UNIT I

1

ELEMENTS OF MODERN NETWORKING

Unit Structure

1.1 OBJECTIVES

After studying this chapter, you should be able to:

- Explain the key elements and their relationships of a modern networking ecosystem, including end users, network providers, application providers and application service providers.
- Discuss the motivation for the typical network hierarchy of access networks, distribution networks, and core networks.
- Present an overview of Ethernet, including a discussion of its application areas and common data rates.
- Present an overview of Wi-Fi, including a discussion of its application areas and common data rates.
- Understand the differences between the five generations of cellular networks.
- Present an overview of cloud computing concepts.
- Describe the Internet of Things.
- Explain the concepts of network convergence and unified communications.

1.2 THE NETWORKING ECOSYSTEM

As enterprises persist to adjust to the ever changing working nature, understanding the long-term effects of the any road-blocker and what actions we all need to take is critical. As technologists modernize their IT infrastructure, they face a host of obstacles, including legacy infrastructure, poor system integration and teams whose programming skills are not up to snuff. Whether IT professionals need to deliver new applications or create a more efficient IT environment, outmoded IT gets in the way. By contrast, a modern infrastructure adapts, helping IT pros keep pace with business needs.

To create a modern and responsive infrastructure, IT teams have virtualized datacenter infrastructure, from servers to storage to networking. Combined with analytics, virtualization helps the network stand up to new demands. As hardware becomes virtualized and programmable through software, IT teams need the skills to enable integration – and the hardware needs to be able to integrate. How IT professionals choose their tools and design their IT environments to be automated, virtualized, programmable, secure, and scalable will surely be the difference between success and failure.

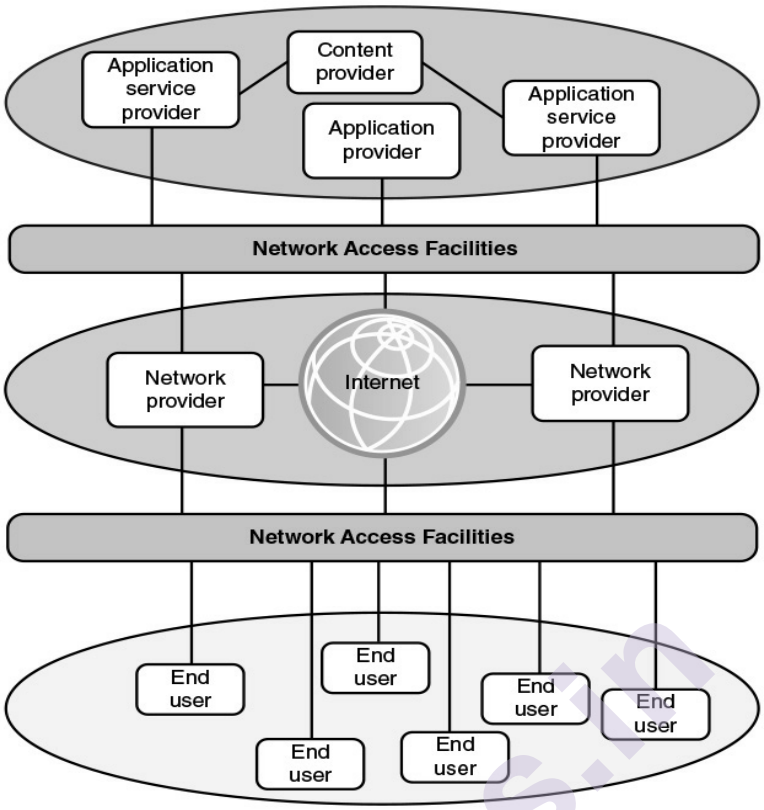


Figure 1.0 The Modern Networking Ecosystem

Figure 1.0 depicts the modern networking ecosystem in very general terms. The entire ecosystem exists to provide services to end users. The term end user, or simply user, is used here as a very general term, to encompass users working within an enterprise or in a public setting or at home. The user platform can be fixed (for example, PC or workstation), portable (for example, laptop), or mobile (for example, tablet or smartphone).

Users connect to network-based services and content through a wide variety of network access facilities. These include digital subscriber line (DSL) and cable modems, Wi-Fi, and Worldwide Interoperability for Microwave Access (WiMAX) wireless modems, and cellular modems. Such network access facilities enable the use to connect directly to the Internet or to a variety of network providers, including Wi-Fi networks, cellular networks, and both private and shared network facilities, such as a premises enterprise network. Ultimately, of course, users want to use network facilities to access applications and content.

Figure 1.0 indicates three broad categories of interest to users. Application providers provide applications, or apps, that run on the user's platform, which is typically a mobile platform. More recently, the concept

of an app store has become available for fixed and portable platforms as well.

A distinct category of provider is the application service provider. Whereas the application provider downloads software to the user's platform, the application service provider acts as a server or host of application software that is executed on the provider's platforms. Traditional examples of such software include web servers, e-mail servers, and database servers. The most prominent example now is the cloud computing provider.

The final (topmost) element shown in Figure 1.0 is the content provider. A content provider serves the data to be consumed on the user device (for example, e-mail, music, video). This data may be commercially provided intellectual property. In some instances, an enterprise may be an application or content provider. Examples of content providers are music record labels and movie studios.

Figure 1.0 is intended to provide a very general depiction of the networking ecosystem. It is worth pointing out here two major elements of modern networking not explicitly depicted in this figure:

- Data center networking: Both large enterprise data centers and cloud provider data centers consist of very large numbers of interconnected servers. Typically, as much as 80 percent of the data traffic is within the data center network, and only 20 percent relies on external networks to reach users.

Results of the 2020 Gartner Magic Quadrant (series of market research reports) for Data Center and Cloud Networking is presented below for reference along with an understanding of the criteria for each category:

- Leaders – Cisco, Arista Networks, Juniper Networks (Typically, innovative giants who excel at both vision and execution)
- Challengers – Huawei (Strong execution but low vision)
- Visionaries – VMWare, Dell EMC, Cumulus Networks, HPE (Aruba) (Good vision but low execution)
- Niche Players – NVIDIA-Mellanox Technologies, Extreme, H3C (Hyper-focused on a small segment, resulting in low vision and low execution)
- IoT or fog networking: An Internet of Things deployed by an enterprise may consist of hundreds, thousands, even millions of devices. The vast bulk of the data traffic to and from these devices is machine to machine, rather than user to machine.

Each of these networking environments creates its own requirements, which are discussed as the book progresses.

1.3 EXAMPLE NETWORK ARCHITECTURES

This section introduces two example network architectures, and with them some of the networking terminology in common use.

1.3.1 A Global Network Architecture:

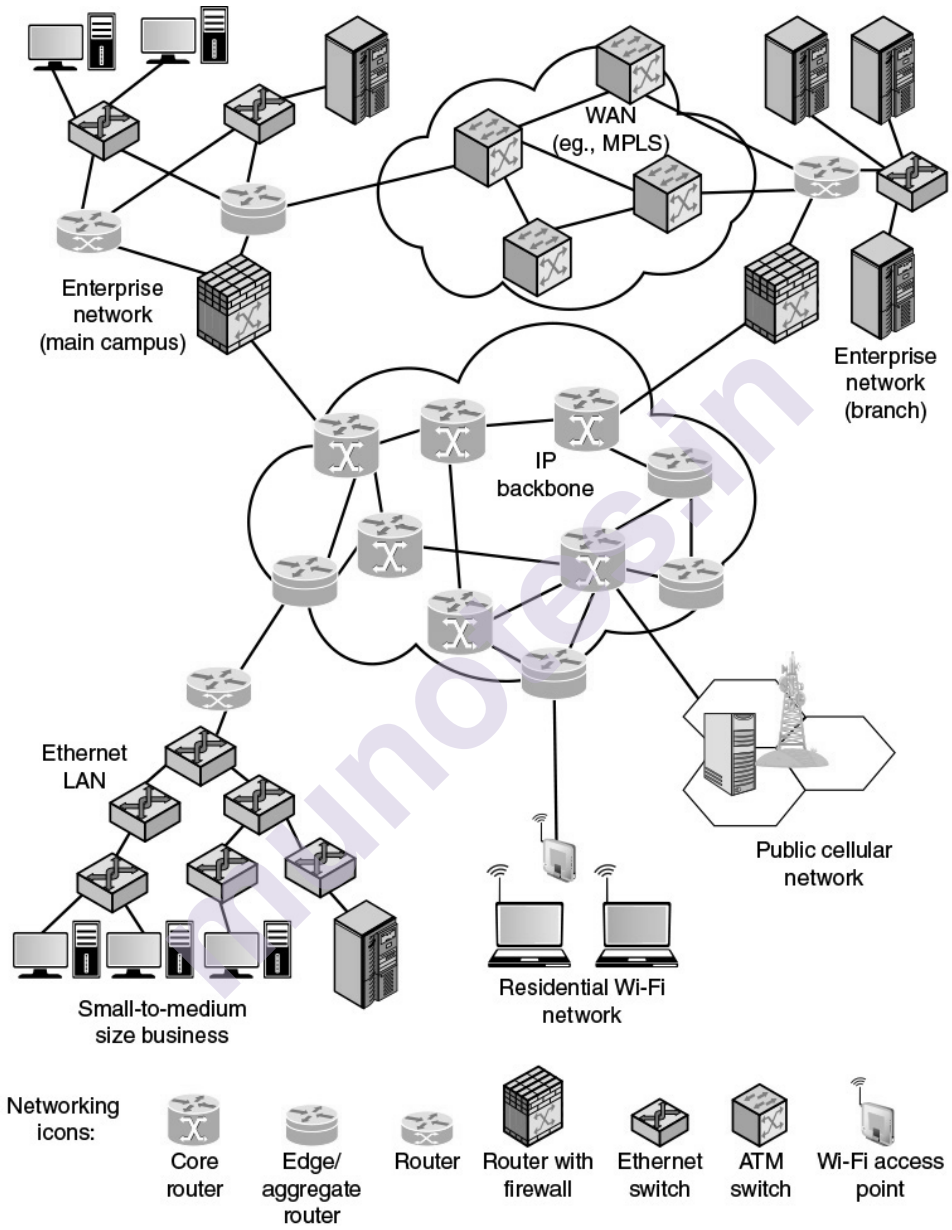


Figure 1.1 A Global Networking Architecture

We begin with an architecture that could represent an enterprise network of national or global extent, or a portion of the Internet with some of its associated networks. Figure 1.1 illustrates some of the typical communications and network elements in use in such a context.

At the center of the figure is an IP backbone, or core, network, which could represent a portion of the Internet or an enterprise IP network. Typically, the backbone consists of high-performance routers, called core routers, interconnected with high-volume optical links. The optical links often make use of what is known as wavelength-division multiplexing (WDM), such that each link has multiple logical channels occupying different portions of the optical bandwidth.

At the periphery of an IP backbone are routers that provide connectivity to external networks and users. These routers are sometimes referred to as edge routers or aggregation routers. Aggregation routers are also used within an enterprise network to connect several routers and switches, to external resources, such as an IP backbone or a high-speed WAN. As an indication of the capacity requirements for core and aggregation routers, the IEEE Ethernet Bandwidth Assessments Group [XI11] reports on an analysis that projects these requirements for Internet backbone providers and large enterprise networks in China. The analysis concludes that aggregation router requirements will be in the range of 200 Gbps to 400 Gbps per optical link by 2020, and 400 Gbps to 1 Tbps per optical link for core routers by 2020.

The upper part of Figure 1.1 depicts a portion of what might be a large enterprise network. The figure shows two sections of the network connected via a private high-speed WAN, with switches interconnected with optical links. MPLS using IP is a common switching protocol used for such WANs; wide-area Ethernet is another option. Enterprise assets are connected to, and protected from, an IP backbone or the Internet via routers with firewall capability, a not uncommon arrangement for implementing the firewall. The lower left of the figure depicts what might be a layout for a small- or medium-size business, which relies on an Ethernet LAN. Connection to the Internet through a router could be through a cable or DSL connection or a dedicated high-speed link.

The lower portion of Figure 1.1 also shows an individual residential user connected to an Internet service provider (ISP) through some sort of subscriber connection. Common examples of such a connection are a DSL, which provides a high-speed link over telephone lines and requires a special DSL modem, and a cable TV facility, which requires a cable modem, or some type of wireless connection. In each case, there are separate issues concerning signal encoding, error control, and the internal structure of the subscriber network. Finally, mobile devices, such as smartphones and tablets, can connect to the Internet through the public cellular network, which has a high-speed connection, typically optical, to the Internet.

1.3.2 A Typical Network Hierarchy:

This section focuses in on a network architecture that, with some variation, is common in many enterprises. As Figure 1.2 illustrates, enterprises often design their network facilities in a three-tier hierarchy: access, distribution, and core.

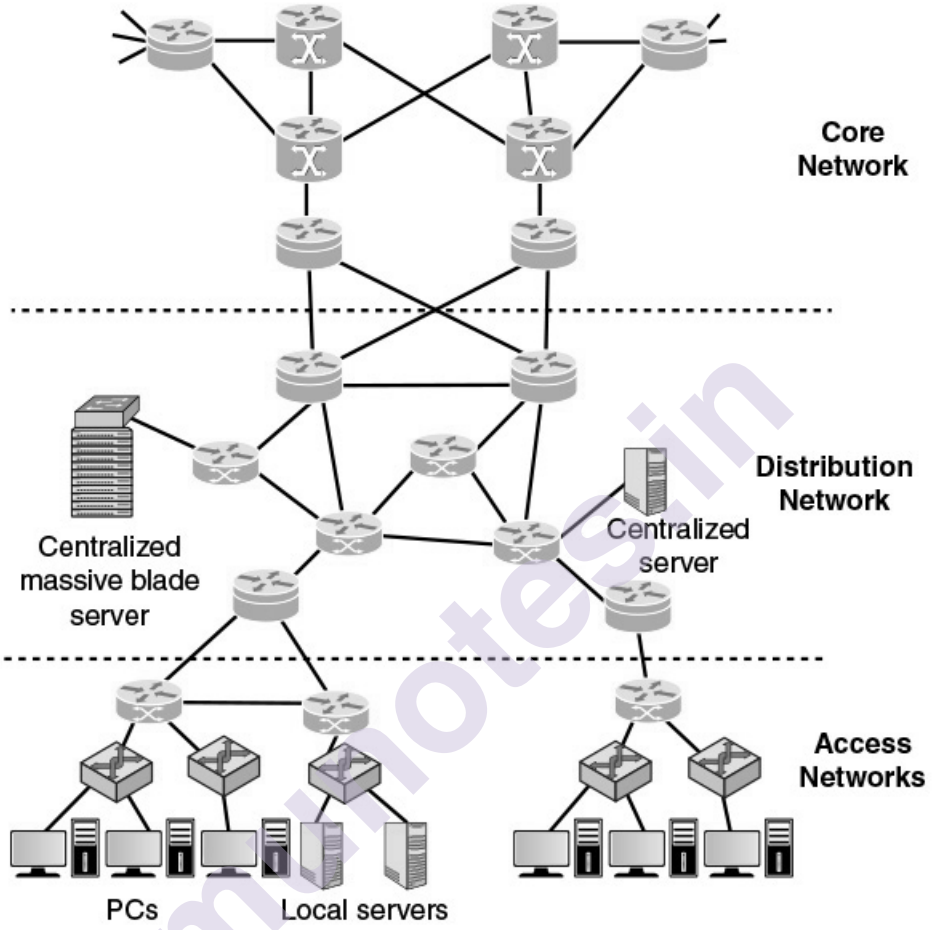


Figure 1.2 A Typical Network Hierarchy

Closest to the end user is the access network. Typically, an access network is a local-area network (LAN) or campus-wide network that consisting of LAN switches (typically Ethernet switches) and, in larger LANs, IP routers that provide connectivity among the switches. Layer 3 switches (not shown) are also commonly used within an LAN. The access network supports end user equipment, such as desktop and laptop computers and mobile devices. The access network also supports local servers that primarily or exclusively serve the users on the local access network.

One or more access routers connect the local assets to the next higher level of the hierarchy, the distribution network. This connection may be via the Internet or some other public or private communications

facility. Thus, as described in the preceding subsection, these access routers function as edge routers that forward traffic into and out of the access network. For a large local facility, there might be additional access routers that provide internal routing but do not function as edge routers (not shown in Figure 1.1).

The distribution network connects access networks with each other and with the core network. An edge router in the distribution network connects to an edge router in an access network to provide connectivity. The two routers are configured to recognize each other and will generally exchange routing and connectivity information and, typically, some traffic-related information. This cooperation between routers is referred to as peering. The distribution network also serves to aggregate traffic destined for the core router, which protects the core from high-density peering. That is, the use of a distribution network limits the number of routers that establish peer relationships with edge routers in the core, saving memory, processing, and transmission capacity. A distribution network may also directly connect servers that are of use to multiple access networks, such as database servers and network management servers.

Again, as with access networks, some of the distribution routers may be purely internal and do not provide an edge router function. The core network, also referred to as a backbone network, connects geographically dispersed distribution networks as well as providing access to other networks that are not part of the enterprise network. Typically, the core network will use very high-performance routers, high-capacity transmission lines, and multiple interconnected routers for increased redundancy and capacity. The core network may also connect to high-performance, high-capacity servers, such as large database servers and private cloud facilities.

Some of the core routers may be purely internal, providing redundancy and additional capacity without serving as edge routers.

A hierarchical network architecture is an example of a good modular design. With this design, the capacity, features, and functionality of network equipment (routers, switches, network management servers) can be optimized for their position in the hierarchy and the requirements at a given hierarchical level.

1.4 ETHERNET

The concept of Ethernet was formulated and introduced by XEROX PARC, now simply known as PARC (Palo Alto Research Centre).This agency proposed to develop a form of system that would permit/allow computers and devices to be connected with one and other

using coaxial cables. Engineers Bob Metcalfe and D.R Boggs developed Ethernet beginning in 1972. In 1976, a connection two computers were made, and data transfer fruitfully took place with the speed of 3MB/second. In 1980, industry standards based on their work were established under IEEE 802.3 set of specifications. In 1990's, fast Ethernet technology came into existence fulfilling the objective of:

a) increasing the performance of previous traditional Ethernet
b) avoiding the need of completely re-cable existing Ethernet networks.

Technologies like Ethernet, Wi-Fi, and 4G/5G cellular networks have evolved to support very high data rates supporting many multimedia applications required by enterprises, consumers and, at the same time, place great demands on network switching equipment and network management facilities.

1.4.1 Applications of Ethernet:

Ethernet is the predominant wired networking technology, used in homes, offices, data centers, enterprises, and WANs. As Ethernet has evolved to support data rates up to 100 Gbps and distances from a few meters to tens of kilometers, it has become essential for supporting personal computers, workstations, servers, and massive data storage devices in organizations large and small.

Ethernet in the Home:

Ethernet has long been used in the home to create a local network of computers with access to the Internet via a broadband modem/router. With the increasing availability of high-speed, low-cost Wi-Fi on computers, tablets, smartphones, modem/routers, and other devices, home reliance on Ethernet has declined. Nevertheless, almost all home networking setups include some use of Ethernet.

Two recent extensions of Ethernet technology have enhanced and broadened the use of Ethernet in the home: powerline carrier (PLC) and Power over Ethernet (PoE). Powerline modems take advantage of existing power lines and use the power wire as a communication channel to transmit Ethernet packets on top of the power signal. This makes it easy to include Ethernet-capable devices throughout the home into the Ethernet network.

PoE acts in a complementary fashion, distributing power over the Ethernet data cable. PoE uses the existing Ethernet cables to distribute power to devices on the network, thus simplifying the wiring for devices such as computers and televisions. With all these Ethernet options, Ethernet will retain a strong presence in home networking, complementing the advantages of Wi-Fi.

Ethernet in the Office:

Ethernet has also long been the dominant network technology for wired local-area networks (LANs) in the office environment. Early on there were some competitors, such as IBM's Token Ring LAN and the Fiber Distributed Data Interface (FDDI), but the simplicity, performance, and wide availability of Ethernet hardware eventually made Ethernet the winner. Today, as with home networks, the wired Ethernet technology exists side by side with the wireless Wi-Fi technology. Much of the traffic in a typical office environment now travels on Wi-Fi, particularly to support mobile devices. Ethernet retains its popularity because it can support many devices at high speeds, is not subject to interference, and provides a security advantage because it is resistant to eavesdropping. Therefore, a combination of Ethernet and Wi-Fi is the most common architecture.

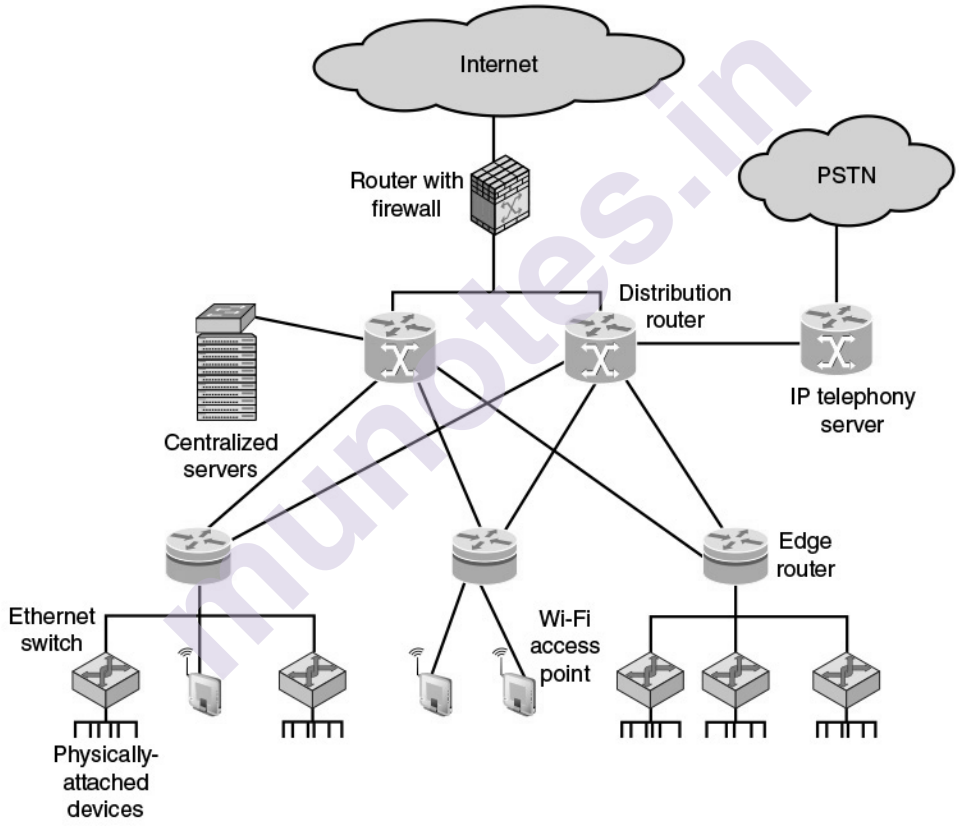


Figure 1.3 A Basic Enterprise LAN architecture

Figure 1.3 provides a simplified example of an enterprise LAN architecture. The LAN connects to the Internet/WANs via a firewall. A hierarchical arrangement of routers and switches provides the interconnection of servers, fixed user devices, and wireless devices. Typically, wireless devices are only attached at the edge or bottom of the hierarchical architecture; the rest of the campus infrastructure is all

Ethernet. There may also be an IP telephony server that provides call control functions (voice switching) for the telephony operations in an enterprise network, with connectivity to the public switched telephone network (PTSN).

Ethernet in the Enterprise:

A tremendous advantage of Ethernet is that it is possible to scale the network, both in terms of distance and data rate, with the same Ethernet protocol and associated quality of service (QoS) and security standards. An enterprise can easily extend an Ethernet network among several buildings on the same campus or even some distance apart, with links ranging from 10 Mbps to 100 Gbps, using a mixture of cable types and Ethernet hardware. Because all the hardware and communications software conform to the same standard, it is easy to mix different speeds and different vendor equipment. The same protocol is used for intensive high-speed interconnections of data servers in a single room, workstations and servers distributed throughout the building, and links to Ethernet networks in other buildings up to 100 km away.

Ethernet in the Data Center:

As in other areas, Ethernet has come to dominate in the data center, where very high data rates are needed to handle massive volumes of data among networked servers and storage units. Historically, data centers have employed various technologies to support high-volume, short-distance needs, including InfiniBand and Fiber Channel. But now that Ethernet can scale up to 100 Gbps, with 400 Gbps on the horizon, the case for a unified protocol approach throughout the enterprise is compelling. Two features of the new Ethernet approach are noteworthy. For co-located servers and storage units, high-speed Ethernet fiber links and switches provided the needed networking infrastructure. Another important version of Ethernet is known as backplane Ethernet. Backplane Ethernet runs over copper jumper cables that can provide up to 100 Gbps over very short distances. This technology is ideal for blade servers, in which multiple server modules are housed in a single chassis.

Ethernet for Wide-Area Networking:

Until recently, Ethernet was not a significant factor in wide-area networking. But gradually, more telecommunications and network providers have switched to Ethernet from alternative schemes to support wide-area access (also referred to as first mile or last mile). Ethernet is supplanting a variety of other wide-area options, such as dedicated T1 lines, synchronous digital hierarchy (SDH) lines, and Asynchronous Transfer Mode (ATM). When used in this fashion, the term carrier Ethernet is applied. The term metro Ethernet, or metropolitan-area network (MAN) Ethernet, is also used. Ethernet has the advantage that it seamlessly fits into the enterprise network for which it provides wide-area

access. But a more important advantage is that carrier Ethernet provides much more flexibility in terms of the data rate capacity that is used, compared to traditional wide-area alternatives. Carrier Ethernet is one of the fastest-growing Ethernet technologies, destined to become the dominant means by which enterprises access wide-area networking and Internet facilities.

1.4.2 Standards:

Within the IEEE 802 LAN standards committee, the 802.3 group is responsible for issuing standards for LANs that are referred to commercially as Ethernet. Complementary to the efforts of the 802.3 committee, the industry consortium known as The Ethernet Alliance supports and originates activities that span from incubation of new Ethernet technologies to interoperability testing to demonstrations to education.

1.4.3 Ethernet Data Rates:

Currently, Ethernet systems are available at speeds up to 100 Gbps. Here is a brief chronology:

- 1983: 10 Mbps (megabit per second, million bits per second)
- 1995: 100 Mbps
- 1998: 1 Gbps (gigabits per second, billion bits per second)
- 2003: 10 Gbps
- 2010: 40 Gbps and 100 Gbps
- 2017: 700 Gbps

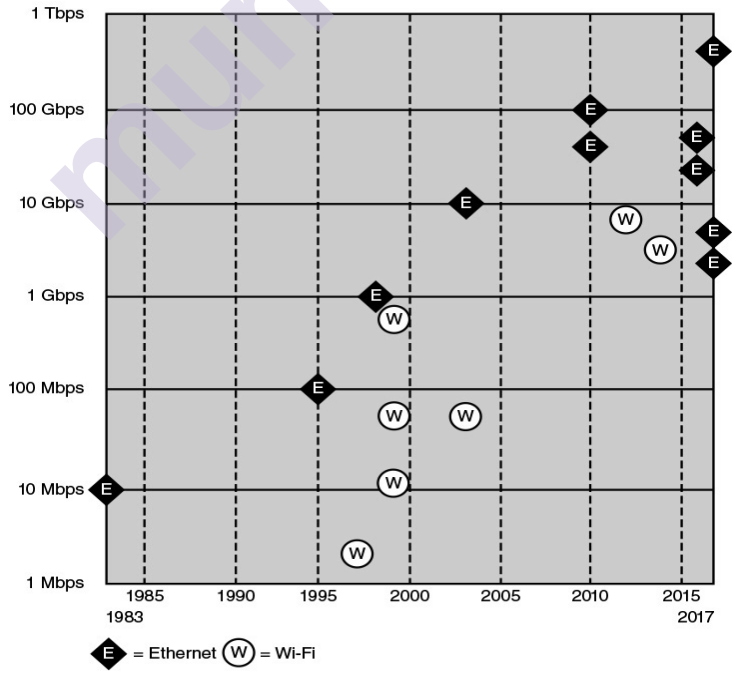


Figure 1.4 Ethernet and Wi-Fi Timelines

1-Gbps Ethernet:

For several years, the initial standard of Ethernet, at 10 Mbps, was adequate for most office environments. By the early 1990s, it was clear that higher data rates were needed to support the growing traffic load on the typical LAN. Key drivers included the following:

- Centralized server farms: In many multimedia applications, there is a need for client system to be able to draw huge amounts of data from multiple, centralized servers, called server farms. As the performance of the servers has increased, the network becomes the bottleneck.
- Power workgroups: These groups typically consist of a small number of cooperating users who need to exchange massive data files across the network. Example applications are software development and computer-aided design.
- High-speed local backbone: As processing demand grows, enterprises develop an architecture of multiple LANs interconnected with a high-speed backbone network.

To meet such needs, the IEEE 802.3 committee developed a set of specifications for Ethernet at 100 Mbps, followed a few years later by a 1-Gbps family of standards. In each case, the new specifications defined transmission media and transmission encoding schemes built on the basic Ethernet framework, making the transition easier than if a completely new specification were issued.

10-Gbps Ethernet:

Even as the ink was drying on the 1-Gbps specification, the continuing increase in local traffic made this specification inadequate for needs in the short-term future. Accordingly, the IEEE 802.3 committee soon issued a standard for 10-Gbps Ethernet. The principle driving requirement for 10-Gbps Ethernet was the increase in intranet (local interconnected networks) and Internet traffic.

Several factors contribute to the explosive growth in both Internet and intranet traffic:

- An increase in the number of network connections
- An increase in the connection speed of each end-station (for example, 10-Mbps users moving to 100 Mbps, analog 56k users moving to DSL and cable modems).
- An increase in the deployment of bandwidth-intensive applications such as high-quality video.
- An increase in web hosting and application hosting traffic.

Initially, network managers used 10-Gbps Ethernet to provide high-speed, local backbone interconnection between large-capacity switches. As the demand for bandwidth increased, 10-Gbps Ethernet began to be deployed throughout the entire network, to include server farm, backbone, and campus-wide connectivity. This technology enables ISPs and network service providers (NSPs) to create very high-speed links at a very low cost between co-located carrier-class switches and routers.

The technology also allows the construction of MANs and WANs that connect geographically dispersed LANs between campuses or points of presence (PoPs).

100-Gbps Ethernet:

The IEEE 802.3 committee soon realized the need for a greater data rate capacity than 10-Gbps Ethernet offers, to support Internet exchanges, high-performance computing, and video-on-demand delivery. The authorization request justified the need for two different data rates in the new standard (40 Gbps and 100 Gbps) by recognizing that aggregate network requirements and end-station requirements are increasing at different rates.

The following are market drivers for 100-Gbps Ethernet:

- **Data center/Internet media providers:** To support the growth of Internet multimedia content and web applications, content providers have been expanding data centers, pushing 10-Gbps Ethernet to its limits. Likely to be high-volume early adopters of 100-Gbps Ethernet.
- **Metro video/service providers:** Video on demand has been driving a new generation of 10-Gbps Ethernet metropolitan/core network buildouts. Likely to be high-volume adopters in the medium term.
- **Enterprise LANs:** Continuing growth in convergence of voice/video/data and in unified communications is driving up network switch demands. However, most enterprises still rely on 1-Gbps or a mix of 1-Gbps and 10-Gbps Ethernet, and adoption of 100-Gbps Ethernet is likely to be slow.
- **Internet exchanges/ISP core routing:** With the massive amount of traffic flowing through these nodes, these installations are likely to be early adopters of 100-Gbps Ethernet.

Figure 1.5 shows an example of the application of 100-Gbps Ethernet. The trend at large data centers, with substantial banks of blade servers, is the deployment of 10-Gbps ports on individual servers to handle the massive multimedia traffic provided by these servers.

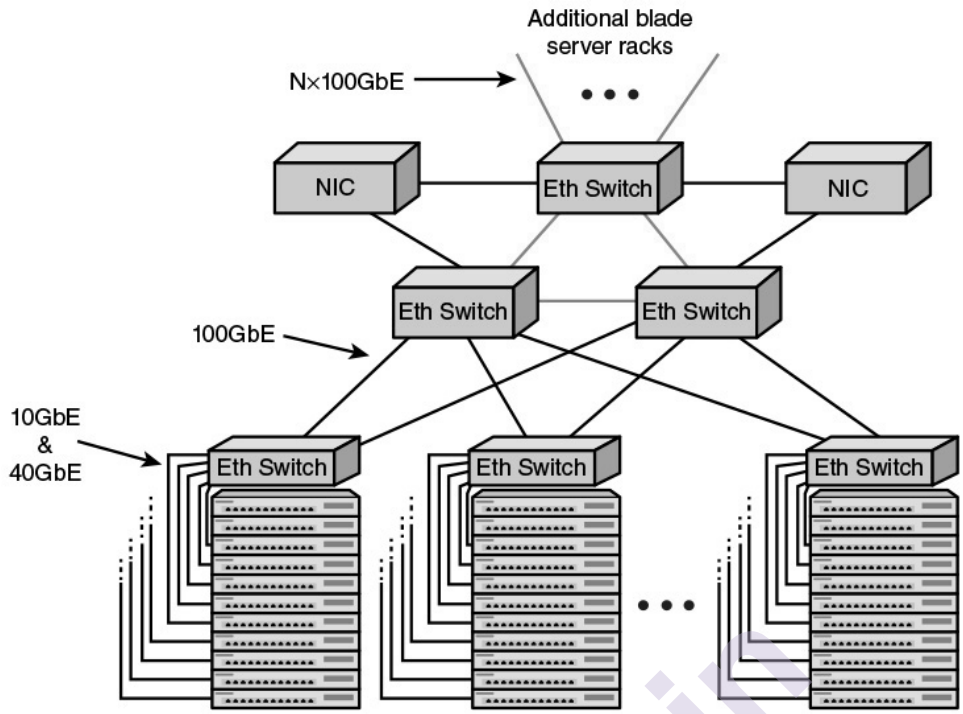


Figure 1.5 Configuration of massive blade server cloud site

Typically, a single blade server rack will contain multiple servers and one or two 10-Gbps Ethernet switches to interconnect all the servers and provide connectivity to the rest of the facility. The switches are often mounted in the rack and referred to as top-of-rack (ToR) switches. The term ToR has become synonymous with server access switch, even if it is not located "top of rack." For very large data centers, such as cloud providers, the interconnection of multiple blade server racks with additional 10-Gbps switches is increasingly inadequate.

To handle the increased traffic load, switches operating at greater than 10 Gbps are needed to support the interconnection of server racks and to provide adequate capacity for connecting offsite through network interface controllers (NICs).

25/50-Gbps Ethernet:

One of the options for implementing 100-Gbps is as four 25-Gbps physical lanes. Therefore, it would be relatively easy to develop standards for 25-Gbps and 50-Gbps Ethernet, using one or two lanes, respectively. Having these two lower-speed alternatives, based on the 100-Gbps technology, would give users more flexibility in meeting existing and near-term demands with a solution that would scale easily to higher data rates. Such considerations have led to the form of the 25 Gigabit Ethernet Consortium by several leading cloud networking providers, including Google and Microsoft. The objective of the Consortium is to support an industry-standard, interoperable Ethernet specification that boosts the performance and slashes the interconnect cost per Gbps between the NIC

and ToR switch. The specification adopted by the Consortium prescribes a single-lane 25-Gbps Ethernet and dual-lane 50-Gbps Ethernet link protocol, enabling up to 2.5 times higher performance per physical lane on twinax copper wire between the rack endpoint and switch compared to 10-Gbps and 40-Gbps Ethernet links. The IEEE 802.3 committee is at work developing the needed standards for 25 Gbps and may include 50 Gbps.

It is too early to say how these various options (25, 40, 50, 100 Gbps) will play out in the marketplace. In the intermediate term, the 100-Gbps switch is likely to predominate at large sites, but the availability of these slower and cheaper alternatives gives enterprises several paths for scaling up to meet increasing demand.

400-Gbps Ethernet:

The growth in demand never lets up. IEEE 802.3 is currently exploring technology options for producing a 400-Gbps Ethernet standard (proposed as 802.3db), although no timetable is yet in place. Looking beyond that milestone, there is widespread acknowledgment that a 1-Tbps (terabits per second, trillion bits per second) standard will eventually be produced.

2.5/5-Gbps Ethernet:

As a testament to the versatility and ubiquity of Ethernet, and while ever higher data rates are being standardized, consensus is developing to standardize two lower rates: 2.5 Gbps and 5 Gbps. These relatively low speeds are also known as Multirate Gigabit BASE-T (MGBASE-T). Currently, the MGBASE-T Alliance is overseeing the development of these standards outside of IEEE. It is likely that the IEEE 802.3 committee will ultimately issue standards based on these industry efforts.

These new data rates are mainly intended to support IEEE 802.11ac wireless traffic into a wired network. IEEE 802.11ac is a 3.2-Gbps Wi-Fi standard that is gaining acceptance where more than 1 Gbps of throughput is needed, such as to support mobile users in the office environment. This new wireless standard overruns 1-Gbps Ethernet link support but may not require the next step up, which is 10 Gbps. If 2.5 and 5 Gbps can be made to work over the same cable that supports 1 Gbps, this would provide a much-needed uplink speed improvement for access points supporting 802.11ac radios with their high bandwidth capabilities.

1.5 Wi-Fi

Just as Ethernet has become the dominant technology for wired LANs, so Wi-Fi, standardized by the IEEE 802.11 committee, has become

the dominant technology for wireless LANs. This overview section discusses applications of Wi-Fi and then looks at standards and performance.

1.5.1 Applications of Wi-Fi:

Wi-Fi is the predominant wireless Internet access technology, used in homes, offices, and public spaces. Wi-Fi in the home now connects computers, tablets, smartphones, and a host of electronic devices, such as video cameras, TVs, and thermostats. Wi-Fi in the enterprise has become an essential means of enhancing worker productivity and network effectiveness. And public Wi-Fi hotspots have expanded dramatically to provide free Internet access in must public places.

Wi-Fi in the Home:

The first important use of Wi-Fi in the home was to replace Ethernet cabling for connecting desktop and laptop computers with each other and with the Internet. A typical layout is a desktop computer with an attached router/modem that provides an interface to the Internet. Other desktop and laptop computers connect either via Ethernet or Wi-Fi to the central router, so that all the home computers can communicate with each other and with the Internet. Wi-Fi greatly simplified the hookup. Not only is there no need for a physical cable hookup, but the laptops can be moved easily from room to room or even outside the house.

Today, the importance of Wi-Fi in the home has expanded tremendously. Wi-Fi remains the default scheme for interconnecting a home computer network. Because both Wi-Fi and cellular capability are now standard on both smartphones and tablets, the home Wi-Fi provides a cost-effective way to the Internet. The smartphone or tablet will automatically use a Wi-Fi connection to the Internet if available, and only switch to the more expensive cellular connection if the Wi-Fi connection is not available. And Wi-Fi is essential to implementing the latest evolution of the Internet: Internet of Things.

Public Wi-Fi:

Access to the Internet via Wi-Fi has expanded dramatically in recent years, as more and more facilities provide a Wi-Fi hotspot, which enables any Wi-Fi device to attach. Wi-Fi hotspots are provided in coffee shops, restaurants, train stations, airports, libraries, hotels, hospitals, department stores, RV parks, and many other places. So many hotspots are available that it is rare to be too far from one. There are now numerous tablet and smartphone apps that increase their convenience.

Even very remote places will be able to support hotspots with the development of the satellite Wi-Fi hotspot. The first company to develop

such a product is the satellite communications company Iridium. The satellite modem will initially provide a relatively low-speed connection, but the data rates will inevitably increase.

Enterprise Wi-Fi:

The economic benefit of Wi-Fi is most clearly seen in the enterprise. Wi-Fi connections to the enterprise network have been offered by many organizations of all sizes, including public and private sector. But in recent years, the use of Wi-Fi has expanded dramatically, to the point that now approximately half of all enterprise network traffic is via Wi-Fi rather than the traditional Ethernet. Two trends have driven the transition to a Wi-Fi-centered enterprise. First, the demand has increased, with more and more employees preferring to use laptops, tablets, and smartphones to connect to the enterprise network, rather than a desktop computer. Second, the arrival of Gigabit Ethernet, especially the IEEE 802.ac standard, allows the enterprise network to support high-speed connections to many mobile devices simultaneously.

Whereas Wi-Fi once merely provided an accessory network designed to cover meetings and public areas, enterprise Wi-Fi deployment now generally provides ubiquitous coverage, to include main offices and remote facilities, and both indoor locations and outdoor spaces surrounding them. Enterprises accepted the need for, and then began to encourage, the practice known as bring your own device (BYOD). The almost universal availability of Wi-Fi capability on laptops, tablets, and smartphones, in addition to the wide availability of home and public Wi-Fi networks, has greatly benefited the organization. Employees can use the same devices and the same applications to continue their work or check their e-mail from wherever they are—home, at their local coffee shop, or while traveling. From the enterprise perspective, this means higher productivity and efficiency and lower costs.

1.5.2 Standards:

Essential to the success of Wi-Fi is interoperability. Wi-Fi-enabled devices must be able to communicate with Wi-Fi access points, such as the home router, the enterprise access point, and public hotspots, regardless of the manufacturer of the device or access point. Such interoperability is guaranteed by two organizations. First, the IEEE 802.11 wireless LAN committee develops the protocol and signaling standards for Wi-Fi. Then, the Wi-Fi Alliance creates test suites to certify interoperability for commercial products that conform to various IEEE 802.11 standards. The term Wi-Fi (wireless fidelity) is used for products certified by the Alliance.

1.5.3 Wi-Fi Data Rates:

Just as businesses and home users have generated a need to extend the Ethernet standard to speeds in the gigabits per second (Gbps) range, the same requirement exists for Wi-Fi. As the technology of antennas, wireless transmission techniques, and wireless protocol design has evolved, the IEEE 802.11 committee has been able to introduce standards for new versions of Wi-Fi at ever-higher speeds. Once the standard is issued, industry quickly develops the products. Here is a brief chronology, starting with the original standard, which was simply called IEEE 802.11, and showing the maximum data rate for each version (Figure 1.4):

- 802.11 (1997): 2 Mbps (megabits per second, million bits per second)
- 802.11a (1999): 54 Mbps
- 802.11b (1999): 11 Mbps
- 802.11n (1999): 600 Mbps
- 802.11g (2003): 54 Mbps
- 802.11ad (2012): 6.76 Gbps (billion bits per second)
- 802.11ac (2014): 3.2 Gbps

IEEE 802.11ac operates in the 5-GHz band, as does the older and slower standards 802.11a and 802.11n. It is designed to provide a smooth evolution from 802.11n. This new standard makes use of advanced technologies in antenna design and signal processing to achieve much greater data rates, at lower battery consumption, all within the same frequency band as the older versions of Wi-Fi.

IEEE 802.11ad is a version of 802.11 operating in the 60- GHz frequency band. This band offers the potential for much wider channel bandwidth than the 5-GHz band, enabling high data rates with relatively simple signal encoding and antenna characteristics. Few devices operate in the 60-GHz band, which means communication experiences less interference than in the other bands used for Wi-Fi.

Because of the inherent transmission limitations of the 60-GHz band, 802.11ad is likely to be useful only within a single room. Because it can support high data rates and, for example, could easily transmit uncompressed high-definition video, it is suitable for applications such as replacing wires in a home entertainment system, or streaming high-definition movies from your cell phone to your television.

Gigabit Wi-Fi holds attractions for both office and residential environments and commercial products are beginning to roll out. In the office environment, the demand for ever greater data rates has led to Ethernet offerings at 10 Gbps, 40 Gbps, and most recently 100 Gbps. These stupendous capacities are needed to support blade servers, heavy

reliance on video and multimedia, and multiple broadband connections offsite. At the same time, the use of wireless LANs has grown dramatically in the office setting to meet needs for mobility and flexibility. With the gigabit-range data rates available on the fixed portion of the office LAN, gigabit Wi-Fi is needed to enable mobile users to effectively use the office resources. IEEE 802.11ac is likely to be the preferred gigabit Wi-Fi option for this environment.

In the consumer and residential market, IEEE 802.11ad is likely to be popular as a low-power, short-distance wireless LAN capability with little likelihood of interfering with other devices. IEEE 802.11ad is also an attractive option in professional media production environments in which massive amounts of data need to be moved short distances.

1.6 4G/5G CELLULAR

Cellular technology is the foundation of mobile wireless communications and supports users in locations that are not easily served by wired networks. Cellular technology is the underlying technology for mobile telephones, personal communications systems, wireless Internet, and wireless web applications, and much more. This section looks at how cellular technology has evolved through four generations and is poised for a fifth generation.

1.6.1 First Generation:

The original cellular networks, now dubbed 1G, provided analog traffic channels and were designed to be an extension of the public switched telephone networks. Users with brick-sized cell phones placed and received calls in the same fashion as landline subscribers. The most widely deployed 1G system was the Advanced Mobile Phone Service (AMPS), developed by AT&T. Voice transmission was purely analog and control signals were sent over a 10-kbps analog channel.

1.6.2 Second Generation:

First-generation cellular networks quickly became highly popular, threatening to swamp available capacity. Second-generation (2G) systems were developed to provide higher-quality signals, higher data rates for support of digital services, and greater capacity. Key differences between 1G and 2G networks include the following:

Digital traffic channels: The most notable difference between the two generations is that 1G systems are almost purely analog, whereas 2G systems are digital. 1G systems are designed to support voice channels; digital traffic is supported only using a modem that converts the digital data into analog form. 2G systems provide digital traffic channels. These

systems readily support digital data; voice traffic is first encoded in digital form before transmitting.

- **Encryption:** Because all the user traffic, and the control traffic, is digitized in 2G systems, it is a relatively simple matter to encrypt all the traffic to prevent eavesdropping. All 2G systems provide this capability, whereas 1G systems send user traffic in the clear, providing no security.
- **Error detection and correction:** The digital traffic stream of 2G systems also lends itself to the use of error detection and correction techniques. The result can be very clear voice reception.
- **Channel access:** In 1G systems, each cell supports several channels. At any given time, a channel is allocated to only one user. 2G systems also provide multiple channels per cell, but each channel is dynamically shared by several users.

1.6.3 Third Generation:

The objective of the third generation (3G) of wireless communication is to provide high-speed wireless communications to support multimedia, data, and video in addition to voice. 3G systems share the following design features:

- **Bandwidth:** An important design goal for all 3G systems is to limit channel usage to 5 MHz There are several reasons for this goal. On the one hand, a bandwidth of 5 MHz or more improves the receiver's ability to resolve multipath when compared to narrower bandwidths. On the other hand, the available spectrum is limited by competing needs, and 5 MHz is a reasonable upper limit on what can be allocated for 3G. Finally, 5 MHz is adequate for supporting data rates of 144 and 384 kbps, the main targets for 3G services.
- **Data rate:** Target data rates are 144 and 384 kbps. Some 3G systems also provide support up to 2 Mbps for office use.
- **Multirate:** The term multirate refers to the provision of multiple fixed-data-rate logical channels to a given user, in which different data rates are provided on different logical channels. Further, the traffic on each logical channel can be switched independently through the wireless and fixed networks to different destinations. The advantage of multirate is that the system can flexibly support multiple simultaneous applications from a given user and can efficiently use available capacity by only providing the capacity required for each service.

1.6.4 Fourth Generation:

The evolution of smartphones and cellular networks has ushered in a new generation of capabilities and standards, which is collectively called

4G. 4G systems provide ultra-broadband Internet access for a variety of mobile devices including laptops, smartphones, and tablets. 4G networks support Mobile web access and high-bandwidth applications such as high-definition mobile TV, mobile video conferencing, and gaming services.

These requirements have led to the development of a fourth generation (4G) of mobile wireless technology that is designed to maximize bandwidth and throughput while also maximizing spectral efficiency. 4G systems have the following characteristics:

- Based on an all-IP packet switched network
- Support peak data rates of up to approximately 100 Mbps for high-mobility mobile access and up to approximately 1 Gbps for low-mobility access such as local wireless access
- Dynamically share and use the network resources to support more simultaneous users per cell
- Support smooth handovers across heterogeneous networks
- Support high QoS for next-generation multimedia applications

In contrast to earlier generations, 4G systems do not support traditional circuit-switched telephone service, providing only IP telephony services.

1.6.5 Fifth Generation:

In telecommunications, 5G is the fifth-generation technology standard for broadband cellular networks, which cellular phone companies began deploying worldwide in 2019, and is the planned successor to the 4G networks which provide connectivity to most current cellphones. Like its predecessors, 5G networks are cellular networks, in which the service area is divided into small geographical areas called cells. All 5G wireless devices in a cell are connected to the Internet and telephone network by radio waves through a local antenna in the cell. The main advantage of the new networks is that they will have greater bandwidth, giving higher download speeds, eventually up to 10 gigabits per second (Gbit/s). Due to the increased bandwidth, it is expected that the new networks will not just serve cellphones like existing cellular networks, but also be used as general internet service providers for laptops and desktop computers, competing with existing ISPs such as cable internet, and also will make possible new applications in internet of things (IoT) and machine to machine areas. Current 4G cellphones will not be able to use the new networks, which will require new 5G enabled wireless devices.

The increased speed is achieved partly by using higher-frequency radio waves than current cellular networks. However, higher-frequency radio waves have a shorter range than the frequencies used by previous cell phone towers, requiring smaller cells. So, to ensure wide service, 5G

networks operate on up to three frequency bands, low, medium, and high. A 5G network will be composed of networks of up to 3 different types of cells, each requiring different antennas, each type giving a different trade-off of download speed vs. distance and service area. 5G cellphones and wireless devices will connect to the network through the highest speed antenna within range at their location:

- Low band 5G uses a similar frequency range to current 4G cellphones, 600-700 MHz, giving download speeds a little higher than 4G: 30-250 megabits per second (Mbit/s). Low-band cell towers will have a range and coverage area like current 4G towers. Mid-band 5G uses microwaves of 2.5-3.7 GHz, currently allowing speeds of 100-900 Mbit/s, with each cell tower providing service up to several miles in radius. This level of service is the most widely deployed and should be available in most metropolitan areas in 2020. Some countries are not implementing low band, making this the minimum service level.

- High band 5G currently uses frequencies of 25-39 GHz, near the bottom of the millimeter wave band, although higher frequencies may be used in the future. It often achieves download speeds of a gigabit per second (Gbit/s), comparable to cable internet. However, millimeter waves (mmWave or mmW) have a more limited range, requiring many small cells. They have trouble passing through some types of walls and windows. Due to their higher costs, current plans are to deploy these cells only in dense urban environments and areas where crowds of people congregate such as sports stadiums and convention centers. The above speeds are those achieved in actual tests in 2020, and speeds are expected to increase during rollout.

1.7 CLOUD COMPUTING

This section provides a brief overview of cloud computing, which is dealt with in greater detail later in the book. Although the general concepts for cloud computing go back to the 1950s, cloud computing services first became available in the early 2000s, particularly targeted at large enterprises. Since then, cloud computing has spread to small- and medium-size businesses, and most recently to consumers. Apple's iCloud was launched in 2012 and had 20 million users within a week of launch. Evernote, the cloud-based note-taking and archiving service, launched in 2008, approached 100 million users in less than six years. In late 2014, Google announced that Google Drive had almost a quarter of a billion active users. Here, we look at the key elements of clouds, including cloud computing, cloud networking, and cloud storage.

1.7.1 Cloud Computing Concepts:

There is an increasingly prominent trend in many organizations to move a substantial portion or even all IT operations to an Internet-

connected infrastructure known as enterprise cloud computing. At the same time, individual users of PCs and mobile devices are relying more and more on cloud computing services to back up data, sync devices, and share, using personal cloud computing.

The National Institute of Standards and Technology (NIST) defines the essential characteristics of cloud computing as follows:

- **Broad network access:** Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (for example, mobile phones, laptops, and personal digital assistants [PDAs]) and other traditional or cloud-based software services.
- **Rapid elasticity:** Cloud computing enables you to expand and reduce resources according to your specific service requirement. For example, you may need many server resources for the duration of a specific task. You can then release these resources upon completion of the task.
- **Measured service:** Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (for example, storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.
- **On-demand self-service:** A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider. Because the service is on demand, the resources are not permanent parts of your IT infrastructure.
- **Resource pooling:** The provider's computing resources are pooled to serve multiple consumers using a multitenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a degree of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources, but may be able to specify location at a higher level of abstraction (for example, country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines. Even private clouds tend to pool resources between different parts of the same organization.

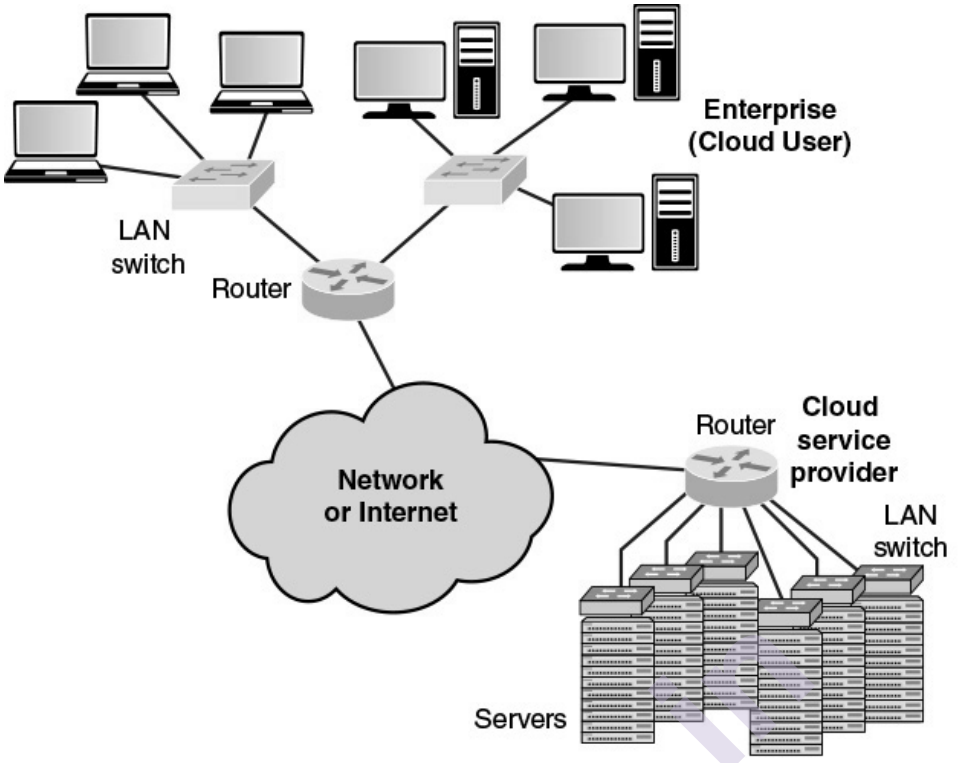


Figure 1.6 Cloud Computing Context

Figure 1.6 illustrates the typical cloud service context. An enterprise maintains workstations within an enterprise LAN or set of LANs, which are connected by a router through a network or the Internet to the cloud service provider. The cloud service provider maintains a massive collection of servers, which it manages with a variety of network management, redundancy, and security tools. In the figure, the cloud infrastructure is shown as a collection of blade servers, which is a common architecture.

1.7.2 The Benefits of Cloud Computing:

Cloud computing benefits include –

a. Flexibility - Users can scale services to fit their needs, customize applications and access cloud services from anywhere with an internet connection. Flexibility further integrates:

- **i Scalability** - Cloud infrastructure scales on demand to support fluctuating workloads.
- **ii Storage options -** Users can choose public, private or hybrid storage offerings, depending on security needs and other considerations.
- **iii Control choices** - Organizations can determine their level of control with as-a-service options. These include software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS).

iv **Tool selection -** Users can select from a menu of prebuilt tools and features to build a solution that fits their specific needs.

v **Security features -** Virtual private cloud, encryption and API keys help keep data secure.

b. Efficiency - Enterprise users can get applications to market quickly, without worrying about underlying infrastructure costs or maintenance. This could be further explained as:

i **Accessibility:** Cloud-based applications and data are accessible from virtually any internet-connected device.

ii **Speed to market:** Developing in the cloud enables users to get their applications to market quickly.

iii **Data security:** Hardware failures do not result in data loss because of networked backups.

iv **Savings on equipment:** Cloud computing uses remote resources, saving organizations the cost of servers and other equipment.

v **Pay structure:** A "utility" pay structure means users only pay for the resources they use.

c. Strategic value: Cloud services give enterprises a competitive advantage by providing the most innovative technology available consisting:

i. **Streamlined work:** Cloud service providers (CSPs) manage underlying infrastructure, enabling organizations to focus on application development and other priorities.

ii. **Regular updates:** Service providers regularly update offerings to give users the most up-to-date technology.

iii. **Collaboration:** Worldwide access means teams can collaborate from widespread locations.

iv. **Competitive edge:** Organizations can move more nimbly than competitors who must devote IT resources to managing infrastructure.

Cloud computing provides economies of scale, professional network management, and professional security management. These features can be attractive to companies large and small, government agencies, and individual PC and mobile users. The individual or company needs to pay only for the storage capacity and services they need.

The user, be it company or individual, does not have the hassle of setting up a database system, acquiring the hardware they need, doing

maintenance, and backup up the data; all this is part of the cloud service. In theory, another big advantage of using cloud computing to store your data and share it with others is that the cloud provider takes care of security.

Alas, the customer is not always protected. There have been several security failures among cloud providers. Evernote made headlines in early 2013 when it told all its users to reset their passwords after an intrusion was discovered.

1.7.3 Cloud Networking:

Cloud networking refers to the networks and network management functionality that must be in place to enable cloud computing. Many cloud computing solutions rely on the Internet, but that is only a piece of the networking infrastructure. One example of cloud networking is the provisioning high-performance/high-reliability networking between the provider and subscriber. In this case, some or all the traffic between an enterprise and the cloud bypasses the Internet and uses dedicated private network facilities owned or leased by the cloud service provider.

More generally, cloud networking refers to the collection of network capabilities required to access a cloud, including making use of specialized services over the Internet, linking enterprise data centers to a cloud, and using firewalls and other network security devices at critical points to enforce access security policies.

1.7.4 Cloud Storage:

We can think of cloud storage as a subset of cloud computing. In essence, cloud storage consists of database storage and database applications hosted remotely on cloud servers. Cloud storage enables small businesses and individual users to take advantage of data storage that scales with their needs and to take advantage of a variety of database applications without having to buy, maintain, and manage the storage assets.

1.8 INTERNET OF THINGS

The Internet of Things (IoT) is the latest development in the long and continuing revolution of computing and communications. Its size, ubiquity, and influence on everyday lives, business, and government dwarf any technical advance that has gone before. This section provides a brief overview of the IoT, which is dealt with in greater detail later in the book.

1.8.1 Things on the Internet of Things:

The Internet of Things (IoT) is a term that refers to the expanding interconnection of smart devices, ranging from appliances to tiny sensors. A dominant theme is the embedding of short-range mobile transceivers into a wide array of gadgets and everyday items, enabling new forms of communication between people and things, and between things themselves.

The Internet now supports the interconnection of billions of industrial and personal objects, usually through cloud systems. The objects deliver sensor information, act on their environment, and in some cases modify themselves, to create overall management of a larger system, like a factory or city.

The IoT is primarily driven by deeply embedded devices. These devices are low-bandwidth, low-repetition data-capture and low-bandwidth data-usage appliances that communicate with each other and provide data via user interfaces.

Embedded appliances, such as high-resolution video security cameras, Video over IP (VoIP) phones, and a handful of others, require high bandwidth streaming capabilities. Yet countless products simply require packets of data to be intermittently delivered.

1.8.2 Evolution:

With reference to end systems supported, the Internet has gone through roughly four generations of deployment culminating in IoT:

1. Information technology (IT): PCs, servers, routers, firewalls, and so on, bought as IT devices by enterprise IT people, primarily using wired connectivity.

2. Operational technology (OT): Machines/appliances with embedded IT built by non-IT companies, such as medical machinery, SCADA (supervisory control and data acquisition), process control, and kiosks, bought as appliances by enterprise OT people and primarily using wired connectivity.

3. Personal technology: Smartphones, tablets, and eBook readers bought as IT devices by consumers (employees) exclusively using wireless connectivity and often multiple forms of wireless connectivity.

4. Sensor/actuator technology: Single-purpose devices bought by consumers, IT, and OT people exclusively using wireless connectivity, generally of a single form, as part of larger systems.

It is the fourth generation that is usually thought of as the IoT, and which is marked using billions of embedded devices.

1.8.3 Layers of the Internet of Things:

Both the business and technical literature often focus on two elements of the Internet of Things - the "things" that are connected, and the Internet that interconnects them. It is better to view the IoT as a massive system, which consists of five layers:

1. Sensors and actuators: Sensors observe their environment and report back quantitative measurements of such variables as temperature, humidity, presence, or absence of some observable, and so on. Actuators operate on their environment, such as changing a thermostat setting or operating a valve.

2. Connectivity: A device may connect via either a wireless or wired link into a network to send collected data to the appropriate data center (sensor) or receive operational commands from a controller site (actuator).

3. Capacity: The network supporting the devices must be able to handle a potentially huge flow of data.
4. Storage: There needs to be a large storage facility to store and maintain backups of all the collected data. This is typically a cloud capability.

5. Data analytics: For large collections of devices, "big data" is generated, requiring a data analytics capability to process the data flow.

All these layers are essential to an effective use of the IoT concept.

1.9 NETWORK CONVERGENCE

Network convergence refers to the merger of previously distinct telephony and information technologies and markets. You can think of this convergence in terms of a three-layer model of enterprise communications:

- **Application convergence:** These are seen by the end users of a business. Convergence integrates communications applications, such as voice calling (telephone), voice mail, e-mail, and instant messaging, with business applications, such as workgroup collaboration, customer relationship management, and back-office functions. With convergence, applications provide rich features that incorporate voice, data, and video in a seamless, organized, and value-added manner. One example is multimedia messaging, which enables a user to use a single interface to access messages from a variety of sources (for example, office voice mail, e-mail, SMS text messages, and mobile voice mail).

- **Enterprise services:** At this level, the manager deals with the information network in terms of the services that must be available to ensure that users can take full advantage of the applications that they use. For example, network managers need to make sure that appropriate privacy mechanisms and authentication services are in place to support convergence-based applications. They may also be able to track user locations to support remote print services and network storage facilities for mobile workers. Enterprise network management services may also include setting up collaborative environments for various users, groups, and applications and QoS provision.

- **Infrastructure:** The network and communications infrastructure consist of the communication links, LANs, WANs, and Internet connections available to the enterprise. Increasingly, enterprise network infrastructure also includes private/public cloud connections to data centers that host high-volume data storage and web services. A key aspect of convergence at this level is the ability to carry voice, image, and video over networks that were originally designed to carry data traffic. Infrastructure convergence has also occurred for networks that were designed for voice traffic. For example, video, image, text, and data are routinely delivered to smartphone users over cell phone networks.

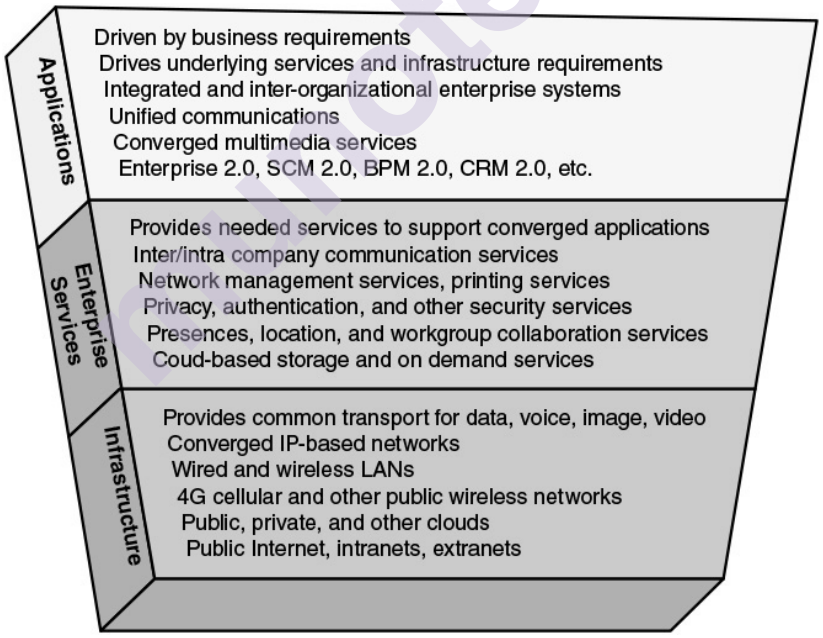


Figure 1.7 Business-driven convergence

Figure 1.7 illustrates the major attributes of the three-layer model of enterprise communications. In simple terms, convergence involves moving an organization's voice, video, and image traffic to a single

network infrastructure. This often involves integrating distinct voice and data networks into a single network infrastructure and extending the infrastructure to support mobile users. The foundation of this convergence is packet-based transmission using the Internet Protocol (IP).

Using IP packets to deliver all varieties of communications traffic, sometimes referred to as everything over IP, enables the underlying infrastructure to deliver a wide range of useful applications to business users.

Convergence brings many benefits, including simplified network management, increased efficiency, and greater flexibility at the application level. For example, a converged network infrastructure provides a predictable platform on which to build new add applications that combine video, data, and voice. This makes it easier for developers to create innovative mashups and other value-added business applications and services.

The following list summarizes three key benefits of IP network convergence:

1. **Cost savings:** A converged network can provide significant double-digit percent reductions in network administration, maintenance, and operating costs; converging legacy networks onto a single IP network enables better use of existing resources, and implementation of centralized capacity planning, asset management, and policy management.

2. **Effectiveness:** The converged environment has the potential to provide users with great flexibility, irrespective of where they are. IP convergence allows companies to create a more mobile workforce. Mobile workers can use a virtual private network (VPN) to remotely access business applications and communication services on the corporate network. A VPN helps maintain enterprise network security by separating business traffic from other Internet traffic.

3. **Transformation:** Because they are modifiable and interoperable, converged IP networks can easily adapt to new functions and features as they become available through technological advancements without having to install new infrastructure. Convergence also enables the enterprise-wide adoption of global standards and best practices, thus providing better data, enhanced real-time decision making, and improved execution of key business processes and operations. The result is enhanced agility and innovation, the key ingredients of business innovation.

These compelling business benefits are motivating companies to invest in converged network infrastructures. Businesses, however, are keenly aware of the downside of convergence: having a single network means a single point of failure. Given their reliance on ICT (information and communications technology), today's converged enterprise network infrastructures typically include redundant components and back up systems to increase network resiliency and lessen the severity of network outages.

1.10 UNIFIED COMMUNICATIONS

While enterprise network convergence focuses on the consolidation of traditionally distinct voice, video, and data communications networks into a common infrastructure, Unified Communications (UC) focuses on the integration of real-time communication services to optimize business processes. As with converged enterprise networks, IP is the cornerstone on which UC systems are built.

Key elements of Unified Communications include the following:

1. UC systems typically provide a unified user interface and consistent user experience across multiple devices and media.
2. UC merges real-time communications services with non-real-time services and business process applications.

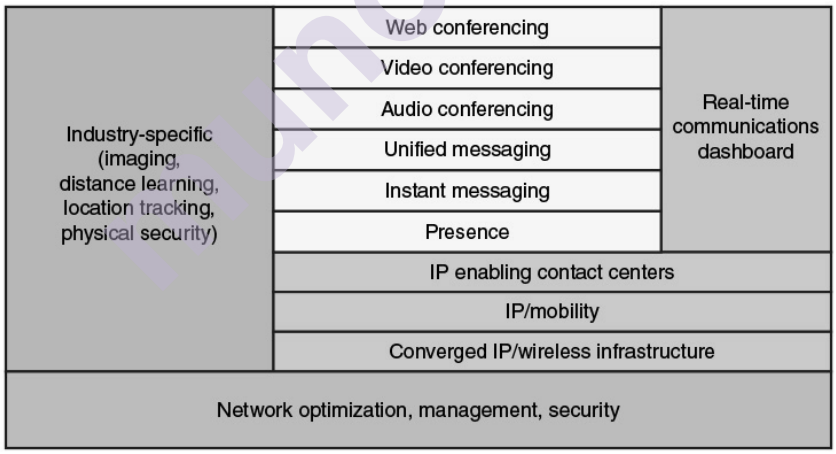


Figure 1.8 Elements of a Unified Communications

Architecture:

Figure 1.8 shows the typical components of a UC architecture and how they relate to one another.

The key elements of this architecture are as follows:

- Real-time communications (RTC) dashboard: An RTC dashboard is a key component of UC architecture. This is the element that provides UC users with a unified user interface across communication devices. Ideally, the user has a consistent interface no matter what communication device the user is currently using, whether it is a cell phone, wireless tablet computer, desktop system, or office telephone attached to the corporate private branch exchange (PBX). As you can see in Figure 1.8, RTC dashboards provide access to real-time communication services such as instant messaging, audio and video conferencing, and interactive whiteboards; RTC dashboards also provide access to non-real-time services such as unified messaging (e-mail, voice mail, fax, and SMS) in unified view. An RTC dashboard includes presence information about co-workers and partners so that users can know on the fly which colleagues are available to communicate or join a collaborative communication session. RTC dashboards have become necessities in organizations that require high levels of communication and collaboration to support business processes.

- **Web conferencing:** Refers to live meetings or presentations in which participants access the meeting or presentation via a mobile device or the web, either over the Internet, or corporate intranet. Web conferences often include data sharing through web-connected interactive white boards (IWBs).

- **Audio conferencing:** Also called conference calling, refers to a live meeting in which participants are linked together for audio transmission and reception. A participant may be on a landline, mobile phone, or at a "softphone" - a computer equipped with microphone and speaker.

- **Unified messaging:** Unified messaging systems provide a common repository for messages from multiple sources. It allows users to retrieve saved e-mail, voice mail, and fax messages from a computer, telephone, or mobile device. Computer users can select and play voice-mail recordings that appear in their unified messaging inboxes. Telephone users can both retrieve voice mail and hear text-to-voice translations of e-mail messages. Messages of any type can be saved, answered, filed, sorted, and forwarded. Unified messaging systems relieve business users from having to monitor multiple voice mailboxes by enabling voicemail messages received by both office phones and cell phones to be saved to the same mailbox. With UC, users can use any device at any time to retrieve e-mail or voicemail from unified messaging mailboxes.

- **Instant messaging (IM):** Real-time text-based messaging between two or more participants. IM is like online chat because it is text-based

and exchanged bidirectionally in real time. IM is distinct from chat in that IM clients use contact (or buddy) lists to facilitate connections between known users, whereas online chat can include text-based exchanges between anonymous users.

- **Video teleconferencing (VTC):** Videoconferencing allows users in two or more locations to interact simultaneously via two-way video and audio transmission. UC systems enable users to participate in video conferences via desktop computers, smartphones, and mobile devices.

- **Presence:** The capability to determine, in real time, where someone is, how that person prefers to be reached, and even what the person is currently doing. Presence information shows the individual's availability state before co-workers attempt to contact them person. It was once considered simply an underlying technology to instant messaging (for example, "available to chat" or "busy") but has been broadened to include whether co-workers are currently on office or mobile phones, logged in to a computer, involved in a video call or in a meeting, or out of the office for lunch or vacation. A co-worker's geographic location is becoming more common as an element in presence information for several business reasons, including the capability to quickly respond to customer emergencies. Business has embraced presence information because it facilitates more efficient and effective communication. It helps eliminate inefficiencies associated with "phone tag" or composing and sending e-mails to someone who could more quickly answer a question over the phone or with a quick meeting.

- **IP enabling contact centers:** Refers to the use of IP-based unified communications to enhance customer contact center functionality and performance. The unified communications infrastructure makes use of presence technology to enable customers and internal enterprise employees to be quickly connected to the required expert or support person.

  In addition, this technology supports mobility, so that call center personnel need not be located at a particular office or remain in a particular place. Finally, the UC infrastructure enables the call center employee to quickly access other employees and information assets, including data, video, image, and audio.

- **IP/mobility:** Refers to the delivery of information to and collection of information from enterprise personnel who are usually mobile, using an IP network infrastructure. In a typical enterprise, upward of 30 percent of employees use some form of weekly remote access technology in the performance of their jobs.

- **Converged IP/wireless infrastructure:** A unified networking and communications-based IP packet transfer to support voice, data, and video transmission and can be extended to include local- and wide-area wireless communications. UC-enabled mobile devices can switch between Wi-Fi and cellular systems in the middle of a communication session.

For example, a UC user could receive a co-worker's call via a smartphone connected to Wi-Fi network at home, continue the conversation while driving to work over a cellular network connection, and could end the call at the office while connected to the business's Wi-Fi network. Both handoffs (home Wi-Fi to cellular and cellular to office Wi-Fi) would take place seamlessly and transparently without dropping the call.

The importance of UC is not only that it integrates communication channels but also that it offers a way to integrate communication functions and business applications. Three major categories of benefits are typically realized by organizations that use UC:

- **Personal productivity gains:** Presence information helps employees find each other and choose the most effective way to communicate in real time. Less time is wasted calling multiple numbers to locate co-workers or checking multiple worked-related voice mailboxes. Calls from VIP contacts can be routed simultaneously to all a UC user's phone devices (office phone, softphone, smartphone, home phone) to ensure faster responsiveness to customers, partners, and co-workers. With mobile presence information capabilities, employees who are geographically closest can be dispatched to address a problem.

- **Workgroup performance gains:** UC systems support real-time collaboration among team members, which facilitates workgroup performance improvements. Examples include the use of presence information to speed identification of an available individual with the right skills a work team needs to address a problem. Enhanced conferencing capabilities with desktop VTC and interactive white boards and automated business rules to route or escalate communications also help to increase workgroup performance.

- **Enterprise-level process improvements:** IP convergence enables UC to be integrated with enterprise-wide and departmental-level applications, business processes, and workflows. UC-enabled enhanced communications with customers, suppliers, and business partners are redefining best practices for customer relationship management (CRM), supply chain management (SCM), and other enterprise-wide applications and are transforming relationships among

members of business networks. Communication-enabled business processes (CEBP) are fueling competition in several industries, including financial services, healthcare, and retail.

1.11 SUMMARY

- Users connect to network-based services and content through a wide variety of network access facilities. These include digital subscriber line (DSL) and cable modems, Wi-Fi, and Worldwide Interoperability for Microwave Access (WiMAX) wireless modems, and cellular modems.
- At the periphery of an IP backbone are routers that provide connectivity to external networks and users.
- Some of the core routers may be purely internal, providing redundancy and additional capacity without serving as edge routers.
- Two recent extensions of Ethernet technology have enhanced and broadened the use of Ethernet in the home: powerline carrier (PLC) and Power over Ethernet (PoE).
- A tremendous advantage of Ethernet is that it is possible to scale the network, both in terms of distance and data rate, with the same Ethernet protocol and associated quality of service (QoS) and security standards.
- The technology also allows the construction of MANs and WANs that connect geographically dispersed LANs between campuses or points of presence (PoPs).
- In the consumer and residential market, IEEE 802.11ad is likely to be popular as a low-power, short-distance wireless LAN capability with little likelihood of interfering with other devices.
- Cloud networking refers to the networks and network management functionality that must be in place to enable cloud computing.
- Convergence brings many benefits, including simplified network management, increased efficiency, and greater flexibility at the application level.
- UC systems typically provide a unified user interface and consistent user experience across multiple devices and media.
- The importance of UC is not only that it integrates communication channels but also that it offers a way to integrate communication functions and business applications.
- Cloud computing benefits include flexibility, efficiency and strategic value.

1.12 UNIT END QUESTION

1. Explain the key elements and their relationships of a modern networking ecosystem, including end users, network providers, application providers and application service providers.
2. Discuss the motivation for the typical network hierarchy of access networks, distribution networks, and core networks.
3. Present an overview of Ethernet, including a discussion of its application areas and common data rates.
4. Present an overview of Wi-Fi, including a discussion of its application areas and common data rates.
5. Understand the differences between the five generations of cellular networks.
6. Present an overview of cloud computing concepts.
7. Describe the Internet of Things.
8. Explain the concepts of network convergence and unified communications.

1.13 REFERENCES

1. Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud, First printing: November 2015 by William Stallings, Copyright © 2016 by Pearson Education, Inc. ISBN-13: 978-0-13-417539-3
2. Research paper on "Use of Ethernet Technology in Computer Network" by Zobair Ullah, published in Global Journal of Computer Science and Technology Network, Web & Security Volume 12 Issue 14 Version 1.0 Year 2012, available online at https://core.ac.uk/download/pdf/231149472.pdf
3. IEEE 802.3 standards working group's information available online at https://en.wikipedia.org/wiki/IEEE_802.3
4. Note on 5G available online at https://en.wikipedia.org/wiki/5G
5. Online information available regarding benefits of cloud computing - https://www.ibm.com/in-en/cloud/learn/benefits-of-cloud-computing
6. Online article "The future of networking: A guide to the intelligent network" by Lauren Horwitz, available on Cisco Digital Network Architecture (DNA) portal online at https://www.cisco.com/c/en/us/solutions/enterprise-networks/future-of-networking.html

*****

2

REQUIREMENTS AND TECHNOLOGY

Unit Structure

2.1 OBJECTIVES

After studying this chapter, you should be able to:

- Present an overview of the major categories of packet traffic on the Internet and internets, including elastic, inelastic, and real-time traffic.

- Discuss the traffic demands placed on contemporary networks by big data, cloud computing, and mobile traffic.
- Explain the concept of quality of service.
- Explain the concept of quality of experience.
- Understand the essential elements of routing.
- Understand the effects of congestion and the types of techniques used for congestion control.
- Compare and contrast software-defined networking and network functions virtualization.

2.2 TYPES OF NETWORK AND INTERNET TRAFFIC

Most of the Internet traffic today is generated by traditional data applications; such traffic is for the most part burst and is well served by the best-effort service that the Internet provides. With the growth and ubiquity of the Internet witnessed in recent years, new applications are being contemplated, introducing new traffic types and new requirements, which in turn require new services from the network which cater to these characteristics and requirements. Furthermore, as the Internet becomes a network on which many businesses rely, it becomes crucial for the network response time to be unaffected by increases in the load on the network.

The phrase traffic classification is used to describe methods of classifying traffic based on features passively observed in the traffic, and according to specific classification goals. One might only have a coarse classification goal, i.e., whether its transaction-oriented, bulk-transfer, or peer-to-peer file sharing. Or one might have a finer-grained classification goal, i.e., the exact application represented by the traffic. Traffic features could include the port number, application payload, or temporal, packet size, and addressing characteristics of the traffic. Methods to classify include exact matching, e.g., of port number or payload, heuristic, or machine learning (statistics).

Traffic on the Internet and enterprise networks can be divided into two broad categories: elastic and inelastic. A consideration of their differing requirements clarifies the need for an enhanced networking architecture.

Elastic Traffic:

Elastic traffic is that which can adjust, over wide ranges, to changes in delay and throughput across an internet and still meet the needs of its applications. This is the traditional type of traffic supported on TCP/IP-based internets and is the type of traffic for which internets were

designed. Applications that generate such traffic typically use Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) as a transport protocol. In the case of UDP, the application will use as much capacity as is available up to the rate that the application generates data. In the case of TCP, the application will use as much capacity as is available up to the maximum rate that the end-to-end receiver can accept data. Also, with TCP, traffic on individual connections adjusts to congestion by reducing the rate at which data are presented to the network. Applications that can be classified as elastic include the common applications that operate over TCP or UDP, including file transfer (File Transfer Protocol / Secure FTP [FTP/SFTP]), electronic mail (Simple Mail Transport Protocol [SMTP]), remote login (Telnet, Secure Shell [SSH]), network management (Simple Network Management Protocol [SNMP]), and web access (Hypertext Transfer Protocol / HTTP Secure [HTTP/HTTPS]). However, there are differences among the requirements of these applications, including the following:

- E-mail is generally insensitive to changes in delay.
- When file transfer is done via user command rather than as an automated background task, the user expects the delay to be proportional to the file size and so is sensitive to changes in throughput.
- With network management, delay is generally not a serious concern. However, if failures in an internet are the cause of congestion, then the need for SNMP messages to get through with minimum delay increases with increased congestion.
- Interactive applications, such as remote logon and web access, are sensitive to delay.

It is important to realize that it is not per-packet delay that is the quantity of interest. Observation of real delays across the Internet suggest that wide variations in delay do not occur. Because of the congestion control mechanisms in TCP, when congestion develops, delays only increase modestly before the arrival rate from the various TCP connections slow down. Instead, the quality of service (QoS) perceived by the user relates to the total elapsed time to transfer an element of the current application. For an interactive Telnet-based application, the element may be a single keystroke or single line. For web access, the element is a web page, which could be as little as a few kilobytes or could be substantially larger for an image-rich page.

For a scientific application, the element could be many megabytes of data. For very small elements, the total elapsed time is dominated by the delay time across the Internet. However, for larger elements, the total elapsed time is dictated by the sliding-window performance of TCP and is therefore dominated by the throughput achieved over the TCP connection.

Thus, for large transfers, the transfer time is proportional to the size of the file and the degree to which the source slows because of congestion.

It should be clear that even if you confine your attention to elastic traffic, some service prioritizing and controlling traffic could be of benefit. Without such a service, routers are dealing evenhandedly with arriving IP packets, with no concern for the type of application and whether a particular packet is part of a large transfer element or a small one. Under such circumstances, and if congestion develops, it is unlikely that resources will be allocated in such a way as to meet the needs of all applications fairly. When inelastic traffic is added to the mix, the results are even more unsatisfactory.

Inelastic Traffic:

Inelastic traffic does not easily adapt, if at all, to changes in delay and throughput across an internet. Examples of inelastic traffic include multimedia transmission, such as voice and video, and high-volume interactive traffic, such as an interactive simulation application (for example, airline pilot simulation). The requirements for inelastic traffic may include the following:

- **Throughput:** A minimum throughput value may be required. Unlike most elastic traffic, which can continue to deliver data with perhaps degraded service, many inelastic applications absolutely require a given minimum throughput.
- **Delay:** Also called latency. An example of a delay-sensitive application is stock trading; someone who consistently receives later service will consistently act later, and with greater disadvantage.
- **Delay jitter:** The magnitude of delay variation, called delay jitter, or simply jitter, is a critical factor in real-time applications. Because of the variable delay imposed by an internet, the interarrival times between packets are not maintained at a fixed interval at the destination. To compensate for this, the incoming packets are buffered, delayed sufficiently to compensate for the jitter, and then released at a constant rate to the software that is expecting a steady real-time stream. The larger the allowable delay variation, the longer the real delay in delivering the data and the greater the size of the delay buffer required at receivers. Real-time interactive applications, such as teleconferencing, may require a reasonable upper bound on jitter.
- **Packet loss:** Real-time applications vary in the amount of packet loss, if any, that they can sustain.

Application Category	Service Class	Traffic Characteristics	Tolerance to Loss	Tolerance to Delay	Tolerance to Jitter
Control	Network control	Variable-size packets, mostly inelastic short messages, but traffic can also burst (BGP)	Low	Low	Yes
	OA&M	Variable-size packets, elastic and inelastic flows	Low	Medium	Yes
Media-Oriented	Telephony	Fixed-size small packets, constant emission rate, inelastic and low-rate flows	Very low	Very low	Very low
	Real-time interactive	RTP/UDP streams, inelastic, mostly variable rate	Low	Very low	Low
	Multimedia conferencing	Variable-size packets, constant transmit interval, rate adaptive, reacts to loss	Low-medium	Very low	Low
	Broadcast video	Constant and variable rate, inelastic,	Very low	Medium	Low
	Multimedia Streaming	Variable-size packets, elastic with variable rate	Low-medium	Medium	Yes
Data	Low-latency data	Variable rate, bursty short-lived elastic flows	Low	Low-medium	Yes
	High-throughput data	Variable rate, bursty long-lived elastic flows	Low	Medium-high	Yes
	Low-priority data	Non-real-time and elastic	High	High	Yes
Best effort	Standard	A bit of everything	Not specified		

BGP = Border Gateway Protocol

OA&M = Operations, administration, and management

RTP = Real-time Transport Protocol

UDP = User Datagram Protocol

Table 2.1 Service Class Characteristics

Table 2.1 above shows the loss, delay, and jitter characteristics of various classes of traffic, as specified in RFC 4594 (Configuration Guidelines for DiffServ Service Classes, August 2006).

Table 2.2 below gives examples of QoS requirements for various media-oriented applications [SZIG14]

Voice	One-way latency ≤ 150 ms
	One-way peak-to-peak jitter ≤ 30 ms
	Per-hop peak-to-peak jitter ≤ 10 ms
	Packet loss ≤ 1 percent
Broadcast video	Packet loss ≤ 0.1 percent
Real-time interactive video	One-way latency ≤ 200 ms
	One-way peak-to-peak jitter ≤ 50 ms
	Per-hop peak-to-peak jitter ≤ 10 ms
	Packet loss ≤ 0.1 percent
Multimedia conferencing	One-way latency ≤ 200 ms
	Packet loss ≤ 1 percent
Multimedia streaming	One-way latency ≤ 400 ms
	Packet loss ≤ 1 percent

Table 2.2 QoS Requirements by Application Class

These requirements are difficult to meet in an environment with variable queuing delays and congestion losses. Accordingly, inelastic traffic introduces two new requirements into the internet architecture. First, some means is needed to give preferential treatment to applications with more demanding requirements. Applications need to be able to state their requirements, either ahead of time in some sort of service request function, or on the fly, by means of fields in the IP packet header. The former approach provides more flexibility in stating requirements, and it enables the network to anticipate demands and deny new requests if the required resources are unavailable. This approach implies the use of some sort of resource reservation protocol.

An additional requirement in supporting inelastic traffic in an internet architecture is that elastic traffic must still be supported. Inelastic applications typically do not back off and reduce demand in the face of congestion, in contrast to TCP-based applications. Therefore, in times of congestion, inelastic traffic will continue to supply a high load, and elastic traffic will be crowded off the internet. A reservation protocol can help control this situation by denying service requests that would leave too few resources available to handle current elastic traffic.

2.2.1 Reat-Time Traffic Characteristics:

As mentioned, a common example of inelastic traffic is real-time traffic. With traditional elastic applications, such as file transfer, electronic mail, and client/server applications including the web, the performance metrics of interest are generally throughput and delay. There is also a concern with reliability, and mechanisms are used to make sure that no

data are lost, corrupted, or misordered during transit. By contrast, real-time applications are concerned with timing issues as well as packet loss. In most cases, there is a requirement that data be delivered at a constant rate equal to the sending rate. In other cases, a deadline is associated with each block of data, such that the data are not usable after the deadline has expired.

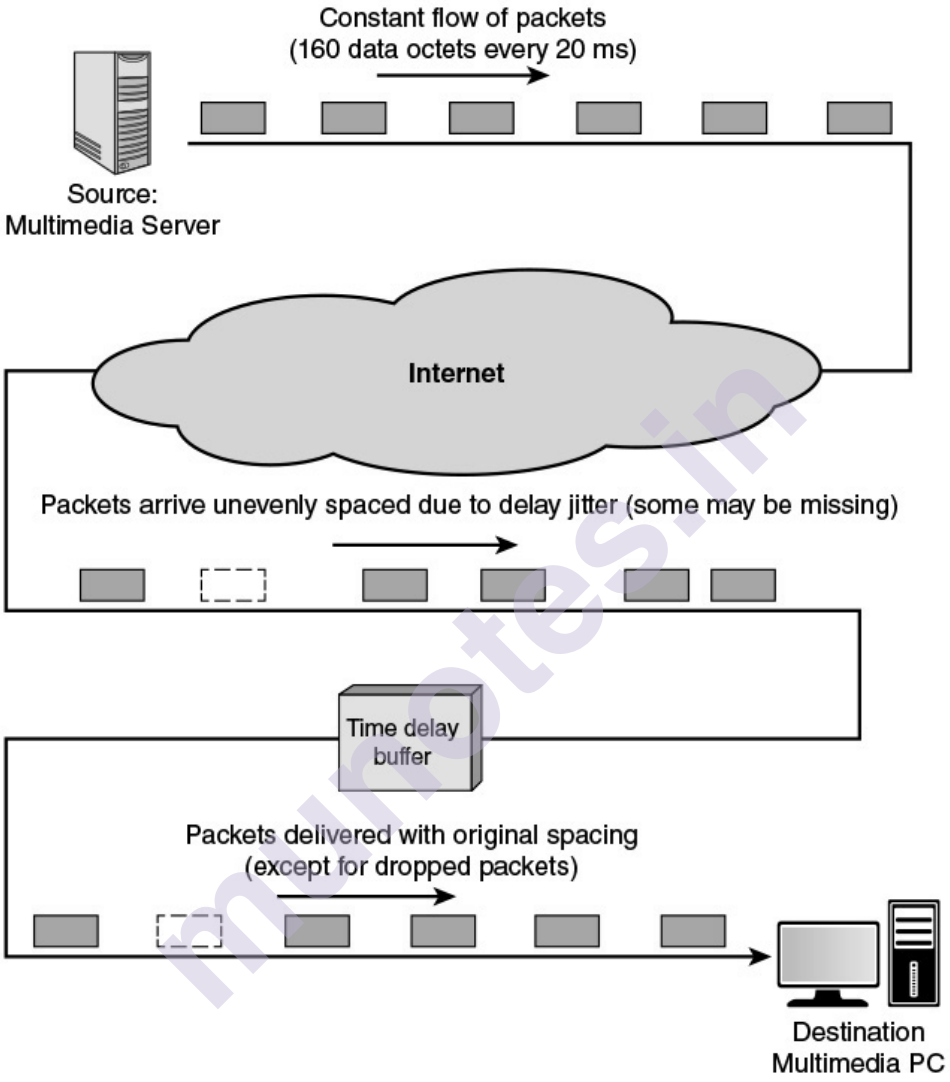


Figure 2.1 Real-Time Traffic

Figure 2.1 illustrates a typical real-time environment. Here, a server is generating audio to be transmitted at 64 kbps. The digitized audio is transmitted in packets containing 160 octets of data, so that one packet is issued every 20 ms. These packets are passed through an internet and delivered to a multimedia PC, which plays the audio in real time as it arrives. However, because of the variable delay imposed by the internet, the interarrival times between packets are not maintained at a fixed 20 ms at the destination. To compensate for this, the incoming packets are

buffered, delayed slightly, and then released at a constant rate to the software that generates the audio. The buffer may be internal to the destination machine or in an external network device.

The compensation provided by the delay buffer is limited. For example, if the minimum end-to-end delay seen by any packet is 1 ms and the maximum is 6 ms, the delay jitter is 5 ms. As long as the time delay buffer delays incoming packets by at least 5 ms, the output of the buffer will include all incoming packets. However, if the buffer delayed packets by only 4 ms, any incoming packets that had experienced a relative delay of more than 4 ms (an absolute delay of more than 5 ms) would have to be discarded so as not to be played back out of order.

The description of real-time traffic so far implies a series of equal-size packets generated at a constant rate. This is not always the profile of the traffic. Figure 2.2 illustrates some of the common possibilities, as described in the list that follows.

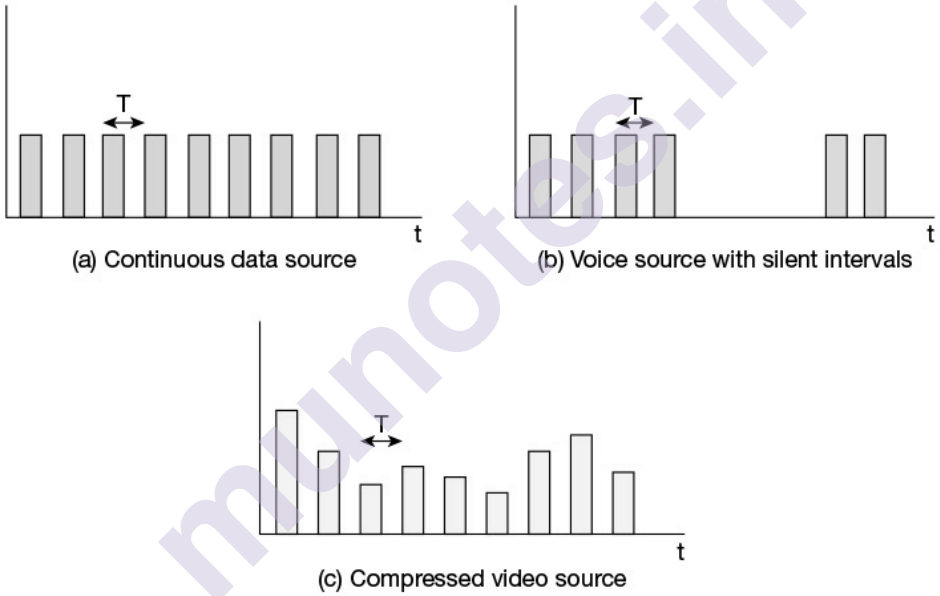


Figure 2.2 Real-Time Packet Transmission

- **Continuous data source:** Fixed-size packets are generated at fixed intervals. This characterizes applications that are constantly generating data, have few redundancies, and that are too important to compress in a lossy way. Examples are air traffic control radar and real-time simulations.

- **On/off source:** The source alternates between periods when fixed-size packets are generated at fixed intervals and periods of inactivity. A voice source, such as in telephony or audio conferencing, fits this profile.

- **Variable packet size:** The source generates variable-length packets at uniform intervals. An example is digitized video in which different frames may experience different compression ratios for the same output quality level.

2.3 DEMAND: BIG DATA, CLOUD COMPUTING, AND MOBILE TRAFFIC

Having looked at the types of traffic presented to the Internet and other IP-based networks, consider the application areas that are generating the greatest stress on network resources and management. Three areas stand out: big data, cloud computing, and mobility. All of these areas suggest the need for using powerful tools such as software-defined networking (SDN) and network functions virtualization (NFV) for network operation and management, and for using comprehensive QoS and quality of experience (QoE) systems for effective delivery of services over IP-based networks.

2.3.1 Big Data:

In simple terms, big data refers to everything that enables an organization to create, manipulate, and manage very large data sets (measured in terabytes, petabytes, exabytes, and so on) and the facilities in which these are stored. Distributed data centers, data warehouses, and cloud-based storage are common aspects of today's enterprise networks. Many factors have contributed to the merging of "big data" and business networks, including continuing declines in storage costs, the maturation of data mining and business intelligence (BI) tools, and government regulations and court cases that have caused organizations to stockpile large masses of structured and unstructured data, including documents, e-mail messages, voice-mail messages, text messages, and social media data. Other data sources being captured, transmitted, and stored include web logs, Internet documents, Internet search indexing, call detail records, scientific research data and results, military surveillance, medical records, video archives, and e-commerce transactions.

Data sets continue to grow with more and more being gathered by remote sensors, mobile devices, cameras, microphones, radio frequency identification (RFID) readers, and similar technologies. One study from a few years ago estimated that 2.5 exabytes (2.5×10 bytes) of data are created each day, and 90 percent of the data in the world was created in the past two years. Those numbers are likely higher today.

Big Data Infrastructure Considerations:

Traditional business data storage and management technologies include relational database management systems (RDBMS), network-attached storage (NAS), storage-area networks (SANs), data warehouses

(DWs), and business intelligence (BI) analytics. Traditional data warehouse and BI analytics systems tend to be highly centralized within an enterprise infrastructure. These often include a central data repository with a RDBMS, high-performance storage, and analytics software, such as online analytical processing (OLAP) tools for mining and visualizing data.

Increasingly, big data applications are becoming a source of competitive value for businesses, especially those that aspire to build data products and services to profit from the huge volumes of data that they capture and store. There is every indication that the exploitation of data will become increasingly important to enterprises in the years ahead as more and more businesses reap the benefits of big data applications.

Big Data Networking Example:

Key elements within the enterprise include the following:

- **Data warehouse:** The DW holds integrated data from multiple data sources, used for reporting and data analysis.
- **Data management servers:** Large banks of servers serve multiple functions with respect to big data. The servers run data analysis applications, such as data integration tools and analytics tools. Other applications integrate and structure data from enterprise activity, such as financial data, point-of-sale data, and e-commerce activity.
- **Workstations / data processing systems:** Other systems involved in the use of big data applications and in the generation of input to big data warehouses.
- **Network management server:** One or more servers responsible for network management, control, and monitoring.

Not shown in Figure 2.3 are other important network devices, including firewalls, intrusion detection/prevention systems (IDS/IPS), LAN switches, and routers.

The enterprise network can involve multiple sites distributed regionally, nationally, or globally. In addition, depending on the nature of the big data system, an enterprise can receive data from other enterprise servers, from dispersed sensors and other devices in an Internet of Things, in addition to multimedia content from content delivery networks.

The networking environment for big data is complex. The impact of big data on an enterprise's networking infrastructure is driven by the so-called three V's:

- Volume (growing amounts of data)
- Velocity (increasing speed in storing and reading data)
- Variability (growing number of data types and sources)

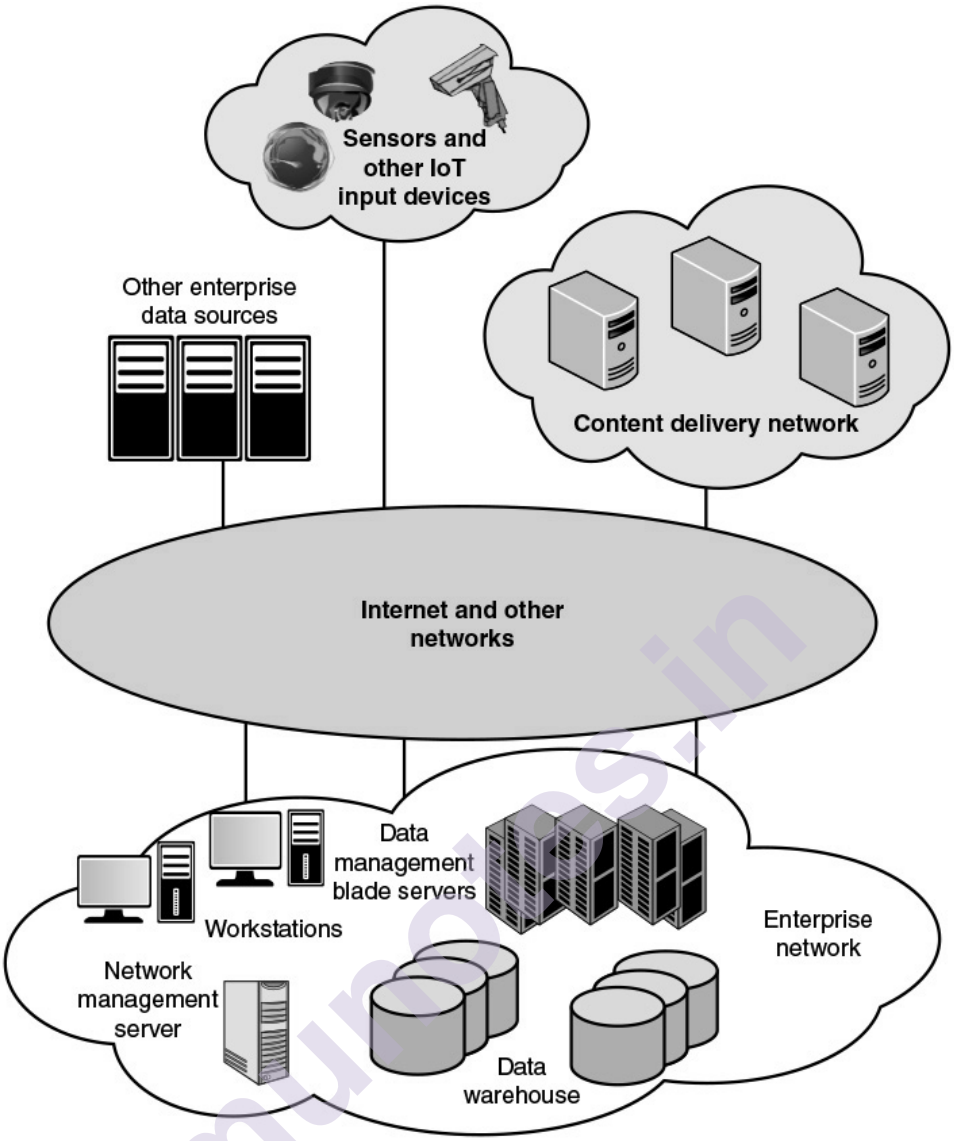


Figure 2.3 Big Data Networking Ecosystem

2.3.2 Cloud Computing:

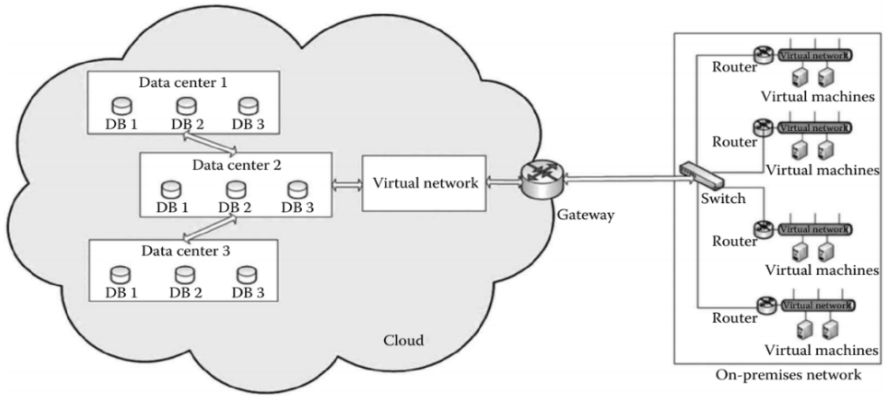


Figure 2.4 Cloud-based Network

A cloud-based network is an enterprise network that can be extended to the cloud shown in Figure 2.4. The cloud-based network allows an enterprise to distribute its network around the world. The cloud significantly simplifies the development of an enterprise network system. In the cloud, the underlying network is constructed by a cloud provider. All an enterprise needs to do is to connect its on-premises network to the network built in the cloud to form a global enterprise-class network system.

There is no initial capital investment in this type of global network system. Unlike the Internet, the cloud-based network provides centralized control over network visibility. Through the cloud-based network, the enterprise can provide a multitenant application, which is a software application that serves multiple tenants. Each tenant subscribes an instance of the application. Each tenant's data are isolated and remain invisible to other tenants. On the other hand, the maintenance and update of the application can be greatly simplified. The cloud-based network enables the enterprise to deploy IT infrastructures to remote locations in minutes.

The cloud-based network targets organizations with many sites around the world. There could be a couple of hundred to ten thousand employees working in multiple sites such as branch offices, schools in a school district, clinics, manufacturing facilities, or retail stores. Through the management tools deployed in the cloud, network administrators can manage the enterprise-distributed networks anywhere and anytime. The management tools can be used to manage cloud-hosted virtual machines and mobile services. They are used to accomplish tasks such as centralized management, remote monitoring, remote software and app installation, remote wiping, and security auditing.

There are three different major implementations of cloud computing. How organizations are using cloud computing is quite different at a granular level, but the uses generally fall into one of these three solutions.

Compute Clouds:
Compute clouds allow access to highly scalable, inexpensive, on-demand computing resources that run the code that they are given. Three examples of compute clouds are

- Amazon's EC2
- Google App Engine
- Berkeley Open Infrastructure for Network Computing (BOINC)

Compute clouds are the most flexible in their offerings and can be used for sundry purposes; it simply depends on the application the user wants to access. You could close this book right now, sign up for a cloud computing account, and get started right away. These applications are

good for any size organization, but large organizations might be at a disadvantage because these applications do not offer the standard management, monitoring, and governance capabilities that these organizations are used to. Enterprises are not shut out, however. Amazon offers enterprise-class support and there are emerging sets of cloud offerings like Terremark's Enterprise Cloud, which are meant for enterprise use.

Cloud Storage:

One of the first cloud offerings was cloud storage and it remains a popular solution. Cloud storage is a big world. There are already more than 100 vendors offering cloud storage. This is an ideal solution if you want to maintain files off-site. Security and cost are the top issues in this field and vary greatly, depending on the vendor you choose. Currently, Amazon's S3 is the top player.

Cloud Applications:

Cloud applications differ from compute clouds in that they utilize software applications that rely on cloud infrastructure. Cloud applications are versions of Software as a Service (SaaS) and include such things as web applications that are delivered to users via a browser or application like Microsoft Online Services. These applications offload hosting and IT management to the cloud.

Cloud applications often eliminate the need to install and run the application on the customer's own computer, thus alleviating the burden of software maintenance, ongoing operation, and support. Some cloud applications include

- Peer-to-peer computing (like Skype)
- Web applications (like MySpace or YouTube)
- SaaS (like Google Apps)
- Software plus services (like Microsoft Online Services)

2.3.3 Mobile Traffic:

Following the extraordinary peak in traffic growth seen in 2018 and the first part of 2019, the growth rate has returned to a more normal level. The quarter-on-quarter growth for Q1 2020 was 14 percent. A change in consumer behavior caused by COVID-19 lockdown restrictions impacted mobile networks by geographically shifting traffic loads; for example, daytime loads moved, to a degree, from city centers to suburban residential areas due to home-working guidance. This effect was most pronounced in areas with limited penetration of fixed residential broadband connections. Generally, the traffic volumes were only modestly affected in mobile networks in markets where fixed network connections

are common. Over the long-term, traffic growth is driven by both the rising number of smartphone subscriptions and an increasing average data volume per subscription, fueled primarily by more viewing of video content.

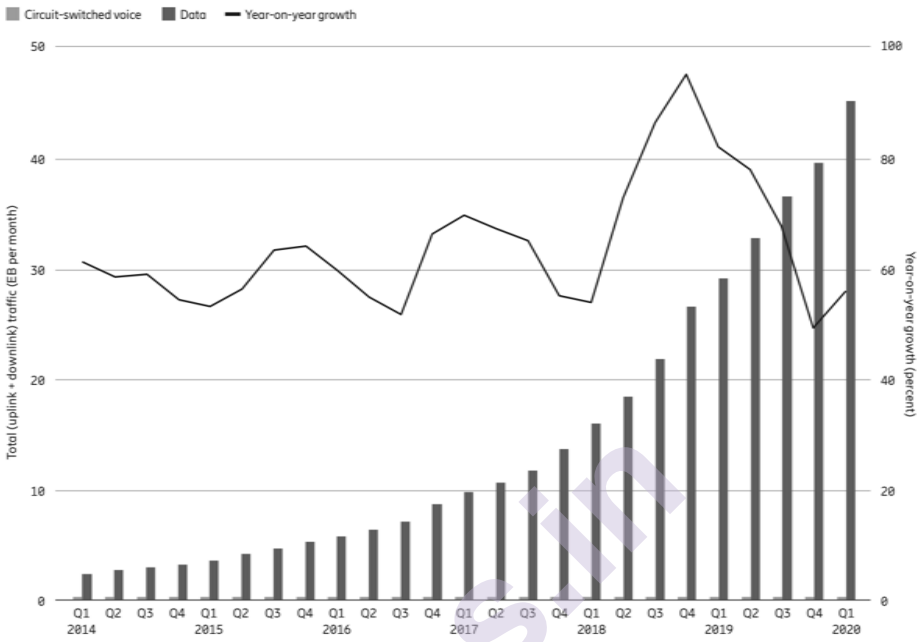


Figure 2.5: Global mobile network data traffic and year-on-year growth (exabytes per month)

Figure 2.5 shows total global monthly network data and voice traffic from Q1 2014 to Q1 2020, along with the year-on-year percentage change for mobile network data traffic. Mobile network data traffic depicted in Figure 2.5 also includes traffic generated by fixed wireless access (FWA) services and does not include DVB-H, Wi-Fi, or Mobile WiMAX. VoIP is included.

Global total mobile data traffic reached around 33 EB per month by the end of 2019 and is projected to grow by a factor close to 5 to reach 164 EB per month in 2025. Figure 2.6 represents the mobile data that will be consumed by over 6 billion people using smartphones, laptops, and a multitude of new devices at that time. (This graph does not include traffic generated by fixed wireless access (FWA) services)

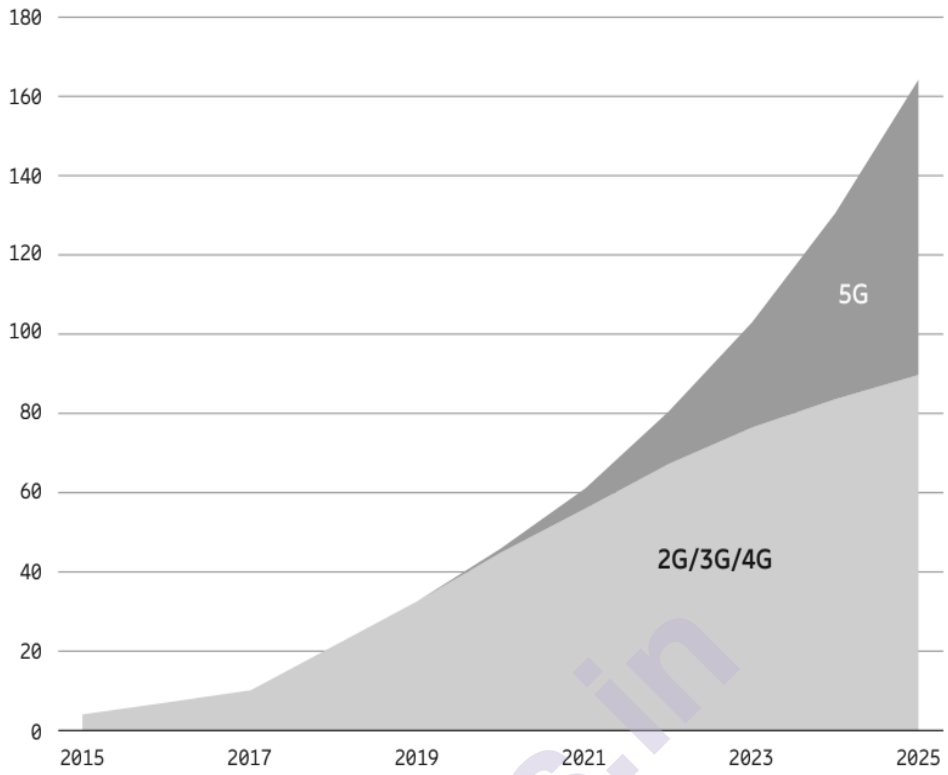


Figure 2.6: Global mobile data traffic (exabytes per month)

Smartphones continue to be at the epicenter of this development as they generate most of the mobile data traffic - about 95 percent, today, a share that is projected to increase throughout the forecast period. Populous markets that launch 5G early are likely to lead traffic growth over the forecast period. By 2025, we expect that 45 percent of total mobile data traffic will be carried by 5G networks.

Large variations in traffic growth across regions

Traffic growth can be very volatile between years, and can also vary significantly between countries, depending on local market dynamics. In the US, the traffic growth rate declined slightly during 2018 but recovered to previously expected rates during 2019. In China, 2018 was a year of record traffic growth. India's traffic growth continued its upward trajectory, and it remains the region with the highest usage per smartphone and per month. Globally, the growth in mobile data traffic per smartphone can be attributed to three main drivers: improved device capabilities, an increase in data-intensive content and more affordable data plans.

Around 410 million additional smartphone users are expected in India by 2025. In the India region, the average monthly mobile data usage per smartphone continues to show robust growth, boosted by the rapid adoption of 4G. Low prices for mobile broadband services, affordable

smartphones and people's changing video viewing habits have continued to drive monthly usage growth in the region. According to GlobalData, India Telecom Operators Country Intelligence Report (2019), only 4 percent of households have fixed broadband, making smartphones the only way to access the internet in many cases.

Total traffic is projected to triple, reaching 21 EB per month in 2025. This comes from two factors: high growth in the number of smartphone users, including growth in rural areas, and an increase in average usage per smartphone. A total of around 410 million additional smartphone users are expected in India by 2025. Even if the traffic per existing smartphone user continues to grow significantly over time, the increase in average traffic per smartphone is expected to moderate as more consumers in India acquire smartphones. The average traffic per smartphone is expected to increase to around 25 GB per month in 2025.

2.4 REQUIREMENTS: QOS AND QOE

The notion of Quality of Service has served as a central research topic in communication networks for more than a decade, however, usually starting from a rather technical view on service quality. Therefore, the notion of Quality of Experience has emerged, redirecting the focus towards the end-user, and trying to quantify user's subjective experience gained from using a service.

2.4.1 Quality of Service:

For at least a decade, Quality of Service (QoS) has been one of the dominating research topics in communication networks. Whereas the Internet originally has been conceived as a best-effort network, the introduction of QoS architectures like Integrated Services or Differentiated Services was supposed to pave the way for high-quality real-time services like Voice-over-IP or video streaming and thus to increase the competitiveness of packet-based TCP/IP networks.

Technology-centered approach:

The technology-centered approach mainly emphasizes the concept of QoS and has its strongest reference from the ITU (International Telecommunications Union).

The ITU Recommendation E.800 [i.31] is the key reference and states that QoS is the: "Totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service.".

Although the ITU definition refers to user satisfaction, QoS is mainly used by technicians to define technical parameters of

telecommunication applications such as network delay and packet loss. In addition, a focus on user satisfaction is rather limited because it is only one of many measures of user behavior with a communication service. For example, other measures include the time taken to perform a communication task (a measure of efficiency) and the accuracy with which a task is completed (a measure of effectiveness).

Commonly specified properties of QoS include:

- **Throughput:** A minimum or average throughput, in bytes per second or bits per second, for a given logical connection or traffic flow.
- **Delay:** The average or maximum delay. Also called latency.
- **Packet jitter:** Typically, the maximum allowable jitter.
- **Error rate:** Typically, maximum error rate, in terms of fraction of bits delivered in error.
- **Packet loss:** Fraction of packets lost.
- **Priority:** A network may offer a given number of levels of priority. The assigned level for various traffic flows influences the way in which the different flows are handled by the network.
- **Availability:** Expressed as a percentage of time available.
- **Security:** Different levels or types of security may be defined.

2.4.2 Quality of Experience:

There are different definitions of Quality of Experience across current ITU, ETSI and other literature. A research document by ETSI defined Quality of Experience (QoE) as: *"A measure of user performance based on both objective and subjective psychological measures of using an ICT service or product."*

NOTE 1: It considers technical parameters (e.g. QoS) and usage context variables (e.g. communication task) and measures both the process and outcomes of communication (e.g. user effectiveness, efficiency, satisfaction, and enjoyment).

NOTE 2: The appropriate psychological measures will be dependent on the communication context.

Objective psychological measures do not rely on the opinion of the user (e.g. task completion time measured in seconds, task accuracy measured in number of errors). Subjective psychological measures are based on the opinion of the user (e.g. perceived quality of medium, satisfaction with a service). For example, a service provider may conclude that a service with a certain level of QoS used for a particular

communication situation offers users excellent QoE, whist with a different level of QoS provides poor QoE.

For practical application, these features need to be converted to quantitative measures. The management of QoE has become a crucial concept in the deployment of future successful applications, services, and products.

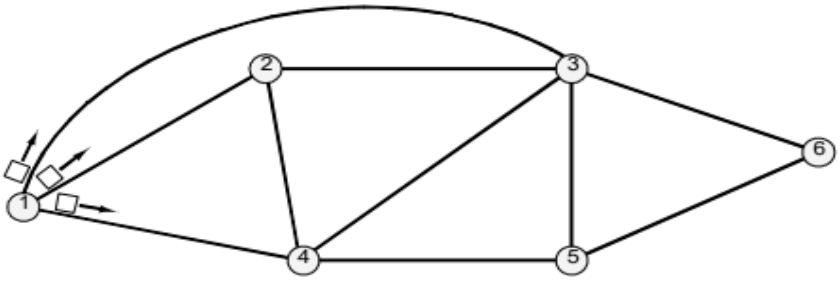
The greatest challenges in providing QoE are developing effective methods for converting QoE features to quantitative measures and translating QoE measures to QoS measures. Whereas QoS can now easily be measured, monitored, and controlled at both the networking and application layers, and at both the end system and network sides, QoE is something that is still quite intricate to manage.

2.5 ROUTING

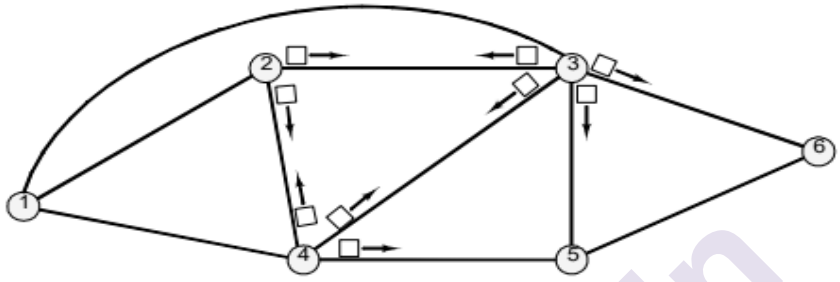
Routing and congestion control are the basic tools needed to support network traffic and to provide QoS and QoE. Both mechanisms are fundamental to the operation of a network and its capability to transmit and deliver packet traffic.

2.5.1 Characteristics:

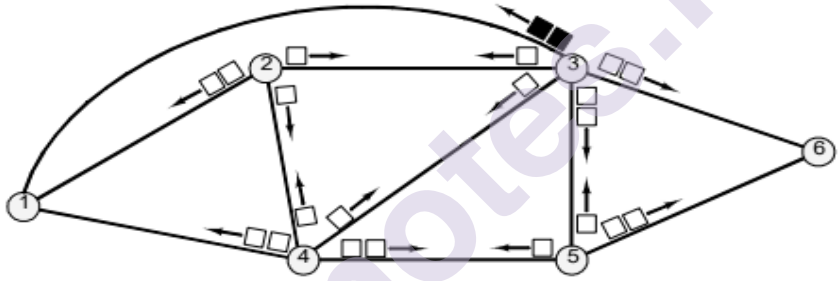
Routers forward packets from the original source to the destination. A router is considered a Layer 3 device because its primary forwarding decision is based on the information in the Layer 3 IP packet, specifically the destination IP address. This is known as routing and the decision is generally based on some performance criterion with the simplest one being minimum-hop route through the network.



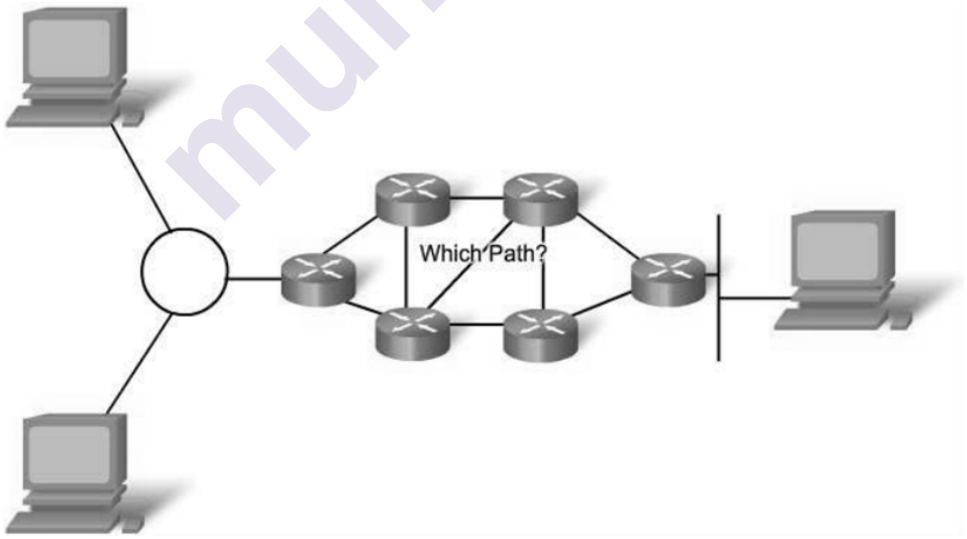
(a) First hop



(b) Second hop



(c) Third hop



(d) Hypothetical Network Architecture

Figure 2.7: Network Architecture Example

A generalization of the minimum-hop criterion is least-cost routing. In this case, a cost is associated with each link, and, for any pair of attached stations, the route through the network that accumulates the least cost is sought. Figure 2.7 illustrates a network with numbers circled are nodes and lines connecting them represent links between these nodes. The shortest path from node 1 to node 6 is node 1 to node 3 to node 6 (1-3-6).

2.5.2 Packet Forwarding:

The key function of any router is to accept incoming packets and forward them. For this purpose, a router maintains forwarding tables. A router's forwarding table shows, for each destination, the identity of the next node on the router. Each router may be responsible for discovering the appropriate routes. Alternatively, a network control center may be responsible for designing routes for all routers and maintaining a central forwarding table, providing each router with individual forwarding tables relevant only to that router.

Additional information is often used to determine the forwarding decision, such as the source address, packet flow identifier, or security level of the packet:

- **Failure:** When a node or link fails, it can no longer be used as part of a route.
- **Congestion:** When a particular portion of the network is heavily congested, it is desirable to route packets around rather than through the area of congestion.
- **Topology change:** The insertion of new links or nodes affects routing.

For adaptive routing to be possible, information about the state of the network must be exchanged among the nodes or between the nodes and a central controller.

2.6 CONGESTION CONTROL

Congestion occurs when the number of packets being transmitted through the network approaches the packet handling capacity of the network. Congestion control aims to keep number of packets below level at which performance falls off dramatically. It basically is reduced quality of service occurring when a network node or link is carrying more data than it can handle.

2.6.1 Effects of Congestion :

Typical effects of congestion include queueing delay, packet loss or the blocking of new connections.

Queueing Delay:

In telecommunication and computer engineering, the queuing delay or queueing delay is the time a job waits in a queue until it can be executed. In a switched network, queuing delay is the time between the completion of signaling by the call originator and the arrival of a ringing signal at the call receiver. Queuing delay may be caused by delays at the originating switch, intermediate switches, or the call receiver servicing switch. In a data network, queuing delay is the sum of the delays between the request for service and the establishment of a circuit to the called data terminal equipment (DTE). In a packet-switched network, queuing delay is the sum of the delays encountered by a packet between the time of insertion into the network and the time of delivery to the address.

This term is most often used about routers. When packets arrive at a router, they must be processed and transmitted. A router can only process one packet at a time. If packets arrive faster than the router can process them (such as in a burst transmission) the router puts them into the queue (also called the buffer) until it can get around to transmitting them. Delay can also vary from packet to packet, so averages and statistics are usually generated when measuring and evaluating queuing delay.

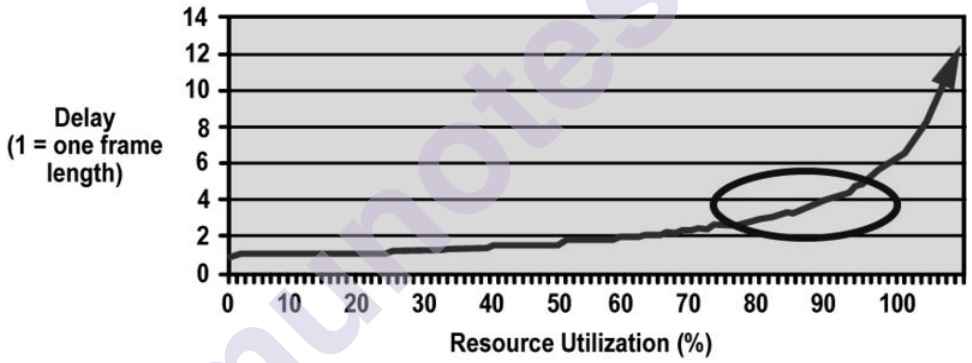


Figure 2.8: queuing delay based on throughput

Figure 2.8 displays the well-known effect of queuing delay based on throughput. As a queue begins to fill up due to traffic arriving faster than it can be processed, the amount of delay a particular packet experiences traversing the queue increases. The speed at which the contents of a queue can be processed is a function of the transmission rate of the facility. This leads to the classic "delay curve" depicted in the image to the right.

Packet Loss:

Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination. Packet loss is either caused by errors in data transmission, typically across wireless networks, or network congestion. Packet loss is measured as a percentage

of packets lost with respect to packets sent. In real-time applications like streaming media or online game, packet loss can affect a user's quality of experience (QoE). High packet loss rate indicates that users sustain undoubtedly a very poor quality.

Packet loss is also closely associated with quality of service (QoS) considerations. The amount of packet loss that is acceptable depends on the type of data being sent. A typical service level agreement (SLA) between Service Provider and Recipient (Company) could also mention a clause on Network packet delivery (reliability) as Average monthly packet loss shall be no greater than 0.1 percent (or successful delivery of 99.9 percent of packets). In such cases, packet loss could be defined as the percentage of packets that are dropped between backbone hubs on the Service Provider's Network.

Packet loss is detected by reliable protocols such as TCP. Reliable protocols react to packet loss automatically, so when a person such as a network administrator needs to detect and diagnose packet loss, they typically use status information from network equipment or purpose-built tools. The Internet Control Message Protocol provides an echo functionality, where a special packet is transmitted that always produces a reply. Tools such as ping, traceroute, and MTR use this protocol to provide a visual representation of the path packets are taking, and to measure packet loss at each hop. Many routers have status pages or logs, where the owner can find the number or percentage of packets dropped over a particular period.

Blocking of new connections:

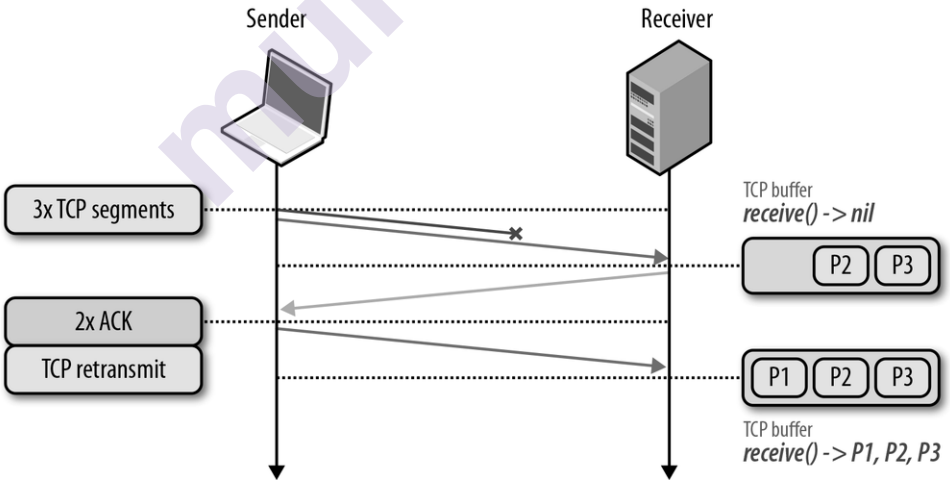


Figure 2.9: TCP Head-of-line blocking

2.6.2 Congestion Control Techniques:

Congestion control techniques can be broadly classified into two categories: open loop congestion control and closed loop congestion control.

A. Open loop congestion control:

Open loop congestion control policies are applied to prevent congestion before it happens. The congestion control is handled either by the source or the destination.

Policies adopted by open loop congestion control:

1. Retransmission Policy:

It is the policy in which retransmission of the packets are taken care. If the sender feels that a sent packet is lost or corrupted, the packet needs to be retransmitted. This transmission may increase the congestion in the network. To prevent congestion, retransmission timers must be designed to prevent congestion as also able to optimize efficiency.

2.Window Policy:

The type of window at the sender side may also affect the congestion. Several packets in the Go-back-n window are resent, although some packets may be received successfully at the receiver side. This duplication may increase the congestion in the network and making it worse. Therefore, Selective repeat window should be adopted as it sends the specific packet that may have been lost.

3.Discarding Policy:

A good discarding policy adopted by the routers is that the routers may prevent congestion and at the same time partially discards the corrupted or less sensitive package as also able to maintain the quality of a message. In case of audio file transmission, routers can discard less-sensitive packets to prevent congestion as also maintain the quality of the audio file.

4.Acknowledgment Policy:

Since acknowledgement are also the part of the load in network, the acknowledgment policy imposed by the receiver may also affect congestion. Several approaches can be used to prevent congestion related to acknowledgment. The receiver should send acknowledgement for N packets rather than sending acknowledgement for a single packet. The receiver should send an acknowledgment only if it must send a packet or a timer expires.

5.Admission Policy:

In admission policy a mechanism should be used to prevent congestion. Switches in a flow should first check the resource requirement of a network flow before transmitting it further. If there is a chance of a congestion or is a congestion in the network, router should deny establishing a virtual network connection to prevent further congestion.

All the above policies are adopted to prevent congestion before it happens in the network.

B. Closed Loop Congestion Control:

Following closed loop congestion control techniques are used to treat or alleviate congestion after it happens.

1. Backpressure:

Backpressure is a technique in which a congested node stop receiving packet from upstream node. This may cause the upstream node or nodes to become congested and rejects receiving data from above nodes. Backpressure is a node-to-node congestion control technique that propagate in the opposite direction of data flow. The backpressure technique can be applied only to virtual circuit where each node has information of its above upstream node.

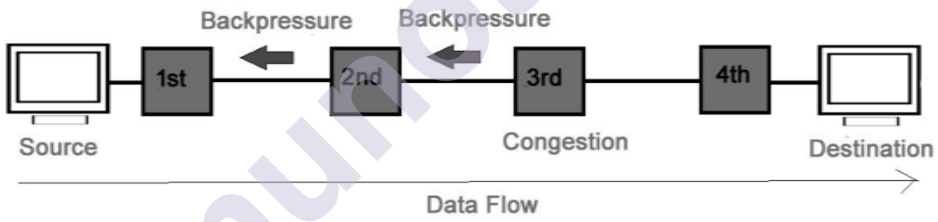


Figure 2.10 Backpressure

In figure 2.10, third node is congested and stops receiving packets as a result second node may be get congested due to slowing down of the output data flow. Similarly, first node may get congested and informs the source to slow down.

2. Choke Packet Technique:

Choke packet technique is applicable to both virtual networks as well as datagram subnets. A choke packet is a packet sent by a node to the source to inform it of congestion. Each router monitors its resources and the utilization at each of its output lines. whenever the resource utilization exceeds the threshold value, which is set by the administrator, the router directly sends a choke packet to the source giving it a feedback to reduce

the traffic. The intermediate nodes through which the packet has travelled are not warned about congestion.

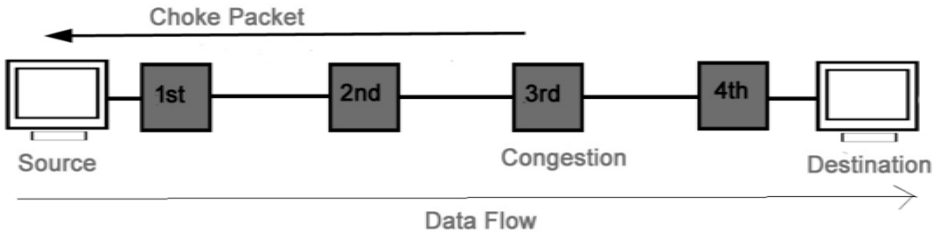


Figure 2.11 Choke Packet

3. Implicit Signaling:

In implicit signaling, there is no communication between the congested nodes and the source. The source guesses that there is congestion in a network. For example, when sender sends several packets and there is no acknowledgment for a while, one assumption is that there is a congestion.

4. Explicit Signaling:

In explicit signaling, if a node experiences congestion it can explicitly sends a packet to the source or destination to inform about congestion. The difference between choke packet and explicit signaling is that the signal is included in the packets that carry data rather than creating different packet as in case of choke packet technique.

Explicit signaling can occur in either forward or backward direction.

- **Forward Signaling:**

A bit can be set in a packet moving in the direction of the congestion. This bit can warn the destination that there is congestion. The receiver in this case can use policies, such as slowing down the acknowledgments, to alleviate the congestion.

- **Backward Signaling:**

A bit can be set in a packet moving in the direction opposite to the congestion. This bit can warn the source that there is congestion and that it needs to slow down to avoid the discarding of packets.

Explicit congestion signaling approaches could be divided into three general categories:

A. Binary:

A bit is set in a data packet as it is forwarded by the congested node. When a source receives a binary indication of congestion on a logical connection, it may reduce its traffic flow.

B. Credit based:

These schemes are based on providing an explicit credit to a source over a logical connection. The credit indicates how many octets or how many packets the source may transmit. When the credit is exhausted, the source must await additional credit before sending additional data. Credit-based schemes are common for end-to-end flow control, in which a destination system uses credit to prevent the source from overflowing the destination buffers, but credit-based schemes have also been considered for congestion control. Credit-based schemes are defined in Frame Relay and ATM networks.

C. Rate based:

These schemes are based on providing an explicit data rate limit to the source over a logical connection. The source may transmit data at a rate up to the set limit. To control congestion, any node along the path of the connection can reduce the data rate limit in a control message to the source.

2.7 SDN AND NFV

The Internet is the midst of a transformation, one that moves away from bundled proprietary devices, and instead embraces disaggregating network hardware (which becomes commodity) from the software that controls it (which scales in the cloud). The transformation is generally known as Software-Defined Networking (SDN).

2.7.1 Software-Defined Networking:

Software-defined networks provide an enhanced level of flexibility and customizability to meet the needs of newer networking and IT trends such as cloud, mobility, social networking, and video.

The SDN Architecture is:

- **Directly programmable**
  Network control is directly programmable because it is decoupled from forwarding functions.

- **Agile**
  Abstracting control from forwarding lets administrators dynamically adjust network-wide traffic flow to meet changing needs.

- **Centrally managed**
  Network intelligence is (logically) centralized in software based SDN controllers that maintain a global view of the network, which appears to applications and policy engines as a single, logical switch.

- **Programmatically configured**
  SDN lets network managers configure, manage, secure, and optimize network resources very quickly via dynamic, automated SDN programs, which they can write themselves because the programs do not depend on proprietary software.

- **Open standards-based and vendor-neutral**
  When implemented through open standards, SDN simplifies network design and operation because instructions are provided by SDN controllers instead of multiple, vendor-specific devices and protocols.

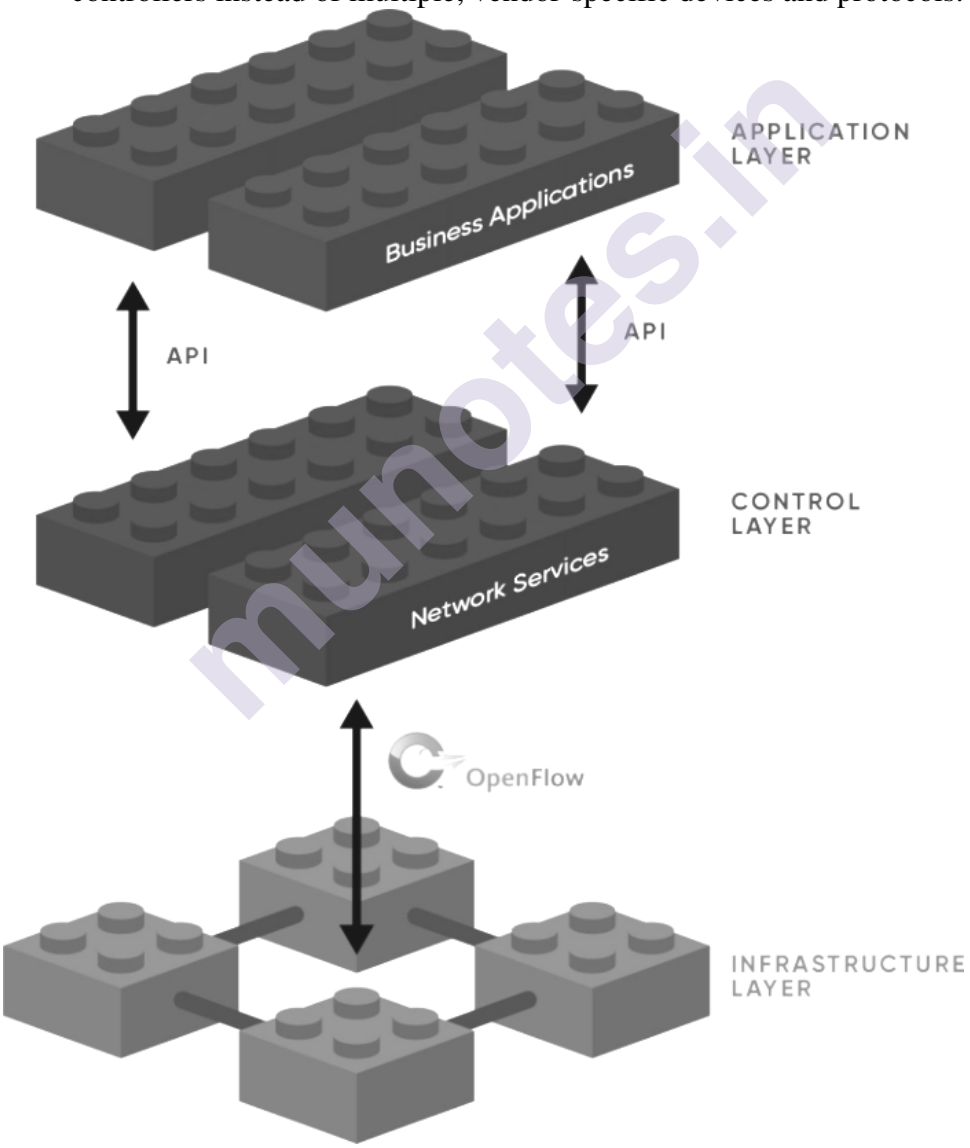


Figure 2.12 SDN Architecture

SDN Functionality:

The two elements involved in forwarding packets through routers are a control function, which decides the route the traffic takes and the relative priority of traffic, and a data function, which forwards data based on control-function policy.

Prior to SDN, these functions were performed in an integrated fashion at each network device (router, bridge, packet switch, and so on). Control in such a traditional network is exercised by means of a routing and control network protocol that is implemented in each network node. This approach is relatively inflexible and requires all the network nodes to implement the same protocols. With SDN, a central controller performs all complex functionality, including routing, naming, policy declaration, and security checks.

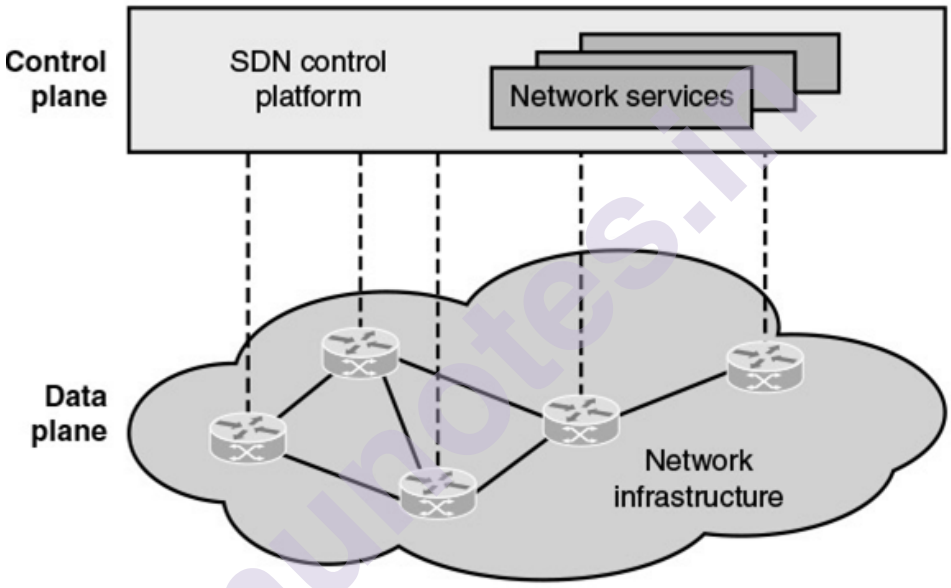


Figure 2.13 Software-Defined Networking

This constitutes the SDN control plane and consists of one or more SDN controllers. The SDN controller defines the data flows that occur in the SDN data plane. Each flow through the network is configured by the controller, which verifies that the communication is permissible by the network policy. If the controller allows a flow requested by an end system, it computes a route for the flow to take, and adds an entry for that flow in each of the switches along the path. With all complex function subsumed by the controller, switches simply manage flow tables whose entries can only be populated by the controller. The switches constitute the data plane. Communication between the controller and the switches uses a standardized protocol.

2.7.2 Network Functions Virtualization:

Virtual machine technology over the Internet or an enterprise network has been used for application-level server functions such as

database servers, cloud servers, web servers, e-mail servers, and so on. This same technology, however, can equally be applied to network devices, such as routers, LAN switches, firewalls, and IDS/IPS servers.

Network Functions Virtualization (NFV) decouples network functions, such as routing, firewalling, intrusion detection, and Network Address Translation from proprietary hardware platforms and implements these functions in software. It utilizes standard virtualization technologies that run on high-performance hardware to virtualize network functions. It is applicable to any data plane processing or control plane function in both wired and wireless network infrastructures. NFV has several features in common with SDN.

They share the following objectives:

- Move functionality to software
- Use commodity hardware platforms instead of proprietary platforms
- Use standardized or open application program interfaces (APIs)
- Support more efficient evolution, deployment, and repositioning of network functions

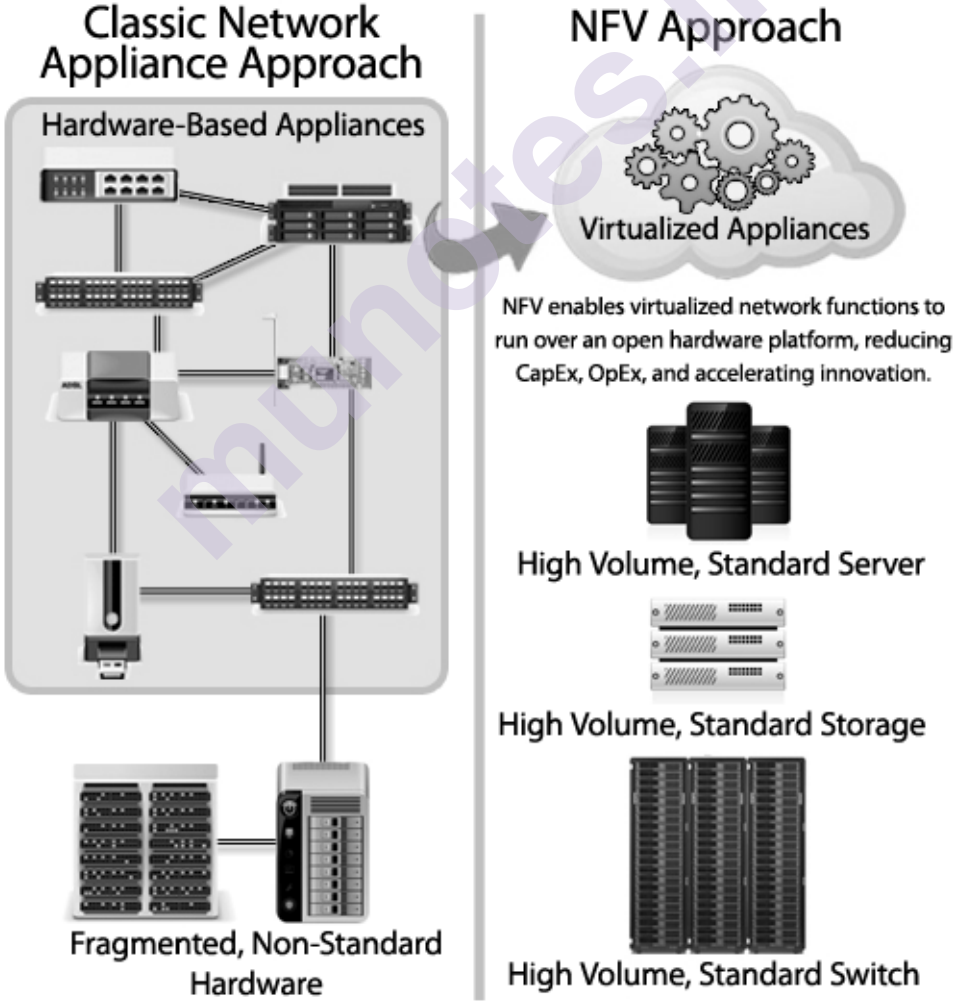


Figure 2.14 Network Functions Virtualization approach

NFV and SDN are independent but complementary schemes. SDN decouples the data and control planes of network traffic control, making the control and routing of network traffic more flexible and efficient. NFV decouples network functions from specific hardware platforms via virtualization to make the provision of these functions more efficient and flexible. Virtualization can be applied to the data plane functions of the routers and other network functions, including SDN controller functions. So, either can be used alone, but the two can be combined to reap greater benefits.

NFV reduces the need for dedicated hardware to deploy and manage networks by offloading network functions into software that can run on industry-standard hardware and can be managed from anywhere within the operator's network.

Separating network functions from hardware yields numerous benefits for the network operator, which include:

- Reduced space needed for network hardware
- Reduce network power consumption
- Reduced network maintenance costs
- Easier network upgrades
- Longer life cycles for network hardware
- Reduced maintenance and hardware costs

2.8 MODERN NETWORKING ELEMENTS

Ultimately, the concern of a network service provider is about the set of network devices (such as routers) and the control and management of the functions they perform (such as packet forwarding). If NFV is used, these network functions are implemented in software and executed on VMs. If instead the network functions are implemented on dedicated machines and SDN is used, the control functions are implemented on central SDN controllers, which interact with the network devices. However, SDN and NFV are not mutually exclusive.

If both SDN and NFV are implemented for a network, the following relationships hold:

- Network data plane functionality is implemented on VMs.
- The control plane functionality may be implemented on a dedicated SDN platform or on an SDN VM.

In either case, the SDN controller interacts with the data plane functions running on VMs.

QoS measures are commonly used to specify the service required by various network customers or users and to dictate the traffic management policies used on the network. The common case, until recently, is that QoS was implemented on network that used neither NFV nor SDN. In this case, routing and traffic control policies must be configured directly on network devices using a variety of automated and manual techniques. If NFV but not SDN is implemented, the QoS settings are communicated to the VMs. With SDN, regardless of whether NFV is used, it is the SDN controller that is responsible for enforcing QoS parameters for the various network users. If QoE considerations come into play, these are used to adjust QoS parameters to satisfy the users' QoE requirements.

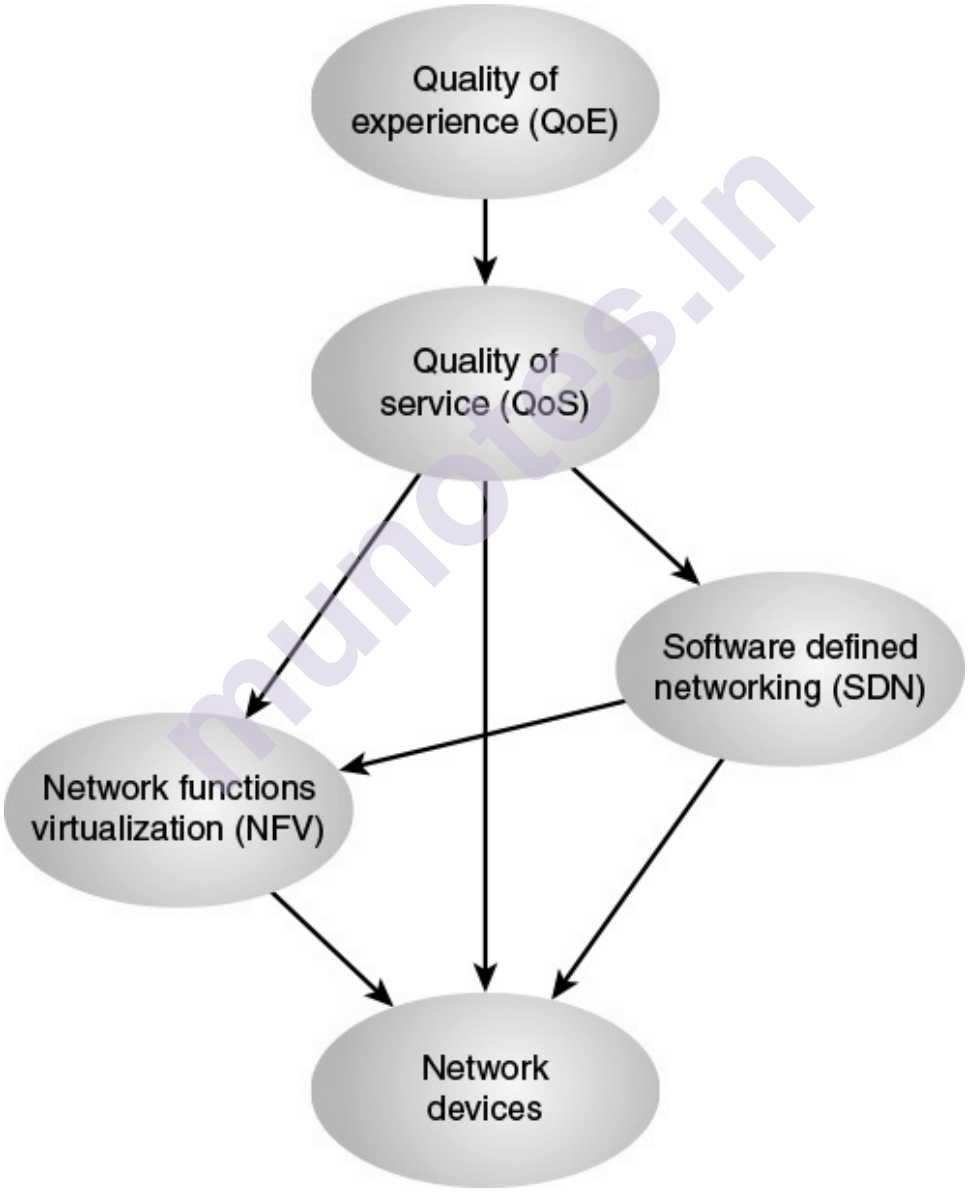


Figure 2.15 Modern Networking Schema

2.9 SUMMARY

- Elastic traffic is that which can adjust, over wide ranges, to changes in delay and throughput across an internet and still meet the needs of its applications
- Inelastic traffic does not easily adapt, if at all, to changes in delay and throughput across an internet
- big data refers to everything that enables an organization to create, manipulate, and manage very large data sets (measured in terabytes, petabytes, exabytes, and so on) and the facilities in which these are stored.
- Traditional business data storage and management technologies include relational database management systems (RDBMS), network-attached storage (NAS), storage-area networks (SANs), data warehouses (DWs), and business intelligence (BI) analytics.
- A cloud-based network is an enterprise network that can be extended to the cloud
- The technology-centered approach mainly emphasizes the concept of QoS and has its strongest reference from the ITU (International Telecommunications Union).
- Routing and congestion control are the basic tools needed to support network traffic and to provide QoS and QoE.
- Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination
- Congestion control techniques can be broadly classified into two categories: open loop congestion control and closed loop congestion control
- Software-defined networks provide an enhanced level of flexibility and customizability to meet the needs of newer networking and IT trends such as cloud, mobility, social networking, and video.
- Network Functions Virtualization (NFV) decouples network functions, such as routing, firewalling, intrusion detection, and Network Address Translation from proprietary hardware platforms and implements these functions in software.

2.10 UNIT END QUESTION

1. Present an overview of the major categories of packet traffic on the Internet and internets, including elastic, inelastic, and real-time traffic.
2. Discuss the traffic demands placed on contemporary networks by big data, cloud computing, and mobile traffic.
3. Explain the concept of quality of service.
4. Explain the concept of quality of experience.

5. Understand the essential elements of routing.
6. Understand the effects of congestion and the types of techniques used for congestion control.
7. Compare and contrast software-defined networking and network functions virtualization.
8. What is congestion? Why does it occur?
9. What is choke packet? How is it used for congestion control?

2.11 REFERENCES

1. Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud, First printing: November 2015 by William Stallings, Copyright © 2016 by Pearson Education, Inc. ISBN-13: 978-0-13-417539-3
2. Research paper "On Traffic Types and Service Classes in the Internet" by Mansour J. Karam, Fouad A. Tobagi, available online at http://mmnetworks.stanford.edu/papers/karam_globecom00.pdf
3. Center for Applied Internet Data Analysis (CAIDA) research material on Internet Traffic Classification published and available online at https://www.caida.org/research/traffic-analysis/classification-overview/
4. Cloud Computing Networking Theory, Practice, and Development by Lee Chao, © 2016 by Taylor & Francis Group, LLC. ISBN-13: 978-1-4822-5482-2
5. Cloud Computing: A Practical Approach by Anthony T. Velte Toby J. Velte, Robert Elsenpeter, Copyright © 2010 by The McGraw-Hill Companies. ISBN-13: 978-0-07-162695-8
6. Mobile network traffic Q1 2020 and Mobile data traffic outlook sections adopted from Ericsson Mobility Report June 2020, Publisher: Fredrik Jejdling, available online at https://www.ericsson.com/49da93/assets/local/mobility-report/documents/2020/june2020-ericsson-mobility-report.pdf
7. Research paper "From Quality of Service to Quality of Experience" by Markus Fiedler, Kalevi Kilkki and Peter Reichl published online at https://drops.dagstuhl.de/opus/volltexte/2009/2235/pdf/09192.SWM.Paper.2235.pdf
8. Technical Report on "Human Factors (HF); Quality of Experience (QoE) requirements for real-time communication services" by ETSI, © European Telecommunications Standards Institute 2009, available online at

https://www.etsi.org/deliver/etsi_tr/102600_102699/102643/01.00.01_60/tr_102643v010001p.pdf

9. Technical Article on Queuing delay, published by hill associates, archive available online at https://web.archive.org/web/20150904041151/http://www.hill2dot0.com/wiki/index.php?title=Queuing_delay
10. Technical Article on Queuing delay, published by Wikipedia available online at https://en.wikipedia.org/wiki/Queuing_delay
11. Technical Article on Packet Loss, published by Wikipedia available online at https://en.wikipedia.org/wiki/Packet_loss
12. Chapter on Building Blocks of TCP, published by O'Reilly at https://www.oreilly.com/library/view/high-performance-browser/9781449344757/ch02.html
13. Computer Network: Lecture Notes, prepared by Mr. Daya Ram Budhathoki, available online at https://dayaramb.files.wordpress.com/2011/03/computer-network-notes-pu.pdf
14. SDN Architecture https://opennetworking.org/sdn-definition/
15. Network Functions Virtualizations approach available online at: https://www.blueplanet.com/resources/What-is-NFV-prx.html

*****

UNIT II

3

SDN: BACKGROUND AND MOTIVATION AND SDN DATA PLANE AND OPENFLOW

Unit Structure

3.0 OBJECTIVES

- Present the view that traditional network architectures are insufficient to meet modern networking requirements.
- List and explain main SDN architecture specifications.

- Present an SDN architecture summary to clarify the significance of northbound and southbound APIs.
- Summarize work on standardization of SDN and NFV by various organizations.

The review of SDNs starts throughout this chapter, and gives some context to and motivation for the SDN approach.

3.1 EVOLVING NETWORK REQUIREMENTS

Network providers and clients are required to redefine existing network architecture approaches. Such developments can be divided into demand, supply and traffic pattern categories.

3.1.1 Demand Is Increasing|:

The demand on corporate network, the internet and other internets is growing in a variety of trends. The following are especially noteworthy:

- **Cloud computing:** Organizations have shifted significantly to both Cloud services in public and private**.**
- **Big data:** The processing of massive sets of data requires major parallel Processing on numbers of servers needing a high degree of interconnection between them. There is therefore a massive and ever-growing demand for data center network capabilities.
- **Mobile traffic:** Increasingly employees access the company network resources through a mobile devices like as mobile phones, tablets and other services Laptops. Such devices requires an complex technologies that process and creates images and videos traffic and put new pressures on the company networks
- **The Internet of Things (IoT):** the majority of IoT "things" are create a modest traffic although there are exceptions including surveillance video cameras. But it's a matter of fact. The large volume of these devices leads in a large number of companies load to the network of the company.

3.1.2 Supply Is Increasing:

With an extraordinary demand on usability of network is increasing day by days, so the capacity of the networking bandwidth load is consuming significantly .we have seen in above chapter briefly introduce the key technologies for wired and wireless technologies, Ethernet and Wi-Fi respectively having sending data in terms of Gigabit per seconds in range. Likewise for wireless cellular networks 4G and 5G provides huge capacity for mobile devices accessing the enterprise network by remote employees instead of Wi-Fi.

So increase in the network transmission technologies capacity has to match with the increase capacity in the performance of the network devices such as LAN Switches, routers, firewalls, intrusion detection system/intrusion prevention systems (IDS/IPS), monitoring and management of networks systems. Year after year such machines becomes very larger, having faster memories capacity, which allows more buffer space, faster buffer access, and faster processors. Speed

3.1.3 Traffic Patterns Are More Complex:

It would seem that today's networks will be capable of dealing to data traffic presently, merely by coping with supply and demand. However, as the patterns of traffic have changed, conventional network architectures become gradually inadequate for demand. The conventional corporate network architecture since currently yet still popular these days was a local or campus-wide system of Ethernet switches with routers connecting wide Ethernet LANs with Internet and WAN connections. This architecture is suitable for the computing model client / server where in prominent in the business environment at one time. The interaction and traffic with this model was primarily between a one client and a server. In this environment it is possible to build and configure networks with largely constant client and server locations and reasonably predictable amounts of traffic between clients and servers.

A number of developments have resulted in far more dynamic and complex traffic patterns within the enterprise data center, local and regional enterprise networks, and carrier networks. They contain the following:

- Applications of clients / server usually have access to multiple servers and databases which must communicate among themselves and generate "horizontal" traffic among the Servers and "vertical" traffic between the clients and servers.
- Voice, data and video of network convergence produces unexpected outcomes traffic patterns, also massive data transfers of multimedia applications.
- Substantially use unified communications (UC) techniques, drives multiple server access mechanism.
- The high use of mobile devices which includes personal bring your own device (BYOD) policies, that allow users access to company features and applications t from any device anywhere at any time. This mobile traffic is an increasingly important component of Corporate Traffic Network.
- The common use of public clouds changed a great deal of local traffic was previously on WANs for several businesses and this led to Increased loads on enterprise routers which are often very unexpected.

- The now traditional application method and virtualization of the database server .The amount of hosts needing high-volume network access increased dramatically and resulted in any physical change in the server resources location.

3.1.4 Traditional Network Architectures are Inadequate:

While there is growing capacity in transmission networks and the improved efficiency of network equipment, with in context of the increasing complexity, variability and high volume enforced on the demand, conventional network architectures are ultimately become inadequate. However, as network's quality of service (QoS) and quality of experience (QoE) needs are extended across the various applications, the traffic load has to be handled increasingly advanced and flexible.

The conventional network design is focused on the architecture of the TCP / IP protocol. The following are three important aspects of this approach:

- Two-level end system addressing
- Routing based on destination
- Distributed, autonomous control

In addition, let's look at each of these functions.

The conventional layout is firmly focused on the identity of the network interface. Devices connecting via the networks may be recognized through hardware-based identifications, including Ethernet MAC addresses, on the physical layer of the TCP/IP model. The framework is a network of networks, including the Internet and private internet. Each connected device has its own physical layer identifier which is identifies by neighbor network and its logical IP address.

The TCP / IP architecture facilitates communication of independent networks with distributed control using this addressing scheme. Such model has strong flexibility and scales for the addition of new networks. Routes may be found and accessed on the Internet using IP and distributed routing protocols. Protocols of transport layers, like TCP, may be used to build distributed and decentralized algorithms with respond to congestion.

Routing was generally focused on the destination address of each packet. The datagram method, subsequent packets across the source and destination that take different internet routes, as routers continuously try to find the shortest delay path for each packet. Most importantly, packets are also handled with flows of packets in order to meet the QoS criteria. The QoS characteristics associated with a given flow have been established for the routing of the whole flow.

Nevertheless, this distributed, independent methodology evolved while networks are largely fixed-location of static and end systems. Dependent on such features, the Open Networking Foundation (ONF) addresses following four constraints on traditional network design

Static, complex architecture: Until date, networking technologies consisted primarily of discrete protocol sets designed to reliably connect hosts over unspecified lengths, connection speeds, and topologies. The industry has developed networking standards to provide better efficiency and durability, broader connectivity, and stronger protection to satisfy technological and business needs over the past decades.

Protocols appear, though, to be described in isolation with each problem solved and without fundamental abstracts. This has led to one of the main constraints of today's networks: complexity. For example, IT will reach a number of switches, routers, firewalls, Web authentication portals, etc., and upgrade device-level management tools to add or transfer devices; ACLs, VLANs, quality of services (QoS); as well as other protocol-based mechanisms. Furthermore, network topology, vendor switch model and software version must be considered. Despite of this difficulty, networks today are largely static, with IT striving to reduce the possibility of disruption.

The static characteristics of networking differ significantly with those of the dynamic characteristics of today's server environments, where the amount of hosts needing network access has grown dramatically and host physical circumstances have changed fundamentally.

Applications were residing on a single server until virtualization and mainly shared traffic with selected clients. Currently applications are distributed through several virtual machines, exchanging traffic between them. VMs migrate to optimize and rebalance server workloads and make adjustments over time at the physical end points of existing flows. VM migration questions many facets of conventional networking from the implementation of schemes and namespaces, to the basic principle of a segmented architecture based on routing.

In addition to adoption of virtualization technology, several organizations today have an IP-converged voice, data and video traffic network. Although existing networks can deliver differentiated QoS rates for various applications, they deliver those resource are highly procedural. IT will independently configure the equipment of each vendor and change parameters like as network bandwidth and QoS on a per-session, per-application basis. The network cannot adapt rapidly to changing traffic, device and user requirements because of its static nature.

Inconsistent policies: IT can need to configure thousands of devices and mechanisms for implementing a network-wide policy. For instance, it can require hours or days for IT to reconfigure ACLs along all the whole network every time you build a new virtual machine. The complexity of existing networks makes it quite difficult for IT to apply to increasingly mobile users a consistent set of access, security, QoS and other policies, making it vulnerable to security breaches, failures to comply with regulations, and other negative consequences.

Inability to scale: The network needs to expand as demands on the data center rise rapidly. But, by incorporating hundreds or thousands of networks the network is much more complex that must be configuring and managing devices. With today's virtualization data centers traffic are dynamics and extremely complicated and often unpredictable. It was therefore relied upon to oversubscribe to scale the network by connecting it to a predetermined traffic pattern. Mega players like Yahoo, Google! and Facebook ,they even face the scalability problems which are even more challenges. Such providers use massively parallel algorithms and related datasets in large-scale in their computing pool. The number of computing elements increases as end-user applications expand (for example, crawling and indexing the whole world wide web to automatically give users search outcomes).

An exchange of data between computer nodes can be possible between petabytes. These companies need so-called hyperscale networks which can provide hundreds of thousands — possibly millions — of physical servers with high-performance low-cost connectivity. The manual setup cannot allow such scaling.

To order to stay competitive, carriers have to provide consumers with more and more value-added services. Multi-tenancy makes their job more difficult as the network needs to accommodate groups of users with particular software and performance criteria. The implementation with existing networks, particularly at carrier level, involves very complex main operations that seem fairly straightforward as to directing the traffic flows of the customer for the distribution or on-demand results. Different devices at the edge of the network are required, thus increasing capital and business investment as well as time to market to incorporate new services.

Vendor dependence: Carriers and companies seek to implement new capabilities and services to respond quickly to changing business requirements or customer requirements. However, the vendors' product cycles of equipment, ranging to or exceeding three years, hinder their ability to respond. The absence of standard, open interfaces reduces the network's ability to tailor the network to their respective environments.

The industry has come to a tipping point with this division between market requirements and network capacity. The industry has therefore designed and has developed the Software Defined Networking (SDN) architecture.

3.2 THE SDN APPROACH

This section provides a description of SDN and demonstrates how it is built to achieve network requirements. The Open Data Center Alliance (ODCA) offers the following useful and concise list of requirements.

- **Adaptability:** networks will rapidly adapt and react to requirements, depend upon corporate policy and network conditions.
- **Automation:** Policy modifications need to be propagated continuously in order to so it may be possible to reduce manual work and errors.
- **Maintainability:** New features and capabilities are introduced (software upgrades, patches) should be smooth and operating disruption minimal.
- **Model management:** the software for Network Management should enable network management at the model level, instead of re-configuring the network elements to implement conceptual changes.
- **Mobility:** Mobility must be controlled including mobile user devices and virtual servers.
- **Integrated security:** Network applications have to integrate seamless security as a basic service rather than as an add-on.
- **On-demand scaling:** Implementations must be equipped to scale up or down the network and its facilities in support of on-demand request.

3.2.1 SDN Architecture:

As shown in figure 3.1, both the control plane and the data plane are attached to the proprietary hardware in the traditional networks. The key feature of a dedicated appliance network is 'dedicated appliance' that refers to one or more switches, routers and/or application delivery controllers. Inside this appliance several of the functionality are implemented in hardware only and are commonly used with Application Specific Integrated Circuit (or: ASIC)

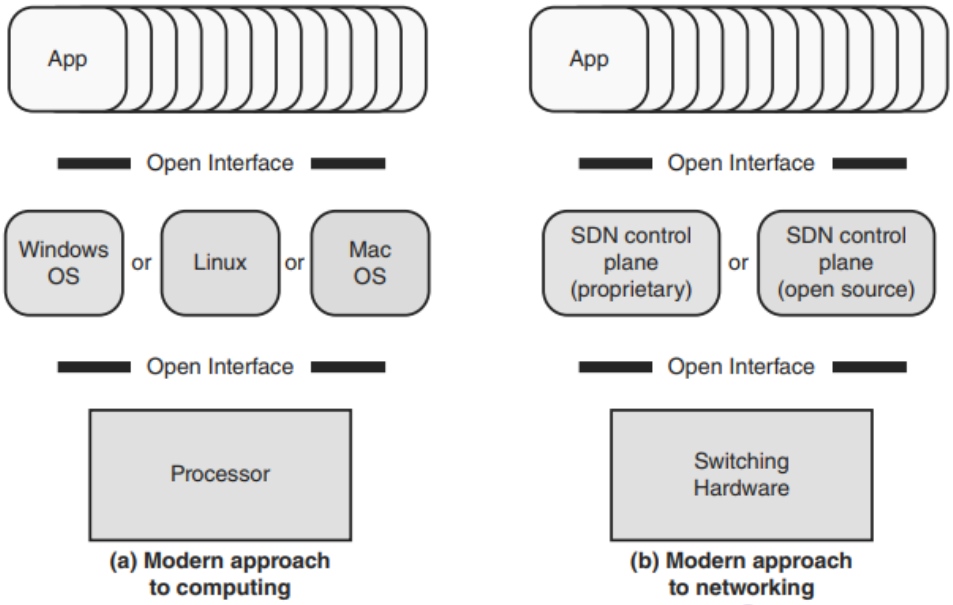


FIGURE 3.1 a) Traditional networks b) SDN approach is implemented

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

In the initial times of computing , vendors such IBM and DEC supplied a completely integrated package, containing proprietary processors' hardware, a specific assembly language, a specific OS and the majority, if not all, applications for the program. In this environment, consumers, in particular large clients, were typically locked in a single vendor which was primarily dependent on the applications of the vendor. Transition to the hardware platform of another vendor led to big catastrophes of the application. The computer environment is today distinguished by total openness and tremendous flexibility for consumers. Most device equipment is made up of x86 and x86-compatible processors in embedded systems for isolated systems and ARM processors. The above makes it much easier to port operating systems build using C, C++, Java and similar language. However proprietary hardware architectures like the IBM enterprise line have standard compilers and programming frameworks, and like Linux can conveniently run from open sources operating systems. Thus, Linux or other open operating software may be switched from one vendor platform to another with ease. Also proprietary platforms including Windows and Mac OS have programming environments for easy application porting. It's likewise enables the creation of virtual machines that can be transferred through hardware and operating systems from one server to another.

In the pre-open era of computing, the current networking environment seems to have a few of the identical limitations. Thus, developing applications that can operate on multiple platforms are not

important issue. Actually, the problem is that the applications and network infrastructure are not compatible each other. The traditional network designs, as seen in the previous section, are insufficient to satisfy rising volumes and traffic diversity.

It is the main concept of SDN's goal clearly to allow developers are able to control over x86 servers for many years and bring the latest developments in cloud computing to the networking industry The method SDN separates the switching function between the data plane and the control plane on individual devices (see Figure 3.2). It is simply the data plane which transmits packets and the control plane provides "intelligence" in designing routes, setting priority parameters that meet QoS and QoE needs and follow changing traffic trends. Open interfaces are specified to ensure that the switching hardware provides a compatible design, irrespective of internal implementation specifics. In turn, open interfaces are set up to connect with networking systems via the SDN controllers.

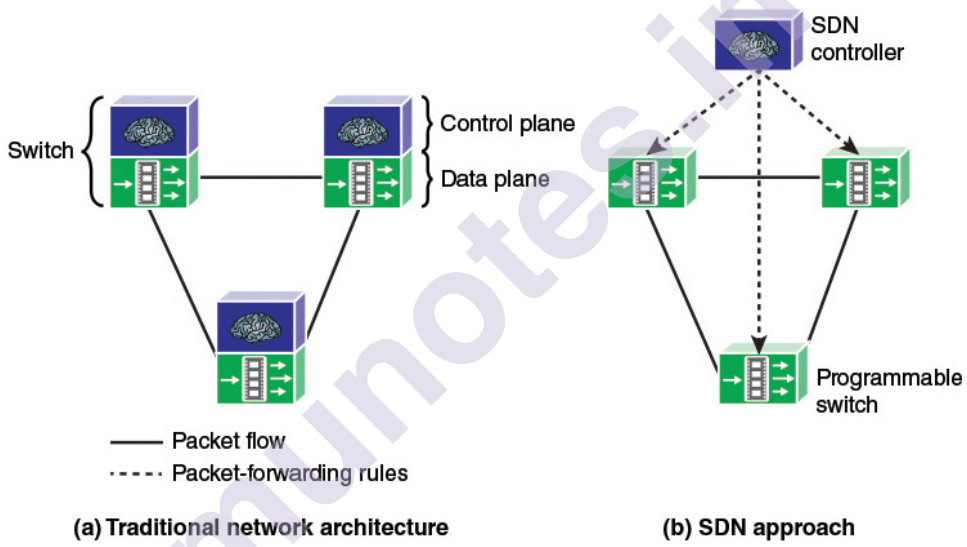


FIGURE 3.2 The Modern Approach to Computing and Networking

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

Figure 3.2 describes the SDN approach in greater depth. SDN is a three-layer architecture; the control layer, which consists of a controller and acts like a brain for the network because the controller manages the traffic flow from the switches using the flow tables.

3.2.2 Characteristics of Software-Defined Networking:

Programmability is Direct: Since the network control is explicitly programmable from the forwarding functions.

Agility: modification of diverse network traffic dynamically to satisfy evolving network requirements is simple.

Management of network is central: Network management is central: Network intelligence is (logically) centralized on software-based SDN controllers that retain an overall network perspective and appears as a single logical switch for applications or policy engines.

Configuration is programmable: SDN lets the network administrators secure, configure, manage and optimize network resources quite easily by dynamic, automated SDN programs so they can write them on their own as proprietary software is not dependent anymore.

Open standards-based and no more vendor-dependency : Open standards implementing SDN make network design and operations easy as the majority of the instructions are provided with SDN controllers (such as POX, Ryu, OpenDaylight etc), rather than Multiple Provider-specific protocols. Both three layers are mutually dependent and communicate via some interfaces. SDN architecture offers an overview perspective of the whole network for the applications it serves; this allows the network much more "smart."

SDN Architecture contains the following three layers

Application Layer: The application layer is made up of applications which communicate with control layer controllers via certain interfaces, called Northbound APIs. The most used API is the REST (representation State Transfer) API for providing Northbound APIs. Applications in SDN can be like Firewall, Load balancer etc.

Control Layer: This is the middle layer of the SDN architecture which is the SDN controller that serves like a network's brain which provides a broad overview of a network defined as the control plane. It is necessary to deploy SDN controllers directly on a server or on a virtual server. For manage the switches in the data plane, Open Flow or some other open API is used. However, controllers using information regarding the capacity and demand generated from the traffic networking equipment which the traffic flows. SDN controllers often open northbound APIs that enable emerging companies and network administrators, many of them untenable previous to SDN's introduction, to deploy a broad range of off-the-shelf and personalized network applications. No standardized northbound API or agreement for an open northbound API is currently available. Several vendors offer a REpresentational State Transfer (REST) API to provide their SDN controller with a programmable interface.

Physical Layer: includes equipment such as switches, also known as Data Plane, used in the network. They include forwarding and switching of packets. Switches only perform the actions according to the controller.

The interface they use to communicate with a control layer is Southbound APIs. OpenFlow Protocol is the most common protocol used to provide Southbound API.

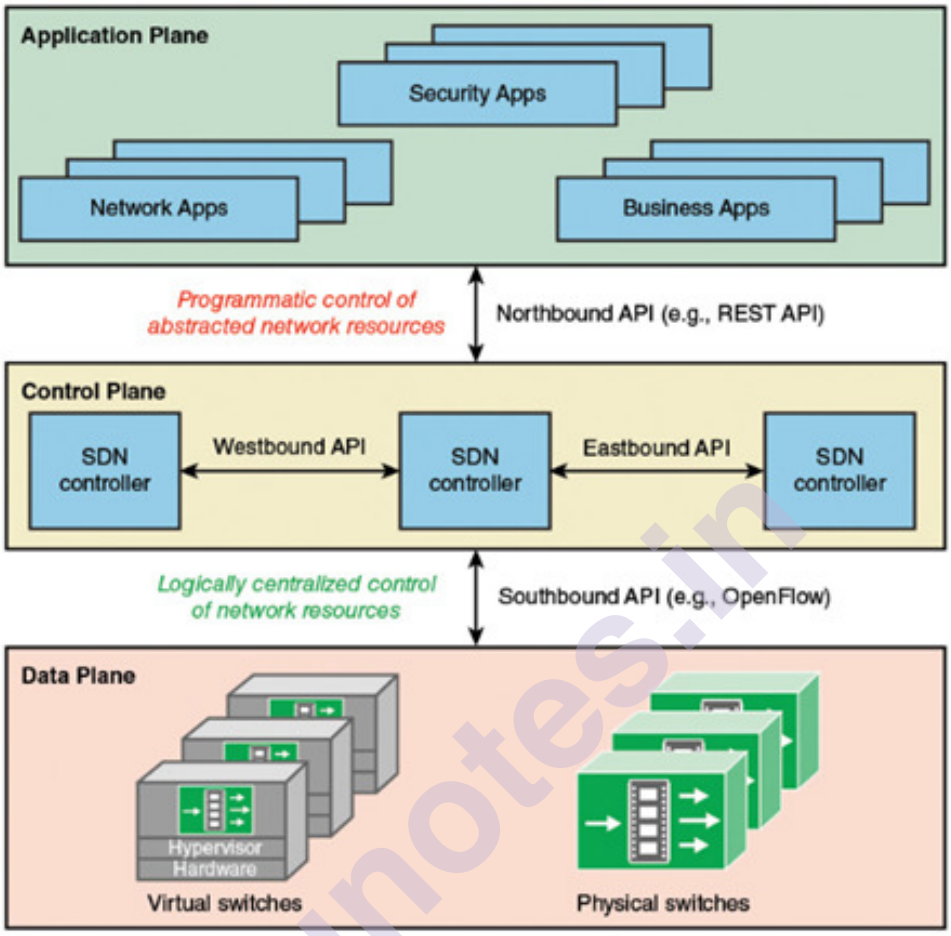


FIGURE 3.3 Software-Defined Architecture

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

3.3 SDN- AND NFV-RELATED STANDARDS

Like other fields of technology, such as Wi-Fi, there is no unified set of guidelines responsible for SDN and NFV to build open standards. It's a wide and evolving collection of SDOs, industrial consortia and open development initiatives involved in developing SDN and NFV standards and guidelines.

Table 3.1 summarized the main SDOs and other organizations participating in the efforts and the main results produced so far. This section deals with some of the most important efforts.

TABLE 3.1 SDN and NFV Open Standards Activities

Organization	Mission	SDN- and NFV-Related Effort
Open Networking Foundation (ONF)	This is a user-driven non-profit organization devoted to speeding up the deployment of SDN and NFV. In 2011, the ONF really powered the SDN movement; it involves companies including Deutsche Telekom, Facebook , Twitter, Microsoft, Verizon and Yahoo! aimed at standardizing SDN and OpenFlow. In 2017, ONF unveiled an Open Innovation pipeline that will lead the industry into the next SDN and NFV generation. Disaggregation of network devices, open source platforms and standard software are primary priorities.	OpenFlow
Internet Engineering Task Force (IETF)	SDN (via RFC 7426 SDN) and a new IETF SDN Standards Group (I2RS) work together to focus on southbound programming and NFV and network service chains.	Interface to routing systems (I2RS) Service function chaining
European Telecommunications Standards Institute (ETSI)	An EU-sponsored standardization organization that develops internationally applicable standards for information and communications technologies.	NFV architecture
OpenDaylight	The OpenDaylight Project is a collaborative open source project hosted by The Linux Foundation	OpenDaylight
International Telecommunication	UN agency that provides guidelines for global	SDN functional requirements and

Union — Telecommunication Standardization Sector (ITU-T)	standardization in telecommunications	architecture
Internet Research Task Force (IRTF) Software Defined Networking Research Group (SDNRG)	Research group within IRTF. Produces SDN-related RFCs	SDN architecture
Broadband Forum (BBF)	The Broadband Forum is a non-profit industry consortium dedicated to developing broadband network specifications	SDN requirements and framework of broadband telecom networks
Metro Ethernet Forum (MEF)	Industry consortium supporting Ethernet use for metropolitan and wide-area applications.	Defining APIs for SDN and NFV service orchestration
IEEE 802	An IEEE committee responsible for the development of LAN standards.	Standardize SDN communication network capabilities.
Optical Internetworking Forum (OIF)	Consortium for industry to promote the development and implementation of interoperable networking products and services.	Transport network requirements of SDN architectures
Open Data Center Alliance (ODCA)	Leading IT companies consortium build interoperable cloud solutions and services.	SDN usage model
Alliance for Telecommunications Industry Solutions (ATIS)	A standard organization that establishes unified communications (UC) standards.	SDN / NFV programmable infrastructure to technical opportunities and challenges
Open Platform for NFV (OPNFV)	OPNFV created a reference platform through system-level integration, deployment, and testing, to accelerate the transformation of enterprise and service provider networks.	NFV infrastructure

3.3.1 Standards-Developing Organizations:

The Internet Community, ITU-T and ETSI make significant contributions to SDN and NFV standardization.

Internet Society:

In the past few years, the next major trend in networking has been the software-defined networking (SDN) and Network functions virtualization (NFV). As a consequence, we saw networking standards development organizations (SDOs) such as ITU, IETF and TMF leap in the bandwagon to comply with SDN and NFV. The two groups most involved in the internet society (ISOC) are: IETF and IRTF. The ISOC is an international non - profit organization which manages the Internet, education and policy development standards. Founded in 1992, the aim of ISOC is to promote open Internet development for organizations and individuals all over the world through enhancement and promotion of internet usage.

The actual work of standards development and publication is being carried out by a variety of organizations under the ISOC.

The Task Force for Internet Engineering (IETF) has SDN working groups in the following areas:

Interface to routing systems (I2RS): Establish capability to communicate with routers and protocols for implementing routing policies

Service function chaining: Build an architecture and controller capabilities for direct network-wide traffic subsets such that each virtual service platform only recognizes the traffic for which it needs to function.

Software-Defined Networking (SDN) Layers and Architecture Terminology was published by the Internet Research Task Force (IRTF) in RFC 7426 on January 2015. The document provides a brief description of current techniques to the architecture of the SDN layer. A valuable discussion of the southbound API (see Figure 3.3) is also available in the Request for Comments (RFC) and some specific APIs, like I2RS.

The Software Defined Networking Research Group (SDNRG) examines SDN from diverse perspectives to identify the approaches to be defined, deployed and applied in the near future and to identify future research challenges.

The International Telecommunications Union (ITU) is the united nation's specialized organization for telecom, information and technology communication (ICT). ITU –T is responsible for researching and

addressing technological, operational tariff questions and issuing Recommendations with a view to standardizing telecommunications globally. Recommendation ITU-T Y.3300 (Framework of Software-Defined Networking, June 2014) defines the software networking (SDN) structure for describing SDN fundamentals. In this Recommendation it discuss concepts, priorities, high-level capabilities, requirements and high-level SDN architecture.

In June 2013, ITU-T TSAG approved the establishment of a Joint Coordination Activity in Software Defined Networking (JCA-SDN) and research begun on SDN standards development.
SDN-related activities involvements comprise four ITU-T research groups (SGs).

Study Group (SG) 13 (Future networks including cloud computing, mobile and next-generation networks):

SG13 is a network and architecture Group that standardizes and incorporates various networking technologies and network specifications. SG13, the lead SDN study Group for ITU-T, develops the SDN framework as a basis for all ITU-T SDN standardization activities, including SDN terminology. Software Defined networking and service-aware networking of the future networks are responsible and discuss network virtualization and ITU-T recommendation Y.3300, Software-Defined Networking Framework, has been developed. Requirements and capabilities for NGN evolution (NGN-e), including IoT support and the use of software-defined network, SDN research and Next Generation Networks (NGN) virtualization aspects from the needs and capacities of the network are expected to apply. NGN Evolution (NGN-e) functional architecture including IoT support and software defined networking usage studies, SDN studies and next generation (NGN) virtualization aspects architecturically.

Study Group (SG) 11 (Signalling requirements, protocols and test specifications):

SG11 is a community to monitor and evaluate Protocols and standardize their criteria for different networks and networking technologies as well as testing specifications. The research with SG13, the signaling and resource control specifications and procedures in evolving communications environments, provides a supplementary document (non-standard document), which defines the SDN signals framework. Protocol procedures covering particular IPv6 services are investigating how SDN technologies are to be extended to IPv6.

Study Group (SG) 15 (Transport, Access and Home):

SG15 is responsible for the production, deployment, maintenance, management, monitoring, instrumentation and measurement and control

plane technologies in the areas of optical transport network, access network, home networks and power utility networks infrastructures, systems, equipment, optical fibers and cables, to facilitate the creation of intelligent transport networks. The group discusses transport aspects of SDN, which are related to the SDN architecture of the Open Network Framework.

Study Group (SG) 16 (Multimedia):

OpenFlow versus H.248 is evaluated as a protocol to control packet flows by Multimedia gateway control architectures and protocols. The multimedia framework, apps and services study networks for the delivery of virtual content.

European Telecommunications Standards Institute:

The Independent Group for standardization is a European Telecommunication Standards Institute ETSI) was essential to the advancement of ICT standards in Europe. The ETSI is an international standardization organization. The ETSI Industry Specification Group for Network Functions Virtualization (ETSI ISG NFV), a group responsible for establishing virtualization specifications and architecture for different functions within telecommunications networks

3.3.2 Industry Consortia:

It wasn't until the end of the 1980s that the consortium process began to be seriously studied the first Consortium Wave was created. In the late 1980s, there was a sense that the SDOs were too slow to have useful standards in the fast changing technology world. Several consortia have recently been involved in developing SDN and NFV standards. Three of the most important efforts are mentioned.

SDN standardization is the most relevant consortium is the Open Networking Foundation (ONF). The Open Networking Foundation (ONF) is a user oriented non-profit organization that is dedicated to promoting the adoption of software-defined networking (SDN) by creating open standards. That allows the control plane (SDN Controller) in an SDN environment to communicate with the forwarding plane (switch / router / chip set).

The Open Data Center Alliance (ODCA) is a network of the world's leading IT companies. Organizations that facilitate the development of interoperable cloud infrastructure technologies and services. ODCA defines SDN and NFV Cloud deployment requirements by developing usage models for SDN and NFV.

The Alliance for Telecommunication Industry Solutions (ATIS) is an organization that is committed to rapidly developing and fostering

technical and operational communications standards, as well as standards development and technical planning. ATIS released a paper outlining operational challenges and opportunities related to growing network programmability using SDN and NFV.

3.3.3 Open Development Initiatives:

A variety of other organizations, not directly formed by leaders of the industry and not official agencies, such as SDOs, exist. These organizations are usually user-focused and motivated, and often strive to build open standards or open source software. Several of these groups have become active in standardization of SDN and NFV. Three main initiatives are listed in this section.

OpenDaylight:

The OpenDaylight Project (ODL), which is run by the Linux Foundation, is an SDN open source project designed to develop SDN through the offer to the OpenDaylight Controller, renamed the OpenDaylight Framework, of a Community and Industry-supported system. It is open to everyone, which include end users and customers, and it offers a common platform to work together to find new solutions for people with SDN goals.

The OpenDaylight controller shows open northbound APIs that are used in applications. Such applications enable the controller to collect network information and to perform analytics using algorithms, and then to create new network rules using the OpenDaylight Controller.

Open Platform for NFV:

OPNFV was established as an open source framework based on community and industry support. Together, SDN and NFV are part of the transformation of business towards network and application virtualization.

OPNFV supports an open source network which brings together businesses and markets with new technologies to accelerate innovation. In order to speed up the development and implementation of NFV, both for enterprises and for service providers, the OPNFV brings together service providers, cloud and technology vendors, developers and consumers to build an open source platform.

OpenStack:

OpenStack is a software project aimed at developing an open source cloud operating system. It offers Multitenant Infrastructure as a Service (IaaS) and is intended to meet the needs of public and private clouds, irrespective of size, by being simple to implement.

Network flexibility and agility are being improved by new technologies, such as NDN and NFV, decoupling control from forwarding

plane in order to enable network service provision, automation, and orchestration. Network Virtualization (NV) aims to align network resources to meet the needs of a rich multi-tenant environment.

OpenStack Neutron is a networking SDN system that focuses on delivery in virtual machine environments of network-as-a-service (NaaS). In OpenStack, Neutron replaced the original Quantum Network Application Program interface (API). Neutron is designed to address shortcomings in cloud-based network technology as well as the lack of tenants control over the topology and the addressing of a network in multi-tenant environments, making advanced networking services difficult to deploy.

3.4 SDN DATA PLANE AND OPENFLOW

The study of Software-Defined Networking (SDN) in depth with a data plane (Figure 3.4). The rest of the chapter consists of OpenFlow, the most used SDN data plane implementation. OpenFlow is a logical data plane functionality structure and a protocol between SDN controllers and network devices.

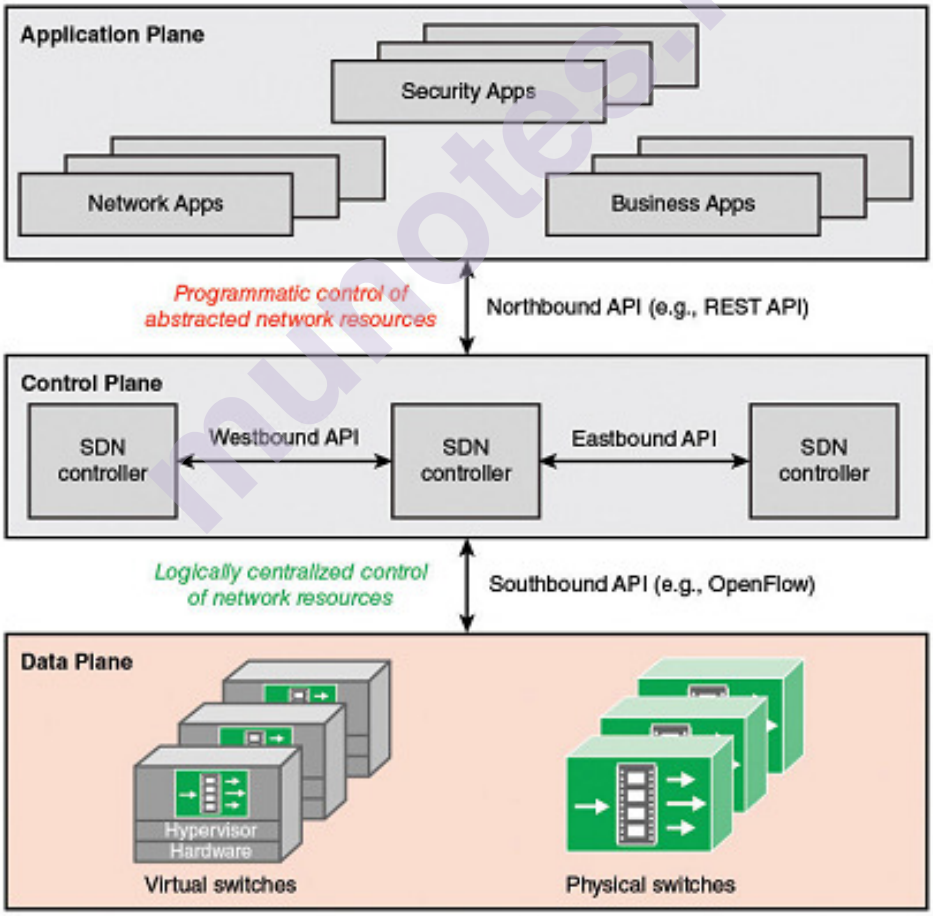


FIGURE 3.4 SDN Architecture

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

3.4.2 SDN DATA PLANE:

The SDN data plane is called the resource layer ITU-T in Y.3300 which is often also named the infrastructure layer. In the SDN architecture, the data plane consist the forwarding hardware. Since the controller requires to interact with network infrastructure, certain protocols are needed to control and manage the interface between different network equipment components. The main feature in an SDN network of network devices is that such devices are simple forwarding function, without embedded software, to take independent decisions.

3.4.2 Data Plane Functions:

The functions conducted by the network data plane devices also called the network elements or switches of the data plane are shown in Figure 3.5. The network device's main functions are as follows:

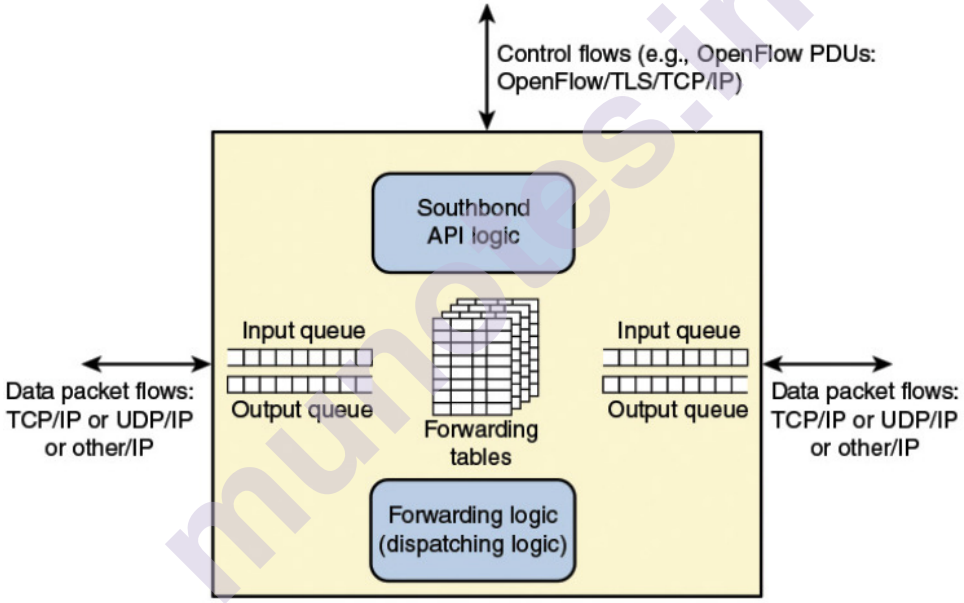


FIGURE 3.5 Data Plane Network Device

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

Control support function: interacts and enables programmability through resource-control interfaces with SDN control layer. The OpenFlow switch communicates with the controller, which is managed by the controller via OpenFlow switch protocol.

Data forwarding function: accept incoming flows of data from other network and end devices and forward them with the data forwarding paths

computed and established in accordance with the rules defined in the SDN applications

The network device forwarding rules are incorporated in the forwarding tables which show certain categories of packets what the next hop in the route should be for. The network device may change the packet header before forwarding or discard the packet, in addition to the simple forwarding of packets. As illustrated, incoming packets can be placed in the input queue awaiting processing by the network device, and forwarded packets are normally placed in an output queue, waiting for transmission.

In Figure 3.5 , the network device shows three I / O ports: one for SDN-controller control communication, and two for data packet input and output. The network device could have more than two I/O ports for packet flows in and out of the device for multiple SDN controllers.

3.4.3 Data Plane Protocols:

Figure 3.5 indicates the protocols supported by means of the network device. Data packet flows encompass streams of IP packets. It may be vital for the forwarding table to outline entries based totally on fields in upper-level protocol headers, along with TCP, UDP or some other protocol. The IP header examines by the network device and likely different headers in every packet and makes a forwarding decision. The other important data traffic flow is via the southbound application programming interface (API), consisting of OpenFlow Protocol Data Unit (PDUs) or a few similar Southbound API protocol traffic.

3.5 OPENFLOW LOGICAL NETWORK DEVICE

The distinguishing properties of SDN must be evaluated. The two key aspects of the SDN are a common logical architecture on SDN devices and an SDN controller protocol with network devices.

That means that the common logical architecture is required for all switches, routers, and other network devices to manage with an SDN controller. The SDN unification feature allows various providers to implement this logical architecture in various ways to construct network devices to operate under an SDN controller with a uniform logical switching function.

The SDN controller and network device require a standard, secure protocol.

The first software-defined networking (SDN) interface is OpenFlow (OF). The SDN Controller initially identified a networking protocol that enables it to deal directly with the forwarding plane of both

physical and virtual network equipment, such as switches and routers, to best suit evolving business needs. OpenFlow is specified by the Open Networking Foundation (ONF) OpenFlow Switch Specifications.

The key elements of an OpenFlow environment consisting of SDN controllers, including OpenFlow software, OpenFlow switches and end systems, are described in Figure 3.6.

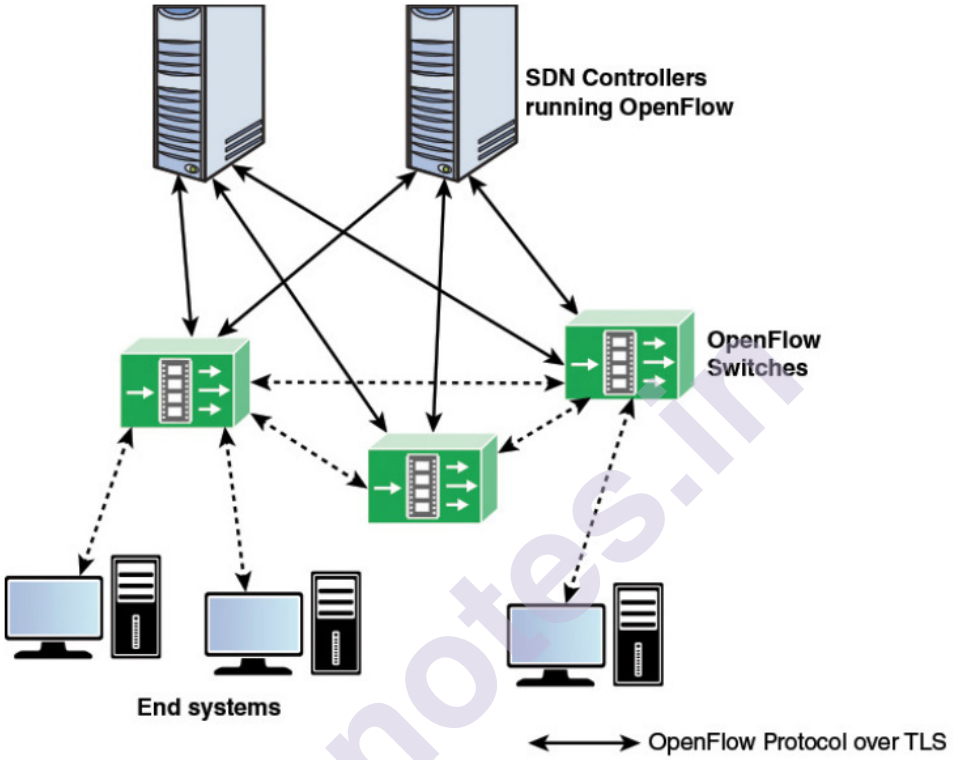


FIGURE 3.6 OpenFlow Switch Context

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

Figure 3.7 displays the main OpenFlow switch components. The communication standard used by SDN, known as OpenFlow, and developed by the Open Network Foundation, does the implementation of the Transport Layer Security (TLS). Each switch connects to other OpenFlow switches and, probably, to consumer devices used for packet flows of the sources and destinations. The Switch contains interface is called OpenFlow channel .Open Flow ports are used for interfaces. The transfer to the SDN controller is also attached to the OpenFlow port.

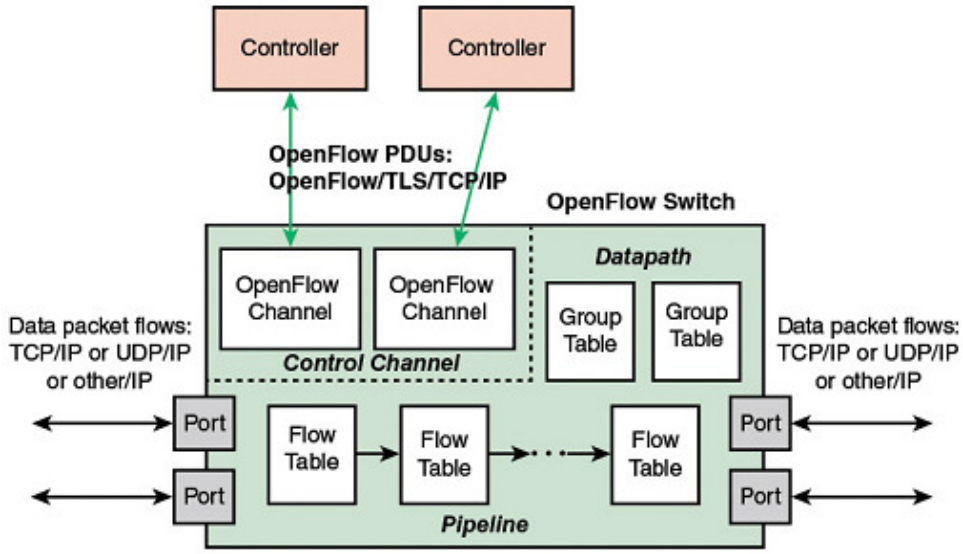


FIGURE 3.7 OpenFlow Switch

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

Three ports are specified by OpenFlow:

Physical ports:

The physical ports in OpenFlow are ports which have been connected to a switch's hardware interface. For instance, physical ports map the Ethernet interfaces one-to-one on an Ethernet switch.

Logical port:

The logical ports for OpenFlow are switched ports that do not meet the hardware of the switch directly. Logical ports are higher level abstracts which can be specified through non-OpenFlow methods (e.g. link aggregation classes, tunnels, loopback interfaces) in a switch.

Logical ports will encapsulate packets and map them through many physical ports. The processing through the logical port will be transparent to OpenFlow processing, and ports such as OpenFlow physical ports can communicate with OpenFlow processing.

Reserved Ports:

This specification defines the OpenFlow reserved ports. They signify basic forwarding actions, such as sending to the controller, flooding or transportation, utilizing non-OpenFlow methods, such as "normal" switch processing.

A series of tables are used within each switch to control packet flows via the switch.

In the logical switch framework the specification OpenFlow describes three types of tables. A flow table corresponds to the incoming packets in one flow and describes the functions on the packets. Multiple flow tables can be used in a pipeline fashion. A flow table may direct a flow relate to a group table that can cause a variety of actions involving one or more flows. A meter table may cause multiple actions related to performance on a flow. The controller will add, update and delete tables of flow entries both responsively and constructively utilizing the OpenFlow switch protocol.

3.5.1 Flow Table Structure:

The flow table represents the fundamental aspect of the logical switch architecture. Each packet entering a switch is loaded with one of the flow tables. A number of rows are specified in the following table, each of which consists of 7 elements, known as entries (see Figure 3.8).

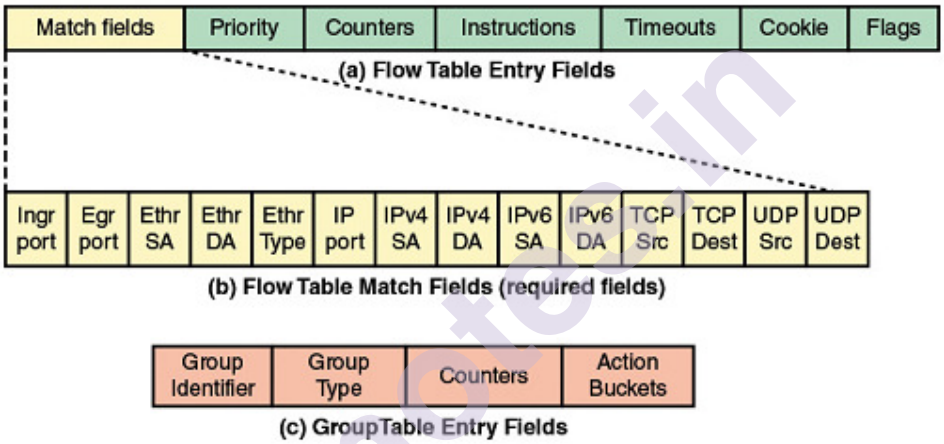


FIGURE 3.8 OpenFlow Table Entry Formats

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"
A flow table consists of flow entries.

match fields: To match the packets. Which comprise the ingress port and packet headers, and metadata listed in a previous table optionally.

Priority: matching precedence of the flow entry.

counters: updated when packets are matched.

instructions: to modify the action set or pipeline processing.

timeouts: maximum amount of time or idle time before flow is expired by the switch.

cookie: opaque data value chosen by the controller. May be used by the controller to filter flow statistics, flow modification and flow deletion. Not used when processing packets.

Flags: Flags modify the handling of flow entries; for instance the flag OFPFF_SEND_FLOW_REM causes messages removed from flow entry.

Counter	Usage	Bit Length
Reference count (active entries)	Per flow table	32
Duration (seconds)	Per flow entry	32
Received packets	Per port	64
Transmitted packets	Per port	64
Duration (seconds)	Per port	32
Transmit packets	Per queue	64
Duration (seconds)	Per queue	32
Duration (seconds)	Per group	32
Duration (seconds)	Per meter	32

TABLE 3.2 Required OpenFlow Counters

Match Fields Component

The match fields component of a table entry consists of the following required fields

Ingr port	Egr port	Ethr SA	Ethr DA	Ethr Type	IP port	IPv4 SA	IPv4 DA	IPv6 SA	IPv6 DA	TCP Src	TCP Dest	UDP Src	UDP Dest

Ingress port: The port identifier on which the packet arrived on this switch. It may be a physical port or a switch-defined virtual port. Required in ingress tables

Egress port: The identifier of the egress port from action set. Required in egress tables.

Ethernet source and destination addresses: Each entry can be an exact address, a bitmasked value for which only some of the address bits are checked, or a wildcard value (match any value).

Ethernet type field: Indicates type of the Ethernet packet payload.

IP: Version 4 or 6

IPv4 or IPv6 source address, and destination address: Each entry can be an exact address, a bitmasked value, a subnet mask value, or a wildcard value.

TCP source and destination ports: Exact match or wildcard value.

UDP source and destination ports: Exact match or wildcard value.

Every OpenFlow-compliant switch will follow the previous match fields. Optionally support can be provided in the following areas.

Physical port: Used to designate underlying physical port when packet is received on a logical port.

Metadata: Additional information that can be passed from one table to another during the processing of a packet. Its use is discussed subsequently.

VLAN ID and VLAN user priority: Fields in the IEEE 802.1Q virtual LAN header.

IPv4 or IPv6 DS and ECN: Differentiated Services and Explicit Congestion Notification fields

SCTP source and destination ports: Exact match or wildcard value for Stream Transmission Control Protocol.

ICMP type and code fields: Exact match or wildcard value.

ARP opcode: Exact match in Ethernet Type field.

Source and target IPv4 addresses in ARP payload: Can be an exact address, a bitmasked value, a subnet mask value, or a wildcard value.

IPv6 flow label: Exact match or wildcard.

ICMPv6 type and code fields: Exact match or wildcard value.

IPv6 neighbor discovery target address: In an IPv6 Neighbor Discovery message.

IPv6 neighbor discovery source and target addresses: Link-layer address options in an IPv6 Neighbor Discovery message.

MPLS label value, traffic class, and BoS: Fields in the top label of an MPLS label stack.

Provider bridge traffic ISID: Service instance identifier.

Tunnel ID: Metadata associated with a logical port.

TCP flags: Flag bits in the TCP header. May be used to detect start and end of TCP connections.

IPv6 extension: Extension header:

While OpenFlow will be used for a range of network protocols services for network traffic. let's remember, Ethernet is only supported on the MAC / link layer. Therefore, OpenFlow cannot control Layer 2 wireless network traffic as currently defined.

A much more appropriate definition of flow can now be provided. The sequence of packets that match a particular flow table entry from the point of view of an individual switch. The definition is packet-oriented, because it is a function of the values in the packet header fields of the flow, not the path through the network. A combination of flow entries on many switches determines a flow connected to a particular path.

The instructions component of a table entry is composed of a set of instructions which are executed if the packet matches the input. They need to specify the words "Action" and "Action Set" before describing the form of instructions. Actions represent packet forwarding, modification of a packet, and group table processing operation. In OpenFlow specification, the following measures are included:

Output: Forward packet to specified port.

Set-Queue: Sets the packet queue ID. The queue Id specifies which queue ueute attached to this Port is used to schedule and forward a packet as the packet is forwarded to a port by utilizing the output action. The forwarding behaviour is defined by the queue configuration and used to enable basic QoS.

Group: Process packet through specified group.

Push-Tag/Pop-Tag: Push or pop a tag field for a VLAN or MPLS packet.

Set-field: Various Set-Field actions by field type are defined and the values of the corresponding header fields in the packet are modified.

Change-TTL: Various actions change-TTL modify the value of IPv4 Time To Live (TTL), IPv6 Hop Limit or MPLS TTL in the packet

Drop: No direct action description of drops is visible. Instead, packets with no action sets have no output action should be dropped

An Action Set is a sequence of activities associated with a packet that are collected during each table is been process and executed as the packet exits the processing pipeline. Instructions of four types:

Direct packet through pipeline: Goto-Table instructions direct the packet to a table in the pipeline. Meter instruction directs the packet to a specified meter.

Perform action on packet: Actions may be performed on the packet when paired with a table entry.

Update action set: Combine actions to the current action set for this packet or clear all actions in the action set.

Update metadata: a packet can have a metadata value. It is used to transmit information from table to other table.

3.5.2 Flow Table Pipeline:

A switch has one or more flow tables in it. If more than one flow table exists, they are organized as a pipeline, with tables labelled with increasing numbers starting with 0. The use of multiple tables in a pipeline and not a single flow table provides considerable flexibility for the SDN controller.

The OpenFlow specification defines two processing stages:

Ingress processing:

The processing of Ingress always takes place from table 0 and uses the input port as an identity. Table 0 could be the only table in which the ingress processing on the same table is simplified and no egress processing is carried out.

Egress processing:

The Egress processing is the processing after the output port has been determined. This occurs in the output port's context. It is an optional stage. When it occurs, one or more tables can be used. The numerical identification of the first egress table indicates the separation of both stages. All tables below the first Egress Table should be used as Input Tables. An Input Table cannot be used as a table with a number above or equal to the first Egress Table.

The processing of pipelines also starts with the first flow table processing; then, the packet has to be matched with the flow inputs of flow Table 0. Depending on the match outcome of the first table, other ingress flow tables can be used. The OpenFlow switch can conduct egress processing in connection with that output port if the result of the ingress processing is to transfer the packet to an output port.

The input includes the packet, ingress port Identity, associated metadata value, and the associated action set if a packet is presented at a matching table. The metadata value for table 0 is blank and the action set is zero. Processing is carried out as follows at each table (see Figure 3.9):

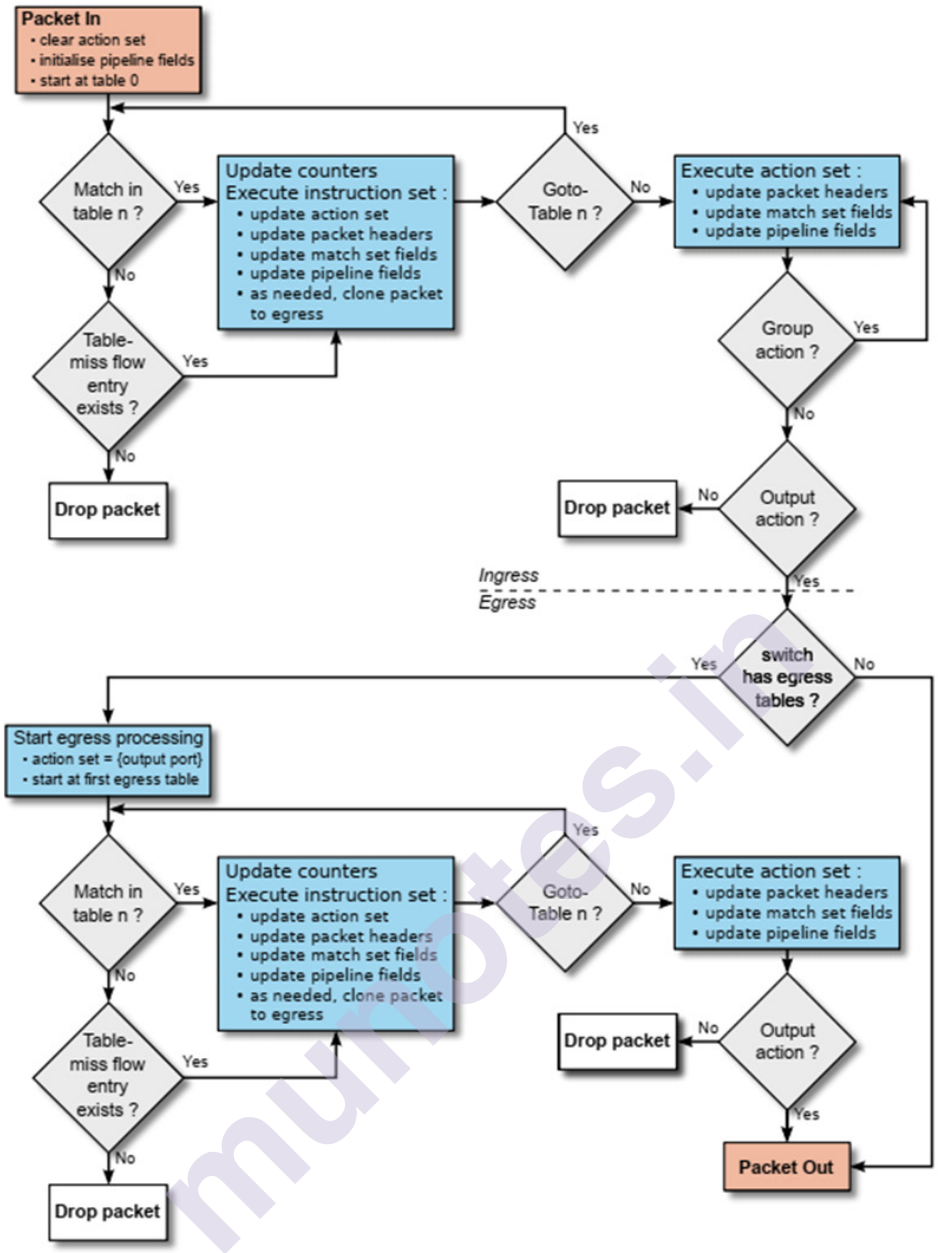


FIGURE 3.9 Simplified Flowchart Detailing Packet Flow Through an OpenFlow Switch

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

The input consists of packet, ID of the ingress port, the associated Metadata value, and the corresponding action set if a packet is introduced to a match table. For Table 0, the value of the metadata is blank and the action set is null. Processing proceeds is the following:

3. When one or more entries other than the table-miss are matched, then the match is defined as the highest priority matching entry. The following actions should be taken:

a. Using this entry, update any counters.

b. Execute any instructions relevant to this entry. These instructions may involve updating the actions set, the metadata value update and the actions are performed.

c. The packet is then forwarded to the flow table downwards, to the group table or the meter table or to the output port

2. Find the flow matching entry with the highest priority. If there is no match and no table-miss entry, the packet will be discarded. If only a table-miss entry matches, then one of three actions is given by that entry:

a. Send the controller to packet. This action allows the controller to define a new packet flow or decide to drop the packet.

b. Down the pipeline, direct the packet to another flow table.

c. Drop the packet.

Forwarding to another flow table for the final table in the pipeline is not an option. If and when a packet is finally sent to the output port, the built-in action set is executed and the packet is queued for output. The complete ingress pipeline process is shown in Figure 1.10.

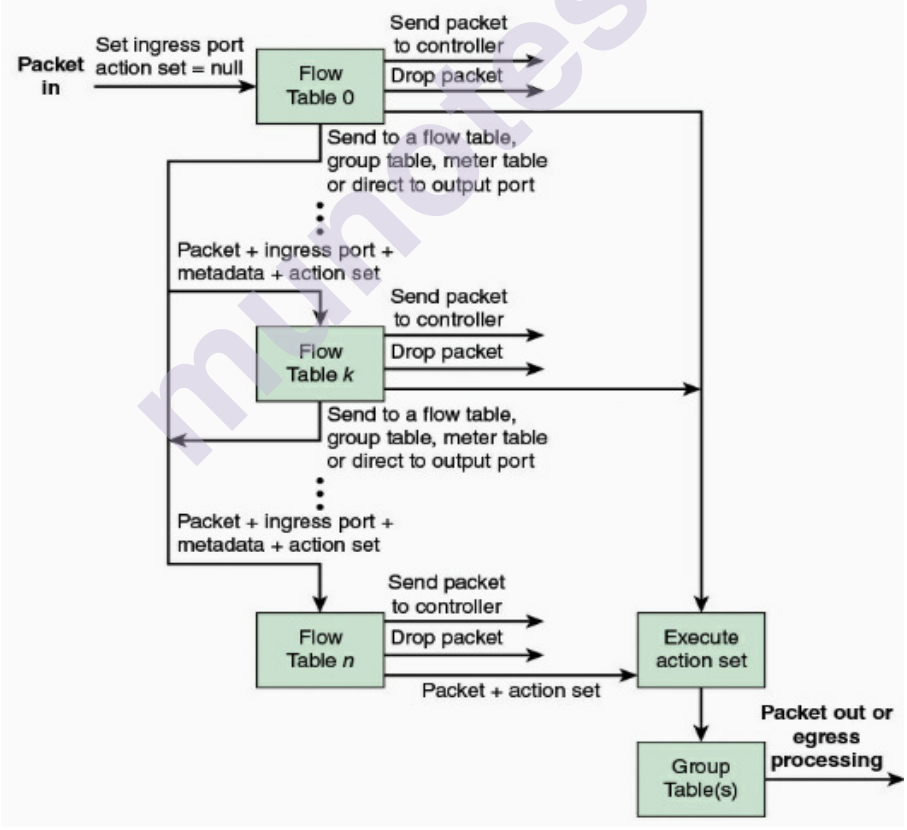


FIGURE 3.10 Packet Flow through an OpenFlow Switch: Ingress Processing

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

As egress processing are connected to a specific output port, the packet will be directed to the first flow table of the egress pipeline until the output port is completed. The processing of the Egress pipelines proceeded in the same way as for the ingress processing apart from that at the end of the egress pipeline there was no group table processing. Figure 1.11 shows the egress processing.

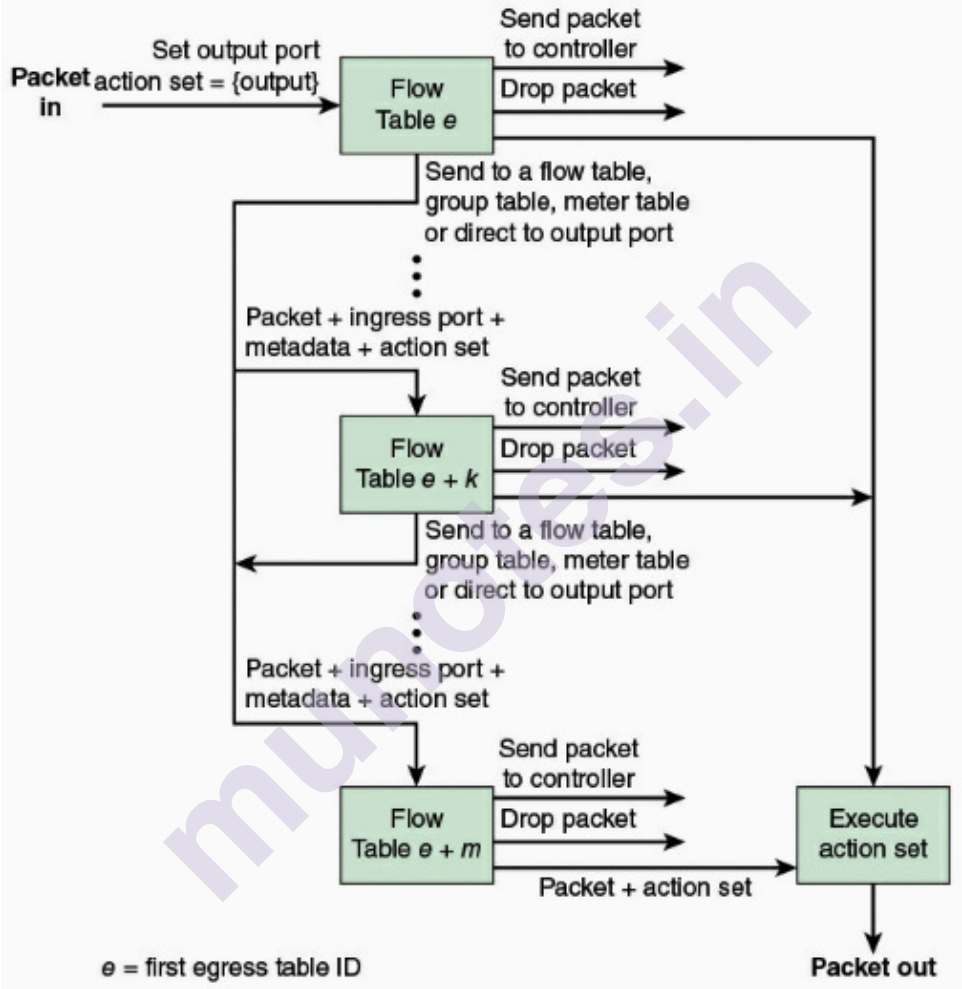


FIGURE 3.11 Packet Flow through OpenFlow Switch: Egress Processing

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

3.5.3 The Use of Multiple Tables:

Using multiple tables, the flow can be nestled or divided into a number of parallel subflows by a single flow. This property is seen in Figure 1.12 . In that example, a Table 0 entry defines a flow from a specific source IP address to a specific IP destination address, comprised

of packets traversing the network. If a cost least route is formed between these two endpoints, all traffic between these two endpoints will make sense in pursuing this route, and the next hop on the route from this switch can be in Table 0 .Different transport layer protocols, such as TCP and UDP, individual entries may be found in Table 1. The same output port can be maintained for such subflows, such that all subflows pursue the same route. TCP, however, involves comprehensive congestion control mechanisms usually not found with UDP, and the parameters correlated with the service quality (QoS) of TCP and UDP subflows should be fairly treated differently. Every Table 1, including any or all the entries may trigger Table 2, splitting some of each subflow. Table 2 that automatically route the respective subflow to the output port. The figure suggests that the TCP subflow may be separated by a protocol running over the TCP, such as the Simple Mail Transfer Protocol (SMTP). The UDP flow may also be subdivided on the basis of UDP-based protocols, such as the Simple Network Management Protocol (SNMP). The table also displays other Table 1 and Table 2 subflows that may be used for other purposes.

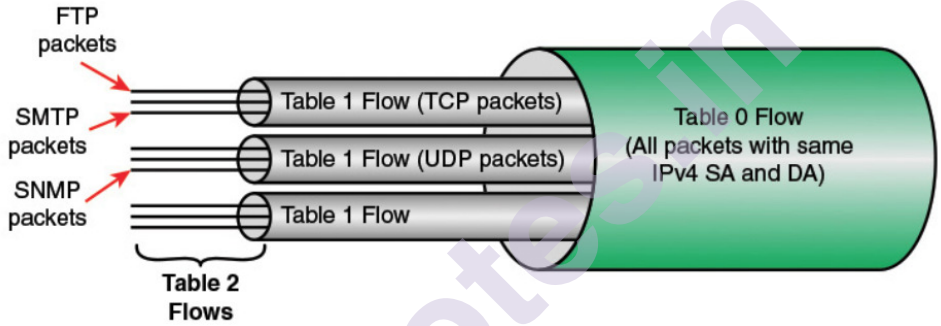


FIGURE 3.12 Example of Nested Flows

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

In this case, each fine-grained subflow could be described in Table 0. Using several tables facilitates the processing in both the SDN controller and OpenFlow switch. Actions like next hop for the combined flow will once be defined by the control and examined and performed once by by the switch. New subflows are added to all levels with less configuration. The usage of pipelines, several tables, improves network efficiency and helps the network to respond to changes at application, user and session level in real time.

3.5.4 Group Table:

There are group entries in a group table. The ability to point to a group for a flow entry enables OpenFlow to demonstrate more forwarding methods (e.g. select and all).

Group Identifier	Group Type	Counters	Action Buckets

Table 2: Main components of a group entry in the group table.

The group identification of each group entry (see Table 2) consists of:

Group Identifier: a 32 bit unsigned integer uniquely identifying the group

Group Type: to determine group semantics

Counters: updated when packets are processed by a group

Action Buckets: an ordered list of action buckets, where each action bucket contains a set of actions to execute and associated parameters

OpenFlow may represent a variety of ports as a single entity for forwarding packets with the new group abstraction. There are different forms of groups that represent various abstractions such as multicasting or multipathing. There are a set group buckets in each group. Each group bucket includes the set of actions to be taken before forwarding them to the port. Groups buckets should also forward to certain groups, which brings chains together.

One of the other types seen in Figure 1.13 is the group: all, select, fast failover and indirect.

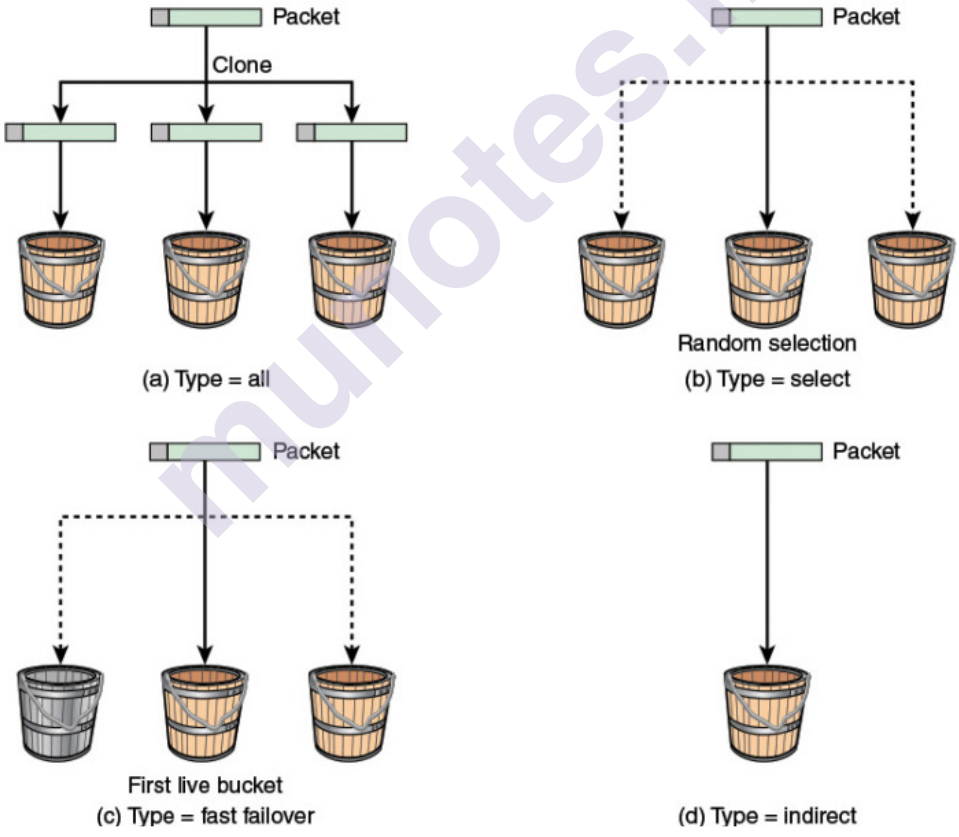


FIGURE 3.13 Group Types

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

The ALL Group

The **ALL** group would begin with each packet received as input and duplicate it to be run independently on the bucket list, through this manner, an ALL group will replicate and then operate on individual copies of the packet, as defined in each bucket by actions. Each bucket can have different and distinct actions allowing for different operations on different packet copies. Each group is used for multicast or broadcast forwarding.

The SELECT Group:

SELECT Group designed specifically for load balancing. Each bucket in a SELECT group is weight-assigned and each packet entering the group is redirected to one bucket. The bucket selection algorithm is not specified and is based on the switch; however, the most obvious option of packet distribution for buckets is weighted round robin. The bucket weight is assigned to each bucket as a special parameter. Each bucket of the group SELECT has a set of actions such that all actions supported by OpenFlow can be included with any bucket, just like the ALL group, the buckets do not have to be uniform.

The INDIRECT Group:

The INDIRECT group can be challenging to comprehend as a "group," because it includes only a single bucket that holds all packets the group receives within this particular bucket. The INDIRECT group contains no bucket list, but rather a single bucket (or a specific list of actions). The INDIRECT group aims at encapsulating a specific series of actions that multiple flows use. For instance, if flows A, B and C match on separate packet headers and common set or sub-set of actions, the flows that send packets to an INDIRECT group instead of duplicating the list of common actions for each flow of each packet header. The INDIRECT group simplifies the deployment of OpenFlow and reduces the memory footprint of a sequence of similar flows.

The FAST-FAILOVER group:

The FAST-FAILOVER group explicitly designed to detect and overcome port faults. The FAST-FAILOVER group, much like the SELECT and ALL groups, has a list of the buckets. Every bucket often has a specific watch port and/or group as a parameter in addition to this set of actions. The watch port / watch group controls the 'liveness' or up / down status of the port / group defined. The bucket would not be included if the liveness is deemed to be decreased. The bucket will be used while the liveness is determined to be up. Only one bucket can be used at a time, and the bucket being used will not be changed without the transition from the watch port / group of the bucket watch being used. When this occurs,

the group FAST-FAILOVER quickly selects the next bucket with a watch / group that is up in the bucket list.

When a failed bucket occurs, there is no guarantee of the transition time. The transformation period depends on the quest period to locate and enforce a control port / group. The rationale for the usage of a FAST-FAILOVER party however is that a control aircraft to monitor the port down case and introduce a fresh flow or collection of flows would almost definitely be faster than consulted. With FAST-FAILOVER classes, the whole data plane is used for failure identification and recovery.

When a failure bucket occurs, there is no guarantee of the transition time. The transition time depends on the search time to locate and implement a control watch port / group. The motivation for the usage of a FAST-FAILOVER group however is that a control plane to manage the port down event and introduce a new flow or set of flows would almost definitely be faster than consulted. With FAST-FAILOVER groups, the whole data plane is used for failure detection and recovery.

3.6 OPENFLOW PROTOCOL

The OpenFlow protocol provides a description of message exchanges between OpenFlow controller and OpenFlow device. In general, the SSL or Transport Layer Security (TLS) protocol is implemented, providing a secure OpenFlow channel.

The OpenFlow protocol helps the controller to add, update, and delete actions to the flow entries in the flow tables. Three messages types are supported (see Table 3):

Controller-to-Device: The controller initiates these messages and requires the device to respond in some cases.

Asynchronous: The controller sends certain forms of messages without a request.

Symmetric: These messages are received from the controller or the device without a request. It is indeed simple yet helpful. Hello messages are usually sent between the controller and the switch when the connection has been established first. The echo request and reply messages may be used to test the latency and bandwidth of a controller-switch connection either by a device or by the controller or merely verifies that the device is operative. The Experimenter message is used to build features for future OpenFlow versions.

In general, three types of information for the management of the network are provided to the SDN controller by the OpenFlow protocol:

Event-based messages: When a connection or port changes occur, send the message to the controller.

Flow statistics: produced by traffic flow-based switch. The controller will monitor traffic, reconfigure the network, and change the flow parameters to satisfy QoS requirements.

Encapsulated packets: The switch sends this packet to the controller, either because it is sent in the flow table or because the switch needs information for the creation of a new flow.

Message	Description
Controller to Switch	
Features	Features Request the capabilities of a switch. Switch responds with a features reply that specifies its capabilities.
Configuration	Configuration Set and query configuration parameters. Switch responds with parameter settings.
Modify-State	Modify-State Add, deiete, and modify fiow/group entries and set switch port properties.
Read-State	Read-State Collect information from switch, such as current configuration, statistics, and capabilities.
Packet-out	Packet-out Direct packet to a specified port on the switch.
Barrier	Barrier request/reply messages are used by the controller to ensure message dependencies have been met or to receive notifications for completed operations.
Role-Request	Role-Request Set or query role of the OpenFlow channel. Useful when switch connects to multiple controllers.
Asynchronous Configuration	Set filter on asynchronous messages or query that filter. Useful when switch connects to multiple controllers.
Asynchronous	
Packet-in	Transfer packet to controller.
Row-Removed	Inform the controller about the removal of a flow entry from a flow table.

Port-Status	Inform the controller of a change on a port
Role-Status	Inform controller of a change of its role for this switch from master controller to slave controller.
Controller-Status	Inform the controller when the status of an OpenFlow channel changes. This can assist failover processing if controllers lose the ability to communicate among themselves
Flow-monitor	Inform the controller of a change in a flow table. Allows a controller to monitor in real time the changes to any subsets of the flow table done by other controllers
Hello	Exchanged between the switch and controller upon connection startup
Echo	Echo request/reply messages can be sent from either the switch or the controller, and must return an echo reply
Error	Used by the switch or the controller to notify problems to the other side of the connection.
Experimenter	For additional functionality,

TABLE 3 OpenFlow Messages

3.7 UNIT END QUESTION

1. Explain the SDN Architecture.
2. Why traditional network architecture are inadequate for transmission for carried a data? How this limitation are solved?

3 Explain the SDN controller.

4. Explain the Software-Defined Architecture.
5. State the characteristics of Software-Defined Networking.

7. How are SDN- and NFV-Related Standards.
8. Explain the different types of Standards-Developing Organizations for developing the Software-Defined Networking.
9. Explain the different types of Industry Consortia involved in the development of SDN and NFV standards.

10 State the SDN Data Plane in the SDN Architecture.

11 Explain the different types of functions performed by the data plane network devices.

12 Explain the Data Plane Protocols.

13 Explain the OpenFlow Logical Network Device.

14 Explain the different types of port in OpenFlow Switch.

15 Explain the Flow Table Structure.

16 Explain the Group Table.

3.8 REFERENCES

- ODCA14: Open Data Center Alliance. Open Data Center Alliance Master Usage Model: Software-Defined Networking Rev. 2.0. White Paper. 2014.
- ONF12: Open Networking Foundation. Software-Defined Networking: The New Norm for Networks. ONF White Paper, April 13, 2012.
- Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud William Stallings Addison- Wesley Professional October 2015
- SDN and NFV Simplified A Visual Guide to Understanding Software Defined Networks and Network Function Virtualization Jim DohertyPearson Education, Inc
- Network Functions Virtualization (NFV) with a Touch of SDN Rajendra Chayapathi Syed Farrukh Hassan Addison- Wesley.
- CCIE and CCDE Evolving Technologies Study Guide Brad dgeworth, Jason Gooley, Ramiro Garza Rios Pearson Education, Inc 2019.

*****

4

SDN CONTROL PLANE AND SDN APPLICATION PLANE

Unit structure

4.0 OBJECTIVES

After you have studied this chapter, you should be able to:

- List and explain the key functions of the SDN control plane.
- Discuss the routing function in the SDN controller.
- Understand the ITU-T Y.3300 layered SDN model.
- Present an overview of OpenDaylight.
- Present an overview of REST.
- Compare centralized and distributed SDN controller architectures.
- Explain the role of BGP in an SDN network.
- Provide an overview of SDN application plane architecture
- Define the abstraction layers of network services.
- List and describe three forms of SDN abstraction.
- List and describe six important SDN application areas.

4.1 SDN CONTROL PLANE ARCHITECTURE

This Chapter remains our studies into the control plane of software-defined networking (SDN) (see Figure). This section provided an overview of the SDN control plane architecture, and describes the functions of a typical SDN control plane. The SDN model, layered ITU-T provide additional insights into the function of the control plane. A summary with one of the most important SDN controller open source efforts known as OpenDaylight. In this Section explains the REST northbound interface, a common feature in SDN implementations. In section addresses cooperation and coordination issues between various SDN controllers.

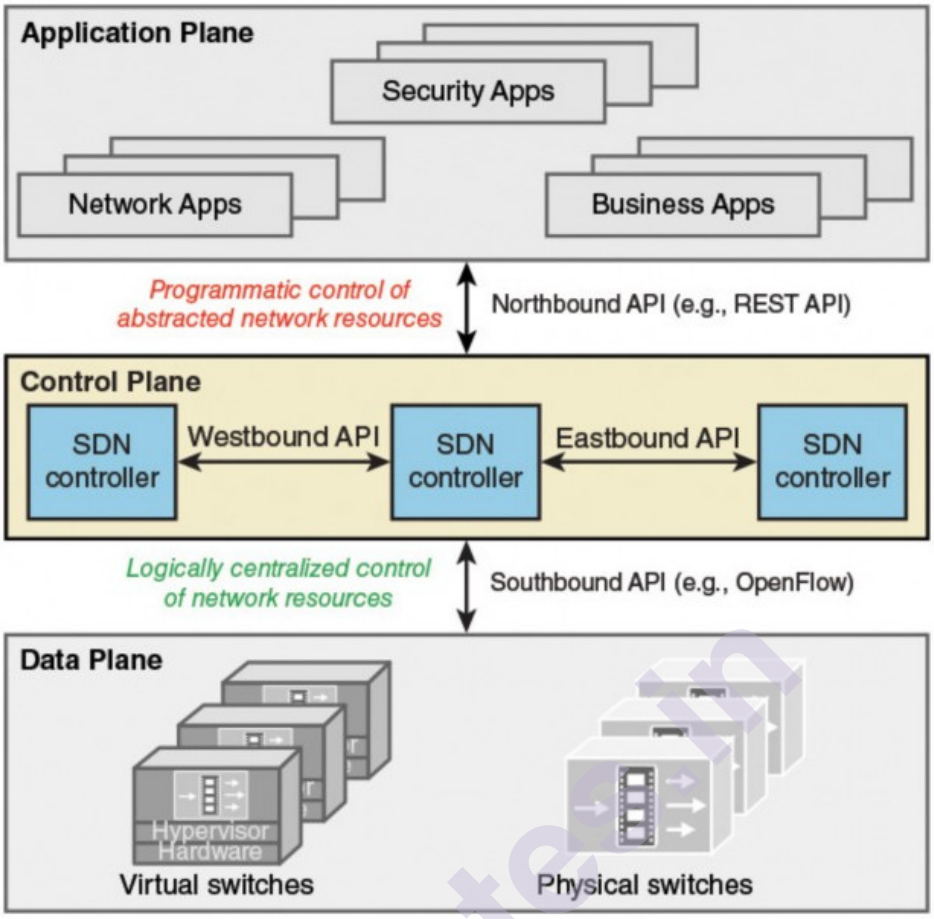


FIGURE 4.1 SDN Architecture

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

The requests for the service of the application layer are mapped to specified commands and directives for data plane switches by the SDN Control Layer. The SDN control layer then offers data plane topology and activity information to applications. The control layer then acts as a server or as a collaborative servers known as SDN controllers.

4.1.1 Control Plane Functions:

The functions performed with SDN controllers as seen in Figure 4.3 a s proposed in a paper by Kreutz , this figure 4.3 demonstrates the important functions that every controller need to provide:

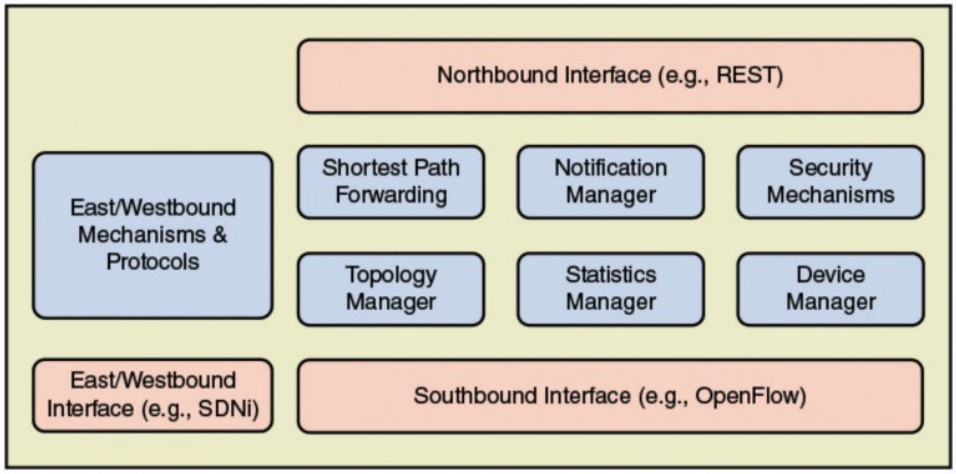


FIGURE 4.2 SDN Control Plane Functions and Interfaces

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

Shortest path forwarding:
Using information gathered from switches in order to classify desired paths.

Notification Manager:
It must be in an able to receive, process and forward events (e.g. Alarm notifications, security alarms, state changes).

Security mechanism:
Critical components for providing basic isolation and security compliance between services and applications.

Topology manager:
It Builds and maintains interconnection topology information.

Statistics manager: It collects traffic data through switches:
SDN controller functionality can be described as a network operating system (NOS). NOS is a software framework running on commodity server technology providing important resources and abstractions that promote programming of forwarding devices focused on an abstract network view that is logically centralized. Therefore, its function is similar to that of a traditional OS. The following addressed Northbound interface offers a standard way of accessing SDN service and conducting network management tasks for application developers and network managers. In addition, well-defined northbound interfaces allow developers to produce software that is not only independent of the data plane, but can be widely used on various SDN controller servers.

A variety of projects, both open source and commercial, have contributed to the introduction of the SDN controller. The list below identifies some significant ones

OpenDaylight:

The largest open source SDN controller, OpenDaylight, helps to handle transformation. ODL is a modular, open platform for the customization and automation of any size and scale of networks. The ODL project originated from the SDN campaign and focused specifically on network programmability. It was developed from the beginning as a basis for commercial solutions, which deal with many applications in existing network environments. The ODL project mission that has been initiated at the beginning of 2013. It was initially led by IBM and Cisco but was then hosted by the Linux Foundation. OpenDaylight can be implemented as one centralized controller which allows for the distribution of controllers where one or more instances can be run on one or more network clustered servers.

Open Network Operating System (ONOS):

The ONOS source code was published in the open source community by the Open Network Lab (ON.Lab) on 5 December 2014 together with other industry partners, including AT&T and NTT Communications. The Linux Foundation stated on 14 October 2015 that ONOS became part of the organizations as one its collaborative project. ONOS is built to be used as a distributed controller and provides an abstraction on multiple distributed controller for partitioning and distributing network state.

POX:

POX is an OpenFlow / Defined Networking (SDN) Controller built in Python. POX offers a framework to communicate using either the OpenFlow or the OVSDB protocol for SDN switches. To build an SDN controller, developers can use POX to use the Python programming language. It is a common method to teach and research software of defined software networks and network applications.

Beacon:

Beacon is a fast, cross-platform-based modular OpenFlow controller, based on Java, which is suitable both for event and threaded operation. Beacon is written in Java and runs from high-end Multi-Core Linux servers to Android phones on various platforms. Beacon is implemented in java and runs on several platforms, from high-end multi-core Linux servers to Android phones. Java and Eclipse make it easier for your application to develop and debug.

Floodlight:

The leading open source OpenFlow Controller is Floodlight. It is sponsored by a developer community with a number of Big Switch Networks engineers. The Floodlight Open SDN controller is a Java-based, Apache-licensed, OpenFlow Controller for enterprise-class applications that is designed to work with traditional JDK- and ANT-tools. There is both a web-based and a Java-based GUI and most of its functions are displayed via a REST API.

Ryu:

Ryu Controller is an open, software-defined SDN network controller intended to improve network agility by enabling the management and adaptation of traffic control. The NTT lab supports Ryu controller and is implemented in laboratories. It is open sourced and fully built in python.

Onix:

Onix is a distributed system operating on a control platform that distributes, collects and transmits information from switches appropriately across different servers and offers a broad range of management applications. VMWare, Google and NTT developed it all together. Onix is an SDN controller that is commercially available.

4.1.2 Southbound Interface:

In SDN, the OpenFlow protocol specification is the southbound interface. The key role of the SDN controller is to allow communication with the network nodes (physical, virtual switches and router) to enable a router to identify a network topology, to establish network flows, and to send the related requests. The usage of a southbound abstraction layer providing a standard framework for control plane functions while supporting multiple Southbound APIs is a more flexible approach.

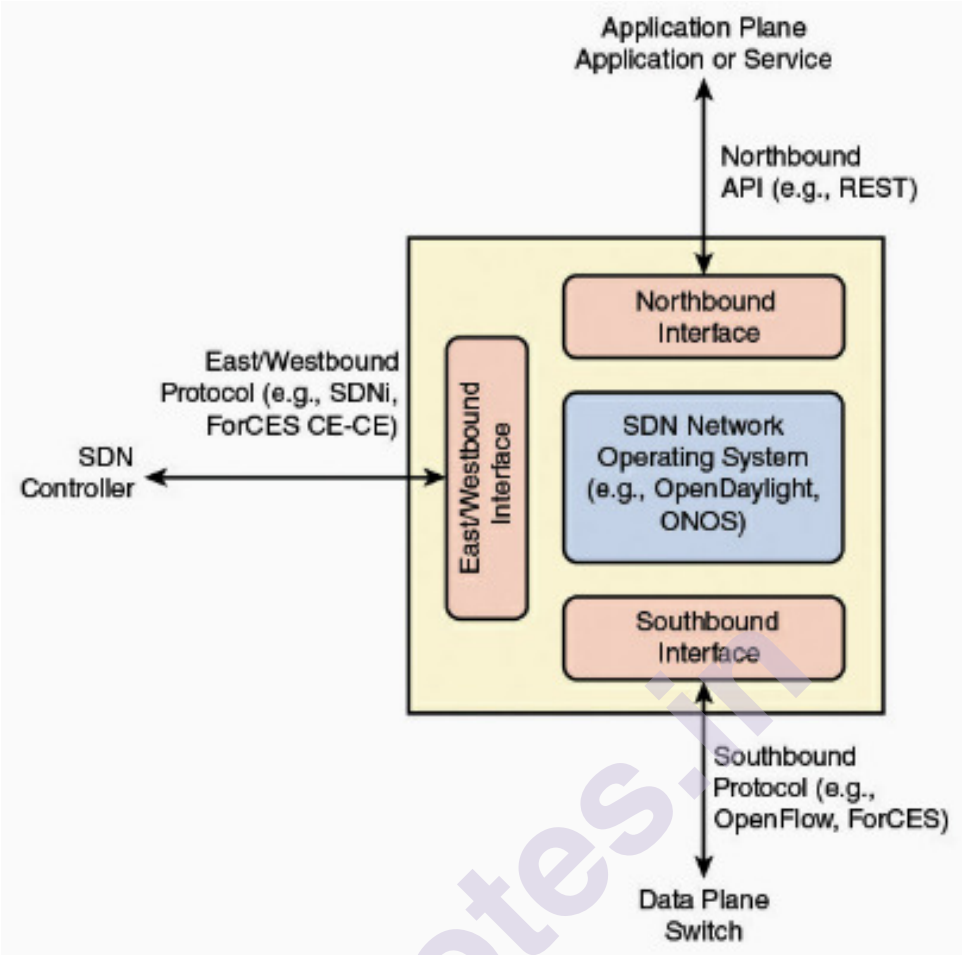


FIGURE 4.3 SDN Controller Interfaces

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

OpenFlow is a popular southbound interface. Other southbound interfaces include:

- Open vSwitch Database Management Protocol (OVSDB): The Open vSwitch Database Management Protocol (OVSDB) is a protocol designed to handle Open vSwitch implementation. OVS is an Open-Flow implementation of the switch side of the most advanced and feature-full open source implementation. OVS was used as a logic control for real hardware switches and a virtual switch inside a hypervisor to enable network virtualization. OVSDB is the protocol used for OVS instances management and configuration.
- Forwarding and Control Element Separation (ForCES): It specifies an architectural structure and related protocols to standardize the information exchange in a ForCES network unit (NE ForCES) between the control plane and the forwarding plane.
- Protocol Oblivious Forwarding (POF): This is an improvement in OpenFlow that simplifies the logic of the data plane to a generic

forwarding element that does not need to be understood field format of the protocol data unit (PDU) at different protocol levels. Instead, the matching is made by blocks (offset, length) in the packet. Intelligence about packet format resides at the control plane level.

4.1.3 Northbound Interface:

Software-oriented Networking (SDN) is a change in network based computation through the breaking down of existing physical boundaries through switches, routers and controllers by well specified APIs. The software's interaction between systems should be described using an Application Programming Interface (API). The programming aspect of the API is what SDN needs. In comparison to another system exchanges, this programming capability gives APIs stability. This, too, makes it ideal for SDNs, for its simplicity and efficiency. Therefore, APIs allow a 'system' to effectively submit a request or to address requests. Northbound APIs the most critical APIs in the SDN environment. To further improve the use of SDN, the ONF developed, but not generally standardized, a working group for the growth of prototypes for the NBI. The Northbound Interfaces Working Group aims to establish an architecture that can deal with various abstract levels and across a broad range of fields. However, it strives to create case-specific interfaces before seeking the universal solution. While it was first developed in 2013, an NBI standardization has not been published yet publicly.

The definition of different API latitudes is demonstrated in Figure 4.5, NBI-WG Charter paper (October 2013). For instance, an application can require one or more APIs which show the controller's functionality directly, manage a network domain and use APIs invoking analytical or reporting services that reside on the controller.

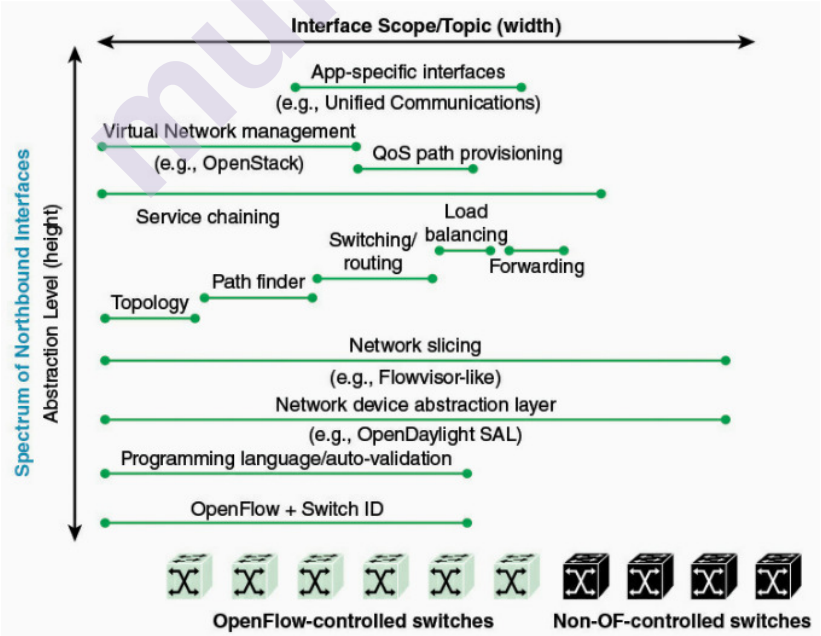


FIGURE 4.5 Latitude of Northbound Interfaces

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

The architecture with several levels of northbound APIs, which is listed in the list below, is seen in Figure 4.6.

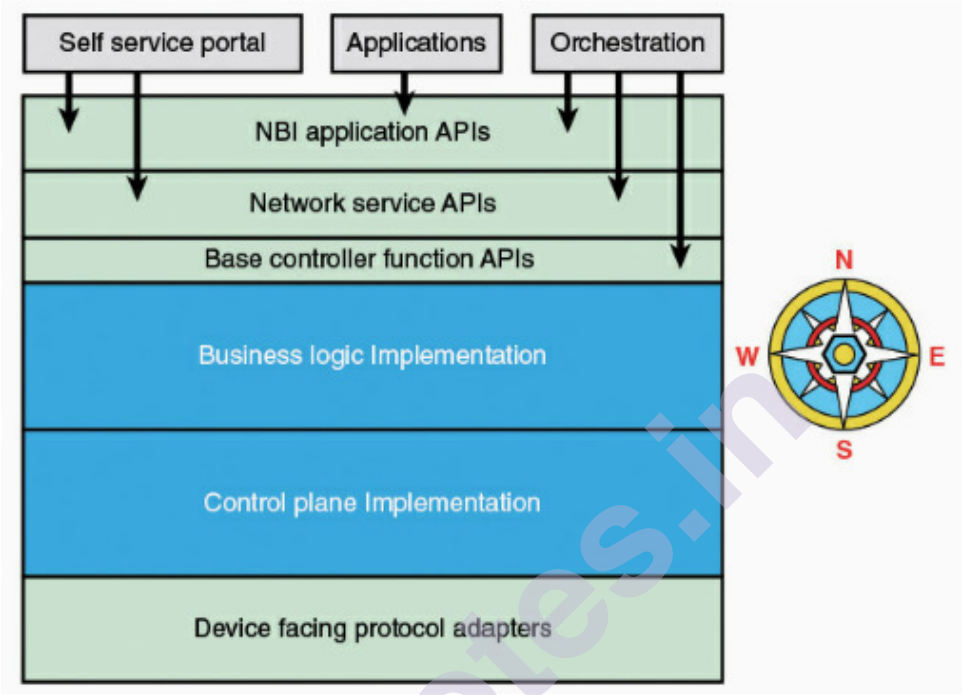


FIGURE 4.6 SDN Controller APIs:

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

Base controller function APIs: This APIs carry out the controller's basic functions and are used to build network services by developers.

Network service APIs: The network services are exposed to the North in these APIs.

Northbound interface application APIs: This APIs show applications services focused on network services.
REpresentational state Transfer is a typical architectural form used to describe Northbound APIs.

4.1.4 Routing:

SDN network needs a routing function, just like with any network or the internet. The routing function constitutes in specific protocol to gather information on network topology and traffic patterns as well as an algorithm for network route layout. Two types of routing protocols exist:

interior router protocols (IRP), operating in an Autonomous System (AS), and exterior router protocols (ERPs) that works between autonomous systems.

An IRP helps to discover the topology of routers inside the AS and deciding on the basis of various measures the optimal path to each destination. Open Shortest Path (OSPF) Protocol and Enhanced Interior Gateway Routing (EIGRP) Protocol are two commonly used IRPs. An ERP does not need to obtain as much data on traffic. Instead, the main problem of an ERP is to assess the reachability of external networks and end-systems outside of an AS. The ERP is therefore normally only performed in edge nodes connecting one AS to another AS. The ERP is commonly used for Border Gateway Protocol (BGP). The routing function is traditionally distributed over network routers. Every router is in responsibility of generating a diagram of the network topology. Each router must also collect connectivity information, delays, and compute the desired path for individual IP destination addresses for interior routing. However, centralization of the routing feature in the SDN controller makes sense in an SDN-controlled network. A consistent network view is developed by the controller for computation of shortest paths, and application aware routing policies are implemented. The data plane switches are reduced by the routing processing and storage costs, thus improving efficiency.

The centralized routing system contains two separate functions: the link discovery and the topology manager. The routing function must know the links between data plane switches for discovery of a connection. Remember, the links among routers in the case of an Internet network are networks, while the links are a direct physical link for Layer 2 switches such as Ethernet switches. Link discovery must also be done in a neighboring domain between a router and a host system and between a routers in that controller's domain. Discovery is caused by unknown traffic from an attached host and the adjacent router accessing the controller's network domain.

The topology manager manages the network topology and calculates the network routes. The calculation of the route involves deciding the shortest route between two nodes or between the node of a data plane and the host.

4.2 ITU-T MODEL

SDN functional architecture is based on the SDN framework [ITU-T Y.3300] (see Figure 4.7). The core features of the framework include logically unified network control, automated management of network infrastructure, network virtualization support and network optimization,

efficient network implementations SDN needs include: isolation of SDN control from network resources, preparation of network resources; integration of network resources by standard information and data models and support for network resource orchestration and SDN applications and operations.

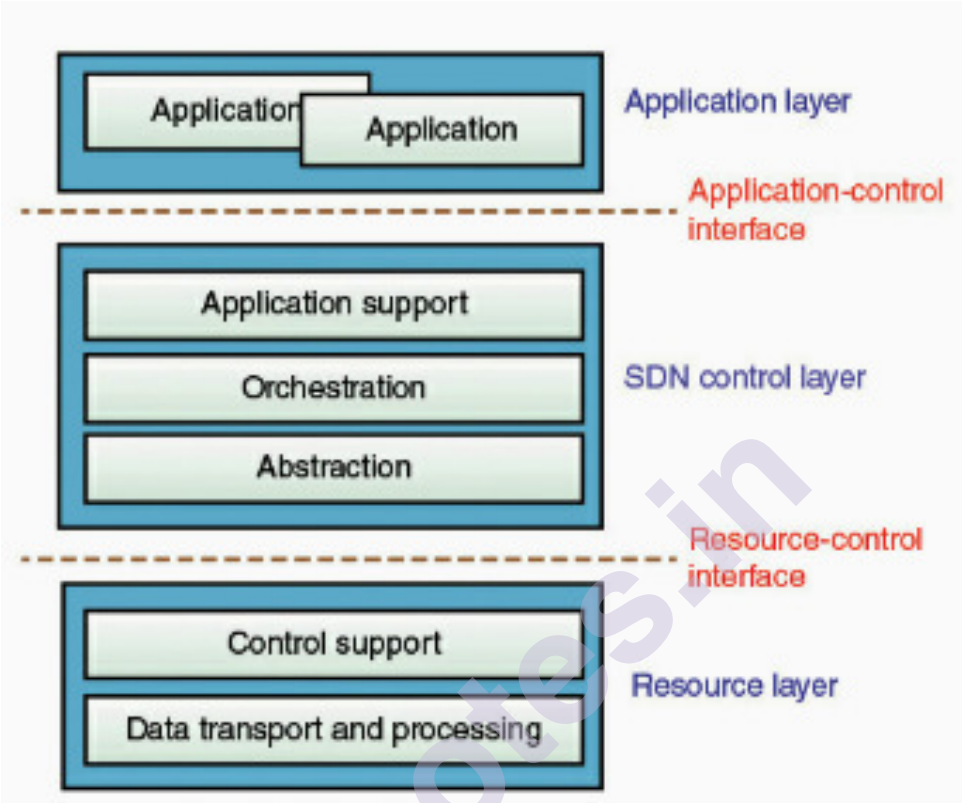


FIGURE 4.7 High-Level Architecture of SDN (ITU-T Y.3300)

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

The control layer provides the means, as directed by the application layer, to dynamically manage network resources. The control layer can be seen as providing the following sublayers:

Application support:

The application support function provides application control interface to SDN applications to access network information details and application-specific behavior of the program

Orchestration:

The orchestration function provides automated network resource controls and management, as well as managing network resource access requests on the basis of a multi-tier management policy or application layer.

The orchestration function provides network infrastructure control and management, such as management of physical and virtual network topologies, network elements and traffic. It integrates with multi-layer management features to manage SDN applications such as user management, service advancement and distribution.

Abstraction:

The Abstraction Function interacts with network resources and gives an overview of network resources, including network capacity and features that support management and orchestration of physical and virtual network resource. This abstraction is based on standard data models and information, and is autonomous of the transport infrastructure that underlies it.

Resource layer:

The resource layer is used to transport and process data packets by the network elements based on the decisions taken by the SDN control layer and distributed through a resource control interface to the resource layer.

Control support:

The control support function interfaces with and maintains the SDN control layer programming through resource-control interfaces

Data transport and processing:

Data transfer and processing function includes data forwarding and data routing functionality.

The data forwarding function manages the incoming data flows to forward them along the data transfer routes which were calculated and computed according to the SDN applications' requirements. The data forwarding functionality is managed by the SDN control layer in order to reduce the data transfer functionality in the resource layer.

4.3 OPENDAYLIGHT

OpenDaylight is a Linux Platform hosting open source SDN controller / framework. It is currently one of the most popular SDN controllers (open source). One of the protocols for the southbound interface is OpenFlow. It is open to everyone, which include end users and customers, and it offers a common platform to work together for SDN objectives and find innovative solutions. Since both the multi-protocol and modular OpenDaylight platform enables users to construct an SDN controller for their particular requirements. It allows SDN to be combined with a fully pluggable controller, interfaces, protocol plug-ins and applications. The central component of the project is the lightweight, pluggable and flexible controller: it's entirely implemented in software and

has its own Java virtual machine (JVM). The interfaces northbound and southbound are well defined and documented APIs and their combinations enable vendors, providers, customers and application developers to use a standard SDN platform that is widely supported.

4.3.1 OpenDaylight Architecture:

The top level perspective of OpenDaylight architecture is shown in Figure 4.8. Therefore, as defined in the following list, it contains five logical layers.

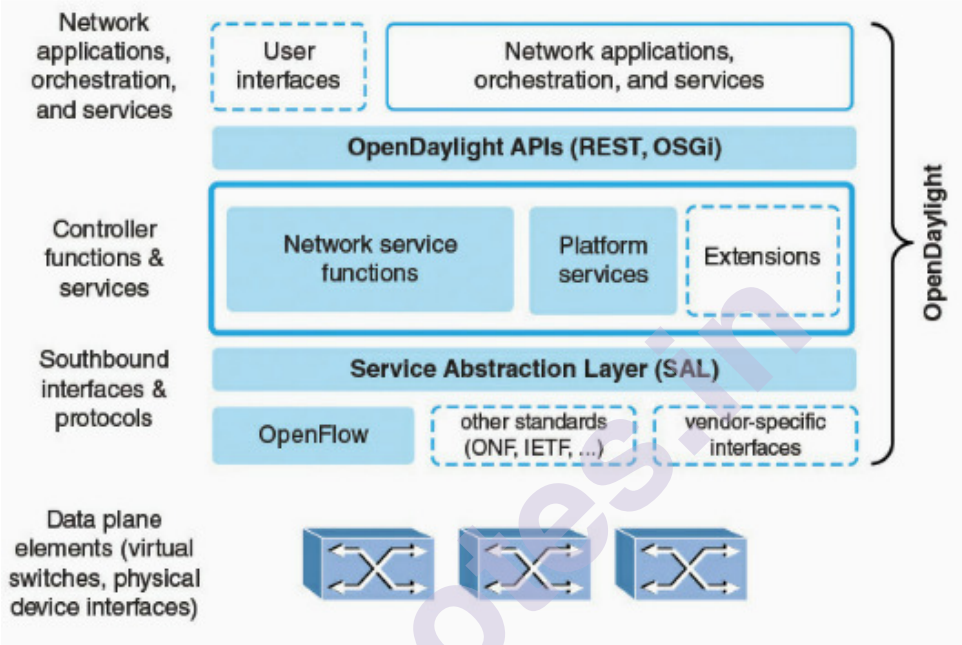


FIGURE 4.8 OpenDaylight Architecture

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

Network applications, orchestration, and services: The top layer includes business and network logic applications which control and monitor the computation of the network. More complex solution orchestration applications for cloud and NFV thread services and engineering network traffic to meet the needs of such environments.

APIs:

A group of standard controller function interfaces for OpenDaylight. The Open Service Gateway Initiative (OSGI) is supported by OpenDaylight. The Northbound APIs support the OSGi framework and bidirectional REST. The OSGi framework is utilized for applications running inside the same address space as that of the controller, whereas the REST (web-based) API for an applications not running in the same address space (or even the same system) as the controller.

Controller functions and services: Functions and services of the SDN control plane.

Service abstraction layer (SAL):

The Service Abstraction Layer (SAL) is the main architecture for OpenDaylight which allows for the abstraction between consumers and producers of services. SAL serves as a broad registry of services offered across different modules and binds them to the appropriate applications. Services or providers modules can register the registry of their APIs. When a customer or an application demands the service through a standardized API, SAL is liable to put the request in a contract, brokered and serviceable by SAL through the binding producer and consumer. The application-driven SAL and the module-driven SAL are two separate ways of implementing this registry.

Southbound interfaces and protocols:

The southbound interface will accept many protocols (such as independent plugins), for instance. OpenFlow 1.0, OpenFlow 1.3, BGP-LS, LISP, SNMP and several more. These modules are attached to the abstract layer of service (SAL) dynamically to decide how the requested service (by applications) is to be implemented independently of the underlying protocol used by the controller and the network devices.

The operation of the SAL is seen in Figure 4.9 .The OSGi framework implies that plug-ins for the available southbound protocols are dynamically linked. Collection of features that can be depended on by control plane services through a services manager of the SAL is a description of the capability of such protocols. In order to map service requests, the service manager maintains a registry. Depending on the service request, SAL maps the necessary plug-ins to interact with a certain network system using the most suitable southbound protocol.

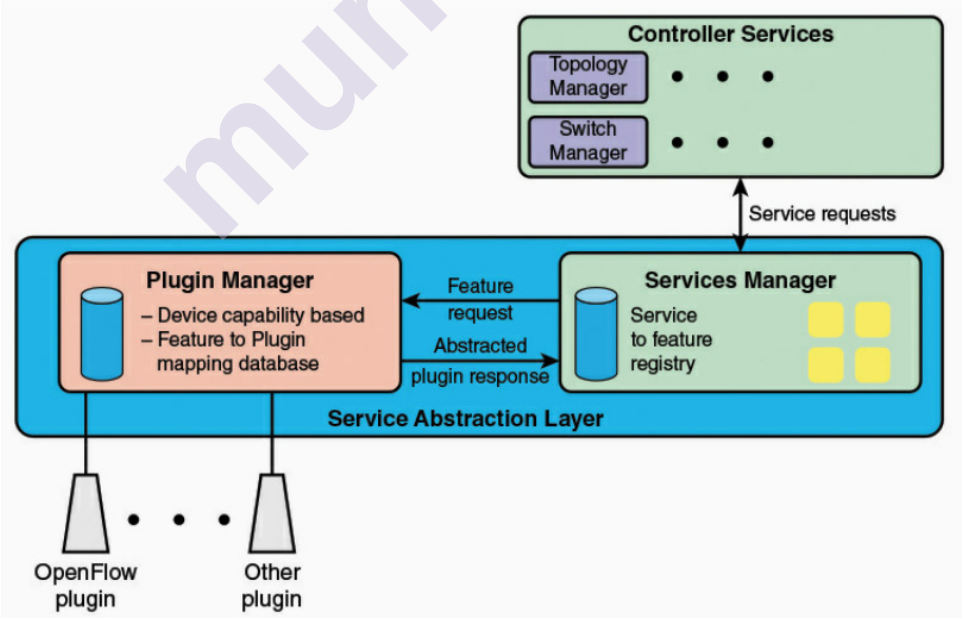


FIGURE 4.9 Service Abstraction Layer Model

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

4.3.2 OpenDaylight Helium:

The OpenDaylight community, the largest of the open source projects supporting SDN controllers and providing documentation. It has given his second stable release, "Helium," which comes with an Apache Karaf container with a new user interface and a customizable installation process. OpenDaylight is the Helium, illustrated in Figure 4.10.

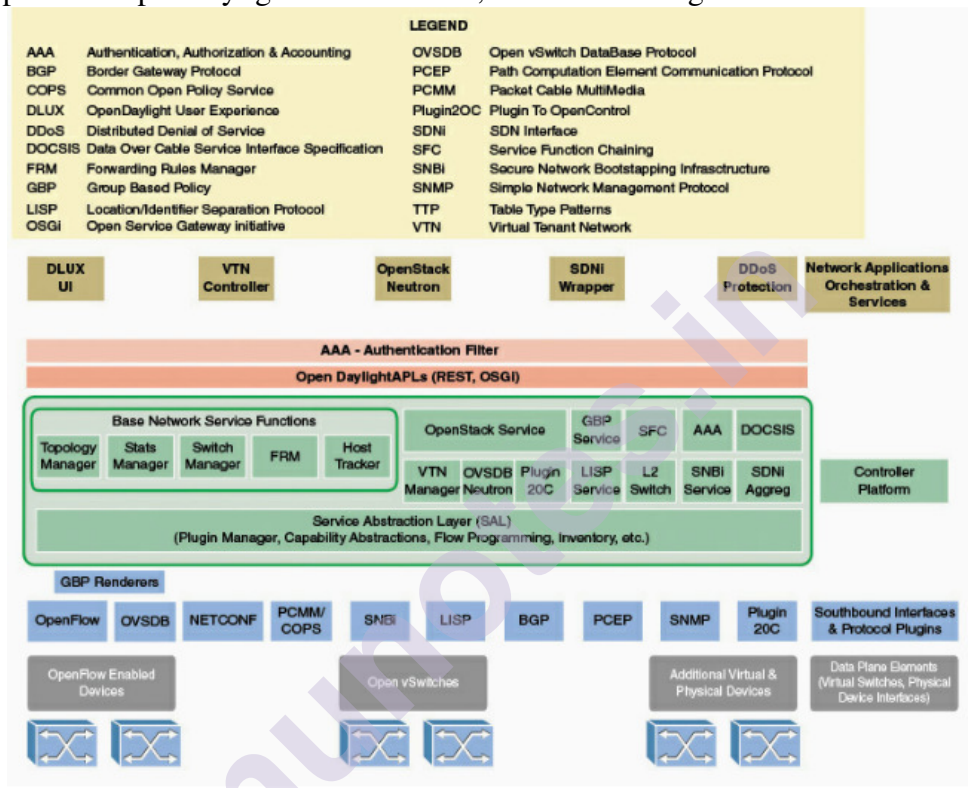


FIGURE 4.10 OpenDaylight Structure (Helium)

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

OpenDaylight architecture:

The components in OpenDaylight Helium include a fully connectable control, interfaces, plugins and protocol applications. There are three main blocks in the helium controller:

Controller Platform:

- The OpenDaylight controller platform
- Northbound applications and services
- Southbound plugins and protocols

A modular architecture (as seen in figure 4.11) is the controller platform and has "northbound" and "southbound "interface. The Northbound interface includes controller services and a collection of standard REST APIs that can be used by applications to manage network infrastructure configuration. Using the authentication and authorization models shown as the top layer of the open-day light architecture in Figure 4.10 you can access the northbound interface.

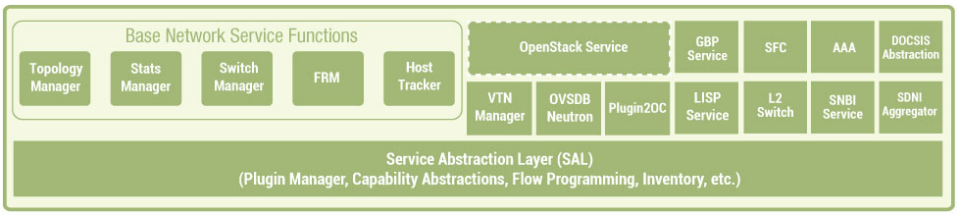


Figure 4.11: Controller components

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

The southbound interface uses protocols to manage the network infrastructure. The southbound level has several plugins which either implement different networking protocols or communicate directly with hardware. The best known configuration and management protocols are OpenFlow, NetConf and SNMP.

The controller platform uses southbound plugins and provides basic network services through a number of managers in the Figure 4.11 base network service functions section including the topology manager, the switching manager, etc. Any custom application can use these network services; for example, OpenStack Networking (Neutron) can use northbound APIs and related components.

Base Network Service Functions:

The following platform managers and components provide the Base network service functions:

Topology Manager: stores and manages networking device information. The topology manager produces the root node in the operating subtree of topology when the controller begins. Then it listens for notifications and updates this sub-tree with topological details, all discovered interconnections and switches. Other modules, such as switch management or device managers' alerts, can also provide important topology information.

Statistics Manager: collects statistics, sends statistical requests to all activated nodes (managed switches), and stores responses in operational sub-trees of the statistics. The Statistics Manager also offers the following information with northbound APIs.

- node-connector (switch port)
- flow
- meter
- table
- group statistics

Further, you can configure the statistical query interval with the Helium release of the Open Daylight controller.

Switch Manager: offers information about network nodes (switches) and node connections (switch ports). The parameters of the control are saved to the Switch Manager data tree as soon as the control discovered network components. The details on discovered nodes and port devices may be obtained by using APIs northbound.

Forwarding Manager Rules (FRM): administers and validates the main forwarding rules (such as OpenFlow rules). In southbound plugins, the forwarding rules manager communication and loads OpenFlow rules into managed switches.

Inventory Manager: queries and changes switch and port information handled by OpenDaylight, ensuring reliable and up-to-date inventory database.

Host Tracker: stores end host information (address of the layer of data, type of switch, type of port, Network address) and offers APIs to retrieve end-node information. Host Tracker can operate dynamically or statically. The Host Tracker uses ARP to monitor the state of the database as complex operations take place. The Host Tracker database is manually supplemented by northbound APIs in static mode.

The Platform Network Service Functions:

The ODL controller provides plug-in oriented services that perform unique SDN functionality -enhancing network functions and other extensions. Some of these services have been listed in the following text:

Affinity Metadata Service: offers an NB API for expressing applications network specifications and communicating the controller workload. The controller can provide network infrastructure to meet these requirements or reduce the resulting workload between endpoints

Virtual Tenant Network (VTN): manager establishes and operates a virtual network of multiple tenants. VTN enables users to build a logical network, no longer the physical network topology.

L2 Switch: offers L2 switching features to include multiple reusable, standardized services such as address tracking, basic spanning tree protocol, modular packet handling and optimal route calculations.

Service Function Chaining (SFC): allows for the specification in an arranged list of the service path for data traffic of a network service chains (like the firewall, the routers, the load balancers).

The Group-based policy (GBP),: through an application-based approach model, distinguishes application connectivity requirements from the underlying information for the network elements. It group network endpoints on the basis of the application requirements and apply the application policy to those classes.

AAA Service: is proposed to provide such a generalized model for AAA functioning in ODL project (specifically, the authentication, authorization and accounting service). The identity of human and computer users is verified by token/claim authentication. Access authorization to resources such as RPCs, notifications, subscriptions and database subsets are reviewed on the basis of role-based access control (RBAC) authorization. Accounting tracks access to all services for various purposes, such as analysis, accounting, diagnostics and security audit.

The Service Abstraction Layer:

SAL allows ODL, as the heart of ODL, to offer a standardized range of services to various modules and network applications and to support many SB protocols (by means of the SB plugins. Device Discovery is a service that SAL provides and uses to form the topology network and to construct element capabilities by the Topology Manager. Based on the SB plugin features, most SAL services are developed. The SAL shall fulfill the service requested for a given switch, regardless of the SB protocol underlies. The plugins of the NB and SB may be service producers or users of services or both. The SAL acts as a register of large services where producers advertise their services via their APIs. When a customer asks for an advertised service via a generic API, SAL links both producer and consumer. Initial SSL was coded with an API-driven SAL architecture by the ODL developers. API-driven SAL or AD-SAL, the developers were required to code the SAL APIs (to route services requests from consumers to suppliers), and adapt the API (if NB is not identical (Service abstract) to its respective SB (Protocol) API) functionality;

The Southbound Interface and Protocols Plugins:

SB protocols are used to ensure secure communication between the controller and network components. These protocols allow the network elements to be managed, configured and monitored. ODL supports SB protocols multiple (via SB plugins). ODL will support heterogeneous

networks and ensure interoperability with other technologies and between suppliers through these SB Protocols. Below are some of the approved SB plug-ins (such as SB protocols):

OpenFlow Plugin: implements the specifications for the OF protocol as it evolves. ODL supports OF 1.0, 1.3 and table-type patterns currently (TTPs) that allow the OF and OF controllers to negotiate and agree on a number of features provided with OF 1.1+ versions.

Open vSwtich Database (OVSDB) Plugin: models OVSDB that manages and configures open vswitches. OVSDB has been released on Ethernet switches firmware for the most recent time.

SNMP Plugin: suggested that a Simple Network Management Protocol SB Plugin be built for controlling Ethernet off-shelf commodity switches. Flow configuration can be done over the forwarding table, ACL, and VLAN table on these switches.

The Java based Border Gateway protocol and path element computation protocol are implemented by BGP-LS/PCEP plugins (PCEP). BGP-LS plugin facilitates the distribution of Link state BGP and has been considered a source of L3 topology information for ODL while the PCEP plugin establishes paths to the entire underlying network.

A plugin built to allow ODL to manage and configure the support of NETCONF protocol in its Network Configuration Protocol (NETCONF) protocol. Furthermore, it helps to discover certain elements and their functions and offers all NETCONF protocol functionalities.

The Network Applications and Services:

The network applications and services that control, manage and monitor's entire network arise out of the top layer of ODL. Many of these applications and services are associated with the respective network platforms, including the VTN coordinator and the VTN manager. This layer also comprises orchestration services, which are used by developers, according to environmental criteria such as the NVF and the cloud. The central platform layer displays open NB APIs that are used in the applications of this layer. ODL supports both OSGi and NB API bi-directional REST APIs. The difference between REST (HTTP-based) and OSGi frameworks. This means that the first is used in applications running in the same space as the ODL, while this second is used in applications not running. There are some of the network implementations of ODL:

The new web-based user interface (UI) for the second ODL Release "Helium" is Open Daylight User eXperience (DLUX). DLUX is an interactive, more dynamic UI built as a simple front end technology with

Angular JS (a JavaScript client-side framework). Only NB (REST) APIs are exposed to the ODL and then consumed via UI.

An external application providing REST API's to construct VTN and coordinating virtual networks across multiple ODL controllers is VTN Coordinator. It also interacts with the user setup plugin of the VTN Manager.

In order to allow inter-sDN control communication, SDNi Wrapper is part of the ODL-SDNi program. The SDNi Rest API is used to collect the controller-specific information. In order to gather topology and other associated information, SDNi aggregator (a platform network service) communicates with controllers switch and host tracker statistics. The SDNi REST API collects the aggregated SDNi aggregator information, which is then collected by the SDNi Wrapper.

Distributed Denial of Service (DDoS) attacks: are an application to detect and mitigate DDoS protection. The NB REST API is used for monitoring and transfer of attacks to predefined attack mitigation systems (AMSs) the computing of secure traffic

4.4 REST

REST is REpresentational State Transfer, and API stands for Application Program Interface. REST is an architectural software style that sets the rules to create web services. The REST-architectural type web services are known as the RESTful web services'. It allows for systems to use a standardized, default collection of standards to access and manipulate web resources. REST-based systems communicate via the Hypertext Transfer Protocol (HTTP).

A Restful system consists of a:

- Client who requests for the resources.
- Server who has the resources.

It is necessary to construct a REST API in line with industry standards that will make development easier and customer adoption easier.

4.4.1 REST Constraints:

Architectural constraints of RESTful API: Six architectural constraints are listed below which make any web service:

- Uniform Interface
- Stateless
- Cacheable

- Client-Server
- Layered System
- Code on Demand

REST architecture's only optional constraint is code on demand. If a service infringes any other constraint, it cannot be strictly called RESTful.

Uniform Interface: This is the main restriction between REST API and Non-Rest API, indicating that a uniform interaction with a server should be defined regardless of device or application type like uniform interface (website, mobile app).

Representation Manipulation of the resource: Client has resource representation and contains sufficient information to change or remove the resource on the server if permission is granted. Example: User usually receives a user ID for a user list request and then use it to remove or change the user.

Self-descriptive Messages: Each message contains enough information to explain the way the message is handled so that a server can quickly analyze your request.

Hypermedia as the Engine of Application State (HATEOAS): Links for each response must be provided so that clients can simply detect other resources.

Stateless: It means that the required state is contained within the request itself and that nothing relevant to the session will be registered by the server. In REST, the client must provide all information for the server, whether as part of query parameters, headers or URIs, to complete the request. The statelessness allows for enhanced availability as the server has no session state to maintain, update, or communicate. The client has a downside if he has to send too much information to the server so that network improvement can be minimized and more bandwidth is needed.

Cacheable: Any response should include whether or not the response is cacheable and how long responses on the client side can be cached. For any subsequent request, the client will return the data from its cache and will not need to redirect the request to the server. Some client-server interactions are avoided, thereby increasing usability and efficiency in part or entirely by a well-managed caching. However, sometimes it is possible for users to get stalled results.

Client-Server: A client-server architecture should be used in the REST application. A client is a resource request person who does not deal with data storage that is internal for every server and is a resource holder who does not deal with the interface of the user or the use state. They can grow

autonomously. The client need not know anything about business logic and the server needs to know nothing about the front user interface.

Layered system: Several layers must be designed into an application architecture. Each layer has no knowledge of anything but the immediate one, and between the client and the end server there can be several intermediate servers. Intermediary servers will allow load balance and provide shared cache solutions to increase system availability.

Code on Demand: This is an option. This also means that servers can provide the client with executable code. For example, compiled components such as Java applets, client-side scripts such as JavaScript can include code on demand.

4.4.2 Example REST API:

It is useful to look at an example to get a feel for the structure of a REST API. In this section, we are discussing a REST API for the Ryu SDN network operating system northbound interface. The specific function of Ryu's API Switch Manager is developed for open-flow switches.

Each and every function that could be conducted by the switch manager on behest of an application is allocated a URI. Consider, for instance, the function to describe all entries in a specific switch's group table.

The URI for this switch function is the following:
/stats/group/<dpid>

Where (stats) statistics apply to a set of APIs for the statistics and parameters of the switches retrieval and update, the group denotes be the function name, and the unique identifier of the switch denotes be the <dpid> (Data Path ID). To invoke Switch 1's function, the application sends a command to the switch manager through the REST API:
GET http://localhost:8080/stats/groupdesc/1

This command's for the localhost part means that the application is operating on the identical server as the Ryu NOS. The URI will be a URL providing remote access via HTTP and a web application when the application is remote. The switch manager addresses this command by means of a message whose message body contains the dpid and then a set of value blocks, one per category group in the dpid switch. The following values are:

- type: All, select, fast failover, or indirect
- group_id : Group table entry identification.
- buckets: A standardized field composed of the following subsections:

- Weight: Bucket's relative weight (only for select type).
- watch_port: Port whose state impacts if the bucket is alive (only required for fast failover groups).
- watch_group: Group whose state affects if this bucket is live (only required for fast failover groups).
- actions: A list of actions, possibly null.

Once for each table of group entry, the buckets in the message body are repeated.

Table 1 describes the API functions and the parameters used by the GET message type for the processing of switch statistics. There are many functions using the POST message type, under which a set of matching parameters is included in the request message body.

Request Type	Response Message Body Attributes
Get all switches	Data path ID
get switch description	Datapath ID, manufacturer description, hardware description, software description, serial number, human readable description of data path.
Get all flows stats of switch	Datapath ID , Length of this entry, Table ID, Time flow has been alive in seconds, Time flow has been alive in nanoseconds beyond duration_sec, Priority of the entry, Number of seconds idle before expiration, Number of seconds before expiration, flags, cookie, packet_count, byte_count, Fields to match, actions
Get aggregate flow stats of switch	Datapath ID, packet_count, byte_count, flow_count
Get ports stats	Number of received packets, Number of transmitted packets ,Number of received bytes, Number of transmitted bytes, Number of packets dropped by RX, Number of packets dropped by TX, Number of receive errors,Number of transmit errors, Number of frame alignment errors, Number of packets with RX overrun ,Number of CRC errors, collisions Number of collisions, ,Time port has been alive in seconds , Time port has been alive in nanoseconds beyond duration_sec
Get ports description	Datapath ID, Port number, Ethernet hardware address, Name of por, Config flags, State flags, Current features,

	Advertised features, Supported features, peer features advertised by peer, Current port bitrate , Max port bitrate
Get queues stats	Datapath ID, Port number,Queue ID, Number of transmitted bytes, Number of transmitted packets, Number of packets dropped due to overrun, Time queue has been alive in seconds, Time queue has been alive in nanoseconds beyond duration_sec.
Get groups stats	Datapath ID, Length of this entry, Group ID ,Number of flows or groups that directly forward to this group, Number of packets processed by group,Number of bytes processed by group, Time group has been alive in seconds, Time group has been alive in nanoseconds beyond duration_sec, bucket_stats, Number of packets processed by bucket, Number of bytes processed by bucket,
Get group description	Datapath ID, type, group_id,buckets(weight, watch_port, watch_group, actions)
Get group features	Datapath ID, types,capabilities, Maximum number of groups for each type,actions values supported
Get meters stats	Datapath ID ,Meter ID, Length in bytes, Number of flows bound to meter, Number of packets in input, Number of bytes in input, Time meter has been alive in seconds ,Time meter has been alive in nanoseconds beyond duration_sec band_stats, Number of packets in band, Number of bytes in band.
Get meter config	Datapath ID ,flags ,Meter ID bands (type, rate, burst_size)
Get meter features	Datapath ID , Maximum number of meters, band_types values supported, Maximum bands per meters, Maximum color value

TABLE .1. Ryu REST APIs for Retrieving Switch Statistics Using GET

The API switch manager also includes switch parameter updates parameters. Everyone uses the POST message type. In this scenario, the message body of the request comprises the parameters and the modified values. API update functions are mentioned in Table 2.

Request Type	Response Message Body Attributes
Add a flow entry	Datapath ID (int) cookie cookie_mask Table ID idle_timeout hard_timeout priority buffer_id flags match actions
Modify all matching flow entries of the switch	Datapath ID (int) cookie cookie_mask Table ID idle_timeout hard_timeout priority buffer_id flags match actions
Delete all matching flow entries of the switch	Datapath ID (int) cookie cookie_mask Table ID idle_timeout hard_timeout priority buffer_id flags match actions
Delete all flow entries of the switch	Datapath ID
Add a group entry to the switch	Datapath ID ,type, group_id,buckets(weight, watch_port, watch_group, actions)
Modify a group entry to the switch	Datapath ID,type, group_id,buckets(weight , watch_port, watch_group, actions)
Delete a group entry to the switch	Datapath ID , group_id
Add a meter entry to the switch	Datapath ID ,flags ,Meter ID bands (type, rate, burst_size)
Modify meter entry to the switch	Datapath ID ,flags ,Meter ID bands (type , rate , burst_size)
Delete meter entry to the switch	Datapath ID,Meter ID

TABLE .2. Ryu REST APIs for Update Switch Statistics Filtered by Fields Using POST

4.5 COOPERATION AND COORDINATION AMONG CONTROLLERS

In addition to northbound and southbound interfaces. , A standard SDN controller is provided with an East/West bound interface that allows communication between various SDN controllers as well as other networks, No major developments have been made on open source or standard East/West protocols or interfaces. This section discusses main problems with the configuration of the East/West bound interface.

4.5.1 Centralized Versus Distributed Controllers

A crucial consideration for the architectural design is if either the dataplane switches will be controlled by a single centralized controller or a distributed controller set. A central controller is a single server which manages all network data plane switches.

The use of a single control unit to manage all network devices in a large enterprise network would prove unmanageable or unnecessary. A more likely circumstance is to split the network into a series of nonoverlapping SDN domains, also known as SDN islands (Figure 4.12), managed by distributed controllers that are the operator of a large enterprise or carrier network. The following list contains reasons for using SDN domains.

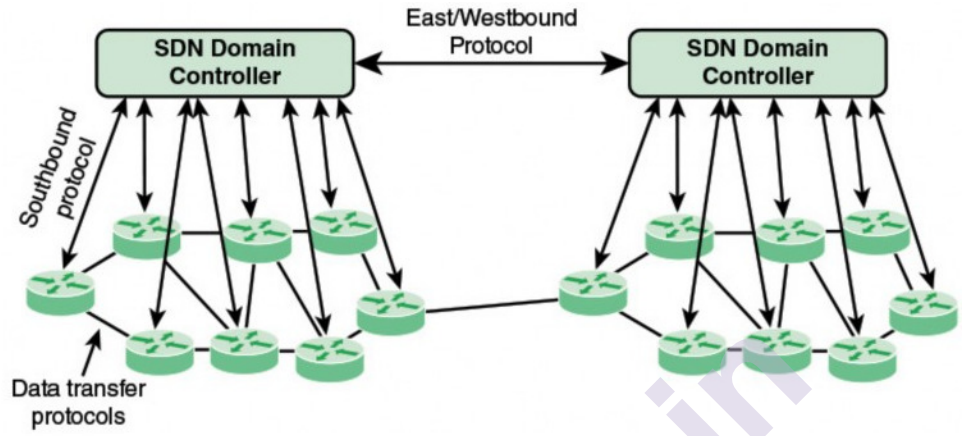


FIGURE 4.12 SDN Domain Structure

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

Scalability:

There is a small number of devices that an SDN controller can handle. Therefore, many SDN controllers can involve a relatively large network.

Reliability: The use of multiple controllers eliminates the possibility of single point of failure.

Privacy:

A carrier may decide in various SDN domains to enforce different privacy policies. For example, a domain may be dedicated to a variety of clients who have their own very customized privacy rules, which would require that any of the network data within this domain should not be revealed to an external entity (for example, network topology.

Incremental deployment:

The network of the transporter will consist of parts of conventional and new infrastructure. The network is divided into many, independently managed SDN domains that enable flexible iterative deployment.

In a very little or widespread region or a mixture of both, distributed controls may be co - located. The proximity controllers provide high

performance and are ideal for data centers while the distributed controllers handle networks with different locations.

In general, controllers are horizontally distributed. This implies that each controller administers an unrelated subset of the switches in the data plane. It is also possible to provide a vertical architecture in which control tasks are spread through various controllers according to such criteria as network view and locational requirements.

A protocol is necessary for communication between controllers in a distributed architecture. A proprietary protocol could in theory be used in this regard, but for the sake of interoperability, an open or standard protocol would obviously be preferable.

For a distributed architecture, functions connected to the East / West bond interface are to maintain a divided or clustered network topology and parameters database and to monitor / monitor functions. Lastly, it involves the monitoring of a controller alive and the arrangement of alterations in the assignment of control switches.

4.5.2 High-Availability Clusters:

A multi-computer architecture comprising redundant network nodes to provide a secondary or back-up services in the absence of a primary service. Such clusters create redundancies in the computer systems so that individual failure points can be removed and multiple network links, redundant data storage volumes, doubled power supplies and other back-up components and capabilities are integrated.

ODL Helium has HA built in and Cisco XNC and the Open Network controller have HA functionality (up to five in a cluster).

4.5.3 Federated SDN Networks:

The distributed SDN architecture mentioned in the above paragraphs refers to a system of SDN domains, all of which belong to one corporate network. The domains can be colloquialized or on different sites. Each case is managed by one specific network management feature for the management of all data plane switches.

SDN networks owned and operated by various organizations, are also able to cooperate via Eastern/Westbound protocols. -Figure 4.13 shows the potential for cooperation between inter-SDN controllers

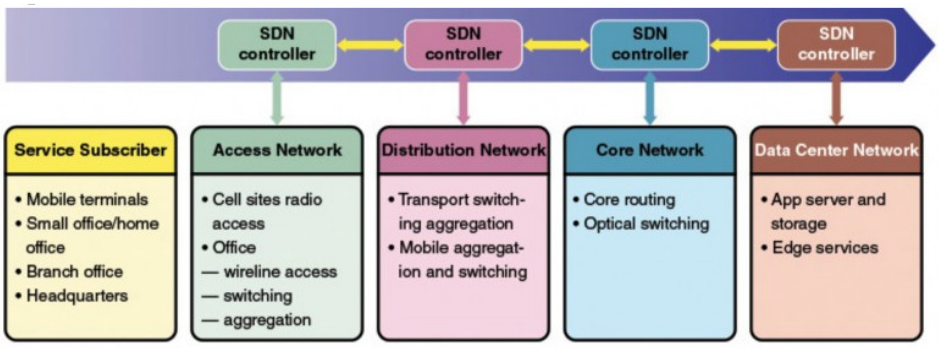


FIGURE 4.13 Federation of SDN Controllers

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

This configuration involves a range of service subscribers to a cloud-based data center network. Subscribers are networked across an access, delivery and core networks hierarchy to the service network. All these intermediate networks may be run by or include other organizations. In this latter scenario, all networks must share similar conventions for plane parameters such as service quality (QoS), policy information and routing information if implementing SDN.

4.5.4 Border Gateway Protocol:

A Border Gateway Protocol (BGP) is standard exterior gateway protocol for the exchange of route and accessibility information among independent (AS) systems on the Internet.

Exterior router protocol (ERP):

Exterior routing protocols serve for exchanging routing among autonomous systems. BGP allows routers, known as standard gateways, to collaborate on the exchange of routing information in different autonomous systems. The protocol works with messages sent via TCP connections. BGP-4 is regarded as the current version.

BGP involves three functional procedures:

- Neighbor acquisition: Every router tries to connect to each of its neighboring routers by sending Neighbor Acquisition Request messages. A neighbor hearing a request will check with a neighboring acquisition to say that he acknowledges the request and wants to communicate. It may refuse the acquisition by replying to the message of rejection of the neighbor acquisition. In order for an EGP connection between a couple of neighbors, each one has to acquire a confirm message in the first case.

- Neighbor reachability: After a neighbor has been acquired, a router inspects to ensure that the neighbor is consistently available and functioning. This is achieved by sending a message from EGP Hello to any neighbor for whom a connection was created. The neighbor responds with a message I Heard You (IHU). The BGP Keepalive message is somewhat similar but is used in matched pairs.

- Network reachability: Router periodically sends poll messages to each neighbor. The neighbor responds with an update message includes details on the networks it can reach. These data are used for updating the system routing tables sent by the poll.

4.5.5 Routing and QoS Between Domains:

The controller establishes a BGP connection to each of the neighboring routers for routing outside a controller domain. Figure 4.14 shows a configuration of two SDN domains, only connected by a non-SDN AS.

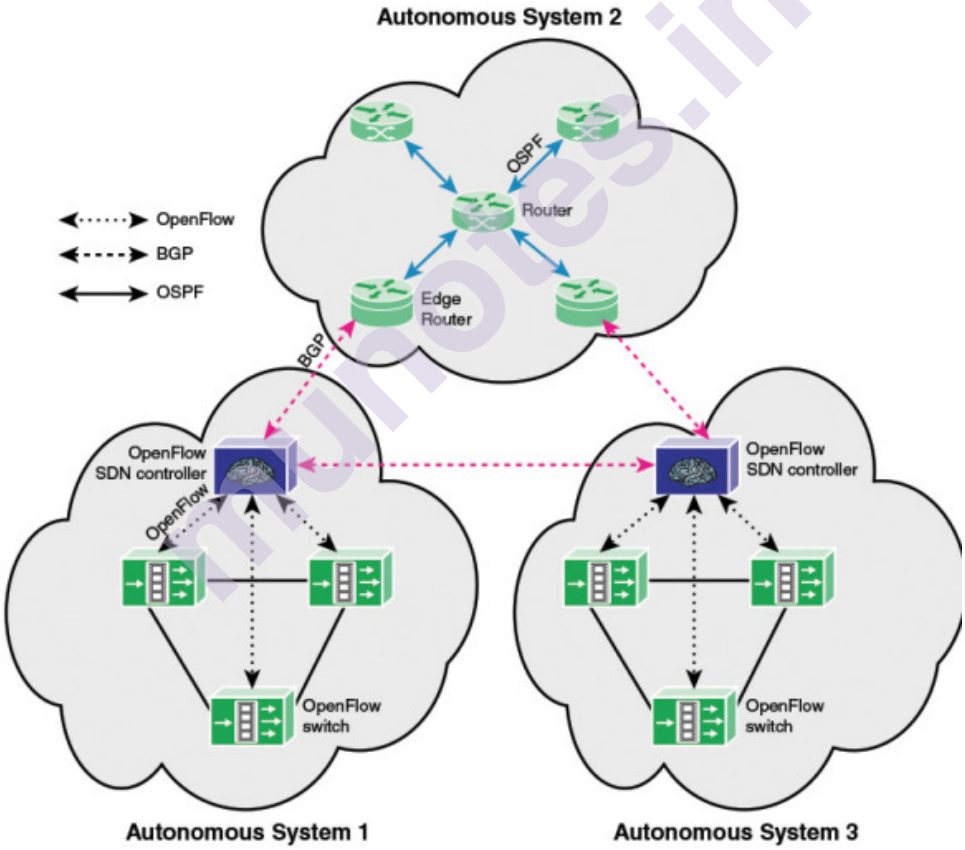


FIGURE 4.14 Heterogeneous Autonomous Systems with OpenFlow and Non-OpenFlow Domains

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

OSPF is used for internal routing within the non-SDN AS. No SDN domain includes OSPF; instead, the routing information is given from each data plane switch using the southbound protocol to the centralized controller (in this case, OpenFlow). BGP is used to share information between each SDN domain and the AS, for example:

Reachability update: Inter-SDN domain routing is enabled by sharing accessibility information. This allows a single flow to travel through multiple SDNs and each controller can choose the most suitable network path.

Flow setup, tear-down, and update requests: Controllers coordinate flow set up requests containing information on multiple SDN domains, such as path, QoS, etc.

Capability Update: In addition to the device and software capabilities within the domain, controllers share information on network-related capabilities, such as bandwidth, QoS and so on.

In relation to figure 4.14, some additional points should be observed:

Each AS is shown as a cloud with linked routers and a control system for an SDN domain. The figure 4.14 shows each AS. The cloud is an internet, so that the network between two routers is an internet network. Likewise, the interface between two adjacent autonomous systems is a network which can be part of one of two adjacent, or a different network.

For an SDN domain, instead of a data plane router, the BGP function is implemented on the SDN controller. This is because the controller controls the topology and makes routing decisions.

A BGP connection between autonomous systems 1 and 3 is shown in the figure 4.14. These networks might not be connected directly to a single network. However, it is beneficial to share additional information relevant to SDN if the two SDN domains form part of a unique SDN system, or if they are federated.

4.5.6 Using BGP for QoS Management:

The best-effort connectivity is a standard procedure for inter-AS interconnections primarily. In other words, traffic forwarding between autonomous systems have not distinguishing of traffic classes and therefore has no assurance of forwarding. Network providers are usually required to restore every IP Packet Traffic Class marking to zero on the AS ingress Router, removing any distinction in traffic. Few providers conduct higher classification at the ingress to estimate and balance transmission requirements their AS internal QoS forwarding policy. The

cross-domain traffic cannot depend on a standardized class collection, no standardized markings (class encoding) or a standardized forwarding behavior. RFC 4594 however offers a series of "quality standards" for these parameters. Network suppliers make separate and uncoordinated decision-making on QoS policy. This specific statement does not include current separate agreements providing a quality interconnection that provides with strict QoS guarantees. Such SLA Agreements are bilateral or multilateral nevertheless, and do not provide a basis for an overall interconnection of 'better than best effort.

IETF is working on a standardized QoS marking scheme with BGP. Additionally, SDN providers are using the extensible BGP to incorporate their own capabilities. In the other case, SDN controller interactions in different domains by using BGP that include the steps shown in Figure 4.15 and defined in the following list.

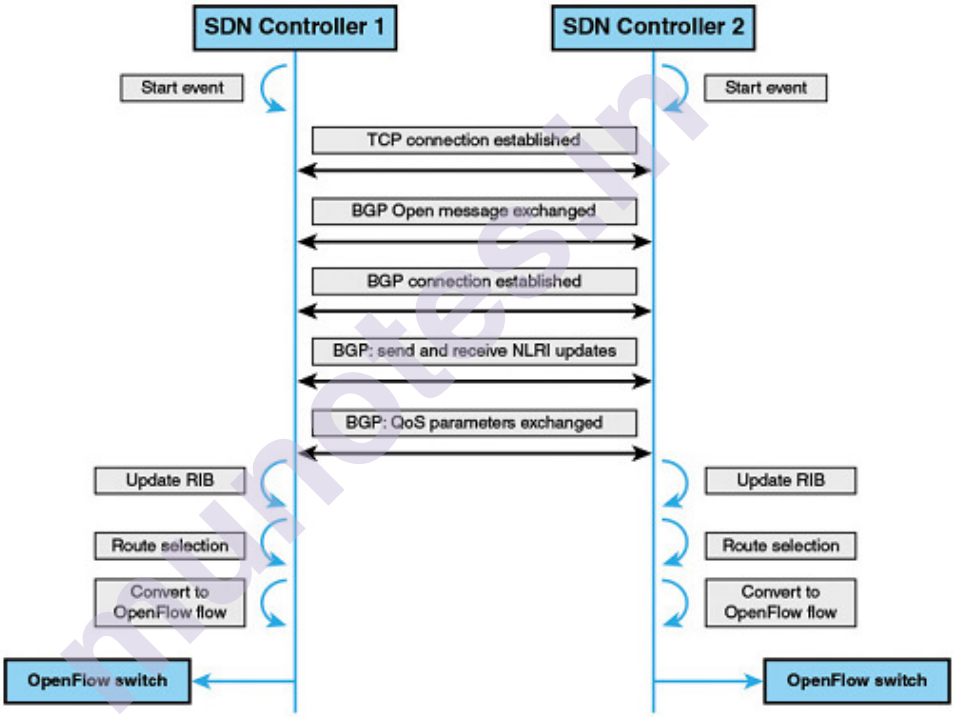


FIGURE 4.15 East-West Connection Establishment, Route, and Flow Setup

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

1. The SDN controller should be customized with BGP and neighboring BGP enables with location information.
2. BGP will be activated with in controller via a start or activation event.
3. A controller BGP entity tries to make a TCP connection with each neighboring BGP entity.

4. The BGP entity of the controller exchanges open messages with the neighbor when a TCP link is established. Capability information with Open messages is exchanged.
5. The exchange shall complete when a BGP connection is created.
6. Update Messages are used to exchange NLRIs (network layer reachability information), showing what networks the entity can reach. The selecting of the most adequate data route among SDN controllers uses Reachability information. The information collected by NLRI is used for updating the controller's Routing Information Base (RIB). This allows the controller to set the based on the flow data on the data plane switches.
7. The update message, like available capacity, could even often been utilized to share QoS information.
8. Route selection is made if the BGP Process Decision allows more than one path available. Packets can travel effectively among two SDN domains when the direction has been determined.

4.5.8 IETF SDNi:

IETF has built a drafted specification that specifies common criteria for coordinating flow configuration and exchanging information about accessibility across multiple domains known as SDNi.SDNi specification describes have not an East/Westbound SDN protocol however incorporates some fundamental concepts for developing such a protocol.

SDNi functionality includes:

Application-based coordinate flow configuration with details such as path requirement, QoS and level of service agreements across multiple SDN domains.

To allow inter-SDN routing, exchange reachability information. It will enable one unified flow to cross several SDNs. When many of these paths are available, each controller determines the most suitable path.

SDNi relies on what resources and features the various controllers in each domain have available and manage. It is therefore necessary to incorporate SDNi in a concise and open manner in order to support new functionality provided by various controller types.

Transiently, the message types for SDNi include:

- Reachability update
- Flow setup/teardown/update request (including application capability requirement such as QoS, data rate, latency, and so on)

- capability Update (including network-related capabilities, such as data rate and QoS, and system and software capabilities available inside the domain)

4.5.9 OpenDaylight SNDi:

The OpenDaylight architecture includes the SDNi capability to connect and exchange topology information with multiple OpenDaylight federated controllers across the network. This feature seems consistent with the SDNi function's IETF specification. The SDNi application on an OpenDaylight controller is composed of three components, as shown in Figure 4.16 and shown in the following list.

SDNi aggregator: Northbound SDNi plug-in functions as an aggregator to gather information from the network such as topology, statistics and host **identifiers:** This plugin can be built to fulfill the demand for the sharing of network data through SDN controllers.

SDNi REST API: The REST APIs from the northbound plug-in (SDNi aggregator) collect added information.

SDNi wrapper: It is the responsibility of the SDNi BGP wrapper to exchange and gather information from/ to federated controllers.

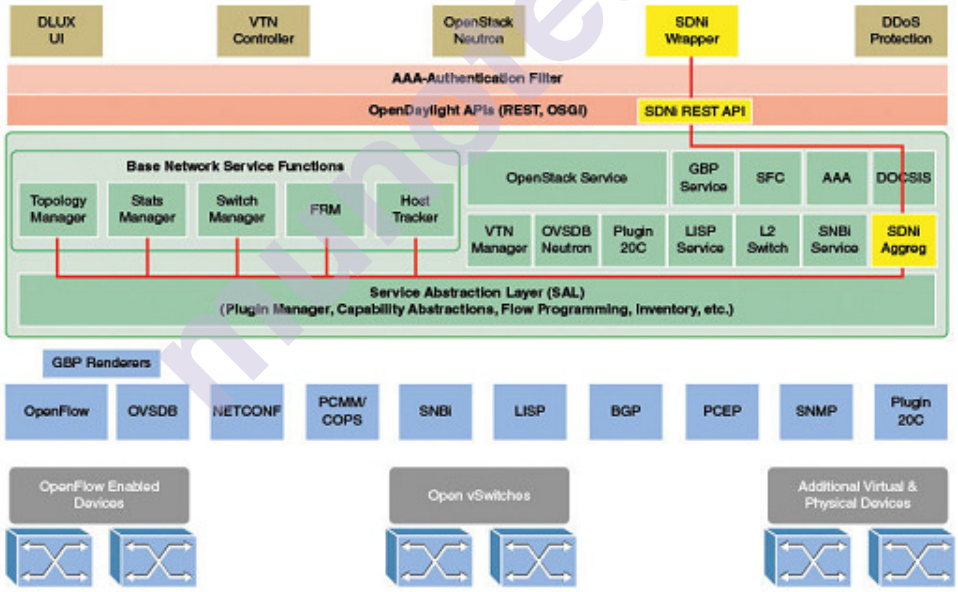


FIGURE 4.16 SDNi Components in OpenDaylight Structure (Helium)

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

Figure 4.17 illustrates the interrelationship between the components by analysing the SDNi wrapper in greater detail. On account of queries

through the REST API, the SDNi aggregator gathers statistics and parameters from the basic network service functions. The focus is on the implementation of the Border Gateway Protocol by OpenDayslight (BGP).

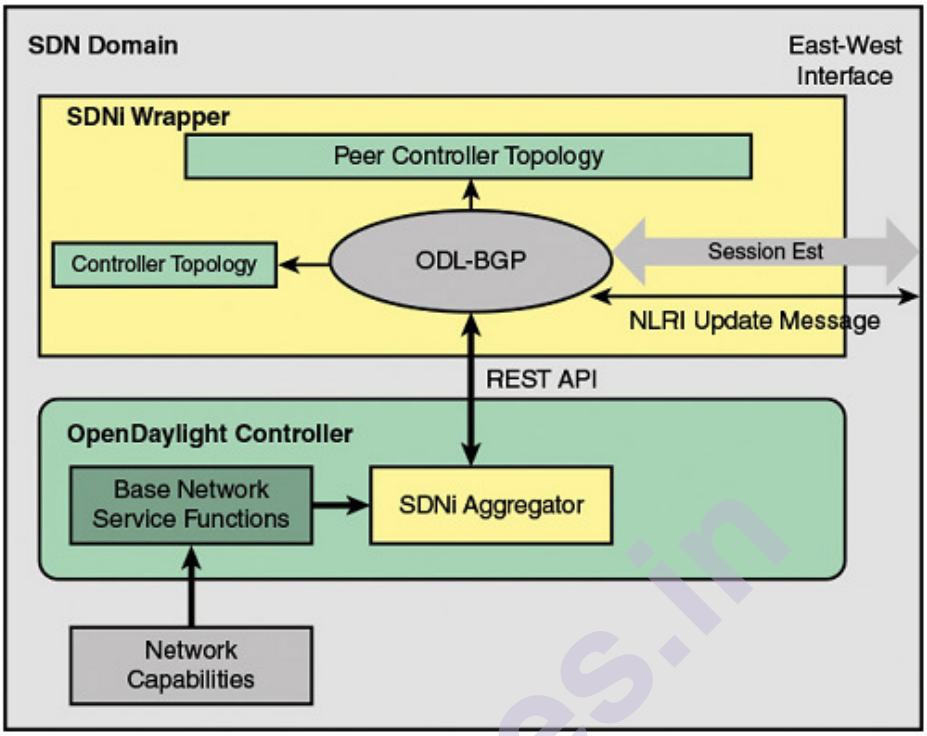


FIGURE 4.17 OpenDaylight SDNi Wrapper

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

SDN Application Plane

Objectives: after you have studied this chapter, you should be able to

- Provide an overview of SDN application plane architecture
- Define the abstraction layers of network services.
- List and describe three forms of SDN abstraction.
- List and describe six important SDN application areas.

SDN includes the ability of software systems to dynamically schedule individual network devices so that the network as a whole is managed. The SDN control plane offers features and services for quick network development and deployment. Although the SDN data and control planes are well known, the design and complexity of the application plane are much less accepted. Applications that explicitly handle the management and control of networks are a minimum part of the implementation strategy. Such applications, or even categories of such applications, are not agreed to. Moreover, the application layer can include tools and services for general purpose network abstraction which could be viewed as part of the control plane functionality.

4.6 SDN APPLICATION PLANE ARCHITECTURE

The application plan includes applications and services that assess, monitor and control resources and behavior of the network. These applications communicate through the application control interfaces with SDN control systems, so that the behavior and properties of network resources can be automatically modified by the SDN control layer. The SDN application programming makes use of the abstract view of network resources that are delivered by the SDN control layer, through the application-control interface, of data and information models. This chapter gives an overview of the functionality of the application plane shown in figure 4.18. The components are evaluated in this figure using a bottom-up approach and following sections include descriptions of particular areas of application.

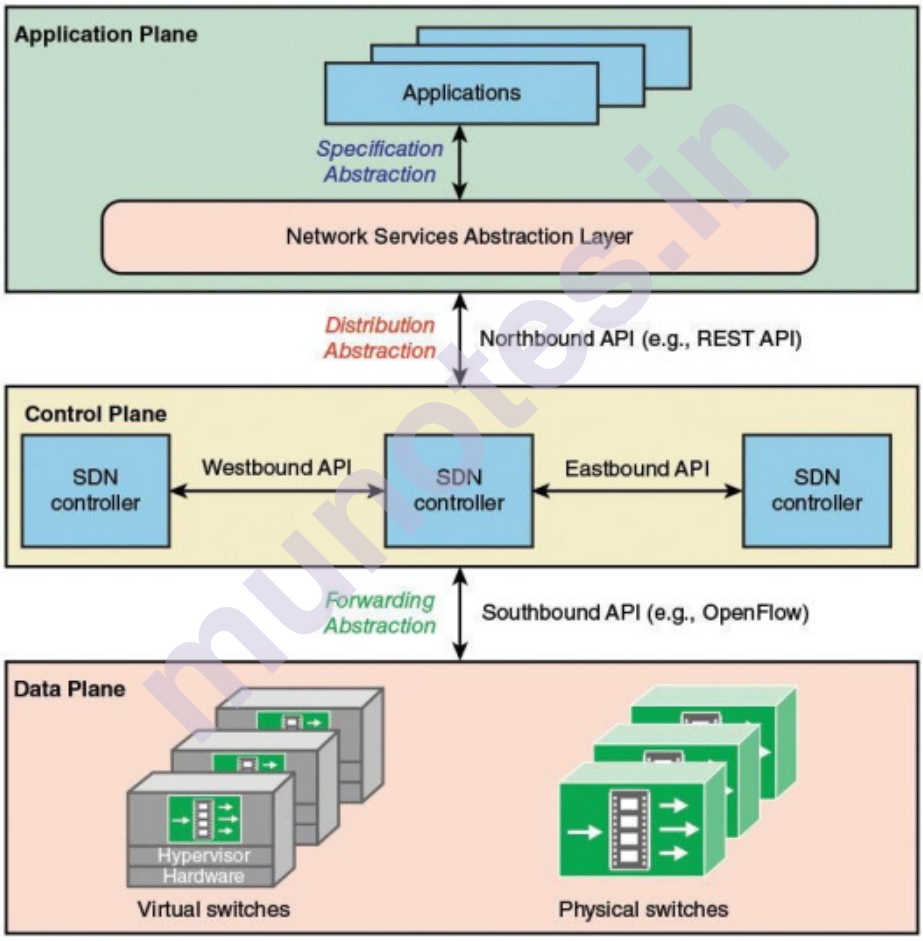


FIGURE 4.18 SDN Application Plane Functions and Interfaces

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

4.6.1 Northbound Interface:

The northbound interface facilitates applications to utilize control plane functionality and services despite knowing about the specifics of the

network switches supporting it. The northbound interface typically offers an abstract view of the software-controlled network resources of the SDN control plane.

Figure 4.18 shows a local or a remote interface for the northbound interface. The SDN applications operate on the identical server for a local interface as the control planes application (controller network operating system). Optionally, applications could even be operate on external machines and the northbound interface is an API that connects the applications on a central server running Network Operating System (NOS) controller. It is likely that both architectures will be implemented. The REST API for the Ryu SDN network operating system is an example of a northernbound interface.

4.6.2 Network Services Abstraction Layer:

The Network Services Abstraction Layer (NSAL) gives users access to applications and services from control, administration and application plans services. The term SAL is overloaded, because it is used in various contexts ranging from the design of systems to service-oriented architectures.

- The placement of such a layer in the SDN architecture indicates multiple functional concepts:
- This layer might give an abstract look at network resources which hides the details of the relating data plane devices.
- This layer could provide a larger perspective of the functionality of control planes so that apps that operate across a number of operating network controller systems could be written.
- This functionality is analogous to the hypervisor or virtual monitor, which provides integrated applications from the underlying OS and the underlying hardware.
- This layer which give a virtualization of the Network which allows various views of the network architecture on the data plane.
  Probably, the abstraction layer for network services may be seen as part of the northbound interface that integrates the functionality of the control plane or the application plane.

Network Applications:

Many network applications for an SDN may be implemented. Various literature SDN surveys have provided different lists of SDN-based network applications and also various basic categories. Figure 4.18 comprises six categories covering most SDN applications.

4.6.4 User Interface: The UI allows a user to customize parameters in SDN apps and communicate with applications supporting user

communication, two possible interfaces take place afterwards. A user who is co - located with the Application SDN server can use the keyboard/display of the server.

4.7 NETWORK SERVICES ABSTRACTION LAYER

Abstraction refers to the amount of information noticeable to higher levels on the lower levels of the model. More abstraction implies less detail and less abstraction implies more detail.

A process for abstracting a high-level request is a low level command needed to execute the request. One of these processes is an API. It preserves the specifics of the implementation of lower software abstraction at a higher level. A network abstraction describes the fundamental characteristics of network organisations such as switches that bind ports and flows, which allows network programmes to concentrate on the functionality required and not have to programme a comprehensive action.

4.7.1 Abstractions in SDN:

Scott Shenker, a member of the Open Network Foundation (ONF) Board and OpenFlow Research, says that SDN can be specified in three main abstractions: forwarding, distributions and specifications, as shown in Figure 4.19.

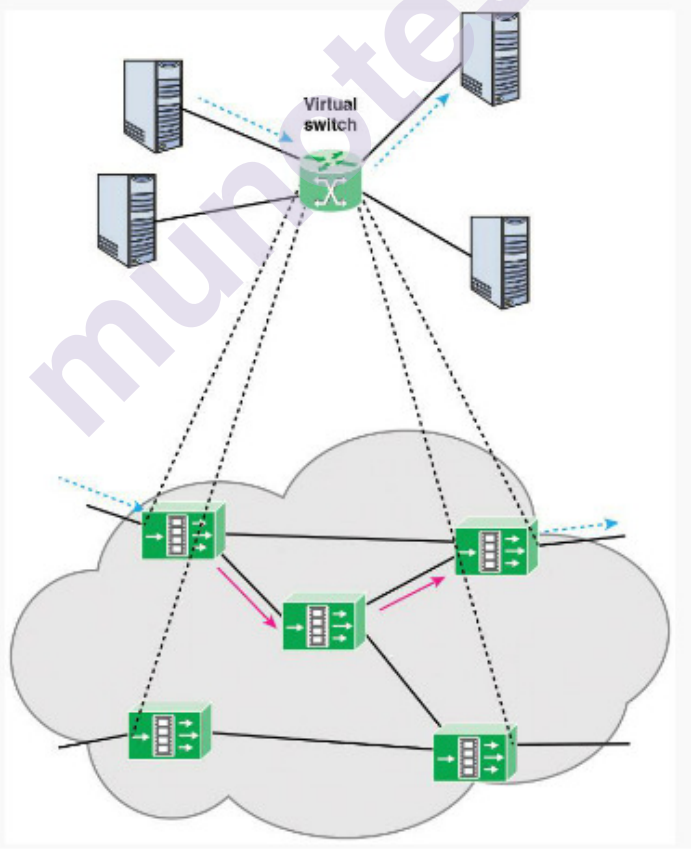


FIGURE 4.19 SDN Architecture and Abstractions

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

The forwarding abstraction enables a control program, which defines the forwarding behaviour of the data planes while hiding specifics of the switching hardware. This abstraction facilitates the forwarding function of data plane. It offers flexibility and distribution neutrality by abstracting from the forwarding hardware.

An example of forwarding abstraction is the openflow API.

Distribution Abstraction:

In the scope of distributed controllers this abstraction is present. A collaborative set of distributed controllers establishes a network and route status description across the networks. The decentralized state of the whole network may include divided data sets for the exchanging of routing data or a multiplicated data set by the controller to ensure that the controllers collaborate to keep the global network consistent.

This abstraction is designed to mask complex distributed structures (which are currently being used by several networks) from the design and implementation of the protocol. It provides an illustrated network graph that is available for control via an API, providing a coherent, global view of the network. An NOS, such as OpenDaylight or Ryu, is an implementation of this abstraction.

Specification Abstraction:

The distribution abstraction offers a global network view that, even though there are several cooperating controllers, there is a single central controller. The abstraction of requirements offers an abstract view of the global network. This view gives the application enough specifics to set objectives, such as routing or security policies, without providing the requisite information to achieve the objectives. The Shenker presentation summarises the following:

Forwarding interface: an abstract forwarding model that protects higher levels of forwarding equipment.

Distribution Interface: a worldwide network view protecting higher layers of state dissemination/collection.

Specification Interface: An abstract network view to protect the application from physical network data

The instance of a specification abstraction is Figure 4.20. The physical network is the set of SDN data plane switches interconnected to it. A single virtual switch is the abstract view. A single SDN domain can

be the physical network. Edge ports that bind to other domains and hosts on a virtual switch are mapped into ports. A module to learn the Media Access Control (MAC) address of hosts can be implemented at the application level. When an unknown host sends a packet, the application module will connect this address directly with the input port and direct future traffic to that host. Likewise, the module floods this packet to all of the output ports if the packet arrives at one virtual switch port with an undefined target address. The abstraction layer conveys these acts to the whole physical network, which carries out internal forwarding with the domain.

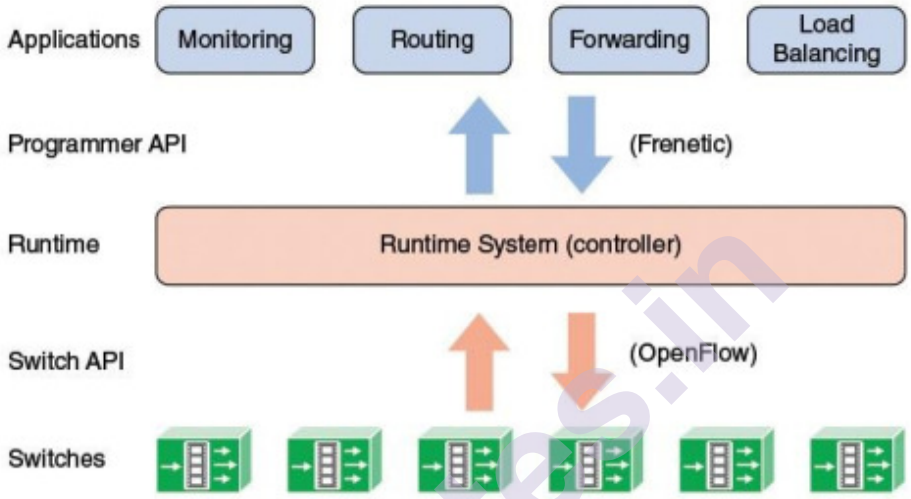


FIGURE 4.20 Virtualization of a Switching Fabric for MAC Learning

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

4.7.2 Frenetic:

The programming language Frenetic is an example of the network services abstraction layer. Instead of manually configuring individual network elements, Frenetic allows network operators to set the entire network. With OpenFlow models, Frenetic was designed to solve problems by dealing with a network level abstraction rather than Openflow explicitly down to a network feature level.

Frenetic contains an embedded query language, offering powerful network status abstractions. The SQL-like language includes segments that select, filter, split, merge and incorporate packet streams. Another unique advantage of this language is its ability to write queries with forwarding policies. The control messages are created by a compiler to query the counters and tab on switches.

As shown in Figure 4.21, Frenetic contains two layers of abstraction. The upper level, the Frenetic API, provides a variety of operators to manipulate network traffic sources. The query langue

provides means to read the network status, merge multiple queries and convey high-level descriptions to classify and filter the packet streams that cross the network, transform and aggregate them. A runtime device running the SDN controller provides the lower level of abstraction. The software translates high level policies and queries into low-level flow rules and issues OpenFlow commands for certain rules to be mounted on the switches.

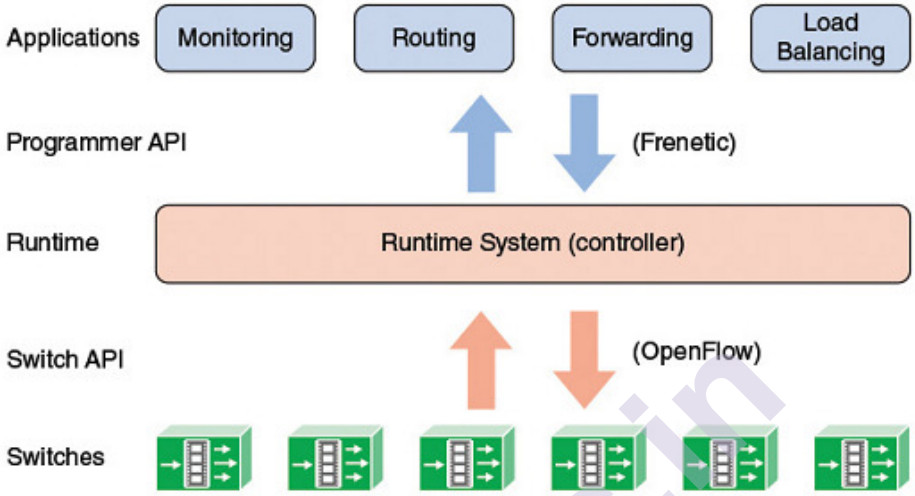


FIGURE 4.21 Frenetic Architecture

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

To gain an insight into the two layers of abstraction, take a simple example, from an IEEE Communications Magazine article by Foster in February 2013. The software integrates forwarding functionality with web traffic control. Take into consideration the following Python program to control OpenFlow switches at run-time:

```
def switch_join(s):
pat1 = {inport:1}
pat2web = {inport:2, srcport:80}
pat2 = {inport:2}
install(s, pat1, DEFAULT, [fwd(2)])
install(s, pat2web, HIGH, [fwd(1)])
install(s, pat2, DEFAULT, [fwd(1)])
query_stats(s, pat2web)
def stats_in(s, xid, pat, pkts, bytes):
print bytes    sleep(30)    query_stats(s, pat)
```

When a switch is added to the network, the program installed three forwarding rules in the switch for three different types of traffic: traffic

arriving on port 1, web traffic arriving on port 2, and other traffic arriving on port 2. The HIGH priority of the second rule precedes the default priority of the third rule. The query_stats call generates a request for the pat2web rule counters. When the controller receives the response, the stats_in handler is invoked. This function prints the statistics on the previous loop iteration, waiting 30 seconds, to the switch for statistics matching the same rule.

The program is written in a way that incorporates the logic for web monitoring and forwarding. This demonstrates the essence of OpenFlow's underlying function. Any improvements or new functionality would have a complex impact on the program. The following can be expressed independently with Frenetic:

```
def repeater():
rules=[Rule(inport:1, [fwd(2)])
Rule(inport:2, [fwd(1)])]
register(rules) def web monitor():
q = (Select(bytes) *
Where(inport=2 & srcport=80) *
Every(30))
q >> Print()
def main():
repeater()
monitor()
```

With this code, changing the monitor program or swapping it out for another monitor program without affecting the repeat code and the repetition program changes will be convenient. Particularly, the functionality of the system delegates the responsibility for installing specific OpenFlow rules which simultaneously enforce the two components. The device runtime for this example will create the same rules as the manually designed rules in the above stated switch join feature.

4.8 TRAFFIC ENGINEERING

Traffic engineering is a means through which data flows in network can be analyzed dynamically, regulated and forecast to optimize performance to fulfill the service level agreements (SLAs). Traffic engineering includes the implementation of QoS-based routing and forwarding policies. In comparison with a non-SDN network, the role of traffic engineering should be significantly simplified. SDN provides a clear global view of heterogeneous devices and effective tools for network switch configuration and management.

It is a significant field in which SDN applications have been developed. Kreutz's SDN survey paper in the IEEE's Proceedings January 2015 lists the following functions of the traffic engineering as SDN applications:

- On-demand virtual private networks
- Load balancing
- Energy-aware routing
- Quality of service (QoS) for broadband access networks Scheduling/optimization
- Traffic engineering with minimal overhead
- Dynamic QoS routing for multimedia apps
- Fast recovery through fast-failover groups
- QoS policy management framework
- QoS enforcement
- QoS over heterogeneous networks
- Multiple packet schedulers
- Queue management for QoS enforcement
- Divide and spread forwarding tables

4.8.1 PolicyCop:

PolicyCop , an automated QoS policy enforcement framework, is an instructive example of an SDN application for traffic engineering. It uses SDN and OpenFlow for its programmability.

- Dynamic Traffic management
- Flexible Flow Level Regulation
- Classes of dynamic traffic
- Adding levels of custom flow

PolicyCop's main features are the network monitoring to detect the policy breaches (depending on a QoS SLA) and networking to strengthen its policy violation.

The policycop contains 11 programme modules and two databases, built both in the application plane and the control plane, as shown in Figure 4.23. PolicyCop uses the SDN control plane to track QoS enforcement and can change control plane rules and flow tables on a data plane, automatically based on the dynamic traffic statistics for the network.

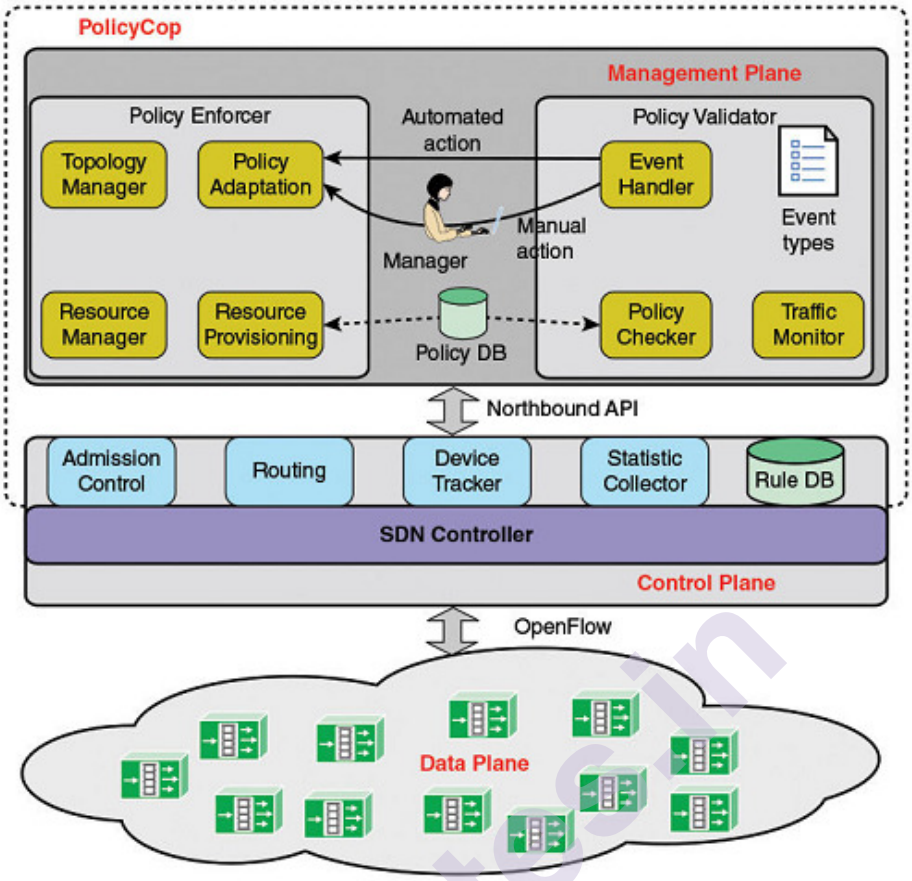


FIGURE 4.22 PolicyCop Architecture

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

PolicyCop contains 4 modules and a database to store control rules in the control plane as defined:

- **Admission Control:** Accept or deny requests for network resource reservation, such as queues, flow tables and capacity in the resource provisioning module.
- **Routing:** Specifies the availability of routes based on the rule database control rules.
- **Device Tracker:** Monitors network switches and their ports up/down status.
- **Collection of Statistics:** Uses a mixture of passive and active control methods for calculating various network metrics.
- **Rule Database:** the application plane translates networking-wide high-level policies in order to manage and storage rules in the rules database.

A Northbound RESTful interface links such control plane frameworks to the application plane frameworks that are organized into two components: a policy validator that tracks the network to detect policy violations, and an enforcer which adjusts control plane regulations to network and top level policy requirements. All components are centered on policy database containing a network manager's QoS policy rules. The following module are:

Traffic Control: The active policy compilation from the policy database and the monitoring duration, network sections and metrics are calculated.

- **Policy Checker:** Policy Violation checks, using policy database details and Traffic Monitor information.
- **Event Handler:** analyses infringement events and either immediately informs the policy enforcer or sends a request to the network manager based on the type of event.
- **Topology Manager:** Holds a large number of nodes based on device tracker data.
- **Resource management:** keeps track of the resources presently assigned by admission control and the collection of statistics.
- **Policy Adaptation:** requires a set of actions for any form of infringement.
- **Provision of resources**: This module allocates new resources, or distributes existing resources or both depending on the infringement case.

The basic functionality of such policy adaptation steps is shown in Table 3. The actions are connectible components which the network manager may specify.

SLA Parameter	PAA Functionality
Packet loss	Modify queue configuration or reroute to a better path
Throughput	Modify rate limiters to throttle misbehaving flows
Latency	Schedule flow through a new path with less congestion and suitable delay
Jitter	Reroute flow through a less congested path
Device failure	Reroute flows through a different path to bypass the failure

TABLE 3 Functionality of Some Example Policy Adaptation Actions (PAAs)

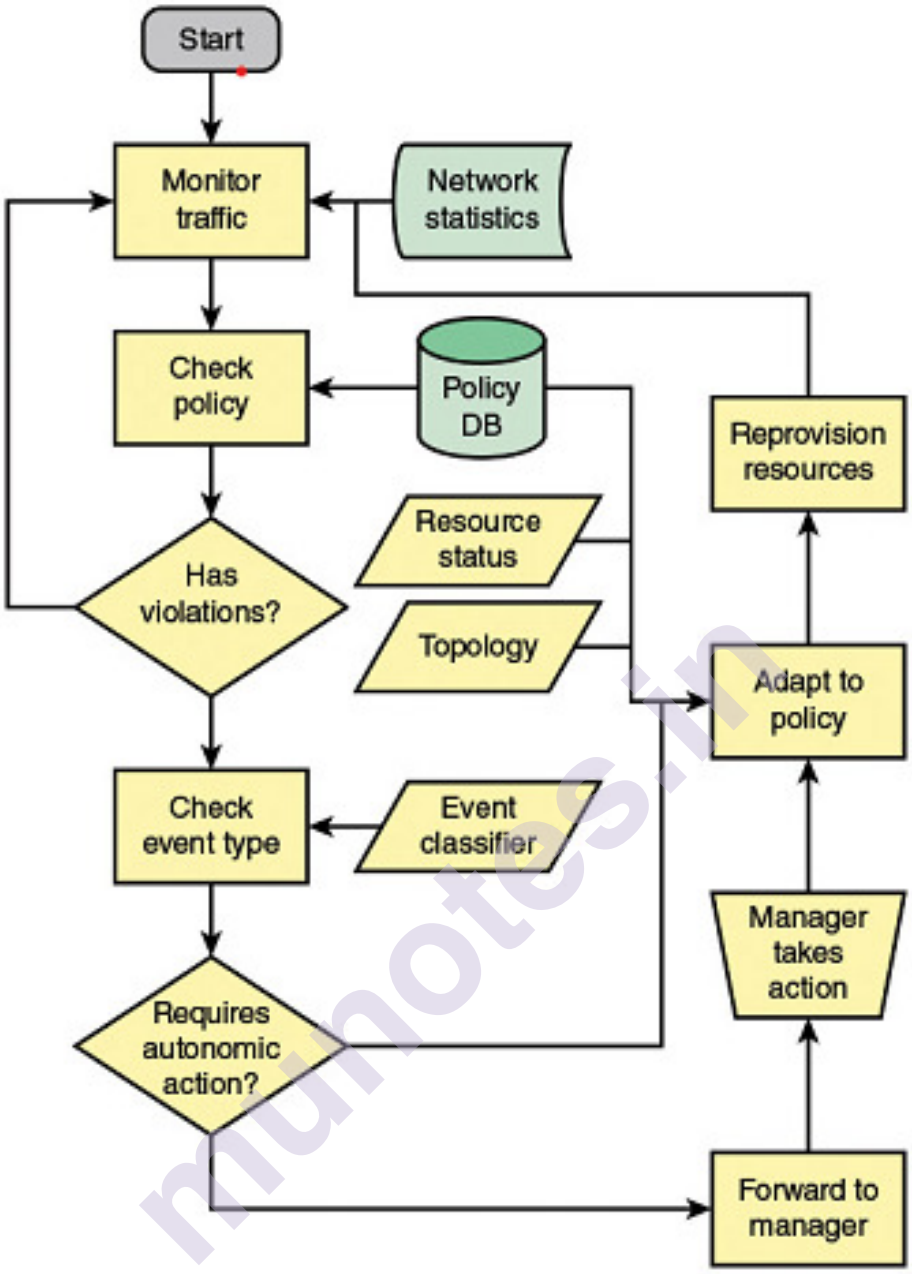


Figure 4.23 shows the process workflow in PolicyCop.

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

4.9 MEASUREMENT AND MONITORING

Applications which provide advanced functionality for many other networking services, and applications which add more value to the OpenFlow based SDN, can be approximately divided into 2 categories for the measuring and surveillance applications. The broadband home connections are an example of the first category. With an SDN

connection, additional functions can be added to monitor traffic and demand from home networks to help the device to respond to changing requirements. The second category usually consists of the application of various sampling methods and calculation methods to minimize the expense of the control plane in gathering statistics on data planes.

4.10 SECURITY

One of two objectives are for applications in this area:

Address security issues surrounding SDN use: SDN has a three-layer architecture (application, control, data) and new distributed data control and encapsulation approaches. All this poses the opportunity for new attack vectors. From any of the three layers or in the communication between layers threats can exist. For the secure use of SDN alone, SDN applications are needed.

To boost network security, use the SDN functionality: SDN offers a platform for development of precise and centre-managed security policies and frameworks for network designers and managers, regardless of new security challenges. SDN enables the development and organization of security services and processes for SDN security controllers and SDN Security Applications.

4.10.1 OpenDaylight DDoS Application:

Through Defense4All, an open SDN security framework incorporated in OpenDaylight, Radware, a vendor of application delivery and application security for virtual and cloud data centres, announced in 2014 its contributions to the OpenDaylight project. Defense4All provides distributors and cloud service providers with the identification and prevention of distributed denial of service (DDoS) as a local network service. With the OpenDaylight SDN controller to incorporate SDN-enabled networks into the DoS/DDoS Protection service, Defense4All provides operators a doS/DDoS protection service either per client or per virtual network segment.

Defense4All uses the following elements as a common method for protecting against attacks by DDoS:

- Defense4All uses the following elements as a common method for protecting against attacks by DDoS:
- A traffic statistics collection and statistical enforcement observed during the peacetime of safe objects. These collected statistics build on the standard traffic baselines for the protected objects.

- DDoS attack patterns are detected as traffic irregularities that deviate from standard base lines.
- Diversion of suspicious traffic from its usual route to AMSs for traffic scrubbing, limited blockage from sources and so forth. Clean traffic is reinjects into the original packet destination.from scrubbing centres.

The overall Defence4All program scope is shown in Figure 4.24. There are numerous data plane switches in the interconnected SDN network which facilitate traffic between client and server devices. Defense4All works with a controller via an OpenDaylight (ODC) API northbound application. Defense4All facilitates a network manager user interface which could either be an interface with a command line or RESTful API. Ultimately, Defense4All has an API for one or more AMSs to communicate.

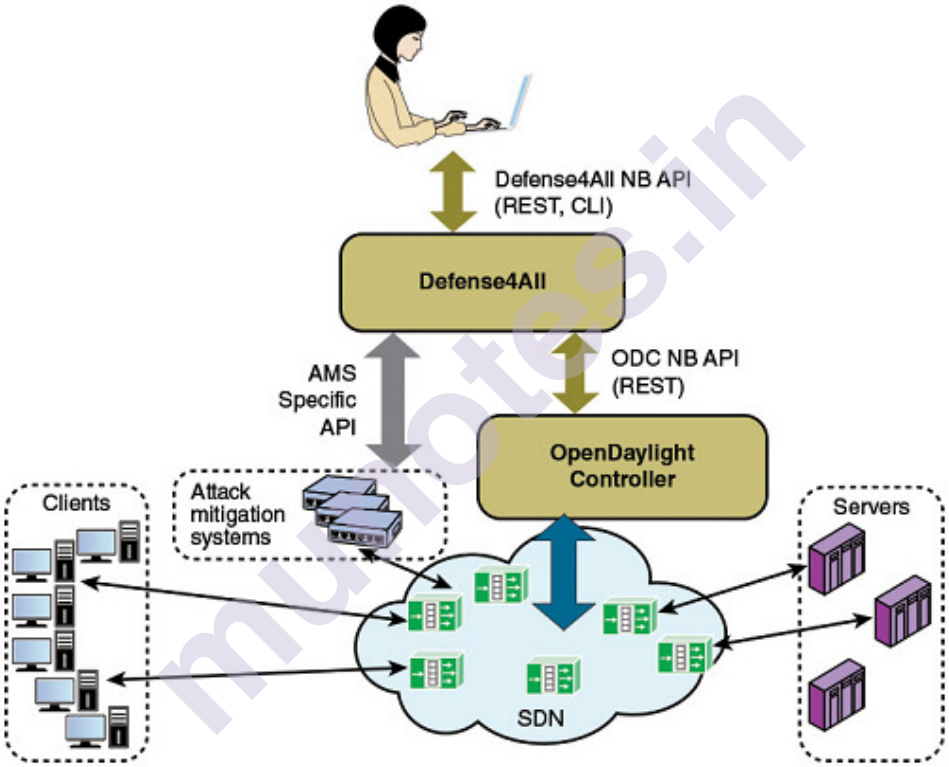


FIGURE 4.24 OpenDaylight DDoS Application

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

Certain networks and servers classified as protected networks and Protected Objects (POs). The application informs the controller to mount traffic counting flows at all network locations wherein the subject PO's traffic is circulating for each protocol of every configured PO.

Defence4All then track traffic from all installed POs and resume readings, rates and averages from each of the respective network sites.

Especially, Defese4All measures the average traffic for the real-time traffic it is calculated with OpenFlow constantly, and an attack is expected, if the real time traffic is 80% different from an average.

Defense4All carries out the following procedure to counteract a detected attack:

1. Verify the presence of the AMS device and choose a live connection. Defense4All is currently designed to operate with the DefensePro AMS of Radware.
2. Configures a security policy and standard traffic rates for the AMS. This gives the AMS the information required to implement the policy of mitigation before the traffic returns to normal rates.
3. It begins to track and log syslogs for subject traffic coming from the AMS. As far as Defense4All still receives alerts for this attack, even if the flow counters on that PO do not suggest any additional attacks, Defence4All tends to divert traffic to AMS.
4. Maps the physical AMS connection to the corresponding PO connection selected. This usually requires the use of OpenFlow to modify connection definitions on a virtual network.
5. t implements the highest priority flow table entries in order to redirect the attack traffic to the AMS and redirect traffic from the AMS to a normal flow route. When Defense4All concludes that the attack ends (no attack information whether from the flow table counters or even from the AMS), the preceding actions will be reversed: it prevents the monitoring of the subject traffic syslogs, eliminates the flow table entries from traffic diversion and eliminates the security setup from the AMS. Defense4All shift to peacetime monitoring.

The primary software components of Defense4All are shown in Figure 4.25. The overall structure of the application, which is called a framework, includes the modules listed in the following list.

- Web (REST) Server: Network Manager s' Interface.
- Framework Main: Beginning, stopping, re-setting mechanism in the framework.
- Framework REST Service: Responses to user requests that are received from the web (REST) server.
- Framework Management Point: Monitor and configure commands coordinate and invoke.
- Defense4All Application: Consequently defined
- Common Classes and Utilities: A simple classification and function library that can be used by any framework or SDN application module.

- Repository Services: The convergence of the computational status from computing logic is an important aspect of the framework philosophy. All permanent states are stored in a series of repositories which can then be replicated, cached and distributed without computer logic awareness (framework or application).

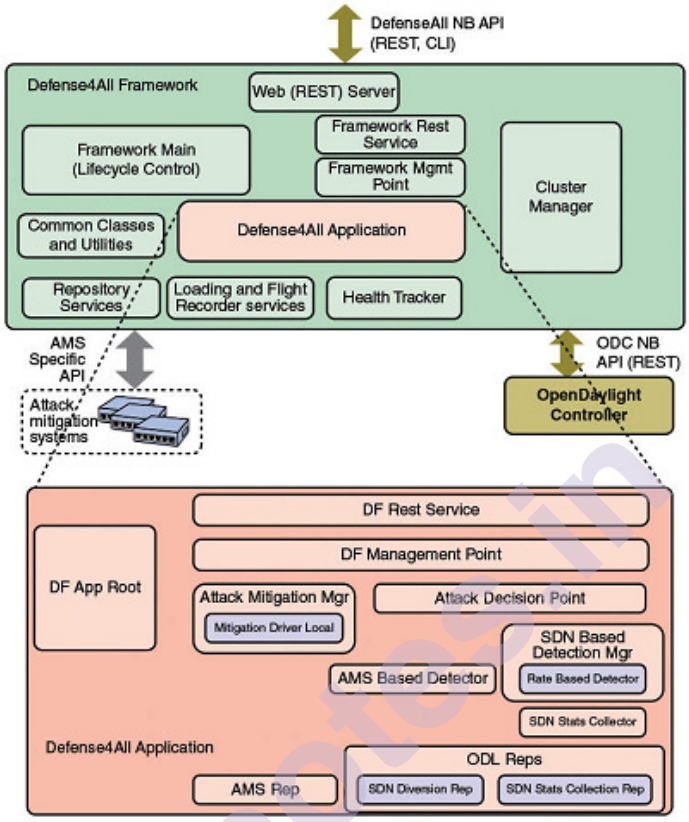


FIGURE 4.25 Defense4All Software Architecture Detail

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

- **Logging and Flight Recorder Services:** logs error, warning, tracking or information message are used by the logging service. Such logs are primarily for all developers of Defense4All. During the run-time of Java apps, the Flight records events and metrics.

- **Health Tracker:** Maintains agglomerate defense4All operational health runtime indicators and acts as a response to severe deterioration in functional or performance.

- Cluster Manager: All entities operating in the cluster mode are responsible for managing coordination with other Defense4All.
  The following elements are included in the Defense4All application module.

- **DF App Root:** The application's root module.

- **DF App Root:** Replies to Defense4All REST requests for application.
- **DF Management Point:** The control and configuration commands is to drive point. In turn, DFMgmtPoint provokes procedures in the right order with the other related modules.

- **ODL Reps:** Set of a plug-in module for various ODC models. Includes two submodular functions: statistical compilation and the diversion of corresponding traffic.

- **SDN Stats Collector:** Taking responsibility for configure "counters" at a given network locations for every PN in physical or logical modes. A counter is a series of ODC-enabling network switches and routers with OpenFlow flow entries. The module gathers stats through the counters regularly and passes it to the SDNBasedDetectionMgr. The module utilizes SDNStatsCollectionRep to configure the controllers and read the latest statistics. The status report includes the reading time, counter description, PN tag, and trafficData collection where the current bytes and packet parameters for flow entries modified in the counter location are included in each TrafficData element. The protocol may be {tcp,udp,icmp, other ip}, port may be any layer 4 and {inbound, outbound} can be the direction.

- **SDN Based Detection Manager:** A container for plug - and - play detectors based on SDN. It supplies statistic reports to plugged-in SDN detectors obtained from the SDNStatsCollector. It also includes all AttackDecisionPoint SDN-based reports of attacks (so as to allow reset of detection mechanisms). For each PN, every detector observes over time its normal course of traffic, and when traffic anomaly is detected it informs AttackDecisionPoint.

- **Attack Decision Point:** Responsible for preserving the lifecycle of an attack, from announcing an attack to terminating diversion when an attack comes to an end.

- **Mitigation Manager:** container is required pluggable prevention drivers. It maintains any lifecycle for every mitigation that can performed by an AMS. Every mitigation driver is accountable for maintaining the mitigation of attacks using AMSs.

- **AMS Based Detector:** The monitoring/querying prevention of attacks by AMSs is responsible for this module.

- **AMS Rep:** Controls the AMS interface.

Figure 4.25 demonstrates the difficulty of the SDN application even comparatively straightforward. Eventually, it should be noted that a commercial version of Defese4All called DefenseFlow has been developed by Radware. DefenseFlow implements more complex attack detection algorithms based on fuzzy logic. The key advantage is

that DefenseFlow is more able to differentiate between an abnormal, yet legit traffic volume.

4.11 DATA CENTER NETWORKING

We have addressed three areas of SDN applications : traffic engineering, measurement and monitoring, and security. The examples given in these applications show the wide range of applications in a variety of networks. The other three areas of application (data center networking, mobility and wireless networking as well as information-centric networking) use cases for particular network types.

The highly scalable and efficient data centers depend heavily on cloud, big data, major corporate networks, and even in many cases smaller company networks. This paper lists the following main criteria for data centers: large and scalable bandwidth and low latency; application-based QoS; resiliency levels; intelligent use of resources to cut energy usage; and overall performance improvements and network resource supply resilience for instance, by means of network virtualization and orchestration with computing and storage.

Many of these criteria are impossible to fulfil with conventional network architectures since the network is complex and rigid.

SDN provides significant improvements in the capabilities to quickly customize the network data centre configuration, adapt to user requirements to ensuring optimal network operation.

The rest of this section looks at two examples of SDN applications for the data centers.

4.11.1 Big Data over SDN:

In the HotSDN'12 Proceedings, a paper by Wang, et al., describe a method of using SDN to improve the networking for the Datacenter. The method builds on SDN's capacity to deliver application-aware networking. It also incorporates standardized Big Data applications and current developments for optical circuits which are rapidly customizable. Most of those applications process data in accordance with established computation patterns for standardized big data applications and often have a centrally managed framework that provides application-level information to be able to leverage to enhance the network. In other words, understanding how the big application can measure the data intelligently over the big-data servers and, most importantly, respond to changing patterns of applications by using SDN to re-configure network flows. The optical switches have the benefits of higher data rates are reduced cable complexity and energy usage in contrast with the electronic switches. A

variety of projects have shown how the traffic data at network level can be obtained and optical circuits between endpoints intelligently allotted for enhanced application performance, such as top-of-rack switches. Even so, the use of the circuit and application outcome cannot be appropriate if traffic demands and dependencies are actually seen at the application level. Combined with the dynamic capabilities of SDN, successful data centers networking configurations could be utilized to meet growing big data needs by incorporating awareness of the large data measurement patterns.

The hybrid electrical and optical data centers network with the OpenFlow enabled high-end (ToR) switches is connected to two aggregation switches in Figure 4.26: Ethernet switch and optical circuit switch (OCS). All switches are controlled through an SDN controller which administers the physical connectivity between ToR switches by setting the optical switch over optical circuits.

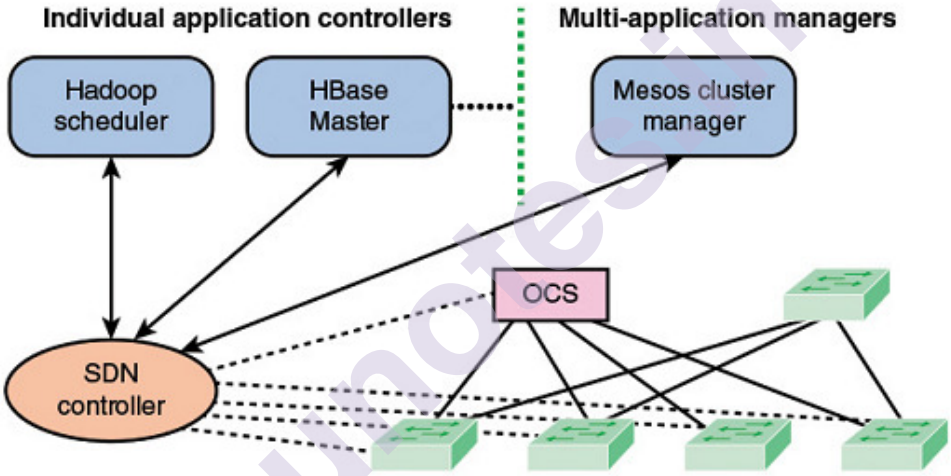


FIGURE 4.26 Integrated Network Control for Big Data Applications [WANG12]

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

The SDN controller is also interconnected to the Hadoop scheduler, that either forms pools of jobs to be scheduled and a relational database controller of HBase master containing data for big data applications. The SDN controller is also attached to a Mesos cluster manager. Mesos is a software package open source that offers scheduling and resources allocation services for distributed applications.

The SDN controller provides the Mesos cluster manager with information about network topology and traffic. The SDN controller accepts requests from Mesos managers for traffic.

Figure 4.26 arranges a scheme that uses the SDN controller to handle the whole task to continuously handle the network by using traffic demands of big data applications.

4.11.2 Cloud Networking over SDN:

Cloud Network as a Service (CloudNaaS) is a cloud networking network system using OpenFlow SDN to provide the Cloud User with more control over the cloud networking functionality. CloudNaaS allow users to use applications with a variety of network features, such as virtual network isolation, customizable addressing, differentiation in service and flexible middlebox interference. CloudNaaS primitives are directly incorporated within the cloud infrastructure with programmable high-speed network components, thus making CloudNaaS extremely effective.

Figure 4.27 shows the main sequence of CloudNaaS events as defined in the following list.

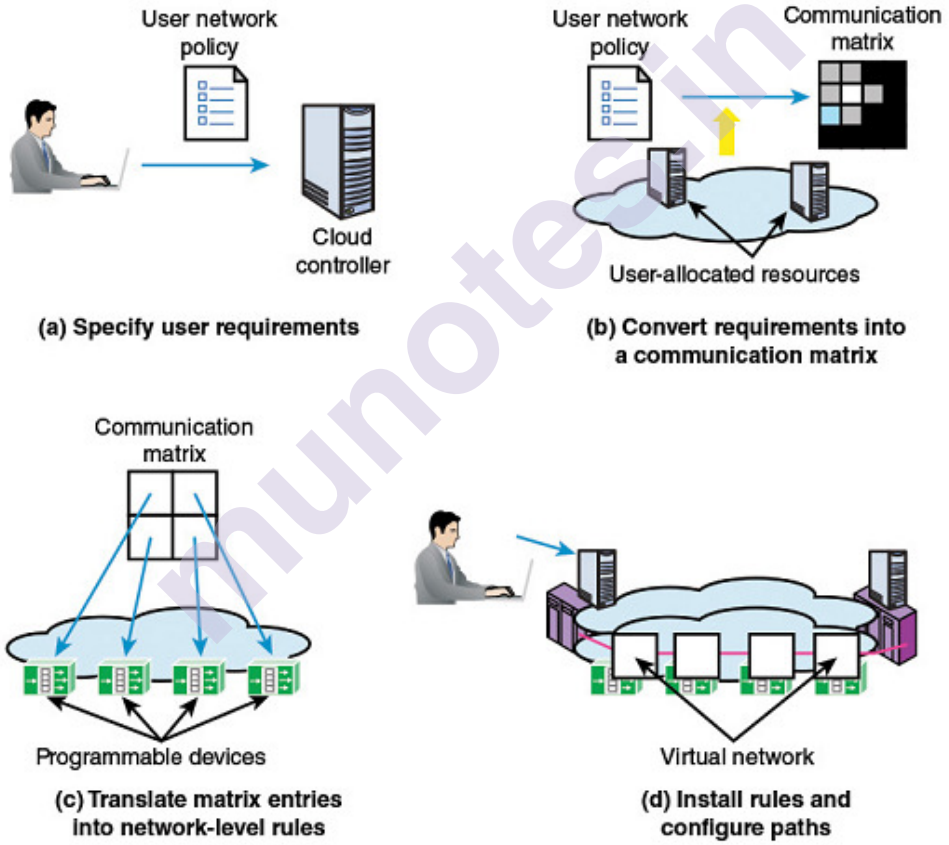


FIGURE 4.27 Various Steps in the CloudNaaS Framework

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

a. A consumer in a cloud uses a simple policy to determine the customer applications' network services. This policy statement will be given to a server operated by the cloud service provider's cloud controller.
b. The cloud controller compares Network Policy to a communication matrix that identifies optimal networking services and communication trends. The matrix is employed to evaluate how virtual machines (VMs) are optimally placed on cloud servers in order for cloud to effectively meet with the largest number of global policies. This is based on the understanding of the requirements and present conditions of many other clients.
c. The logical communication matrix for data plane forwarding elements is converted into network-level guidelines. By establishing and placing the required number of VMs when customer's VM appliances are deployed.
d. The directives at network level are installed via OpenFlow on the network devices.

The client's abstract network model consists of VMs and virtual network segments connecting VMs. Constructs the policy language describe a set of VMs which include an application and define various functions and capabilities associated with virtual network segments. The primary buildings are the following:

Address: Define the VM's customised customer-visible address.

Construct one or more VM's logical group. The aggregation of VMs with similar functions enables improvements to be made across the whole group without the need to modify the service associated with individual VMs.

Middlebox: Name a new virtual middlebox with the name and config file. The cloud providers include the collection of middleboxes accessible and their configuration syntax. E.g. intrusion detection and audit compliance system.

Networkservice: Define functionality for connecting to a virtual network segment such as a broadcast domain, Layer 2 link QoS, and the middlebox collection to traverse.

Virtualnet: Virtual network segments are connected with the network services and connect VM groups to each other. One or two classes are protected by a virtual network. The service extends with one individual group to traffic between several VM pairs in the group. The service is extended between any VM during the first group as well as any VM in the second category with such a pair of classes. Virtual networks can also be accessed by predefined classes, for example EXTERNAL, which contains any endpoints outside the cloud.

The description of CloudNaaS architecture is shown in Figure 4.28. The two principal components are a cloud and a network controller. The cloud controller offers a simple Infrastructure as a Service (IaaS) for the management of VM instances. Standard IaaS requests including VMs and storage can be communicated with the user. The network policy frameworks also allow the user to determine VM 's virtual network capabilities. The Cloud controller administers a virtual software programmable switch from each physical cloud server that enables network services for users, including user-dependent virtual network segments. The cloud controller creates and transmits the communication matrix to the network controller.

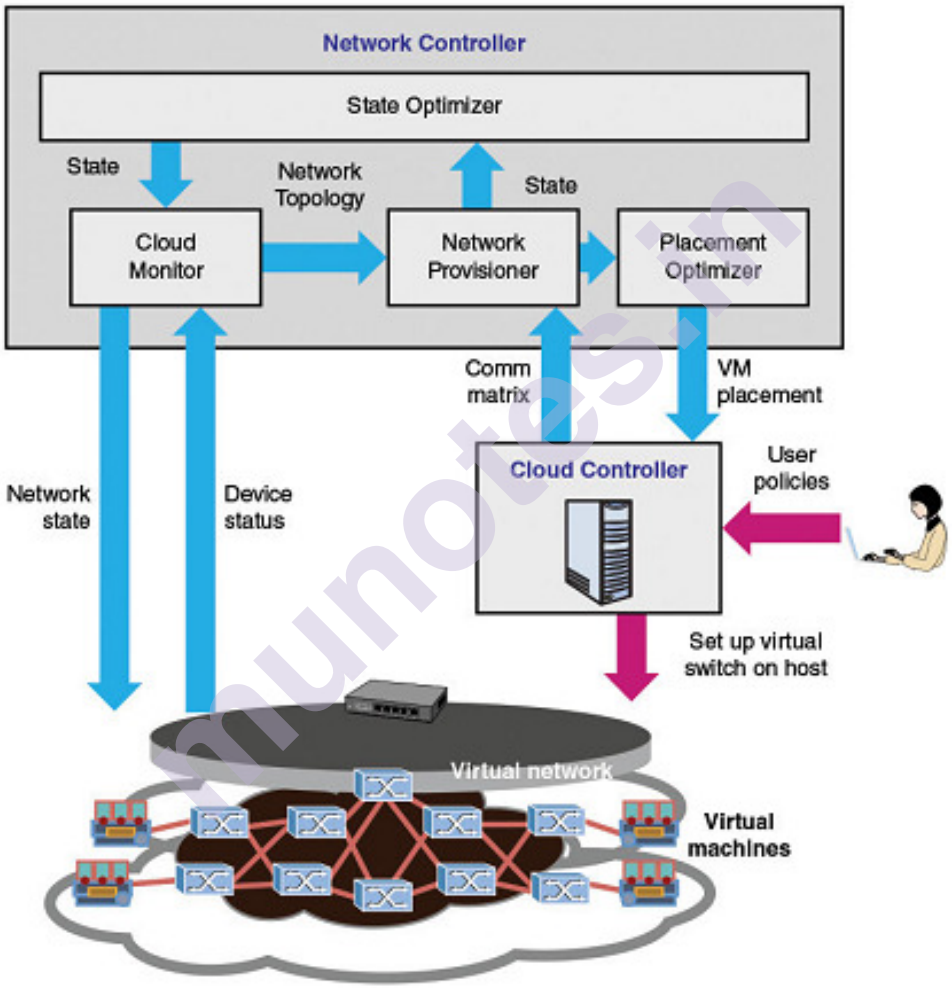


FIGURE 4.28 CloudNaaS Architecture

Above Image from the reference book “Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings”

The network controller configures physical and virtual switches by using the communication matrix. It creates virtual networks across VMs which provides the cloud controller with VM placing guidelines. It tracks

cloud data plane switches on traffic and reliability and makes modifications to the Network System required to increase the utilization of resources in order for tenants to compliance. The controller uses the positioning optimizer to assess the optimum location for VMs in the cloud, which it reports to the cloud controller for distribution. After this the controller enables the network supplier module to create a set of configurations at each of the programmable devices connected to the network and set them up to implement a virtual network segment for the tenant accordingly. Thus, CloudNaaS gives cloud clients the potential to expand beyond basic demands for processing and storage capacity, identifying a virtual VM network and managing the virtual network's service and QoS specifications.

4.12 MOBILITY AND WIRELESS

Wireless networks enforce a wide range of new criteria and barriers, as well as conventional efficiency, protection and reliability requirements of wired networks. Mobile users continually create demand for high quality new services and efficient content distribution regardless of location. Network providers have to resolve issues of spectrum management, implementation of handover mechanisms, efficient load balance performance, QoS and QoE response and security management. SDN will provide the mobile network operator with much needed resources and a range of SDN-based applications have been created in recent times for the wireless network providers. Among other things, it contains the following SDN applications: Efficient transfers for smooth mobility on-demand development of virtual access points, load balancing downlink scheduling, dynamic spectrum use, improved customer / base station resource block allocations, efficient management of interdependent communication technology, interoperability between the networks, wireless shared infrastructures, and management of QoS and access control policies

4.13 INFORMATION-CENTRIC NETWORKING

Information-centric networking (ICN), also identified as content-centric networking, would also have obtained substantial interest in latest years, largely due to the fact that distribution and manipulation of information has now become the primary function of the Internet today. Unlike the conventional host-centric networking paradigm in which information is gathered by contacting designated named hosts, ICN aims to provide indigenous network primitives for effective information extraction by directly naming and operating information objects.

In ICN, location and identification are differentiated, so the sources are decoupled. The principle of this method is that sources of

information can be positioned and information to users can find more information everywhere on the network, even though the information is named, addressed and paired autonomously of its location. In ICN, rather than defining a source host pair for information exchange, a bit of information is labelled. When a request has been sent to ICN, the network is accountable for finding the best source that would provide the relevant information.

It is a challenge to implement ICNs on conventional networks, because current routing devices must be upgraded or replaced with routing devices that are ICN-enabled. In addition, the distribution model of ICN is shifted from host to server to user. This requires a clear difference between the demand for information and the supply task and the transmission task. SDN is able to provide the requisite infrastructure for the ICN implementation as it allows transmission elements to be configured and control and data planes to be separated.

A variety of projects suggested to incorporate ICNs through the use of SDN capabilities. The achievement of this connection between SDN and ICN has no consensus approach. The suggested approaches include major upgrades and amendments to the OpenFlow protocol; the creation of a map of names in IP addresses using a hash function; a name field with the IP option header; and the use of abstraction layers from an OpenFlow (OF) switch to an ICN router, to allow the layer, OF switch, and ICN router run as a single ICN programmable.

This method provides ICN functionality to OF switches without modifying the OF switches. The method is based on an open protocol specification and the deployment of ICN software reference known as CCNx. A brief context on CCNx is needed before looking at the abstraction layer approach.

4.13.1 CCNx:

CCNx has been built as an open source project from the Palo Alto Research Center (PARC), which has numerous implementations have been experimentally deployed. Two kinds of packets are used to communicate to CCN: interest packets and content packets. CCN communications are by two packet types: packets of interest and packages of material. A customer requests content delivered via a Interest packet. Every CCN node receiving interest and naming data matching the interest responds with a content packet also called as a Content. Content seems to be of interest unless the name of the Interest packet corresponds the name of the Content Object packet. When a CCN node is issued an interest, and no copy of the requested content already occurred, it could transmit the interest to a content source. There are forwarding tables in the CCN node which decide which way the interest should be sent. An interest provider for whom the named content correlates responds with a contents packet.

The choice of any intermediate node is to store the content object, and then the next time an interest packet with the same name is received, it may respond to that cached copy of the content object.

The fundamental working of a CCN node is like the IP node. CCN nodes are receiving and transmitting packets through faces. A face is an application connection point or a CCN node or a channel of a different type. A face can have characteristics showing the expected throughput and latency, distribution or multi-cast capacity, or other important elements. Three major data structures are contained in a CCN node:

Content Store: Keeps a table with packages that have already been viewed

Forwarding Information Base (FIB): Which used forward Interest packets to possible sources of data.

Pending Interest Table (PIT): used to track CCN nodes up-stream so that the packets of content obtained later can be returned to their applicants.

The details about how origins of content are known and how destinations are developed through the CCN network is beyond our reach. Shortly, through a cooperation among all the CCN nodes, contents providers advertise content names and routes across the CCN network.

ICN relies heavily on the internet caching—that is to cache content from the content provider to the requester route. This on-the-go caching works well overall, but is not ideal as the content can be reiterated on routers to decrease the total amount of cached content. Off-path caching, which assigns content to specified off-path caches inside the network and sweeps traffic from the optimum route towards to the caches distributed around the network, are capable of overcoming such restriction. Off-path caching increases the global hit ratio by using caching capacity from across network and reducing bandwidth utilization of egress links.

4.13.2 Use of an Abstraction Layer:

The core design problem with the use of an SDN switch to act as an ICN router, in general an OF switches forward on an IP address based on the IP packet fields and an ICN router forward based on a content name. The proposed solution effectively hashes the name within the fields via an OF switch. The overall method architecture is shown in Figure 4.29. A wrapper abstraction layer is used to connect a module to the software of the CCNx node with an OF switch. In CCN messages the wrapper couples an interface to the CCNx, decodes and has contents in fields which an OF-Switch may process (for example, IP addresses, port

numbers). The broad naming space provided by these fields limits the likelihood that two separate content names will collision. The forwarding tables in the OF switch are based on the contents of the hashed fields. The switch may not "know" whether the contents of such fields no longer are legal IP addresses, TCP port numbers, etc. As always, it goes forward, based on the values in the relevant IP packet areas.

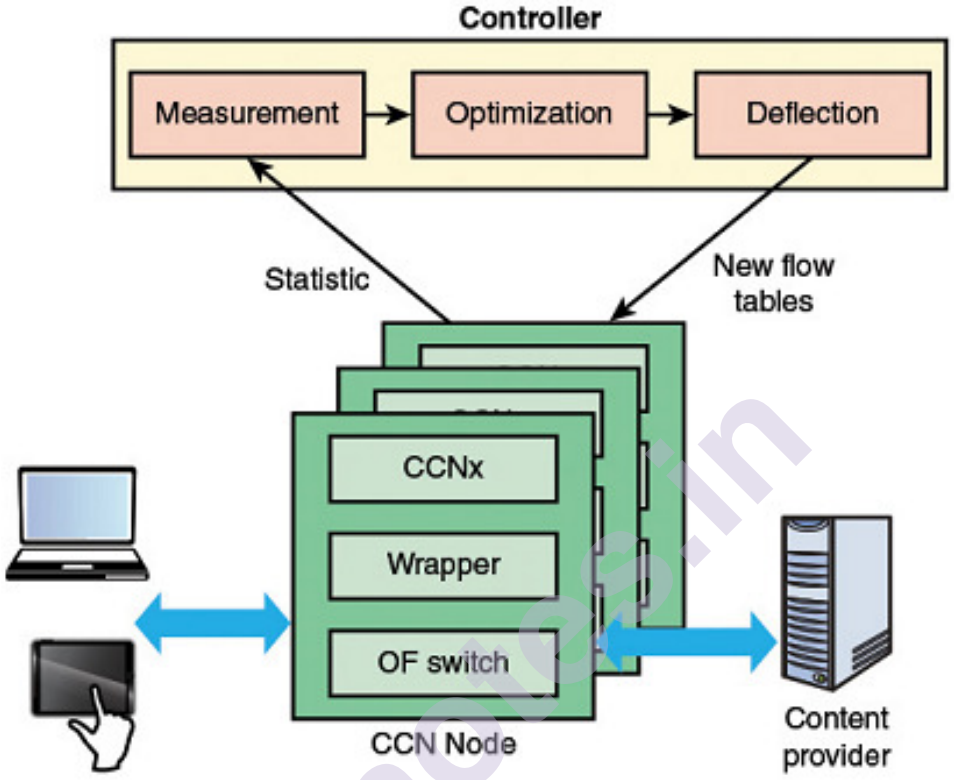


FIGURE 4.29 ICN Wrapper Approach

Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

The abstraction layer eliminates the issue through the use of current OF switches to provide CCN features. Two more challenges must be solved for proper performance: how to accurately calculate content popularity without a high overhead and how routing tables can be designed and optimized in order to deflect them. The architecture needs three new modules in the SDN controller in order to overcome these challenges:

Measurement: The popularity of content can be explicitly derived from OF flow statistics. The measurement module routinely checks statistics from incoming OF switches and processes them to return the popular content list.

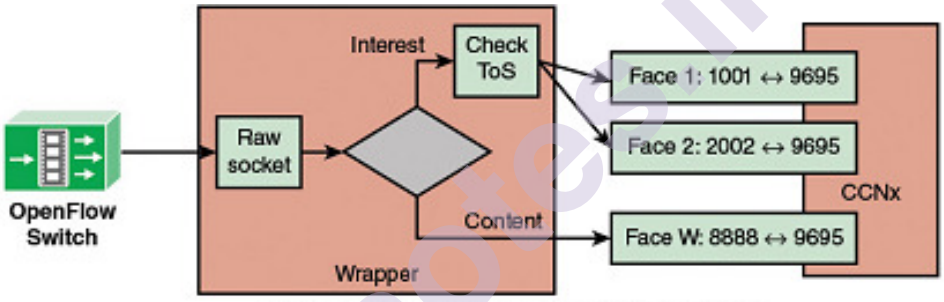
Optimization: The most common contents are used as an input to the optimization algorithm. The purpose is to decrease the total delays in

deflected contents under the following limitations: 1) the content of cache is at node no higher than node availability, and 2) the cache does not cause the link congestion. (3) Every common content is cached at precisely one node.

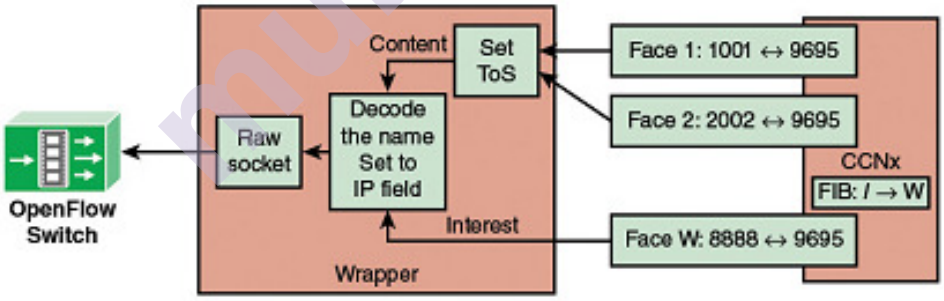
Deflection: use the results of the optimization to construct, for every contents, a map between the content name and the output interface of the content node (for instance, ip.destination = hath(content name), action = forward to interface 1)

Mappings are built on the flow tables of switches via the OF protocol, which can forward the following Interest packet to suitable caches.

Packet flow shows in Figure 4.30. Any packet obtained from other ports will be forwarded to your wrapper by the openflow switch and forwarded to the CCNx module. The OpenFlow switch must be used to identify the packet's switch source port. This is achieved by selecting the ToS value of all packets received to the relevant port value and then forwarding them all to the wrapper port.



(a) Packet flow from OpenFlow Switch to CCNx



(b) Packet flow from CCNx to OpenFlow Switch

FIGURE 4.30 Packet Flow Between CCNx and OpenFlow Switch
Above Image from the reference book "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings"

The wrapper maps a CCNx face to an interface (i.e. the port) where OpenFlow switches have to be used with ToS. Face W is a special face from the CCNx module to the wrapper. W accepts every packet of content from the wrapper and sends each packet of interest from CCNx to the wrapper.

Part 1 of Figure 4.30 shows how the wrapper manages the incoming parts of the OF switch. The wrapper helps in extracting the face value from the ToS field for a packet of interest and transfers the packet to the respective CCNx face. If the CCNx node retains a copy of the content requested, the Content packet is made up and returned to the requested contents. Anything else, this interest will be forwarded to face W and the PIT revised accordingly. The wrapper forwards it directly to face W after the Content packet arrival from the OF switch.

The wrapper function for the packets obtained from the CCNx module is shown in Part b of Figure 4.30. The ToS field defines the output port for content packets. In order to retrieve the packet content name for any packet, the packet is then decoded. The name is hashed and the packet's IP source is set to fit the hashed value. The wrapper then moves the packets to OF switches. The packets of contents will be returned to their incoming face. The ToS value is set to 0 for the packets of interest to be passed by the OF switch to the next hop.

The use of the abstract wrapper layer therefore offers simple ICN functionality including the deflection functionality without modifying the CCNx module or OpenFlow switch.

4.14 UNIT END QUESTION

1. List and explain the key functions of the SDN control plane.
2. Discuss the routing function in the SDN controller.
3. Explain the ITU-T Y.3300 layered SDN model.
4. Explain the OpenDaylight Architecture.
5. Write a short note on REST.
6. Differentiate between centralized and distributed SDN controller architectures.
7. Explain the role of BGP in an SDN network.
8. Write a short note OpenDaylight Helium
9. Explain the Service Abstraction Layer Model in OpenDaylight Architecture.
10. State the six REST constraints and example of REST API.
11. Write a Short note on IETF SDNi.
12. Explain the OpenDaylight SNDi.
13. Describe the overview of the SDN application plane architecture.

14. Define the network services abstraction layers.
15. List and explain three forms of abstraction in SDN.
16. List and describe six major application areas of interest for SDN.
17. Explain the Frenetic Architecture.
18. what is Traffic Engineering?
19. what is PolicyCop's SDN application of traffic engineering? Explain the PolicyCop Architecture.
20. Explain the OpenDaylight DDoS Application.
21. what is Cloud Network as a Service (CloudNaaS)? Explain the Various Steps in the CloudNaaS Framework.
22. Write a Short note on CCNx.

4.15 REFERENCE

- Foundations of Modern Networking: SDN, NFV, QoE, IoT, and CloudWilliam Stallings Addison- Wesley Professional October 2015.
- SDN and NFV Simplified A Visual Guide to Understanding Software Defined Networks and Network Function VirtualizationJim DohertyPearson Education, Inc.
- Network Functions Virtualization (NFV) with a Touch of SDN Rajendra Chayapathi Syed Farrukh Hassan Addison- Wesley.
- CCIE and CCDE Evolving Technologies Study Guide Brad dgeworth, Jason Gooley, Ramiro Garza Rios Pearson Education, Inc 2019.

*****

UNIT III

5

VIRTUALIZATION

Virtualization, Network Functions Virtualization: Concepts and Architecture, Background and Motivation for NFV, Virtual Machines The Virtual Machine Monitor, Architectural Approaches Container Virtualization, NFV Concepts Simple Example of the Use of NFV, NFV Principles High-Level NFV Framework, NFV Benefits and Requirements NFV Benefits, NFV Requirements, NFV Reference Architecture NFV Management and Orchestration, Reference Points Implementation, NFV Functionality, NFV Infrastructure, Container Interface, Deployment of NFVI Containers, Logical Structure of NFVI Domains, Compute Domain, Hypervisor Domain, Infrastructure Network Domain, Virtualized Network Functions, VNF Interfaces, VNFC to VNFC Communication, VNF Scaling, NFV Management and Orchestration, Virtualized Infrastructure Manager, Virtual Network Function Manager, NFV Orchestrator, Repositories, Element Management, OSS/BSS, NFV Use Cases Architectural Use Cases, Service-Oriented Use Cases, SDN and NFV Network Virtualization, Virtual LANs ,The Use of Virtual LANs, Defining VLANs, Communicating VLAN Membership, IEEE 802.1Q VLAN Standard, Nested VLANs, OpenFlow VLAN Support, Virtual Private Networks, IPsec VPNs, MPLS VPNs, Network Virtualization, Simplified Example, Network Virtualization Architecture, Benefits of Network Virtualization, OpenDaylight's Virtual Tenant Network, Software-Defined Infrastructure, Software-Defined Storage, SDI Architecture

Unit Structure

5.1 NETWORK FUNCTIONS VIRTUALIZATION (NFV): CONCEPTS AND ARCHITECTURE

5.1.1 Virtual Machines:

Traditionally, applications have run directly on an operating system (OS) on a personal computer (PC) or on a server. Each PC or server would run only one OS at a time. Therefore, application vendors had to rewrite parts of its applications for each OS/platform they would run on and support, which increased time to market for new features/functions, increased the likelihood of defects, increased quality testing efforts, and usually led to increased price. To support multiple operating systems, application vendors needed to create, manage, and support multiple hardware and operating system infrastructures, a costly and resource-intensive process. One effective strategy for dealing with this problem is known as **hardware virtualization**. Virtualization technology enables a single PC or server to simultaneously run multiple operating systems or multiple sessions of a single OS. A machine running virtualization software can host numerous applications, including those that run on different operating systems, on a single hardware platform. In essence, the host operating system can support a number of **virtual machines (VMs)**, each of which has the characteristics of a particular OS and, in some versions of virtualization, the characteristics of a particular hardware platform.

The Virtual Machine Monitor:

The solution that enables virtualization is a **virtual machine monitor (VMM)**, or **hypervisor**. This software sits between the hardware and the VMs acting as a resource broker (see Figure 7.1). Simply put, the hypervisor allows multiple VMs to safely coexist on a single physical server host and share that host's resources. The number of guests that can exist on a single host is measured as a **consolidation ratio**. For example, a host that is supporting six VMs is said to have a consolidation ration of 6 to 1, also written as 6:1 (see Figure 2 below). If a company virtualized all

of their servers, they could remove 75 percent of the servers from their data centers. More important, they could remove the cost as well, which often ran into the millions or tens of millions of dollars annually. With fewer physical servers, less power and less cooling was needed. Also this leads to fewer cables, fewer network switches, and less floor space.

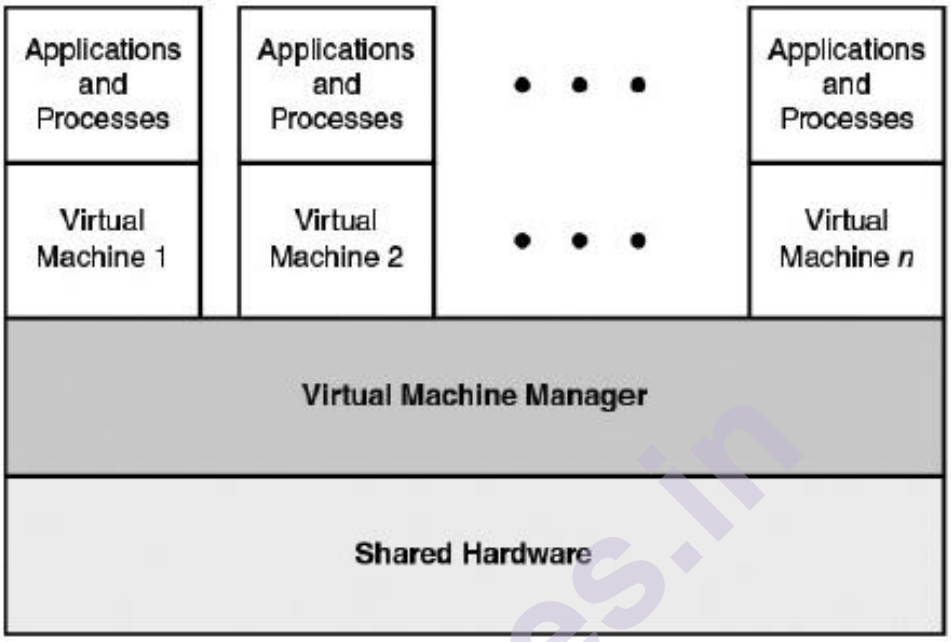


FIGURE 1 Virtual Machine Concept

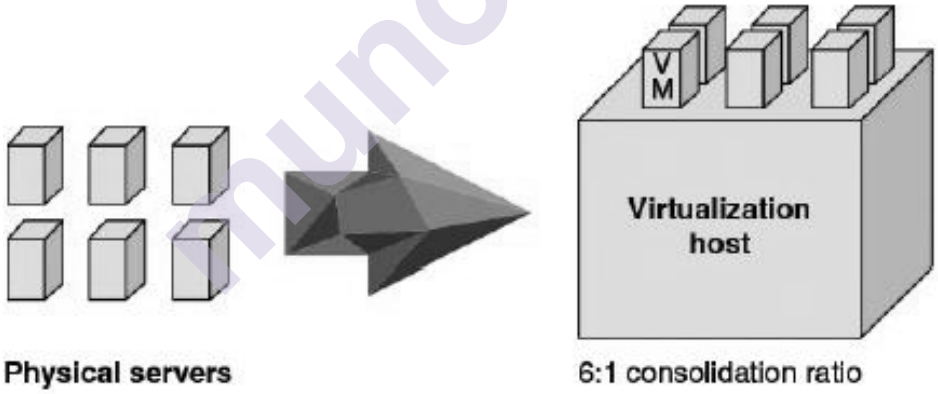


FIGURE 2 Virtual Machine Consolidation

The VM approach is a common way for businesses and individuals to deal with legacy applications and to optimize their hardware usage by maximizing the various kinds of applications that a single computer can handle. Commercial hypervisor offerings by companies such as VMware and Microsoft are widely used, with millions of copies having been sold. A key aspect of server virtualization is that, in addition to the capability of running multiple VMs on one machine, VMs can be viewed as network

resources. Server virtualization has become a central element in dealing with big data applications and in implementing cloud computing infrastructures.

Architectural Approaches:

Virtualization is all about abstraction. Virtualization abstracts the physical hardware from the VMs it supports.

A VM is a software construct that mimics the characteristics of a physical server. It is configured with some number of processors, some amount of RAM, storage resources, and connectivity through the network ports. Once that VM is created, it can be powered on like a physical server, loaded with an operating system and applications, and used in the manner of a physical server. The hypervisor facilitates the translation of I/O from the VM to the physical server devices, and back again to the correct VM. To achieve this, certain privileged instructions that a "native" operating system would be executing on its host's hardware now trigger a hardware trap and are run by the hypervisor as a proxy for the VM. This creates some performance degradation in the virtualization process though over time both hardware and software improvements have minimalized this overhead.

VMs are made up of files. There is a configuration file that describes the attributes of the VM. It contains the server definition, how many virtual processors (vCPUs) are allocated to this VM, how much RAM is allocated, which I/O devices the VM has access to, how many network interface cards (NICs) are in the virtual server, and more. When a VM is powered on, or instantiated, additional files are created for logging, for memory paging, and other functions.

To create a copy of a physical server, additional hardware needs to be acquired, installed, configured, loaded with an operating system, applications, and data, and then patched to the latest revisions, before being turned over to the users.

Because a VM consists of files, by duplicating those files, in a virtual environment there is a perfect copy of the server available in a matter of minutes. There are a few configuration changes to make (server name and IP address to name two), but administrators routinely stand up new VMs in minutes or hours, as opposed to months.

In addition to consolidation and rapid provisioning, virtual environments have become the new model for data center infrastructures for many reasons. One of these is increased availability. VM hosts are clustered together to form pools of computer resources. Multiple VMs are hosted on each of these servers and in the case of a physical server failure,

the VMs on the failed host can be quickly and automatically restarted on another host in the cluster. Compared with providing this type of availability for a physical server, virtual environments can provide higher availability at significantly lower cost and less complexity.

There are two types of hypervisors, distinguished by whether there is another operating system between the hypervisor and the host. A Type 1 hypervisor (see part a of Figure 7.3) is loaded as a thin software layer directly into a physical server, much like an operating system is loaded. Once it is installed and configured, usually within a matter of minutes, the server can then support VMs as guests. Some examples of Type 1 hypervisors are VMware ESXi, Microsoft Hyper-V, and the various open source Xen variants. They are more comfortable with a solution that works as a traditional application, program code that is loaded on top of a Microsoft Windows or UNIX/Linux operating system environment. This is exactly how a Type 2 hypervisor (see part b of Figure 3) is deployed. Some examples of Type 2 hypervisors are VMware Workstation and Oracle VM Virtual Box.

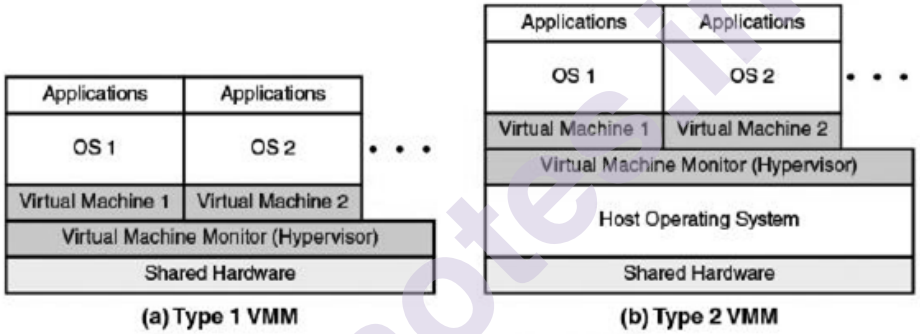


FIGURE 3. Type 1 and Type 2 Virtual Machine Monitors

There are some important differences between the Type 1 and the Type 2 hypervisors. A Type 1 hypervisor is deployed on a physical host and can directly control the physical resources of that host, whereas a Type 2 hypervisor has an operating system between itself and those resources and relies on the operating system to handle all the hardware interactions on the hypervisor's behalf. Typically, Type 1 hypervisors perform better than Type 2 because Type 1 hypervisors do not have that extra layer. Because a Type 1 hypervisor doesn't compete for resources with an operating system, there are more resources available on the host, and by extension, more VMs can be hosted on a virtualization server using a Type 1 hypervisor.

Container Virtualization:

A relatively recent approach to virtualization is known as **container virtualization**. In this approach, software, known as a virtualization **container**, runs on top of the host OS kernel and provides an execution environment for applications (Figure 7.4). Unlike hypervisor-

based VMs, containers do not aim to emulate physical servers. Instead, all containerized applications on a host share a common OS kernel. This eliminates the resources needed to run a separate OS for each application and can greatly reduce overhead.

Because the containers execute on the same kernel, thus sharing most of the base OS, containers are much smaller and lighter weight compared to a hypervisor/guest OS VM arrangement.

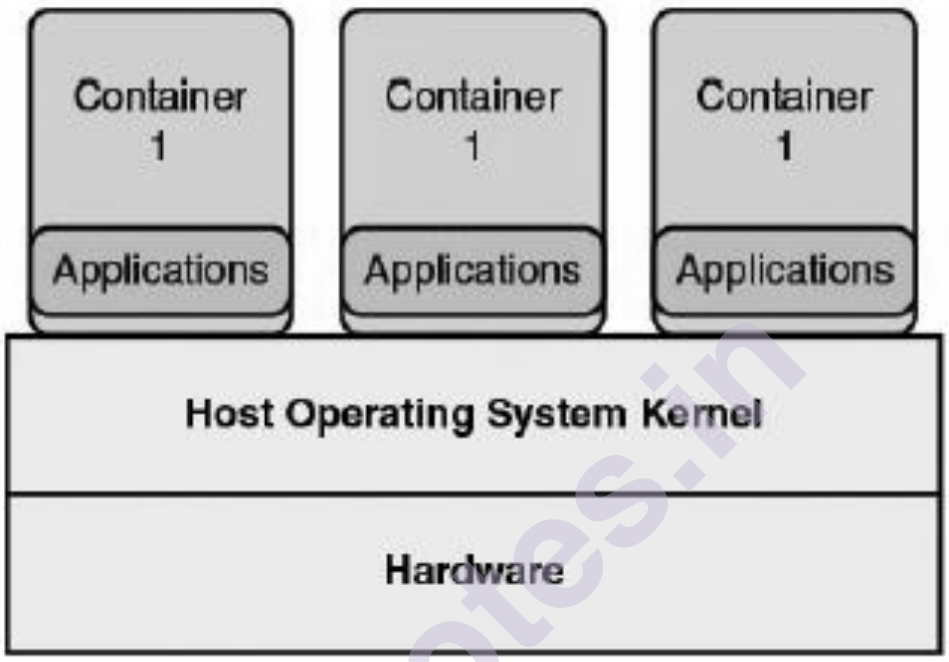


FIGURE 4 Container Virtualization

5.1.2 Nfv Concepts:

Network functions virtualization (NFV) is the virtualization of network functions by implementing these functions in software and running them on VMs. NFV decouples network functions, such as Network Address Translation (NAT), firewalling, intrusion detection, Domain Name Service (DNS), and caching, from proprietary hardware appliances so that they can run in software on VMs. NFV builds on standard VM technologies, extending their use into the networking domain.

- **Network function devices:** Such as switches, routers, network access points, customer premises equipment (CPE), and deep packet inspectors
- **Network-related compute devices:** Such as firewalls, intrusion detection systems, and network management systems.
- **Network-attached storage:** File and database servers attached to the network.

In traditional networks, all devices are deployed on proprietary/closed platforms. All network elements are enclosed boxes, and hardware cannot be shared. Each device requires additional hardware for increased capacity, but this hardware is idle when the system is running below capacity. With NFV, however, network elements are independent applications that are flexibly deployed on a unified platform comprising standard servers, storage devices, and switches. In this way, software and hardware are decoupled, and capacity for each application is increased or decreased by adding or reducing virtual resources (figure 5).

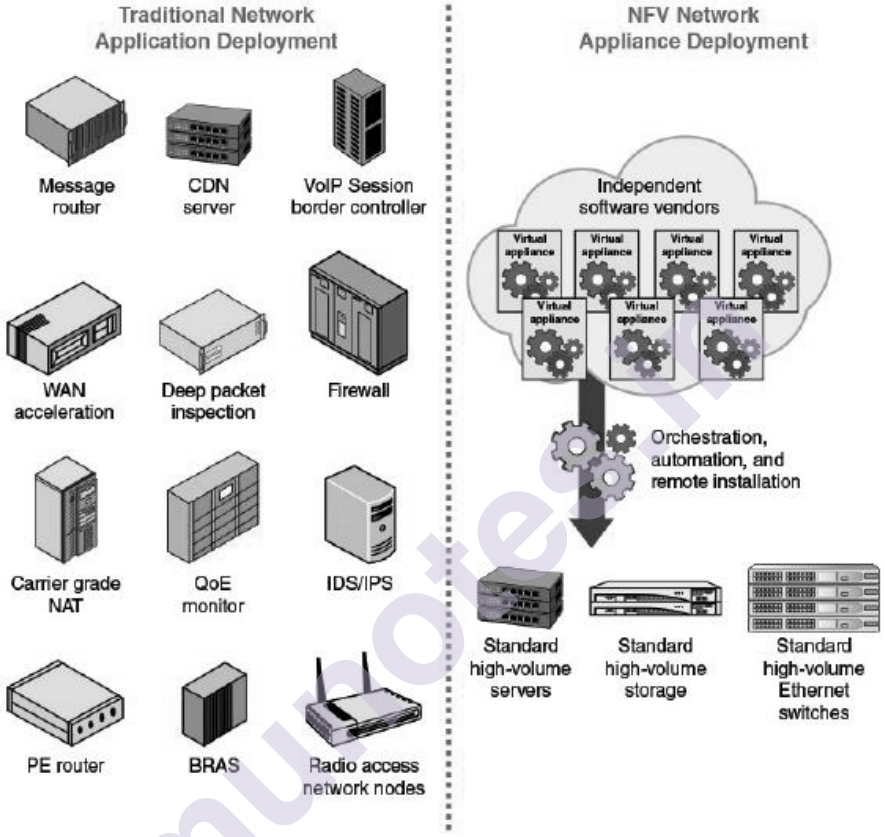


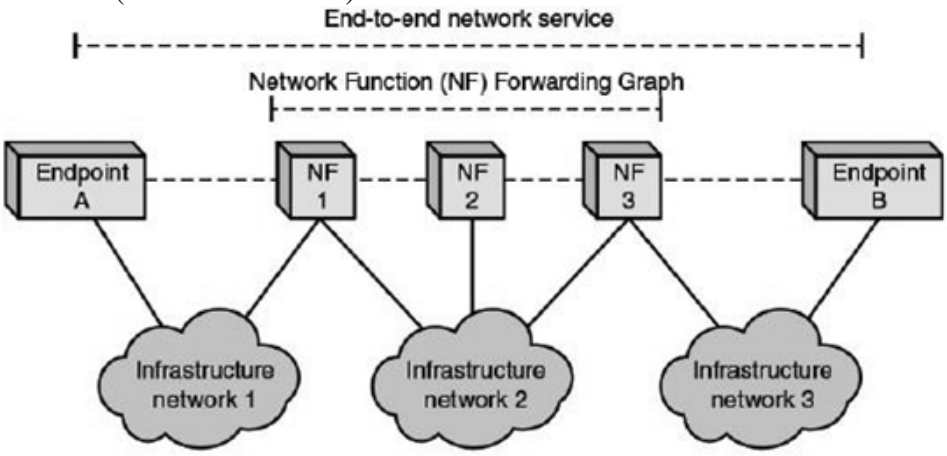
FIGURE 5 Vision for Network Functions Visualization

Simple Example of the Use of NFV:

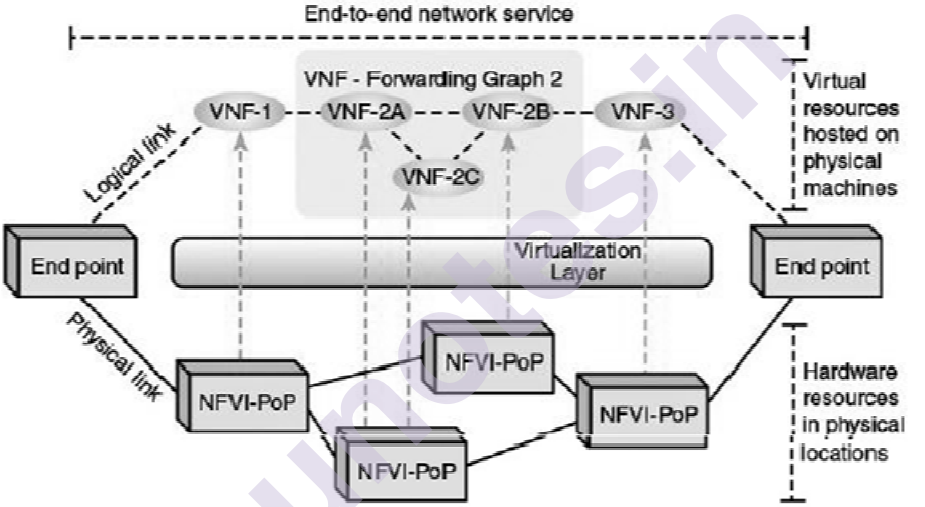
This section considers a simple example from the NFV Architectural Framework document. Part a of Figure 6 shows a physical realization of a network service. At a top level, the network service consists of endpoints connected by a forwarding graph of network functional blocks, called network functions (NFs). Examples of NFs are firewalls, load balancers, and wireless network access points. In the Architectural Framework, NFs are viewed as distinct physical nodes. The endpoints are beyond the scope of the NFV specifications and include all customer-owned devices. So, in the figure, endpoint A could be a smartphone and endpoint B a content delivery network (CDN) server.

Part a of Figure 6 highlights the network functions that are relevant to the service provider and customer. The interconnections among the NFs and endpoints are depicted by dashed lines, representing logical links.

These logical links are supported by physical paths through infrastructure networks (wired or wireless).



(a) Graph representation of an end-to-end network service



(b) Example of an end-to-end network service with VNFs and nested forwarding graphs

FIGURE 6 A Simple NFV Configuration Example

Part b of Figure 6 illustrates a virtualized network service configuration that could be implemented on the physical configuration of part a of Figure 6. VNF-1 provides network access for endpoint A, and VNF-2 provides network access for B. The figure also depicts the case of a nested VNF forwarding graph (VNF-FG-2) constructed from other VNFs (that is, VNF-2A, VNF-2B and VNF-2C). All of these VNFs run as VMs on physical machines, called points of presence (PoPs). VNF-FG-2 consists of three VNFs even though ultimately all the traffic transiting VNF-FG-2 is between VNF-1 and VNF-3. The reason for this is that three separate and distinct network functions are being performed.

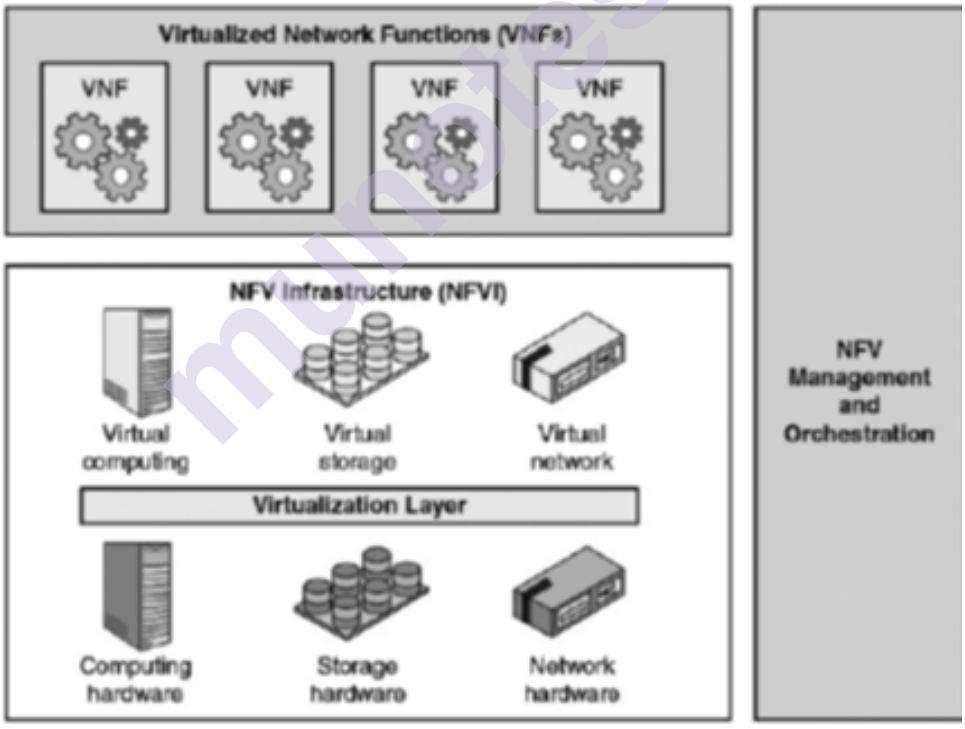
NFV Principles:

Three key NFV principles are involved in creating practical network services:

- **Service chaining:** VNFs are modular and each VNF provides limited functionality on its own. For a given traffic flow within a given application, the service provider steers the flow through multiple VNFs to achieve the desired network functionality. This is referred to as service chaining.
- **Management and orchestration (MANO):** This involves deploying and managing the lifecycle of VNF instances. Examples include VNF instance creation, VNF service chaining, monitoring, relocation, shutdown, and billing. MANO also manages the NFV infrastructure elements.
- **Distributed architecture:** A VNF may be made up of one or more VNF components (VNFC), each of which implements a subset of the VNF's functionality. Each VNFC may be deployed in one or multiple instances. These instances may be deployed on separate, distributed hosts to provide scalability and redundancy.

High-Level NFV Framework:

Figure 7 shows a high-level view of the NFV framework defined by ISG NFV. This framework supports the implementation of network functions as software-only VNFs. We use this to provide an overview of the NFV architecture.



High Level NTV framework

The NFV framework consists of three domains of operation:

- **Virtualized network functions:** The collection of VNFs, implemented in software that run over the NFVI.
- **NFV infrastructure (NFVI):** The NFVI performs a virtualization function on the three main categories of devices in the network service environment**:** computer devices, storage devices, and network devices.
- **NFV management and orchestration:** Encompasses the orchestration and lifecycle management of physical/software resources that support the infrastructure virtualization, and the lifecycle management of VNFs.

Two types of relations between VNFs are supported:

- **VNF forwarding graph (VNF FG):** Covers the case where network connectivity between VNFs is specified, such as a chain of VNFs on the path to a web server tier (for example, firewall, network address translator, load balancer).

- **VNF set:** Covers the case where the connectivity between VNFs is not specified, such as a web server pool.

5.1.3-Nfv Benefits And Requirements:

NFV Benefits:

The following are the most important potential benefits:

- Reduced **CapEx**, by using commodity servers and switches, consolidating equipment, exploiting economies of scale, and supporting pay-as-you grow models to eliminate wasteful overprovisioning.

- Reduced **OpEx**, in terms of power consumption and space usage, by using commodity servers and switches, consolidating equipment, and exploiting economies of scale, and reduced network management and control expenses. Reduced CapeX and OpEx are perhaps the main drivers for NFV.

- The ability to innovate and roll out services quickly, reducing the time to deploy new networking services to support changing business requirements, seize new market opportunities, and improve return on investment of new services. Also lowers the risks associated with rolling out new services, allowing providers to easily trial and evolve services to determine what best meets the needs of customers.

- Ease of interoperability because of standardized and open interfaces.

- Use of a single platform for different applications, users and tenants. This allows network operators to share resources across services and across different customer bases.

- Provided agility and flexibility, by quickly scaling up or down services to address changing demands.

- Targeted service introduction based on geography or customer sets is possible. Services can be rapidly scaled up/down as required.

- A wide variety of ecosystems and encourages openness. It opens the virtual appliance market to pure software entrants, small players and academia, encouraging more innovation to bring new services and new revenue streams quickly at much lower risk.

NFV Requirements:

NFV must be designed and implemented to meet a number of requirements and technical challenges, including the following:

- **Portability/interoperability:** The capability to load and execute VNFs provided by different vendors on a variety of standardized hardware platforms. The challenge is to define a unified interface that clearly decouples the software instances from the underlying hardware, as represented by VMs and their hypervisors.

- **Performance trade-off:** Because the NFV approach is based on industry standard hardware (that is, avoiding any proprietary hardware such as acceleration engines), a probable decrease in performance has to be taken into account. The challenge is how to keep the performance degradation as small as possible by using appropriate hypervisors and modern software technologies, so that the effects on latency, throughput, and processing overhead are minimized.

- **Migration and coexistence with respect to legacy equipment:** The NFV architecture must support a migration path from today's proprietary physical network appliance-based solutions to more open standards- based virtual network appliance solutions. In other words, NFV must work in a hybrid network composed of classical physical network appliances and virtual network appliances.

- **Management and orchestration:** A consistent management and orchestration architecture is required. NFV presents an opportunity, through the flexibility afforded by software network appliances operating in an open and standardized infrastructure, to rapidly align

management and orchestration northbound interfaces to well defined standards and abstract specifications.

- **Automation:** NFV will scale only if all the functions can be automated. Automation of process is paramount to success.

- **Security and resilience:** The security, resilience, and availability of their networks should not be impaired when VNFs are introduced.

- **Network stability:** Ensuring stability of the network is not impacted when managing and orchestrating a large number of virtual appliances between different hardware vendors and hypervisors. This is particularly important when, for example, virtual functions are relocated, or during reconfiguration events (for example, because of hardware and software failures) or because of cyber-attack.

- **Simplicity:** Ensuring that virtualized network platforms will be simpler to operate than those that exist today. A significant focus for network operators is simplification of the plethora of complex network platforms and support systems that have evolved over decades of network technology evolution, while maintaining continuity to support important revenue generating services.

- **Integration:** Network operators need to be able to "mix and match" servers from different vendors, hypervisors from different vendors, and virtual appliances from different vendors without incurring significant integration costs and avoiding lock-in. The ecosystem must offer integration services and maintenance and third-party support; it must be possible to resolve integration issues between several parties.

5.1.4 Nfv Reference Architecture:

Figure 8 shows a more detailed look at the ISG NFV reference architectural framework. You can view this architecture as consisting of four major blocks.

- **NFV infrastructure (NFVI):** Comprises the hardware and software resources that create the environment in which VNFs are deployed. NFVI virtualizes physical computing, storage, and networking and places them into resource pools.
- **VNF/EMS:** The collection of VNFs implemented in software to run on virtual computing, storage, and networking resources, together with a collection of element management systems (EMS) that manage the VNFs.
- **NFV management and orchestration (NFV-MANO):** Framework for the management and orchestration of all resources in the NFV

environment. This includes computing, networking, storage, and VM resources.

- **OSS/BSS:** Operational and business support systems implemented by the VNF service provider.

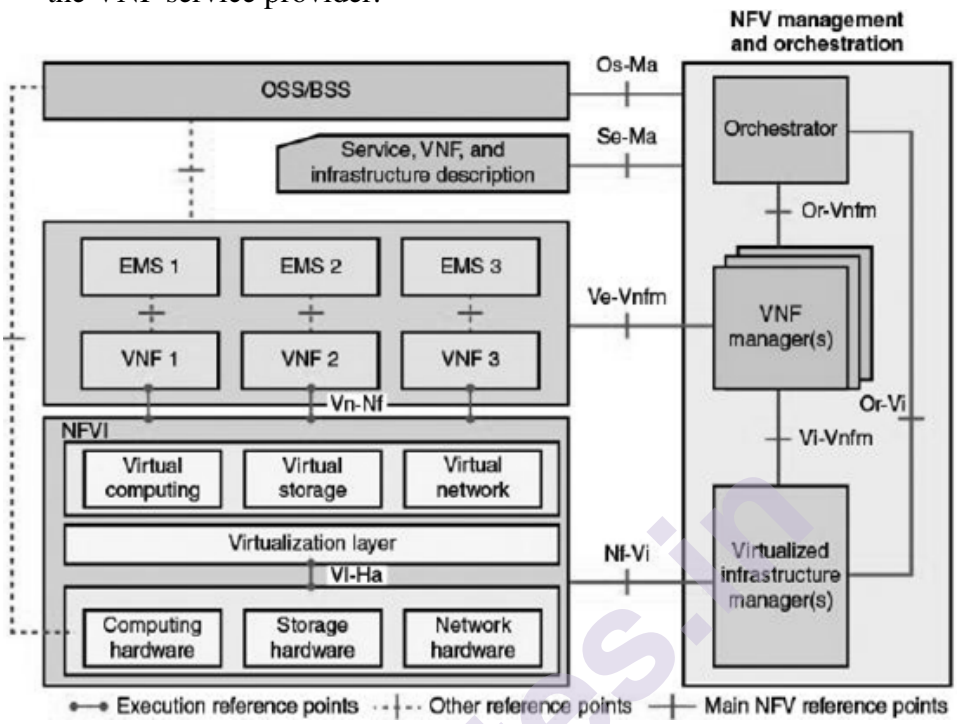


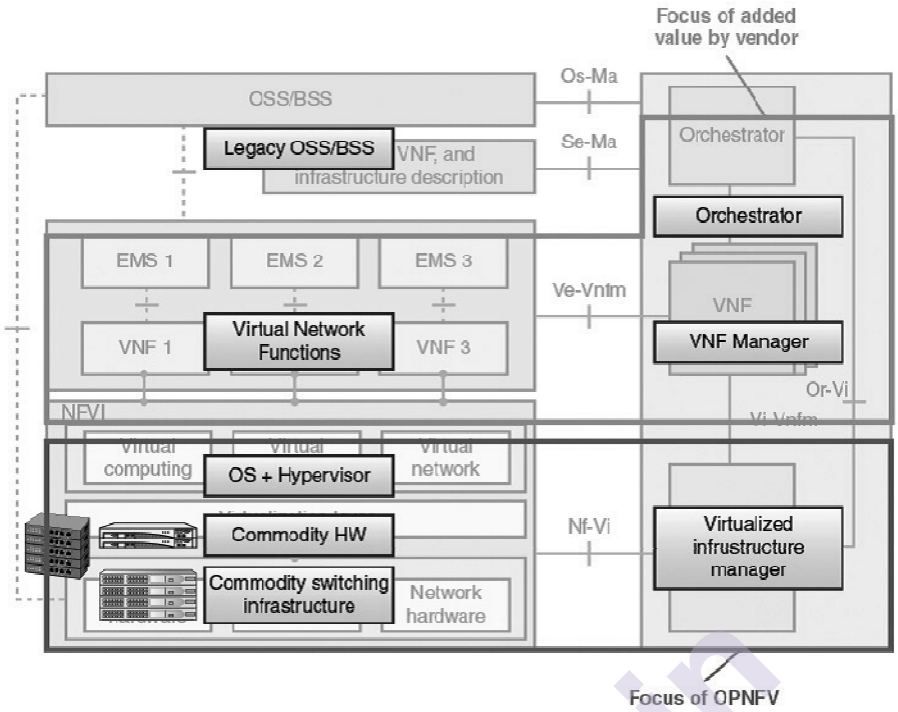
Figure 8 NFV Reference Architectural Framework

It is also useful to view the architecture as consisting of three layers. The NFVI together with the virtualized infrastructure manager provide and manage the virtual resource environment and its underlying physical resources. The VNF layer provides the software implementation of network functions, together with element management systems and one or more VNF managers. Finally, there is a management, orchestration, and control layer consisting of OSS/BSS and the NFV orchestrator.

NNFV Management and Orchestration:

The NFV management and orchestration facility includes the following functional blocks:

- **NFV orchestrator:** Responsible for installing and configuring new network services (NS) and virtual network function (VNF) packages, NS lifecycle management, global resource management, and validation and authorization of NFVI resource requests.
- **VNF manager:** Oversees lifecycle management of VNF instances.
- **Virtualized infrastructure manager:** Controls and manages the interaction of a VNF with computing, storage, and network resources under its authority, in addition to their virtualization.



Reference Points

The main reference points include the following considerations:

- **Vi-Ha:** Marks interfaces to the physical hardware. A well-defined interface specification will facilitate for operators sharing physical resources for different purposes, reassigning resources for different purposes, evolving software and hardware independently, and obtaining software and hardware component from different vendors.
- **Vn-Nf:** These interfaces are APIs used by VNFs to execute on the virtual infrastructure. Application developers, whether migrating existing network functions or developing new VNFs, require a consistent interface the provides functionality and the ability to specify performance, reliability, and scalability requirements.
- **Nf-Vi:** Marks interfaces between the NFVI and the virtualized infrastructure manager (VIM). This interface can facilitate specification of the capabilities that the NFVI provides for the VIM. The VIM must be able to manage all the NFVI virtual resources, including allocation, monitoring of system utilization, and fault management.
- **Or-Vnfm:** This reference point is used for sending configuration information to the VNF manager and collecting state information of the VNFs necessary for network service lifecycle management.
- **Vi-Vnfm:** Used for resource allocation requests by the VNF manager and the exchange of resource configuration and state information.
- **Or-Vi:** Used for resource allocation requests by the NFV orchestrator and the exchange of resource configuration and state information.

- **Os-Ma:** Used for interaction between the orchestrator and the OSS/BSS systems.
- **Ve-Vnfm:** Used for requests for VNF lifecycle management and exchange of configuration and state information.
- **Se-Ma:** Interface between the orchestrator and a data set that provides information regarding the VNF deployment template, VNF forwarding graph, service-related information, and NFV infrastructure information models.

Implementation:

The key objectives of OPNFV are as follows:

- Develop an integrated and tested open source platform that can be used to investigate and demonstrate core NFV functionality.
- Secure proactive participation of leading end users to validate that OPNFV releases address participating operators' needs.
- Influence and contribute to the relevant open source projects that will be adopted in the OPNFV reference platform.
- Establish an open ecosystem for NFV solutions based on open standards and open source software.
- Promote OPNFV as the preferred open reference platform to avoid unnecessary and costly duplication of effort.

The initial scope of OPNFV will be on building NFVI, VIM, and including application programmable interfaces (APIs) to other NFV elements, which together form the basic infrastructure required for VNFs and MANO components. This scope is highlighted in Figure 7.9 as consisting of NFVI and VMI. With this platform as a common base, vendors can add value by developing VNF software packages and associated VNF manager and orchestrator software.

6

NFV FUNCTIONALITY

6.2.1-Nfv Infrastructure:

The heart of the NFV architecture is a collection of resources and functions known as the NFV infrastructure (NFVI). The NFVI encompasses three domains, as illustrated in Figure 8.1 and described in the list that follow-

- **Compute domain:** Provides commercial off-the-shelf (COTS) high-volume servers and storage.
- **Hypervisor domain:** Mediates the resources of the compute domain to the VMs of the software appliances, providing an abstraction of the hardware.
- **Infrastructure network domain (IND):** Comprises all the generic high volume switches interconnected into a network that can be configured to supply infrastructure network services.

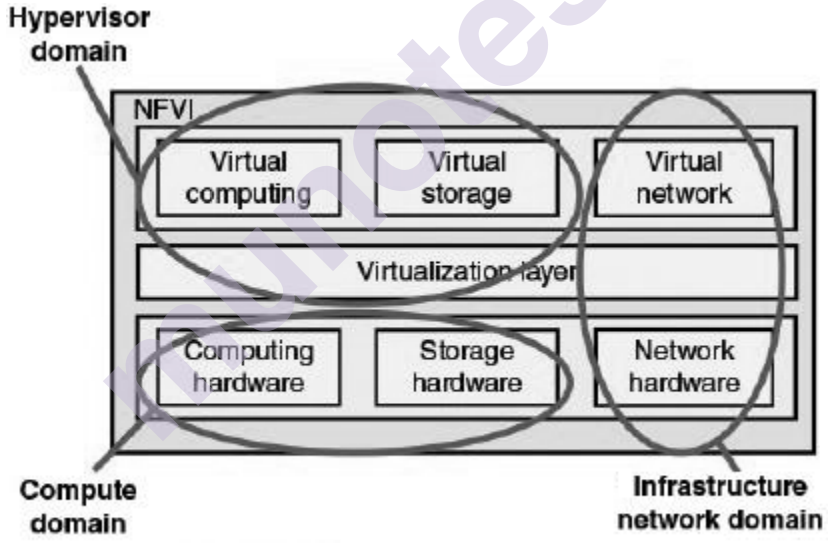


FIGURE [1] NFV Domains

Container Interface:

The ETSI documents make a distinction between a functional block interface and a container interface, as follows:

- **Functional block interface:** An interface between two blocks of software that perform separate (perhaps identical) functions. The interface allows communication between the two blocks. The two functional blocks may or may not be on the same physical host.

- **Container interface:** An execution environment on a host system within which a functional block executes. The functional block is on the same physical host as the container that provides the container interface.

Figure 2 relates container and functional block interfaces to the domain structure of NFVI.

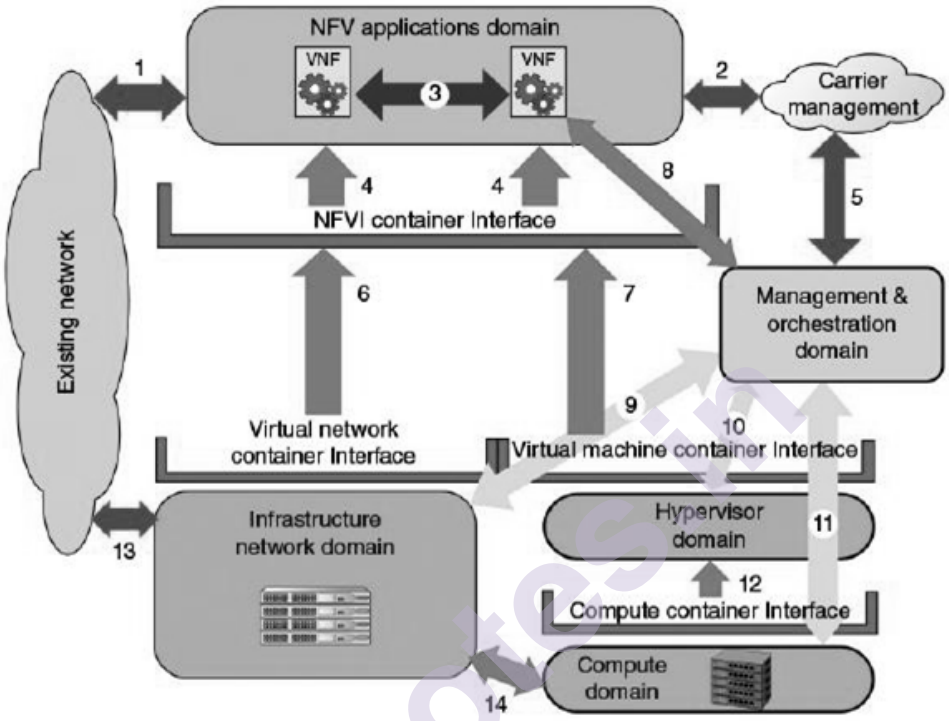


Fig 2: General Domain Architecture and Associated Interface

The ETSI NFVI Architecture Overview document makes the following points concerning this figure:

- The architecture of the VNFs is separated from the architecture hosting the VNFs (that is, the NFVI).
- The architecture of the VNFs may be divided into a number of domains with consequences for the NFVI and vice versa.
- Given the current technology and industrial structure, compute (including storage), hypervisors, and infrastructure networking are already largely separate domains and are maintained as separate domains within the NFVI.
- Management and orchestration tends to be sufficiently distinct from the NFVI as to warrant being defined as its own domain; however, the boundary between the two is often only loosely defined with functions such as element management functions in an area of overlap.
- The interface between the VNF domains and the NFVI is a container interface and not a functional block interface.
- The management and orchestration functions are also likely to be hosted in the NFVI (as VMs) and therefore also likely to sit on a container interface.

Deployment of NFVI Containers:

A single compute or network host can host multiple virtual machines (VMs), each of which can host a single VNF. The single VNF hosted on a VM is referred to as a VNF component (VNFC). A network function may be virtualized by a single VNFC, or multiple VNFCs may be combined to form a single VNF. Part a of Figure 3 shows the organization of VNFCs on a single compute node. The compute container interface hosts a hypervisor, which in turn can host multiple VMs, each hosting a VNFC.

When a VNF is composed of multiple VNFCs, it is not necessary that all the VNFCs execute in the same host. As shown in part b of Figure 3, the VNFCs can be distributed across multiple compute nodes interconnected by network hosts forming the infrastructure network domain.

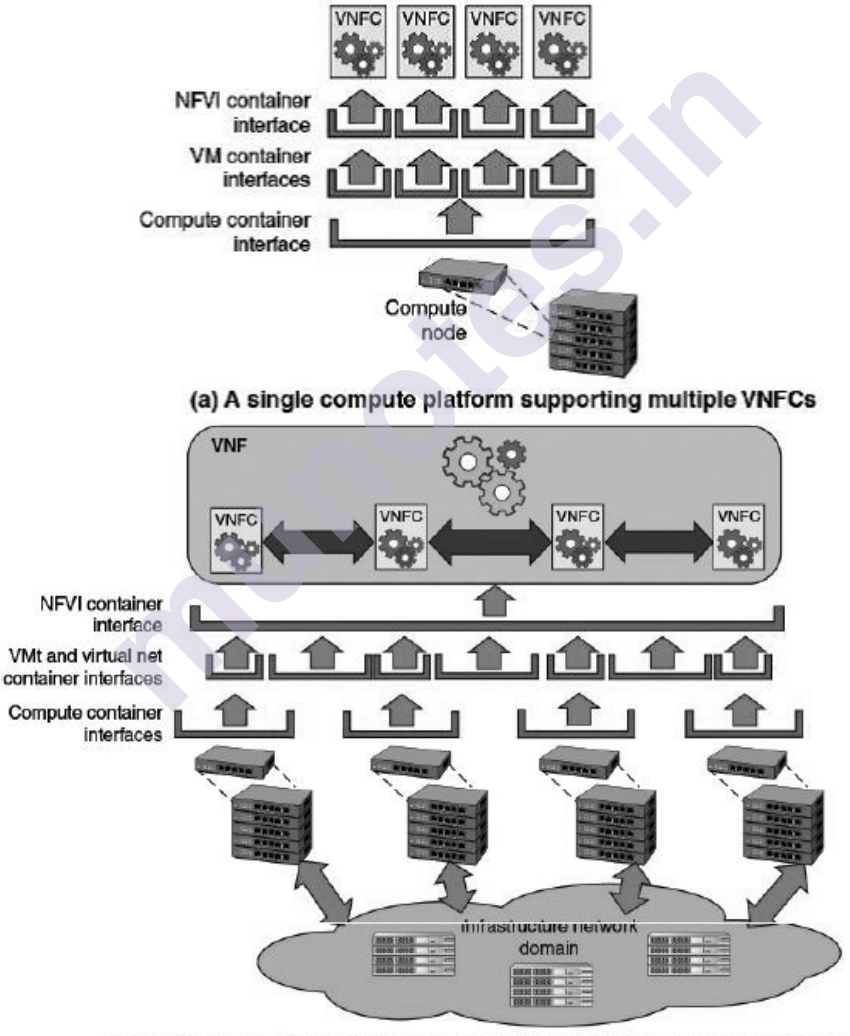
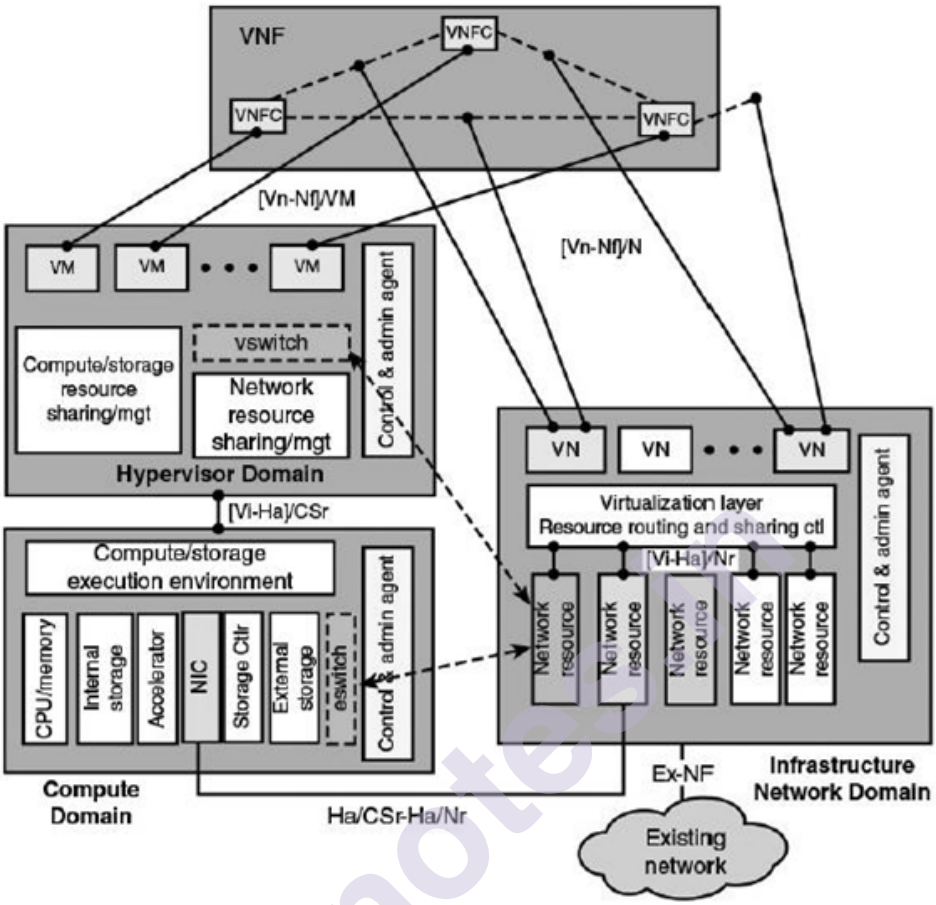


FIGURE 3 Deployment of NVFI Containers

Deployment of NVFI containes
Logical Structure of NFVI Domains

The NFVI domain logical structure provides a framework for such development and identifies the interfaces between the main components, as shown in Figure 5.



Compute Domain

The principal elements in a typical compute domain may include the following:

- **CPU/memory:** A COTS processor, with main memory, that executes the code of the VNFC.
- **Internal storage:** Nonvolatile storage housed in the same physical structure as the processor, such as flash memory.
- **Accelerator:** Accelerator functions for security, networking, and packet processing may also be included.
- **External storage with storage controller:** Access to secondary memory devices.
- **Network interface card (NIC):** Provides the physical interconnection with the infrastructure network domain.
- **Control and admin agent:** Connects to the virtualized infrastructure manager (VIM).

- **Eswitch:** Server embedded switch. However, functionally it forms an integral part of the infrastructure network domain.
- **Compute/storage execution environment:** This is the execution environment presented to the hypervisor software by the server or storage device.
- **Control plane workloads:** Concerned with signaling and control plane protocols such as BGP. Typically, these workloads are more processor rather than I/O intensive and do not place a significant burden on the I/O system.
- **Data plane workloads:** Concerned with the routing, switching, relaying or processing of network traffic payloads. Such workloads can require high I/O throughput.

NFVI Implementation Using Compute Domain Nodes:

The VNFCs run as software on hypervisor domain containers that in turn run on hardware in the compute domain. Although virtual links and networks are defined through the infrastructure network domain, the actual implementation of network functions at the VNF level consists of software on compute domain nodes. The IND interfaces with the compute domain and not directly with the hypervisor domain or the VNFs

An **NFVI-Node** as collection of physical devices deployed and managed as a single entity, providing the NFVI functions required to support the execution environment for VNFs. NFVI nodes are in the compute domain and encompass the following types of compute domain nodes:

- **Compute node:** A functional entity which is capable of executing a generic computational instruction set (each instruction be being fully atomic and deterministic) in such a way that the execution cycle time is of the order of units to tens of nanoseconds irrespective of what specific state is required for cycle execution. In practical terms, this defines a compute node in terms of memory access time
- **Gateway node:** A single identifiable, addressable, and manageable element within an NFVI-Node that implements gateway functions. Gateway functions provide the interconnection between NFVI-PoPs and the transport networks. A gateway may operate at the transport level, dealing with IP and data-link packets, or at the application level.
- **Storage node:** A single identifiable, addressable, and manageable element within an NFVI-Node that provides storage resource using compute, storage, and networking functions. Storage may be physically implemented in a variety of ways. An example of such a storage node may be a physical device accessible via a remote storage technology, such as Network File System (NFS) and Fibre Channel.
- **Network node:** A single identifiable, addressable, and manageable element within an NFVI-Node that provides networking

(switching/routing) resources using compute, storage, and network forwarding functions.

A compute domain within an NFVI node will often be deployed as a number of interconnected physical devices. Physical compute domain nodes may include a number of physical resources, such as a multicore processor, memory subsystems, and NICs. An interconnected set of these nodes comprise one NFVI-Node and constitutes one NFVI point of presence (NFVI-PoP).

The deployment scenarios include the following:

- **Monolithic operator:** One organization owns and houses the hardware equipment and deploys and operates the VNFs and the hypervisors they run on. A private cloud or a data center are examples of this deployment model.
- **Network operator hosting virtual network operators:** Based on the monolithic operator scenario, with the addition that the monolithic operator host other virtual network operators within the same facility. A hybrid cloud is an example of this deployment model.
- **Hosted network operator:** An IT services organization (for example, HP, Fujitsu) operates the compute hardware, infrastructure network, and hypervisors on which a separate network operator (for example, BT, Verizon) runs VNFs. These are physically secured by the IT services organization.
- **Hosted communications providers:** Similar to the hosted network operator scenario, but in this case multiple communications providers are hosted. A community cloud is an example of this deployment model.
- **Hosted communications and application providers:** Similar to the previous scenario. In addition to host network and communications providers, servers in a data center facility are offered to the public for deploying virtualized applications. A public cloud is an example of this deployment model.
- **Managed network service on customer premises:** Similar to the monolithic operator scenario. In this case, the NFV provider's equipment is housed on the customer's premises. One example of this model is a remotely managed gateway in a residential or enterprise location.
- **Managed network service on customer equipment:** Similar to the monolithic operator scenario. In this case, the equipment is housed on the customer's premises on customer equipment. This scenario could be used for managing an enterprise network.

Hypervisor Domain:

The hypervisor domain is a software environment that abstracts hardware and implements services, such as starting a VM, terminating a

VM, acting on policies, scaling, live migration, and high availability. The principal elements in the hypervisor domain are the following:

- **Compute/storage resource sharing/management:** Manages these resources and provides virtualized resource access for VMs.
- **Network resource sharing/management:** Manages these resources and provides virtualized resource access for VMs.
- **Virtual machine management and API:** This provides the execution environment of a single VNFC instance.
- **Control and admin agent:** Connects to the virtualized infrastructure manager (VIM).
- **Vswitch:** The vswitch function, described in the next paragraph, is implemented in the hypervisor domain. However, functionally it forms an integral part of the infrastructure network domain.

Infrastructure Network Domain:

The infrastructure network domain (IND) performs a number of roles. It provides

- The communication channel between the VNFCs of a distributed VNF
- The communications channel between different VNFs
- The communication channel between VNFs and their orchestration and management
- The communication channel between components of the NFVI and their orchestration and management
- The means of remote deployment of VNFCs
- The means of interconnection with the existing carrier network

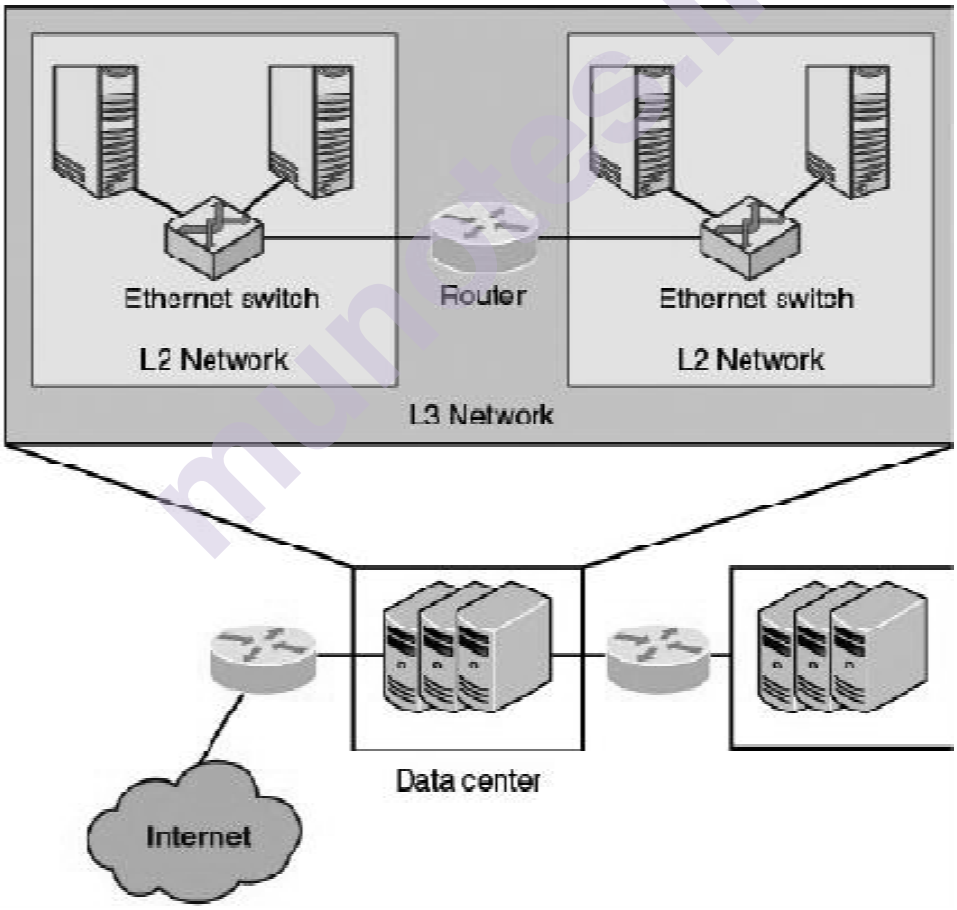
Virtualization in IND creates virtual networks for interconnecting VNFCs with each other and with network nodes outside the NFV ecosystem.

Virtual Networks:

In general terms, a virtual network is an abstraction of physical network resources as seen by some upper software layer. Virtual network technology enables a network provider to support multiple virtual networks that are isolated from one another. Users of a single virtual network are not aware of the details of the underlying physical network or of the other virtual network traffic sharing the physical network resources. Two common approaches for creating virtual networks are (1) protocol-based methods that define virtual networks based on fields in protocol headers, and (2) virtual-machine-based methods, in which networks are created among a set of VMs by the hypervisor. The NFVI network virtualization combines both these forms.

L2 Versus L3 Virtual Networks:

Protocol-based virtual networks can be classified by whether they are defined at protocol Layer 2 (L2), which is typically the LAN media access control (MAC) layer, or Layer 3 (L3), which is typically the Internet Protocol (IP). With an L2 VN, a virtual LAN is identified by a field in the MAC header, such as the MAC address or a virtual LAN ID field inserted into the header. So, for example, within a data center, all the servers and end systems connected to a single Ethernet switch could support virtual LANs among the connected devices. Now suppose there are IP routers connecting segments of the data center, as illustrated in Figure - 5. Normally, an IP router will strip off the MAC header of incoming Ethernet frames and insert a new MAC header when forwarding the packet to the next network. The L2 VN could be extended across this router only if the router had additional capability to support the L2 VN, such as being able to reinsert the virtual LAN ID field in the outgoing MAC frame. Similarly, if an enterprise had two data centers connected by a router and a dedicated line, that router would need the L2 VN capability to extend a VN.



6.2.2Virtualized Network Functions:

A VNF is a virtualized implementation of a traditional network function. Below table contains examples of functions that could be virtualized.

Network Element	Function
Switching elements	Broadband network gateways, carrier grade Network Address Translation (NAT), routers
Mobile network nodes	Home Location Register/Home Subscriber Server, gateway, GPRS support node, radio network controller, various node B functions
Customer premise equipment	Home routers, set-top boxes
Tunneling gateway elements	IPSec/SSL virtual private network gateways
Traffic analysis	Deep packet inspection (DPI), quality of experience measurement
Assurance	Service assurance, service level agreement (SLA) monitoring, testing and diagnostics
Signaling	Session border controllers, IP Multimedia Subsystem components
Control plane/access functions	AAA servers, policy control and charging platforms
Application optimization	Content delivery networks, cache servers, load balancers, accelerators
Security	Firewalls, virus scanners, intrusion detection systems, spam protection

VNF Interfaces:

As discussed earlier, a VNF consists of one or more VNF components (VNFCs). The VNFCs of a single VNF are connected internal to the VNF. This internal structure is not visible to other VNFs or to the VNF user.

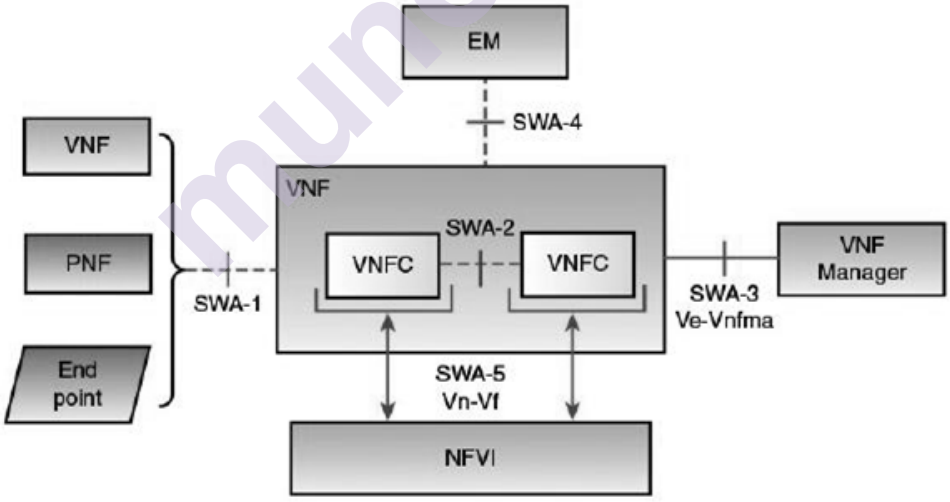


Figure 6 VNF Functional View

Figure 6 shows the interfaces relevant to a discussion of VNFs as described in the list that follows.

- **SWA-1:** This interface enables communication between a VNF and other VNFs, PNFs, and endpoints. Note that the interface is to the VNF as a whole and not to individual VNFCs. SWA-1 interfaces are

logical interfaces that primarily make use of the network connectivity services available at the SWA-5 interface.

- **SWA-2:** This interface enables communications between VNFCs within a VNF. This interface is vendor specific and therefore not a subject for standardization. This interface may also make use of the network connectivity services available at the SWA-5 interface. However, if two VNFCs within a VNF are deployed on the same host, other technologies may be used to minimize latency and enhance throughput, as described below.
- **SWA-3:** This is the interface to the VNF manager within the NFV management and orchestration module. The VNF manager is responsible for lifecycle management (creation, scaling, termination, and so on). The interface typically is implemented as a network connection using IP.
- **SWA-4:** This is the interface for runtime management of the VNF by the element manager.
- **SWA-5:** This interface describes the execution environment for a deployable instance of a VNF. Each VNFC maps to a virtualized container interface to a VM.

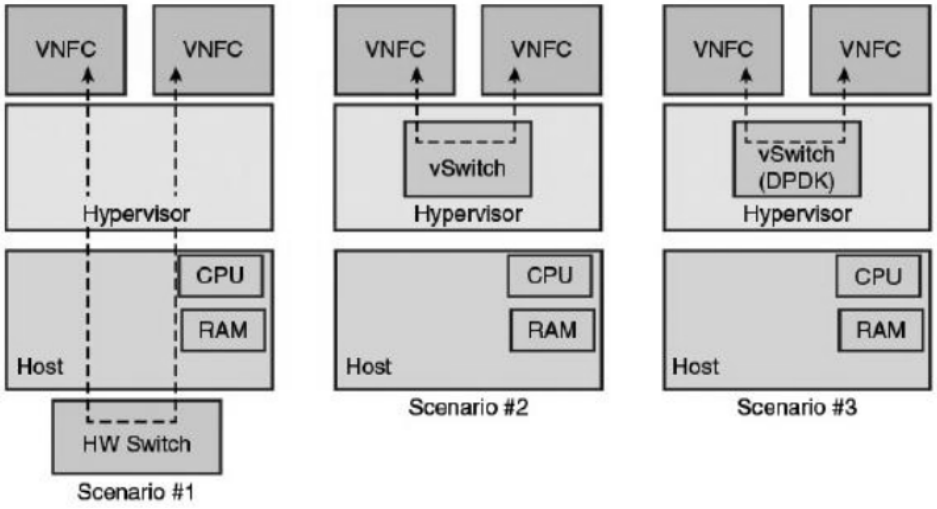
VNFC to VNFC Communication:

The VNF appears as a single functional system in the network it supports. However, internal connectivity between VNFCs within the same VNF or across co-located VNFs needs to be specified by the VNF provider, supported by the NFVI, and managed by the VNF manager. The VNF Architecture document describes a number of architecture design models that are intended to provide desired performance and quality of service (QoS), such as access to storage or compute resources.

Figure 7, from the ETSI VNF Architecture document, illustrates six scenarios using different network technologies to support communication between VNFCs.

1. Communication through a hardware switch. In this case, the VMs supporting the VNFCs bypass the hypervisor to directly access the physical NIC. This provides enhanced performance for VNFCs on different physical hosts.
2. Communication through the vswitch in the hypervisor. This is the basic method of communication between co-located VNFCs but does not provide the QoS or performance that may be required for some VNFs.
3. Greater performance can be achieved by using appropriate data processing acceleration libraries and drivers compatible with the CPU being used. The library is called from the vswitch. An example of a suitable commercial product is the Data Plane Development Kit (DPDK), which is a set of data plane libraries and network interface

controller drivers for fast packet processing on Intel architecture platforms. Scenario 3 assumes a Type 1 hypervisor .



4. Communication through an embedded switch (eswitch) deployed in the NIC with Single Root I/O Virtualization (SR-IOV). SR-IOV is a PCI-SIG specification that defines a method to split a device into multiple PCI express requester IDs (virtual functions) in a fashion that allows an I/O memory management unit (MMU) to distinguish different traffic streams and apply memory and interrupt translations so that these traffic streams can be delivered directly to the appropriate VM, and in a way that prevents nonprivileged traffic flows from impacting other VMs.
5. Embedded switch deployed in the NIC hardware with SR-IOV, and with data plane acceleration software deployed in the VNFC.
6. A serial bus connects directly two VNFCs that have extreme workloads or very low-latency requirements. This is essentially an I/O channel means of communication rather than a NIC means.

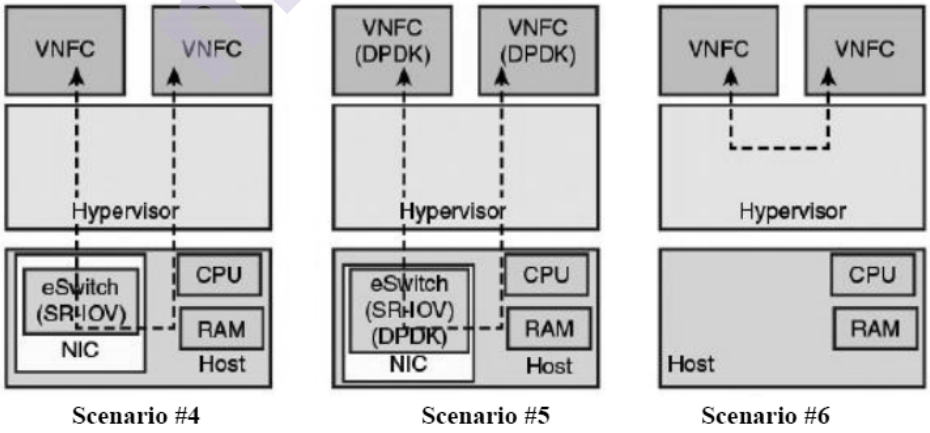


FIGURE 7: VNFC to VNFC Communication

VNF Scaling:

An important property of VNFs is referred to as elasticity, which simply means the ability to **scale up**/down or **scale out**/in. Every VNF has associated with it an elasticity parameter of no elasticity, scale up/down only, scale out/in only, or both scale up/down and scale out/in.

A VNF is scaled by scaling one or more of its constituent VNFCs. Scale out/in is implemented by adding/removing VNFC instances that belong to the VNF being scaled. Scale up/down is implemented by adding/removing resources from existing VNFC instances that belong to the VNF being scaled.

6.2.3 Nfv Management and Orchestration:

The NFV management and orchestration (MANO) component of NFV has as its primary function the management and orchestration of an NFV environment. Figure 8.8, from the ETSI MANO document, shows the basic structure of NFV-MANO and its key interfaces. As can be seen, there are five management blocks: three within NFV-MANO, EMS associated with VNFs, and OSS/BSS.

Virtualized Infrastructure Manager:

Virtualized infrastructure management (VIM) comprises the functions that are used to control and manage the interaction of a VNF with computing, storage, and network resources under its authority, as well as their virtualization.

A VIM performs the following:

- Resource management, in charge of the
  - Inventory of software (for example, hypervisors), computing, storage and network resources dedicated to NFV infrastructure.
  - Allocation of virtualization enablers, for example, VMs onto hypervisors, compute resources, storage, and relevant network connectivity
  - Management of infrastructure resource and allocation, for example, increase resources to VMs, improve energy efficiency, and resource reclamation
- Operations, for
- Visibility into and management of the NFV infrastructure
- Root cause analysis of performance issues from the NFV infrastructure perspective Collection of infrastructure fault information
- Collection of information for capacity planning, monitoring, and optimization

Virtual Network Function Manager:

A VNF manager (VNFM) is responsible for VNFs. Multiple VNFMs may be deployed; a VNFM may be deployed for each VNF, or a VNFM may serve multiple VNFs. Among the functions that a VNFM performs are the following:

- VNF instantiation, including VNF configuration if required by the VNF deployment template (for example, VNF initial configuration with IP addresses before completion of the VNF instantiation operation)
- VNF instantiation feasibility checking, if required
- VNF instance software update/upgrade
- VNF instance modification
- VNF instance scaling out/in and up/down
- VNF instance-related collection of NFVI performance measurement results and faults/events information, and correlation to VNF instance-related events/faults
- VNF instance assisted or automated healing
- VNF instance termination
- VNF lifecycle management change notification
- Management of the integrity of the VNF instance through its lifecycle
- Overall coordination and adaptation role for configuration and event reporting between the VIM and the EM.

NFV Orchestrator:

The NFV orchestrator (NFVO) is responsible for resource orchestration and network service orchestration. Resource orchestration manages and coordinates the resources under the management of different VIMs.

NFVO coordinates, authorizes, releases and engages NFVI resources among different PoPs or within one PoP. This does so by engaging with the VIMs directly through their northbound APIs instead of engaging with the NFVI resources directly.

Network services orchestration manages/coordinates the creation of an end-to-end service that involves VNFs from different VNFMs domains. Service orchestration does this in the following way:

- It creates end-to-end service between different VNFs. It achieves this by coordinating with the respective VNFMs so that it does not need to talk to VNFs directly. An example is creating a service between the base station VNFs of one vendor and core node VNFs of another vendor.
- It can instantiate VNFMs, where applicable.

- It does the topology management of the network services instances (also called VNF forwarding graphs).

Repositories:

Associated with NFVO are four repositories of information needed for the management and orchestration functions:

- **Network services catalog:** List of the usable network services. A deployment template for a network service in terms of VNFs and description of their connectivity through virtual links is stored in NS catalog for future use.
- **VNF catalog:** Database of all usable VNF descriptors. A VNF descriptor (VNFD) describes a VNF in terms of its deployment and operational behavior requirements. It is primarily used by VNFM in the process of VNF instantiation and lifecycle management of a VNF instance. The information provided in the VNFD is also used by the NFVO to manage and orchestrate network services and virtualized resources on NFVI.
- **NFV instances:** List containing details about network services instances and related VNF instances.
- **NFVI resources:** List of NFVI resources utilized for the purpose of establishing NFV services.

Element Management:

The element management is responsible for fault, configuration, accounting, performance, and security (FCAPS) management functionality for a VNF. These management functions are also the responsibility of the VNFM. But EM can do it through a proprietary interface with the VNF in contrast to VNFM. However, EM needs to make sure that it exchanges information with VNFM through open reference point (VeEm-Vnfm). The EM may be aware of virtualization and collaborate with VNFM to perform those functions that require exchange of information regarding the NFVI resources associated with VNF. EM functions include the following:

- Configuration for the network functions provided by the VNF
- Fault management for the network functions provided by the VNF
- Accounting for the usage of VNF functions
- Collecting performance measurement results for the functions provided by the VNF Security management for the VNF functions

OSS/BSS:

The OSS/BSS are the combination of the operator's other operations and business support functions that are not otherwise explicitly captured in the present architectural framework, but are expected to have information exchanges with functional blocks in the NFV-MANO

architectural framework. OSS/BSS functions may provide management and orchestration of legacy systems and may have full end-to-end visibility of services provided by legacy network functions in an operator's network.

In principle, it would be possible to extend the functionalities of existing OSS/BSS to manage VNFs and NFVI directly, but that may be a proprietary implementation of a vendor. Because NFV is an open platform, managing NFV entities through open interfaces (as that in MANO) makes more sense. The existing OSS/BBS, however, can add value to the NFV MANO by offering additional functions if they are not supported by a certain implementation of NFV MANO. This is done through an open reference point (Os-Ma) between NFV MANO and existing OSS/BSS.

6.2.4 Nfv Use Casesp:

There are currently nine use cases, which can be divided into the categories of architectural use cases and service-oriented use cases.

1 Architectural Use Cases:

The four architectural use cases focus on providing general-purpose services and applications based on the NFVI architecture.

i.NFVI as a Service: NFVIaaS is a scenario in which a service provider implements and deploys an NFVI that may be used to support VNFs both by the NFVIaaS provider and by other network service providers. For the NFVIaaS **p**rovider, this service provides for economies of scale. The infrastructure is sized to support the provider's own needs for deploying VNFs and extra capacity that can be sold to other providers. The NFVIaaS customer can offer services using the NFVI of another service provider. The NFVIaaS customer has flexibility in rapidly deploying VNFs, either for new services or to scale out existing services. Cloud computing providers may find this service particularly attractive.

ii. VNF as a Service: Whereas NFVIaaS is similar to the cloud model of Infrastructure as a Service (IaaS), VNFaaS corresponds to the cloud model of Software as a Service (SaaS). NFVIaaS provides the virtualization infrastructure to enable a network service provider to develop and deploy VNFs with reduced cost and time compared to implementing the NFVI and the VNFs. With VNFaaS, a provider develops VNFs that are then available off the shelf to customers. This model is well suited to virtualizing customer premises equipment such as routers and firewalls.

iii.Virtual Network Platform as a Service: VNPaaS is similar to an NFVIaaS that includes VNFs as components of the virtual network infrastructure. The primary differences are the programmability and development tools of the VNPaaS that allow the subscriber to create and

configure custom ETSI NFV-compliant VNFs to augment the catalog of VNFs offered by the service provider.

iv.VNF Forwarding Graphs: VNF FG allows virtual appliances to be chained together in a flexible manner. This technique is called **service chaining**. For example, a flow may pass through a network monitoring VNF, a load-balancing VNF, and finally a firewall VNF in passing from one endpoint to another.

2-Service-Oriented Use Cases:

These use cases focus on the provision of services to end customers, in which the underlying infrastructure is transparent.

i.Virtualization of Mobile Core Network and IP Multimedia Subsystem:

Mobile cellular networks have evolved to contain a variety of interconnected network function elements, typically involving a large variety of proprietary hardware appliances. NFV aims at reducing the network complexity and related operational issues by leveraging standard IT virtualization technologies to consolidate different types of network equipment onto industry standard high-volume servers, switches, and storage, located in NFVI-PoPs.

ii.Virtualization of Mobile Base Station: The focus of this use case is radio access network (RAN) equipment in mobile networks. RAN is the part of a telecommunications system that implements a wireless technology to access the core network of the mobile network service provider. At minimum, it involves hardware on the customer premises or in the mobile device and equipment forming a base station for access to the mobile network.

iii. Virtualization of the Home Environment: This use case deals with network provider equipment located as customer premises equipment (CPE) in a residential location. These CPE devices mark the operator/service provider presence at the customer premises and usually include a residential gateway (RGW) for Internet and Voice over IP (VoIP) services (for example, a modem/router for digital subscriber line [DSL] or cable), and a set-top box (STB) for media services normally supporting local storage for personal video recording (PVR) services.

iv. Virtualization of CDNs: Delivery of content, especially of video, is one of the major challenges of all operator networks because of the massive growing amount of traffic to be delivered to end customers of the network. The growth of video traffic is driven by the shift from broadcast to unicast delivery via IP, by the variety of devices used for video consumption and by increasing quality of video delivered via IP networks in resolution and frame rate. Some Internet service providers (ISPs) are deploying proprietary Content Delivery Network (CDN) cache nodes in their networks to improve delivery of video and other high-bandwidth services to their customers.

V. Fixed Access Network Functions Virtualization: NFV offers the potential to virtualize remote functions in the hybrid fiber/copper access network and passive optical network (PON) fiber to the home and hybrid fiber/wireless access networks. This use case has the potential for cost savings by moving complex processing closer to the network. An additional benefit is that virtualization supports multiple tenancy, in which more than one organizational entity can either be allocated, or given direct control of, a dedicated partition of a virtual access node.

6.2.5 Sdn and Nfv:

The relationship between SDN and NFV is perhaps viewed as SDN functioning as an enabler of NFV. A major challenge with NFV is to best enable the user to configure a network so that VNFs running on servers are connected to the network at the appropriate place, with the appropriate connectivity to other VNFs, and with desired QoS. With SDN, users and orchestration software can dynamically configure the network and the distribution and connectivity of VNFs. Without SDN, NFV requires much more manual intervention, especially when resources beyond the scope of NFVI are part of the environment.

Some of the ways that ETSI believes that NFV and SDN complement each other include the following:

- The SDN controller fits well into the broader concept of a network controller in an NFVI network domain.
- SDN can play a significant role in the orchestration of the NFVI resources, both physical and virtual, enabling functionality such as provisioning, configuration of network connectivity, bandwidth allocation, automation of operations, monitoring, security, and policy control.
- SDN can provide the network virtualization required to support multitenant NFVIs.
- Forwarding graphs can be implemented using the SDN controller to provide automated provisioning of service chains, while ensuring strong and consistent implementation of security and other policies.
- The SDN controller can be run as a VNF, possibly as part of a service chain including other VNFs. For example, applications and services originally developed to run on the SDN controller could also be implemented as separate VNFs.

Figure 10, from the ETSI VNF Architecture document, indicates the potential relationship between SDN and NFV. The arrows can be described as follows-

- SDN enabled switch/NEs include physical switches, hypervisor virtual switches, and embedded switches on the NICs.
- Virtual networks created using an infrastructure network SDN controller provide connectivity services between VNFC instances.

- SDN controller can be virtualized, running as a VNF with its EM and VNF manager. Note that there may be SDN controllers for the physical infrastructure, the virtual infrastructure, and the virtual and physical network functions. As such, some of these SDN controllers may reside in the NFVI or management and orchestration (MANO) functional blocks (not shown in figure).
- SDN enabled VNF includes any VNF that may be under the control of an SDN controller (for example, virtual router, virtual firewall).
-

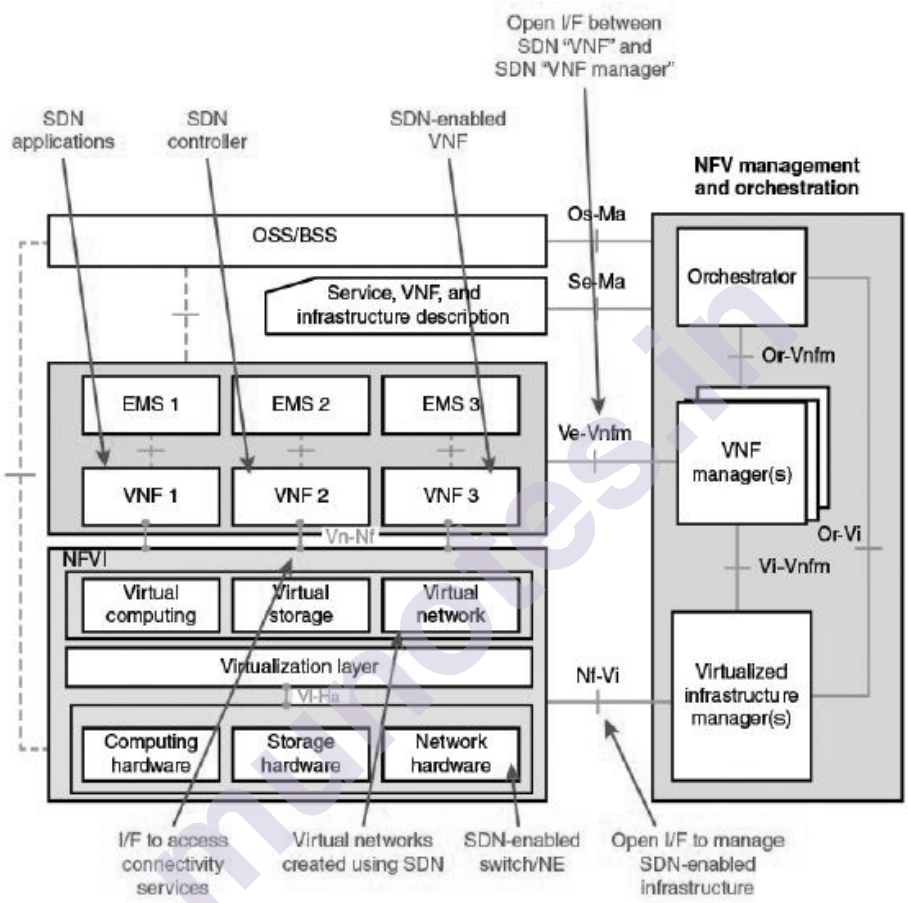


FIGURE 10 Mapping of SDN Components with NFV Architecture

- SDN applications, for example service chaining applications, can be VNF themselves.
- Nf-Vi interface allows management of the SDN enabled infrastructure.
- Ve-Vnfm interface is used between the SDN VNF (SDN controller VNF, SDN network functions VNF, SDN applications VNF) and their respective VNF Manager for lifecycle management.
- Vn-Nf allows SDN VNFs to access connectivity services between VNFC interfaces.

6.3 NETWORK VIRTUALIZATION

Virtual networks have two important benefits:

- They enable the user to construct and manage networks independent of the underlying physical network and with assurance of isolation from other virtual networks using the same physical network.
- They enable network providers to efficiently use network resources to support a wide range of user requirements.

6.3.1 Virtual Lans:

Figure 5.1 shows a relatively common type of hierarchical LAN configuration. In this example, the devices on the LAN are organized into four segments, each served by a LAN switch. The **LAN switch** is a store-and- forward packet-forwarding device used to interconnect a number of end systems to form a LAN segment. The switch can forward a **media access control (MAC) frame:** from a source-attached device to a destination- attached device. It can also broadcast a frame from a source-attached device to all other attached devices. Multiples switches can be interconnected so that multiple LAN segments form a larger LAN. A LAN switch can also connect to a transmission link or a router or other network device to provide connectivity to the Internet or other WANs.

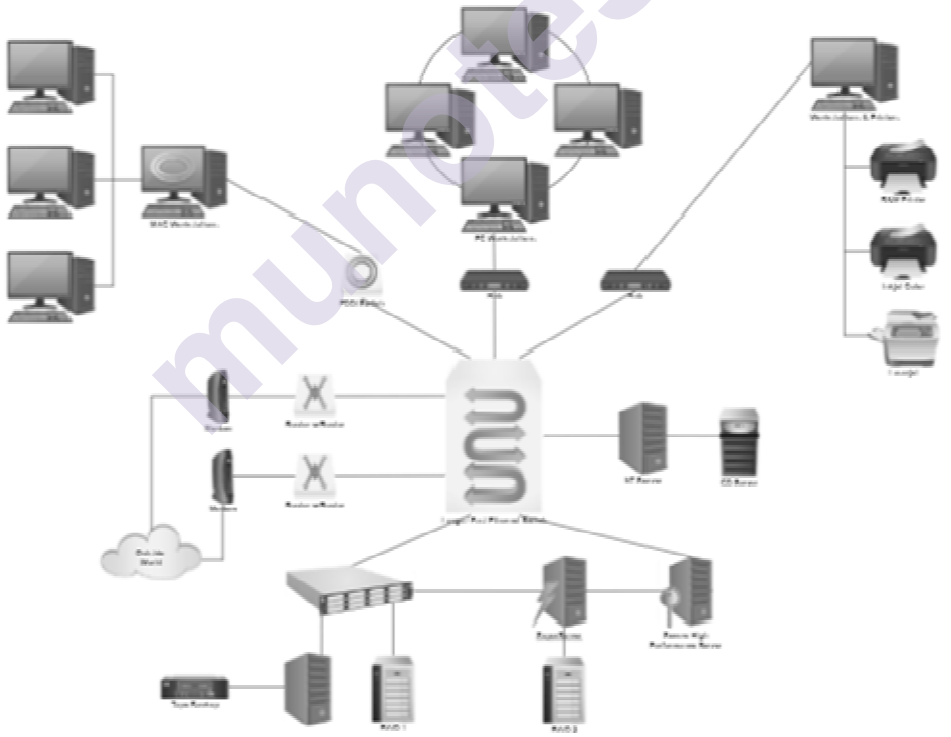


FIGURE 1 A LAN Configuration

Traditionally, a LAN switch operated exclusively at the MAC level. Contemporary LAN switches generally provide greater functionality,

including multilayer awareness (Layers 3, 4, application), quality of service (QoS) support, and trunking for wide-area networking.

The three lower groups in Figure 1 might correspond to different departments, which are physically separated, and the upper group could correspond to a centralized server farm that is used by all the departments.

Consider the transmission of a single MAC frame from workstation X. Suppose the destination MAC address in the frame is workstation Y. This frame is transmitted from X to the local switch, which then directs the frame along the link to Y. If X transmits a frame addressed to Z or W, its local switch forwards the MAC frame through the appropriate switches to the intended destination. All these are examples of **unicast addressing**, in which the destination address in the MAC frame designates a unique destination. A MAC frame may also contain a **broadcast address**, in which case the destination MAC address indicates that all devices on the LAN should receive a copy of the frame. Thus, if X transmits a frame with a broadcast destination address, all the devices on all the switches in Figure 1 receive a copy of the frame. The total collection of devices that receive broadcast frames from each other is referred to as a **broadcast domain**.

In many situations, a broadcast frame is used for a purpose, such as network management or the transmission of some type of alert, with a relatively local significance. Thus, in Figure 5.1, if a broadcast frame has information that is useful only to a particular department, transmission capacity is wasted on the other portions of the LAN and on the other switches.

The Use of Virtual LANs:

A more effective alternative is the creation of VLANs. In essence, a **virtual local-area network (VLAN):** is a logical subgroup within a LAN that is created by software rather than by physically moving and separating devices. It combines user stations and network devices into a single broadcast domain regardless of the physical LAN segment they are attached to and allows traffic to flow more efficiently within populations of mutual interest. The VLAN logic is implemented in LAN switches and functions at the MAC layer. Because the objective is to isolate traffic within the VLAN, a router is required to link from one VLAN to another. Routers can be implemented as separate devices, so that traffic from one VLAN to another is directed to a router, or the router logic can be implemented as part of the LAN switch, as shown in Figure 2.

VLANs enable any organization to be physically dispersed throughout the company while maintaining its group identity. For example, accounting personnel can be located on the shop floor, in the research and development center, in the cash disbursement office, and in

the corporate offices, while all members reside on the same virtual network, sharing traffic only with each other.

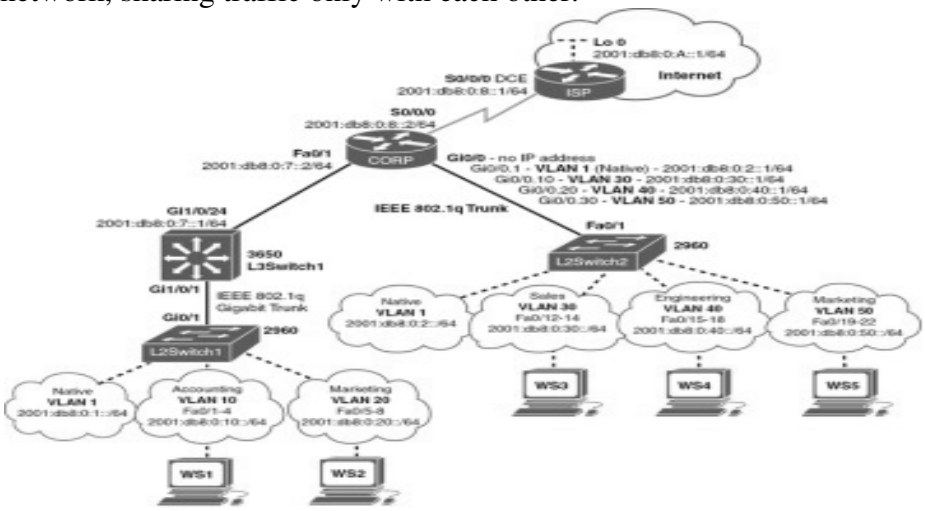


Figure 2 VLAN:

Figure 2 shows five defined VLANs. A transmission from workstation X to server Z is within the same VLAN, so it is efficiently switched at the MAC level. A broadcast MAC frame from X is transmitted to all devices in all portions of the same VLAN. But a transmission from X to printer Y goes from one VLAN to another. Accordingly, router logic at the IP level is required to move the IP packet from X to Y. Figure 2 shows that logic integrated into the switch, so that the switch determines whether the incoming MAC frame is destined for another device on the same VLAN. If not, the switch routes the enclosed IP packet at the IP level.

Defining VLANs:

A VLAN is a broadcast domain consisting of a group of end stations, perhaps on multiple physical LAN segments, that are not constrained by their physical location and can communicate as if they were on a common LAN. A number of different approaches have been used for defining membership, including the following:

- **Membership by port group**: Each switch in the LAN configuration contains two types of ports: a trunk port, which connects two switches; and an end port, which connects the switch to an end system. A VLAN can be defined by assigning each end port to a specific VLAN. This approach has the advantage that it is relatively easy to configure. The principle disadvantage is that the network manager must reconfigure VLAN membership when an end system moves from one port to another.
- **Membership by MAC address**: Because MAC layer addresses are hardwired into the workstation's network interface card (NIC), VLANs based on MAC addresses enable network managers to move a

workstation to a different physical location on the network and have that workstation automatically retain its VLAN membership. The main problem with this method is that VLAN membership must be assigned initially. In networks with thousands of users, this is no easy task. Also, in environments where notebook PCs are used, the MAC address is associated with the docking station and not with the notebook PC. Consequently, when a notebook PC is moved to a different docking station, its VLAN membership must be reconfigured.

- **Membership based on protocol information**: VLAN membership can be assigned based on IP address, transport protocol information, or even higher-layer protocol information. This is a quite flexible approach, but it does require switches to examine portions of the MAC frame above the MAC layer, which may have a performance impact.

Communicating VLAN Membership:

Switches must have a way of understanding VLAN membership (that is, which stations belong to which VLAN) when network traffic arrives from other switches; otherwise, VLANs would be limited to a single switch. One possibility is to configure the information manually or with some type of network management signaling protocol, so that switches can associate incoming frames with the appropriate VLAN.

A more common approach is frame tagging, in which a header is typically inserted into each frame on inter-switch trunks to uniquely identify to which VLAN a particular MAC-layer frame belongs.

IEEE 802.1Q VLAN Standard:

The IEEE 802.1Q standard, defines the operation of VLAN bridges and switches that permits the definition, operation, and administration of VLAN topologies within a bridged/switched LAN infrastructure.

Recall that a VLAN is an administratively configured broadcast domain, consisting of a subset of end stations attached to a LAN. A VLAN is not limited to one switch but can span multiple interconnected switches. In that case, traffic between switches must indicate VLAN membership. This is accomplished in 802.1Q by inserting a tag with a VLAN identifier (VID) with a value in the range from 1 to 4094. Each VLAN in a LAN configuration is assigned a globally unique VID. By assigning the same VID to end systems on many switches, one or more VLAN broadcast domains can be extended across a large network.

Figure 6.4 shows the position and content of the 802.1 tag, referred to as Tag Control Information (TCI). The presence of the two-octet TCI field is indicated by inserting a Length/Type field in the 802.3 MAC frame

with a value of 8100 hex. The TCI consists of three subfields, as described in the list that follows.

- **User priority (3 bits):** The priority level for this frame.
- **Canonical format indicator (1 bit):** Is always set to 0 for Ethernet switches. CFI is used for compatibility between Ethernet type networks and Token Ring type networks. If a frame received at an Ethernet port has a CFI set to 1, that frame should not be forwarded as it is to an untagged port.
- **VLAN identifier (12 bits):** The identification of the VLAN. Of the 4096 possible VIDs, a VID of 0 is used to identify that the TCI contains only a priority value, and 4095 (0xFFF) is reserved, so the maximum possible number of VLAN configurations is 4094.

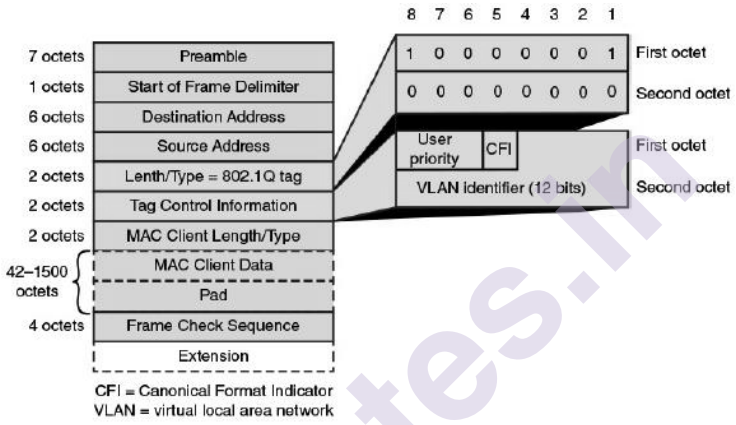


FIGURE 4 Tagged IEEE 802.3 MAC Frame Format

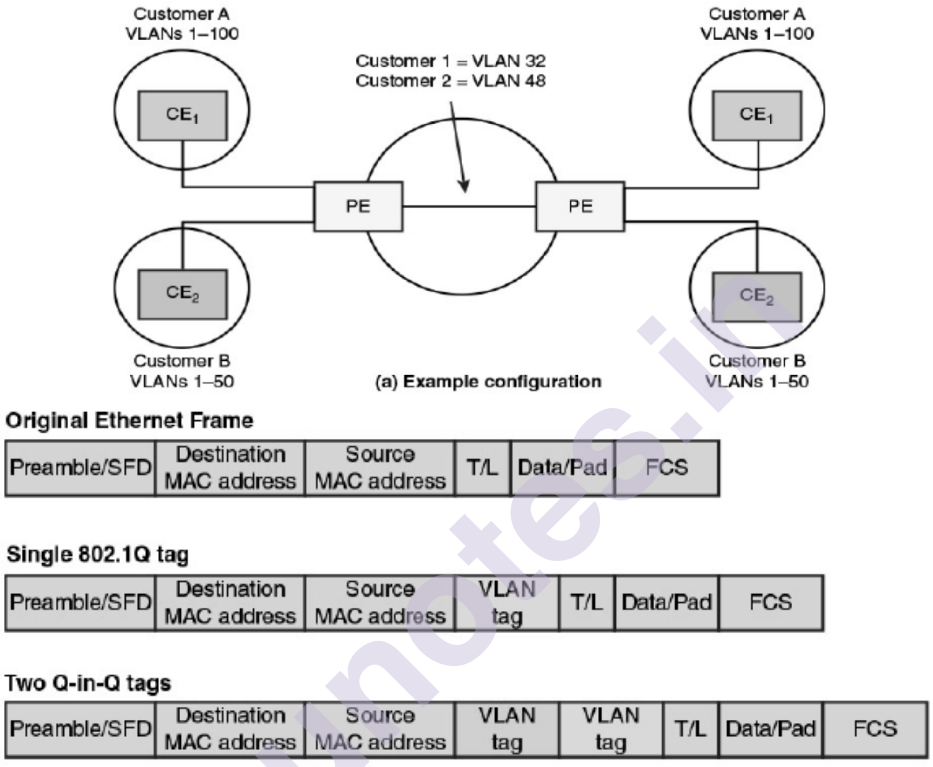
Nested VLANs:

The original 802.1Q specification allowed for a single VLAN tag field to be inserted into an Ethernet MAC frame. More recent versions of the standard allow for the insertion of two VLAN tag fields, allowing the definition of multiple sub-VLANs within a single VLAN.

For example, a single VLAN level suffices for an Ethernet configuration entirely on a single premise. However, it is not uncommon for an enterprise to make use of a network service provider to interconnect multiple LAN locations, and to use metropolitan area Ethernet links to connect to the provider. Multiple customers of the service provider may wish to use the 802.1Q tagging facility across the service provider network (SPN).

One possible approach is for the customer's VLANs to be visible to the service provider. In that case, the service provider could support a total of only 4094 VLANs for all its customers. Instead, the service provider inserts a second VLAN tag into Ethernet frames. For example, consider two customers with multiple sites, both of which use the same SPN (Refer part A of Figure 6). Customer A has configured VLANs 1 to

100 at their sites, and similarly Customer B has configured VLANs 1 to 50 at their sites. The tagged data frames belonging to the customers must be kept separate while they traverse the service provider's network. The customer's data frame can be identified and kept separate by associating another VLAN for that customer's traffic. This results in the tagged customer data frame being tagged again with a VLAN tag, when it traverses the SPN (see part b of Figure 6). The additional tag is removed at the edge of the SPN when the data enters the customer's network again. Packed VLAN tagging is known as VLAN stacking or as Q-in-Q.



b) Position of tags in Ethernet frame

FIGURE 6 Use of Stacked VLAN Tags

6.3.2 Openflow Vlan Support:

A traditional 802.1Q VLAN requires that the network switches have a complete knowledge of the VLAN mapping. Another drawback is related to the choice of one of three ways of defining group membership (port group, MAC address, protocol information). The network administrator must evaluate the trade-offs according to the type of network they wish to deploy and choose one of the possible approaches. It would be difficult to deploy a more flexible definition of a VLAN or even a custom definition (for example, use a combination of IP addresses and ports) with traditional networking devices. Reconfiguring VLANs is also a daunting task for administrators: Multiple switches and routers have to be reconfigured whenever VMs are relocated.

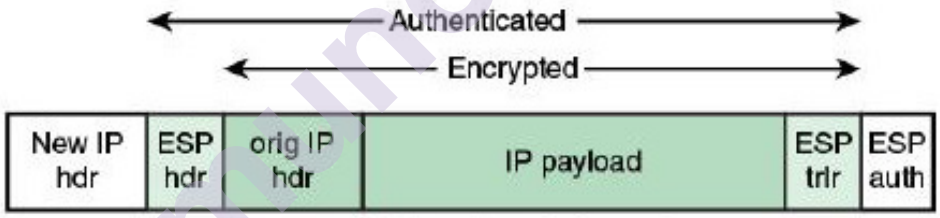
SDN, and in particular OpenFlow, allows for much more flexible management and control of VLANs. It should be clear how OpenFlow can set up flow table entries for forwarding based on one or both VLAN tags, and how tags can be added, modified, and removed.

6.3.3 Virtual Private Networks (Vpn):

A VPN is a private network that is configured within a public network (a carrier's network or the Internet) to take advantage of the economies of scale and management facilities of large networks. VPNs are widely used by enterprises to create WANs that span large geographic areas, to provide site-to-site connections to branch offices, and to allow mobile users to dial up their company LANs. Traffic designated as VPN traffic can only go from a VPN source to a destination in the same VPN. It is often the case that encryption and authentication facilities are provided for the VPN.

A typical scenario for an enterprise that uses VPNs is the following. At each corporate site, one or more LANs link workstations, servers, and databases. The LANs are under the control of the enterprise and can be configured and tuned for cost-effective performance. VPNs over the Internet or some other public network can be used to interconnect sites, providing a cost savings over the use of a private network and offloading the WAN management task to the public network provider. That same public network provides an access path for telecommuters and other mobile employees to log on to corporate systems from remote sites.

1-IPsec VPNs

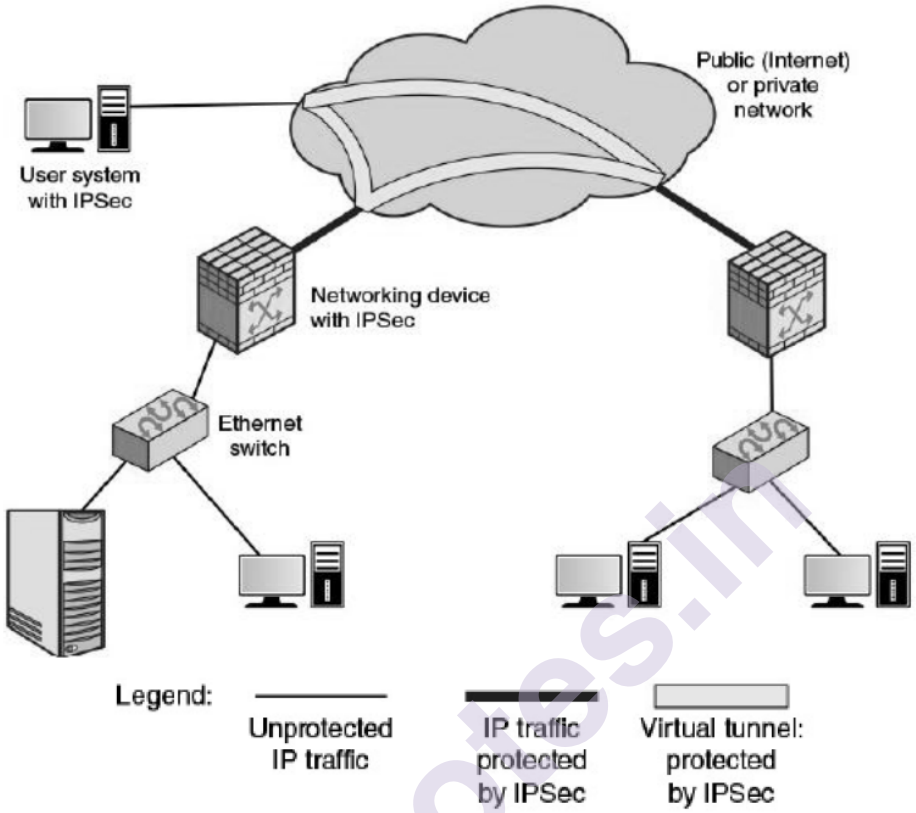


(a) Tunnel-mode format

Use of a shared network, such as the Internet or a public carrier network, as part of an enterprise network architecture exposes corporate traffic to eavesdropping and provides an entry point for unauthorized users. To counter this problem, IPsec can be used to construct VPNs. The principal feature of IPsec that enables it to support these varied applications is that it can encrypt/authenticate traffic at the IP level. Therefore, all distributed applications, including remote logon, client/server, e-mail, file transfer, web access, and so on, can be secured.

Part a of Figure 5.7 shows the packet format for an IPsec option known as tunnel mode. Tunnel mode makes use of the combined authentication/encryption function IPsec called Encapsulating Security Payload (ESP), and a key exchange function. For VPNs, both

authentication and encryption are generally desired, because it is important both to (1) ensure that unauthorized users do not penetrate the VPN, and (2) ensure that eavesdroppers on the Internet cannot read messages sent over the VPN.



(b) Example configuration

Part b of Figure above is a typical scenario of IPsec usage. An organization maintains LANs at dispersed locations. Nonsecure IP traffic is conducted on each LAN. For traffic offsite, through some sort of private or public WAN, IPsec protocols are used.These protocols operate in networking devices, such as a router or firewall, that connect each LAN to the outside world. The IPsec networking device will typically encrypt all traffic going into the WAN, and decrypt and authenticate traffic coming from the WAN; these operations are transparent to workstations and servers on the LAN. Secure transmission is also possible with individual users who connect to the WAN.

Using IPsec to construct a VPN has the following benefits:

- When IPsec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter. Traffic within a company or workgroup does not incur the overhead of security-related processing.

- IPsec in a firewall is resistant to bypass if all traffic from the outside must use IP and the firewall is the only means of entrance from the Internet into the organization.
- IPsec is below the transport layer (TCP, UDP) and so is transparent to applications. There is no need to change software on a user or server system when IPsec is implemented in the firewall or router. Even if IPsec is implemented in end systems, upper-layer software, including applications, is not affected.
- IPsec can be transparent to end users. There is no need to train users on security mechanisms, issue keying material on a per-user basis, or revoke keying material when users leave the organization.
- IPsec can provide security for individual users if needed. This is useful for offsite workers and for setting up a secure virtual subnetwork within an organization for sensitive applications.

2 MPLS Vpns:

An alternative, and popular, means of constructing VPNs is using MPLS. Multiprotocol Label Switching (MPLS) is a set of Internet Engineering Task Force (IETF) specifications for including routing and traffic engineering information in packets. MPLS comprises a number of interrelated protocols, which can be referred to as the MPLS protocol suite. It can be used in IP networks but also in other types of packet-switching networks. MPLS is used to ensure that all packets in a particular flow take the same route over a backbone.

In essence, MPLS is an efficient technique for forwarding and routing packets. MPLS was designed with IP networks in mind, but the technology can be used without IP to construct a network with any link-level protocol. In an MPLS network, a fixed-length label encapsulates an IP packet or a data link frame. The MPLS label contains all the information needed by an MPLS-enabled router to perform routing, delivery, QoS, and traffic management functions. Unlike IP, MPLS is connection oriented.

An MPLS network or internet consists of a set of nodes, called **label-switching routers (LSRs)** capable of switching and routing packets on the basis of a label appended to each packet. Labels define a flow of packets between two endpoints or, in the case of multicast, between a source endpoint and a multicast group of destination endpoints. For each distinct flow, called a **forwarding equivalence class (FEC)**, a specific path through the network of LSRs is defined, called a **label-switched path (LSP)**. All packets in an FEC receive the same treatment en route to the destination. These packets follow the same path and receive the same QoS treatment at each hop. In contrast to forwarding in ordinary IP networks,

the assignment of a particular packet to a particular FEC is done just once, when the packet enters the network of MPLS routers.

Layer 2 MPLS VPN:

With a Layer 2 MPLS VPN, there is mutual transparency between the customer network and the provider network. In effect, the customer requests a mesh of unicast LSPs among customer switches that attach to the provider network. Each LSP is viewed as a Layer 2 circuit by the customer. In an L2VPN, the provider's equipment forwards customer data based on information in the Layer 2 headers, such as an Ethernet MAC address.

Figure 8 depicts key elements in an L2VPN. Customers connect to the provider by means of a Layer 2 device, such as an Ethernet switch; the customer device that connects to the MPLS network is generally referred to as a customer edge (CE) device. The MPLS edge router is referred to as a provider edge (PE) device. The link between the CE and the PE operates at the link layer (for example, Ethernet), and is referred to as an attachment circuit (AC). The MPLS network then sets up an LSP that acts as a tunnel between two edge routers (that is, two PEs) that attach to two networks of the same enterprise. This tunnel can carry multiple virtual channels (VCs) using label stacking. In a manner very similar to VLAN stacking, the use of multiple MPLS labels enables the nesting of VCs.

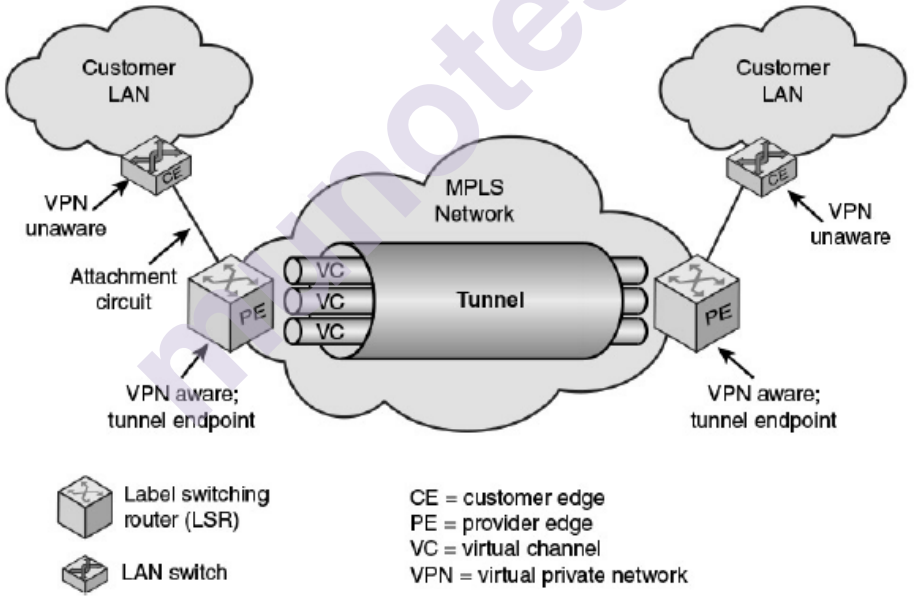


FIGURE 8 MPLS Layer 2 VPN Concepts

When a link-layer frame arrives at the PE from the CE, the PE creates an MPLS packet. The PE pushes a label that corresponds to the VC assigned to this frame. Then the PE pushes a second label onto the label stack for this packet that corresponds to the tunnel between the source and destination PE for this VC. The packet is then routed across the LSP associated with this tunnel, using the top label for label switched routing. At the destination edge, the destination PE pops the tunnel label and

examines the VC label. This tells the PE how to construct a link-layer frame to deliver the payload across to the destination CE.

Layer 3 MPLS VPN:

Whereas L2VPNs are constructed based on link-level addresses (for example, MAC addresses), L3VPNs are based on VPN routes between CEs based on IP addresses. As with an L2VPN, an MPLS-based L3VPN typically uses a stack of two labels. The inner label identifies a specific VPN instance; the outer label identifies a tunnel or route through the MPLS provider network. The tunnel label is associated with an LSP and is used for label swapping and forwarding. At the egress PE, the tunnel label is stripped off, and the VPN label is used to direct the packet to the proper CE and to the proper logical flow at that CE.

For an L3VPN, the CE implements IP and is thus a router. The CE routers advertise their networks to the provider. The provider network can then use an enhanced version of Border Gateway Protocol (BGP) to establish VPNs between CEs. Inside the provider network, MPLS tools are used to establish routes between edge PEs supporting a VPN. Thus, the provider's routers participate in the customer's L3 routing function.

6.3.4 Network Virtualization:

NV is a far broader concept than VPNs, which only provide traffic isolation, or VLANs, which provide a basic form of topology management. NV implies full administrative control for customizing virtual networks both in terms of the physical resources used and the functionalities provided by the virtual networks.

The virtual network presents an abstracted network view whose virtual resources provide users with services similar to those provided by physical networks. Because the virtual resources are software defined, the manager or administrator of a virtual network potentially has a great deal of flexibility in altering topologies, moving resources, and changing the properties and service of various resources. In addition, virtual network users can include not only users of services or applications but also service providers. For example, a cloud service provider can quickly add new services or expanded coverage by leasing virtual networks as needed.

A Simplified Example:

Figure 9 shows a network consisting of three servers and five switches. One server is a trusted platform with a secure operating system that hosts firewall software. All the servers run a hypervisor (virtual machine monitor) enabling them to support multiple VMs. The resources for one enterprise (Enterprise 1) are hosted across the servers and consist of three VMs (VM1a, VM1b, and VM1c) on physical server 1, two VMs (VM1d and VM1e) on physical server 2, and firewall 1 on physical server 3. The virtual switches are used to set up any desired connectivity between

the VMs across the servers through the physical switches. The physical switches provide the connectivity between the physical servers. Each enterprise network is layered as a separate virtual network on top of the physical network. Thus, the virtual network for Enterprise 1 is indicated in Figure 9 by a dashed circle and labeled VN1. The labeled circle VN2 indicates another virtual network.

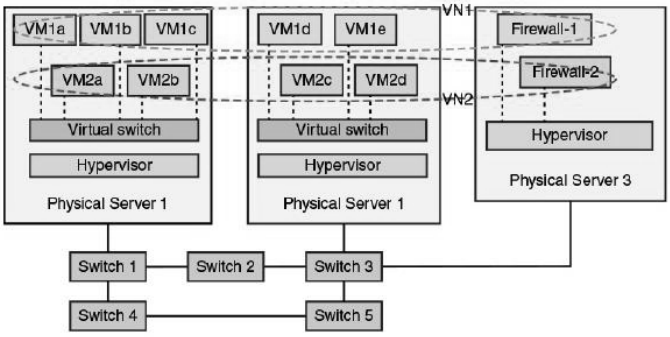


FIGURE 9 Simple Network with Virtual Machines Assigned to Different Administrative Groups

Network Virtualization Architecture:

An excellent overview of the many elements that contribute to an NV environment is provided by the conceptual architecture defined in Y.3011 and shown in Figure 11.

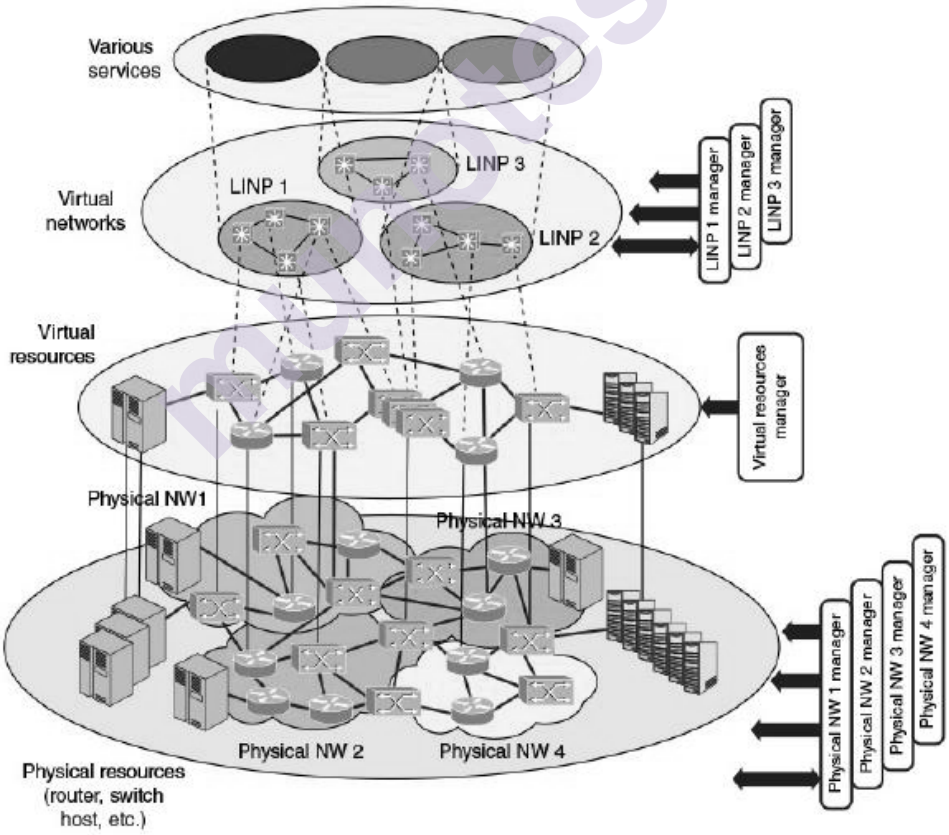


FIGURE 11 Conceptual Architecture of Network Virtualization (Y.3011)

The architecture depicts NV as consisting of four levels:

i. Physical resources
ii. Virtual resources
iii. Virtual networks
iv. Services

A single physical resource can be shared among multiple virtual resources. In turn, each LINP (virtual network) consists of multiple virtual resources and provides a set of services to users.

Various management and control functions are performed at each level, not necessarily by the same provider. There are management functions associated with each physical network and its associated resources. A virtual resource manager (VRM) manages a pool of virtual resources created from the physical resources. A VRM interacts with physical network managers (PNMs) to obtain resource commitments. The VRM constructs LINPs, and an LINP manager is allocated to each LINP.

Benefits of Network Virtualization:

Following are the benefits of NV.

- **Flexibility:** NV enables the network to be quickly moved, provisioned, and scaled to meet the ever- changing needs of virtualized compute and storage infrastructures.
- **Operational cost savings:** Virtualization of the infrastructure streamlines the operational processes and equipment used to manage the network. Similarly, base software can be unified and more easily supported, with a single unified infrastructure to manage services. This unified infrastructure also allows for automation and orchestration within and between different services and components. From a single set of management components, administrators can coordinate resource availability and automate the procedures necessary to make services available, reducing the need for human operators to manage the process and reducing the potential for error.
- **Agility:** Modifications to the network's topology or how traffic is handled can be tried in different ways, without needing to modify the existing physical networks.
- **Scalability:** A virtual network can be rapidly scaled to respond to shifting demands by adding or removing physical resources from the pool of available resources.
- **Capital cost savings:** A virtualized deployment can reduce the number of devices needed, providing capital as well as operational costs savings.
- **Rapid service provisioning/time to market:** Physical resources can be allocated to virtual networks on demand, so that within an

enterprise resources can be quickly shifted as demand by different users or applications changes. From a user perspective, resources can be acquired and released to minimize utilization demand on the system. New services require minimal training and can be deployed with minimal disruption to the network infrastructure.

- **Equipment consolidation:** NV enables the more efficient use of network resources, thus allowing for consolidating equipment purchases to fewer, more off-the-shelf products.

6.3.5 Opendaylight's Virtual Tenant Network:

Virtual Tenant Network (VTN) is an OpenDaylight (ODL) plug-in developed by NEC. It provides multitenant virtual networks on an SDN, using VLAN technology. The VTN abstraction functionality enables users to design and deploy a virtual network without knowing the physical network topology or bandwidth restrictions. VTN allows the users to define the network with a look and feel of a conventional L2/L3 (LAN switch/IP router) network. Once the network is designed on VTN, it is automatically mapped onto the underlying physical network, and then configured on the individual switches leveraging the SDN control protocol.

VTN consists of two components:

- **VTN Manager:** An ODL controller plug-in that interacts with other modules to implement the components of the VTN model. It also provides a REST interface to configure VTN components in the controller.
- **VTN Coordinator:** An external application that provides a REST interface to users for VTN virtualization. It interacts with VTN Manager plug-in to implement the user configuration. It is also capable of multiple controller orchestration.
- Below table shows the elements that are building blocks for constructing a virtual network

Name of Element	Description
vBridge	The logical representation of L2 switch function.
vRouter	The logical representation of router function.
vTep	The logical representation of Tunnel End Point - TEP.
vTunnel	The logical representation of Tunnel.
vBypass	The logical representation of connectivity between controlled networks.
Virtual interface	The representation of end point on the virtual node.
Virtual Linkv(vLink)	The logical representation of L1 connectivity between virtual interfaces.

*****

UNIT IV

7

QUALITY OF SERVICE (QoS) AND USER QUALITY OF EXPERIENCE (QoE)

Unit Structure

7.0 OBJECTIVES

- Describe the ITU-T QoS architectural framework.
- Summarize the key concepts of the Integrated Services Architecture.
- Compare and contrast elastic and inelastic traffic.
- Explain the concept of differentiated services.
- Understand the use of service level agreements.
- Describe IP performance metrics.
- Present an overview of OpenFlow QoS support
- Explain the motivations for QoE.
- Define QoE.
- Explain the factors that could influence QoE.
- Present an overview of how QoE can be measured, including a discussion of the differences between
- Subjective and objective assessment.
- Discuss the various application areas of QoE.

7.1 INTRODUCTION

Fundamental to the acceptance and success of any complex shared networking architecture is that it meets users expectation for performance. Traditionally, the means of defining expected performance, measuring it, providing it, and entering into well-defined agreements relating to it has been the concept of quality of service (QoS).

The Internet and enterprise IP-based networks continue to see rapid growth in the volume and variety of data traffic. Cloud computing, big data, the pervasive use of mobile devices on enterprise networks, and the increasing use of video streaming all contribute to the increasing difficulty in maintaining satisfactory network performance. Two key tools in measuring the network performance that an enterprise desires to achieve are quality of service (QoS) and quality of experience (QoE).

QoS and QoE enable the network manager to determine whether the network is meeting user needs and to diagnose problem areas that require adjustment to network management and network traffic control.

There is a strong need to be able to support a variety of traffic, with a variety of QoS requirements, on IP- based networks.

7.2 BACKGROUND

The Internet and other IP-based networks provided a **best effort** delivery service. This means that the network attempts to allocate its resources with equal availability and priority to all traffic flows, with no regard for application priorities, traffic patterns and load, and customer requirements. To protect the network from congestion collapse and to guarantee that some flows do not crowd out other flows, congestion control mechanisms were introduced, which tended to throttle traffic that consumed excessive resources.

One of the most important congestion control techniques, introduced early on and still in wide use, is the TCP congestion control mechanism. TCP congestion control has become increasingly complex and sophisticated, but it is worth briefly summarizing the principles involved here. For each TCP connection between two end systems across a network, in each direction, a concept known as sliding window is used. TCP segments on a connection are numbered sequentially. The sending and receiving TCP entities maintain a window, or buffer, that defines the range of sequence numbered segments that may be transmitted. As segments arrive and are processed by the receiver, the receiver returns an acknowledgment indicating which segments have been received and implicitly indicated to the sender that the window of sequence numbers has advanced to allow more segments to be sent. Various algorithms are used by the sender to deduce the amount of congestion on a connection based on the round-trip delay for acknowledgments plus whether an acknowledgment is even received for a particular segment. As congestion is detected, the sending TCP entity reduces its transmission of segments to help ease congestion on the intervening network.

Although TCP congestion control and other network congestion control techniques can reduce the risk of excessive congestion, these techniques do not directly address QoS requirements. As the intensity and variety of traffic increased, various QoS mechanisms were developed, including Integrated Services Architecture (ISA) and differentiated services (DiffServ), accompanied by service level agreements (SLAs) so that the service provided to various customers was tunable and somewhat predictable. These mechanisms and services serve two purposes:

- Allocate network resources efficiently so as to maximize effective capacity
- Enable networks to offer different levels of QoS to customers on the basis of customer requirements

In this more sophisticated environment, the term ***best effort refers*** not to the network service as a whole but to a class of traffic treated in best effort fashion. All packets in the best effort traffic class are transmitted with no guarantee regarding the speed with which the packets will be transmitted to the recipient or that the data will even be delivered entirely.

7.3 QoS ARCHITECTURAL FRAMEWORK

The Y.1291 framework consists of a set of generic network mechanisms for controlling the network service response to a service request, which can be specific to a network element, or for signaling between network elements, or for controlling and administering traffic across a network. Figure 7.1 shows the relationship among these elements, which are organized into three planes: data, control, and management.

7.3.1 Data Plane:

The data plane includes those mechanisms that operate directly on flows of data.

- **Traffic classification** refers to the assignment of packets to a traffic class by the ingress router at the ingress edge of the network. Typically, the classification entity looks at multiple fields of a packet, such as source and destination address, application payload, and QoS markings, and determines the aggregate to which the packet belongs. The flow label in the IPv6 header can be used for traffic classification. Other routers en route perform a classification function as well, but the classification does not change as the packets traverse the network.

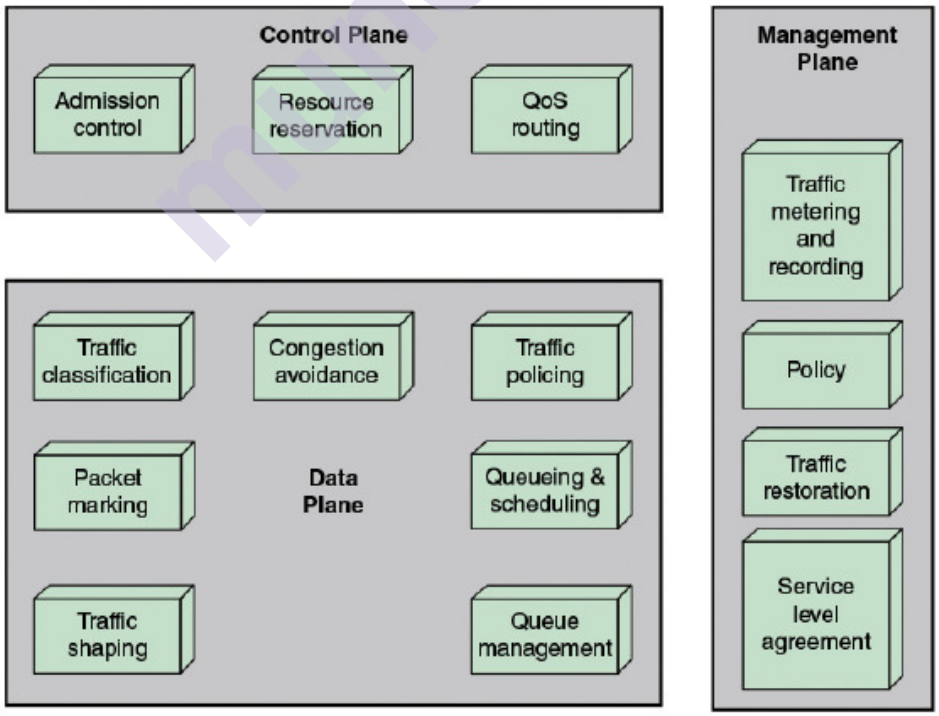


Figure 7.1 Architectural Framework for QoS Support

- **Packet marking** encompasses two distinct functions. First, packets may be marked by ingress edge nodes of a network to indicate some form of QoS that the packet should receive. An example is the Differentiated Services (DiffServ) field in the IPv4 and IPv6 packets and the Traffic Class field in MPLS labels. An ingress edge node can set the values in these fields to indicate a desired QoS. Such markings may be used by intermediate nodes to provide differential treatment to incoming packets. Second, packet marking can also be used to mark packets as nonconformant, either by the ingress node or intermediate nodes, which may be dropped later if congestion is experienced.
- **Traffic shaping** controls the rate and volume of traffic entering and transiting the network on a per-flow basis. The entity responsible for traffic shaping buffers nonconformant packets until it brings the respective aggregate in compliance with the traffic limitations for this flow.
- **Congestion avoidance** deals with means for keeping the load of the network under its capacity such that it can operate at an acceptable performance level. The specific objectives are to avoid significant queuing delays and, especially, to avoid congestion collapse. A typical congestion avoidance scheme acts by senders reducing the amount of traffic entering the network upon an indication that network congestion is occurring (or about to occur).
- **Traffic policing** determines whether the traffic being presented is, on a hop-by-hop basis, compliant with prenegotiated policies or contracts. Nonconformant packets may be dropped, delayed, or labeled as nonconformant.
- **Queuing and scheduling** algorithms, also referred to as queuing discipline algorithms, determine which packet to send next and are used primarily to manage the allocation of transmission capacity among flows.
- **Queue management** algorithms manage the length of packet queues by dropping packets when necessary or appropriate. Active management of queues is concerned primarily with congestion avoidance. One noteworthy example of queue management is random early detection (RED). RED drops incoming packets probabilistically based on an estimated average queue size. The probability for dropping increases as the estimated average queue size grows. There are a number of variants of RED that are in more common use than the original RED, with weighted RED (WRED) perhaps the most commonly implemented.

7.3.2 Control Plane:

The control plane is concerned with creating and managing the pathways through which user data flows. It includes admission control, QoS routing, and resource reservation.

- **Admission control** determines what user traffic may enter the network. This may be in part determined by the QoS requirements of a data flow compared to the current resource commitment within the network. But beyond balancing QoS requests with available capacity to determine whether to accept a request, there are other considerations in admission control.
- **QoS routing** determines a network path that is likely to accommodate the requested QoS of a flow. This contrasts with the philosophy of the traditional routing protocols, which generally are looking for a least-cost path through the network.
- **Resource reservation** is a mechanism that reserves network resources on demand for delivering desired network performance to a requesting flow. An example of a protocol that uses this capability is the Resource Reservation Protocol (RSVP).

7.3.3 Management Plane:

The management plane contains mechanisms that affect both control plane and data plane mechanisms. The control plane deals with the operation, administration, and management aspects of the network. It includes SLAs, traffic restoration, traffic metering and recording, and policy.

A service level agreement (SLA) typically represents the agreement between a customer and a provider of a service that specifies the level of availability, serviceability, performance, operation, or other attributes of the service.

- **Traffic metering and recording** concerns monitoring the dynamic properties of a traffic stream using performance metrics such as data rate and packet loss rate. It involves observing traffic characteristics at a given network point and collecting and storing the traffic information for analysis and further action. Depending on the conformance level, a meter can invoke necessary treatment (for example, dropping or shaping) for the packet stream.
- **Traffic restoration** refers to the network response to failures. This encompasses a number of protocol layers and techniques.
- **Policy** is a category that refers to a set of rules for administering, managing, and controlling access to network resources. They can be specific to the needs of the service provider or reflect the agreement between the customer and service provider, which may include

reliability and availability requirements over a period of time and other QoS requirements.

7.4 INTEGRATED SERVICES ARCHITECTURE (ISA)

7.4.1 ISA Approach:

The purpose of ISA is to enable the provision of QoS support over IP-based internets. The central design issue for ISA is how to share the available capacity in times of congestion. For an IP-based Internet that provides only a best effort service, the tools for controlling congestion and providing service are limited. In essence, routers have two mechanisms to work with:

- **Routing algorithm:** Some routing protocols in use in internets allow routes to be selected to minimize delay. Routers exchange information to get a picture of the delays throughout the Internet. Minimum-delay routing helps to balance loads, thus decreasing local congestion, and helps to reduce delays seen by individual TCP connections.
- **Packet discard:** When a router's buffer overflows, it discards packets. Typically, the most recent packet is discarded. The effect of lost packets on a TCP connection is that the sending TCP entity backs off and reduces its load, thus helping to alleviate Internet congestion.

In ISA, each IP packet can be associated with a flow. A flow is a distinguishable stream of related IP packets that results from a single user activity and requires the same QoS. ISA makes use of the following functions to manage congestion and provide QoS transport:

- **Admission control:** For QoS transport (other than default best effort transport), ISA requires that a reservation be made for a new flow. If the routers collectively determine that there are insufficient resources to guarantee the requested QoS, the flow is not admitted. The protocol RSVP is used to make reservations.
- **Routing algorithm:** The routing decision may be based on a variety of QoS parameters, not just minimum delay.
- **Queuing discipline:** A vital element of the ISA is an effective queuing policy that takes into account the differing requirements of different flows.
- **Discard policy**: A discard policy determines which packets to drop when a buffer is full and new packets arrive.

7.4.2 ISA Components:

Figure 7.2 is a general depiction of the implementation architecture for ISA within a router. Below the thick horizontal line are the forwarding

functions of the router; these are executed for each packet and therefore must be highly optimized. The remaining functions, above the line, are background functions that create data structures used by the forwarding functions.

The principal background functions are as follows:

- **Reservation protocol:** This protocol reserves resources for a new flow at a given level of QoS. It is used among routers and between routers and end systems. The reservation protocol is responsible for maintaining flow-specific state information at the end systems and at the routers along the path of the flow. RSVP is used for this purpose.

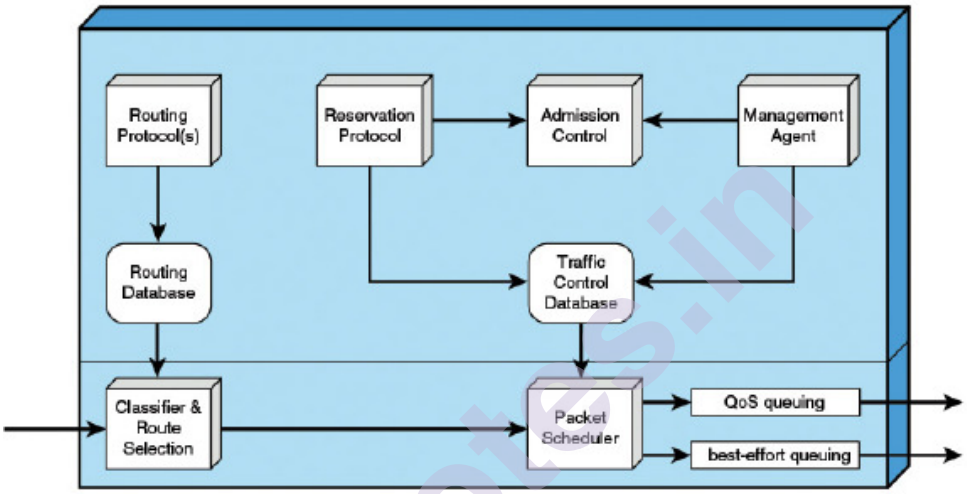


Figure 7.2 Integrated Services Architecture Implemented in Router

- **Admission control**: When a new flow is requested, the reservation protocol invokes the admission control function. This function determines if sufficient resources are available for this flow at the requested QoS. This determination is based on the current level of commitment to other reservations or on the current load on the network.
- **Management agent**: A network management agent can modify the traffic control database and to direct the admission control module to set admission control policies.
- **Routing protocol**: The routing protocol is responsible for maintaining a routing database that gives the next hop to be taken for each destination address and each flow.

These background functions support the main task of the router, which is the forwarding of packets. The two principal functional areas that accomplish forwarding are the following:

1. **Classifier and route selection**: For the purposes of forwarding and traffic control, incoming packets must be mapped into classes. A class

may correspond to a single flow or to a set of flows with the same QoS requirements. For example, the packets of all video flows or the packets of all flows attributable to a particular organization may be treated identically for purposes of resource allocation and queuing discipline. The selection of class is based on fields in the IP header. Based on the packet's class and its destination IP address, this function determines the next-hop address for this packet.

2. **Packet scheduler**: This function manages one or more queues for each output port. It determines the order in which queued packets are transmitted and the selection of packets for discard, if necessary. Decisions are made based on a packet's class, the contents of the traffic control database, and current and past activity on this outgoing port. Part of the packet scheduler's task is that of policing, which is the function of determining whether the packet traffic in a given flow exceeds the requested capacity and, if so, deciding how to treat the excess packets.

7.4.3 ISA Services:

ISA service for a flow of packets is defined on two levels. First, a number of general categories of service are provided, each of which provides a certain general type of service guarantees. Second, within each category, the service for a particular flow is specified by the values of certain parameters; together, these values are referred to as a traffic specification (TSpec). Three categories of service are defined:

1. Guaranteed
2. Controlled load
3. Best effort

An application can request a reservation for a flow for a guaranteed or controlled load QoS, with a TSpec that defines the exact amount of service required. If the reservation is accepted, the TSpec is part of the contract between the data flow and the service.

Guaranteed Service:

The key elements of the guaranteed service are as follows:

- The service provides assured capacity, or data rate.
- There is a specified upper bound on the queuing delay through the network. This must be added to the propagation delay, or latency, to arrive at the bound on total delay through the network.
- There are no queuing losses. That is, no packets are lost because of buffer overflow; packets may be lost because of failures in the network or changes in routing paths.

With this service, an application provides a characterization of its expected traffic profile, and the service determines the end-to-end delay that it can guarantee. The guaranteed service is the most demanding service provided by ISA. Because the delay bound is firm, the delay has to be set at a large value to cover rare cases of long queuing delays.

Controlled Load:

The key elements of the controlled load service are as follows:

- The service tightly approximates the behavior visible to applications receiving best effort service under unloaded conditions.
- There is no specified upper bound on the queuing delay through the network. However, the service ensures that a very high percentage of the packets do not experience delays that greatly exceed the minimum transit delay.
- A very high percentage of transmitted packets will be successfully delivered.

Best Effort :

The risk in an internet that provides QoS for real-time applications is that best effort traffic is crowded out. This is because best effort types of applications are assigned a low priority and their traffic is throttled in the face of congestion and delays. The controlled load service guarantees that the network will set aside sufficient resources so that an application that receives this service will see a network that responds as if these real-time applications were not present and competing for resources.

7.4.4 Queuing Discipline:

An important component of an ISA implementation is the queuing discipline used at the routers. The simplest approach that can be used by a router is a first-in, first-out (FIFO) queuing discipline at each output port. As long as the queue is not empty, the router transmits packets from the queue, taking the oldest remaining packet next.

There are several drawbacks to the FIFO queuing discipline:

- No special treatment is given to packets from flows that are of higher priority or are more delay sensitive. If a number of packets from different flows are ready to be forwarded, they are handled strictly in FIFO order.
- If a number of smaller packets are queued behind a long packet, FIFO queuing results in a larger average delay per packet than if the shorter packets were transmitted before the longer packet. In general, flows of larger packets get better service.

- A selfish TCP connection, which ignores the TCP congestion control rules, can crowd out conforming connections. If congestion occurs and one TCP connection fails to back off, other connections along the same path segment must back off more than they would otherwise have to do.

To overcome the drawbacks of FIFO queuing, a number of more complex routing algorithms have been implemented in routers. These algorithms involve the use of multiple queues at each output port and some method of prioritizing the traffic to provide better service.:

- Priority queuing (PQ)
- Custom queuing (CQ)
- Flow-based weighted fair queuing (WFQ)
- Class-based weighted fair queuing (CBWFQ)

For **priority queuing**, each packet is assigned a priority level, and there is one queue for each priority level. In the Cisco implementation, four levels are used: high, medium, normal, and low. Packets not otherwise classified are assigned to the normal priority. PQ can flexibly prioritize according to network protocol, incoming interface, packet size, source/destination address, or other parameters. The queuing discipline gives absolute preference based on priority.

Custom queuing is designed to allow various applications or organizations to share the network among applications with specific minimum throughput or latency requirements. For CQ, there are multiple queues, with each having a configured byte count. The queues are serviced in round-robin fashion. As each queue is visited, a number of packets are dispatched up to the configured byte count. By providing different byte counts for different queues, traffic on each queue is guaranteed a minimum fraction of the overall capacity. Application or protocol traffic can then be assigned to the desired queue.

The remaining queuing algorithms on the preceding list are based on a mechanism known as fair queuing. With simple fair queuing, each incoming packet is placed in the queue for its flow. The queues are serviced in round-robin fashion, taking one packet from each nonempty queue in turn.

The term *weighted fair queuing* (WFQ) is used in the literature to refer to a class of scheduling algorithms that use multiple queues to support capacity allocation and delay bounds. WFQ may also take into account the amount of service requested by each traffic flow and adjust the queuing discipline accordingly.

Flow-based WFQ, which Cisco simply refers to as WFQ, creates flows based on a number of characteristics in a packet, including source and destination addresses, socket numbers, and session identifiers. The flows are assigned different weights to based on IP precedent bits to provide greater service for certain queues.

Class-based WFQ (CBWFQ) allows a network administrator to create minimum guaranteed bandwidth classes. Instead of providing a queue for each individual flow, a class is defined that consists of one or more flows. Each class can be guaranteed a minimum amount of bandwidth.

7.5 DIFFERENTIATED SERVICES

The differentiated services (DiffServ) architecture is designed to provide a simple, easy-to- implement, low-overhead tool to support a range of network services that are differentiated on the basis of performance. Several key characteristics of DiffServ contribute to its efficiency and ease of deployment:

- IP packets are labeled for differing QoS treatment using the existing IPv4 or IPv6 DSField. Thus, no change is required to IP.
- A service level specification (SLS) is established between the service provider (Internet domain) and the customer prior to the use of DiffServ. This avoids the need to incorporate DiffServ mechanisms in applications. Therefore, existing applications need not be modified to use DiffServ. The SLS is a set of parameters and their values that together define the service offered to a traffic stream by a DiffServ domain.
- A traffic conditioning specification (TCS) is a part of the SLS that specifies traffic classifier rules and any corresponding traffic profiles and metering, marking, discarding/shaping rules which are to apply to the traffic stream.
- DiffServ provides a built-in aggregation mechanism. All traffic with the same DiffServ octet is treated the same by the network service. For example, multiple voice connections are not handled individually but in the aggregate. This provides for good scaling to larger networks and traffic loads.
- DiffServ is implemented in individual routers by queuing and forwarding packets based on the DiffServ octet. Routers deal with each packet individually and do not have to save state information on packet flows.

7.5.1 Services:

The DiffServ type of service is provided within a DiffServ domain, which is defined as a contiguous portion of the Internet over which a consistent

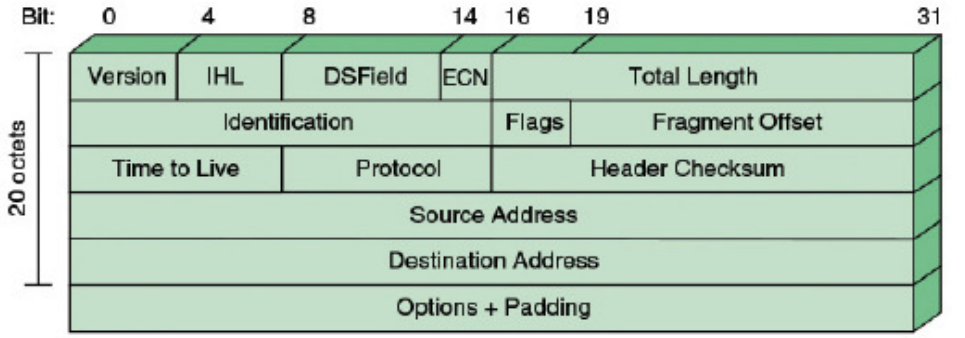
set of DiffServ policies are administered. Typically, a DiffServ domain would be under the control of one administrative entity. The services provided across a DiffServ domain are defined in an SLA, which is a service contract between a customer and the service provider that specifies the forwarding service that the customer should receive for various classes of packets. A customer may be a user organization or another DiffServ domain. Once the SLA is established, the customer submits packets with the DiffServ octet marked to indicate the packet class. The service provider must ensure that the customer gets at least the agreed QoS for each packet class. To provide that QoS, the service provider must configure the appropriate forwarding policies at each router and must measure the performance being provided for each class on an ongoing basis.

A DiffServ framework document lists the following detailed performance parameters that might be included in an SLA:

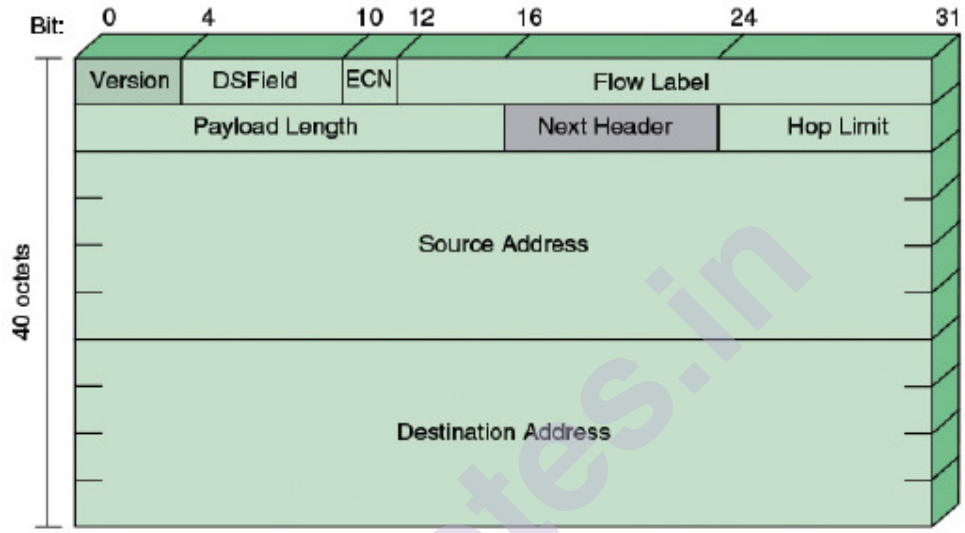
- Detailed service performance parameters such as expected throughput, drop probability, and latency.
- Constraints on the ingress and egress points at which the service is provided, indicating the scope of the service.
- Traffic profiles that must be adhered to for the requested service to be provided, such as token bucket parameters.
- Disposition of traffic submitted in excess of the specified profile.
- The framework document also gives some examples of services that might be provided:
- Traffic offered at service level A will be delivered with low latency.
- Traffic offered at service level B will be delivered with low loss.
- 90 percent of in-profile traffic delivered at service level C will experience no more than 50 ms latency.
- 95 percent of in-profile traffic delivered at service level D will be delivered.
- Traffic offered at service level E will be allotted twice the bandwidth of traffic delivered at service level F. Traffic with drop precedence X has a higher probability of delivery than traffic with drop precedence Y.

7.5.2 DiffServ Field:

Packets are labeled for service handling by means of the 6-bit DSField in the IPv4 header or the IPv6 header. The value of the DSField, referred to as the DiffServ codepoint (DSCP), is the label used to classify packets for differentiated services.



(a) IPv4 Header



(b) IPv6 Header

Figure 7.3

With a 6-bit codepoint, there are in principle 64 different classes of traffic that could be defined. These 64 codepoints are allocated across three pools of codepoints, as follows:

- Codepoints of the form xxxxx0, where x is either 0 or 1, are reserved for assignment as standards.
- Codepoints of the form xxxx11 are reserved for experimental or local use.
- Codepoints of the form xxxx01 are also reserved for experimental or local use.

7.5.3 DiffServ Configuration:

Figure 7.4 illustrates the type of configuration envisioned in the DiffServ documents. A DiffServ domain consists of a set of contiguous routers; that is, it is possible to get from any router in the domain to any other router in the domain by a path that does not include routers outside the domain.

Within a domain, the interpretation of DS codepoints is uniform, so that a uniform, consistent service is provided.

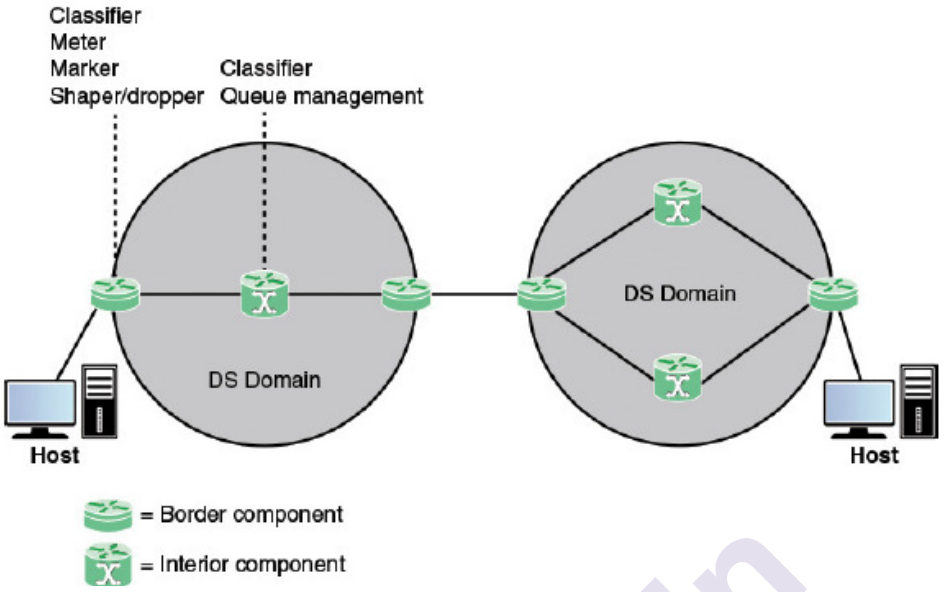


Figure 7.4 DS Domains

Routers in a DiffServ domain are either boundary nodes or interior nodes. Typically, the interior nodes implement simple mechanisms for handling packets based on their DS codepoint values. The DiffServ specifications refer to the forwarding treatment provided at a router as per-hop behavior (PHB). This PHB must be available at all routers, and typically PHB is the only part of DiffServ implemented in interior routers.

The traffic conditioning function consists of five elements:

1. Classifier: Separates submitted packets into different classes. This is the foundation of providing differentiated services. A classifier may separate traffic only on the basis of the DS codepoint (behavior aggregate classifier) or based on multiple fields within the packet header or even the packet payload (multifield classifier).

2. Meter: Measures submitted traffic for conformance to a profile. The meter determines whether a given packet stream class is within or exceeds the service level guaranteed for that class.

3. Marker: Re-marks packets with a different codepoint as needed. This may be done for packets that exceed the profile; for example, if a given throughput is guaranteed for a particular service class, any packets in that class that exceed the throughput in some defined time interval may be re-marked for best effort handling. Also, re-marking may be required at the boundary between two DiffServ domains.

1. **Shaper:** Delays packets as necessary so that the packet stream in a given class does not exceed the traffic rate specified in the profile for that class.
2. **Dropper:** Drops packets when the rate of packets of a given class exceeds that specified in the profile for that class.

7.5.4 DiffServ Operation:

Figure 7.5 illustrates the relationship between the elements of traffic conditioning. After a flow is classified, its resource consumption must be measured. The metering function measures the volume of packets over a particular time interval to determine a flow's compliance with the traffic agreement. If the host is bursty, a simple data rate or packet rate may not be sufficient to capture the desired traffic characteristics. A **token bucket** scheme is an example of a way to define a traffic profile to take into account both packet rate and burstiness.

If a traffic flow exceeds some profile, several approaches can be taken. Individual packets in excess of the profile may be re-marked for lower-quality handling and allowed to pass into the DiffServ domain. A traffic shaper may absorb a burst of packets in a buffer and pace the packets over a longer period. A dropper may drop packets if the buffer used for pacing becomes saturated.

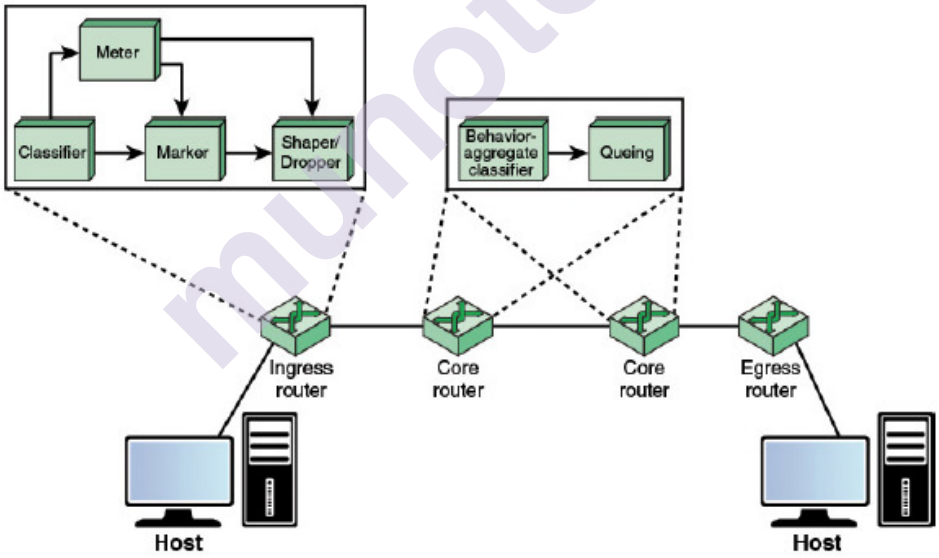


Figure 7.5 DS Functions

7.5.5 Per-Hop Behavior:

DiffServ is a general architecture that can be used to implement a variety of services. As part of the DS standardization effort, specific types of PHB need to be defined, which can be associated with specific differentiated services. The four behavior classes are as follows:

1. Default forwarding (DF) for elastic traffic
2. Assured forwarding (AF) for general QoS requirements
3. Expedited forwarding (EF) for real-time (inelastic) traffic
4. Class selector for historical codepoint definitions and PHB requirements

Figure 7.6 shows the DSCP encodings corresponding to the four classes.

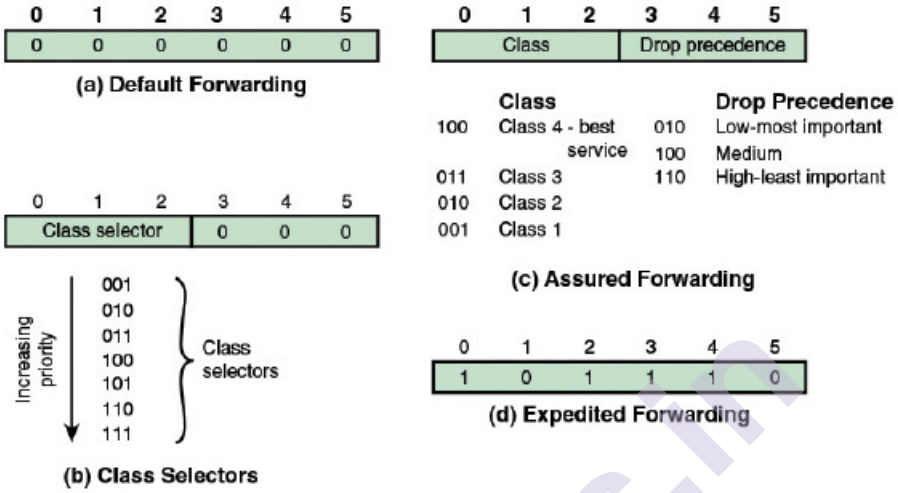


Figure 7.6 DiffServ Forwarding Behavior Classes and Corresponding DSField Encoding

Default Forwarding PHB:

The default class, referred to as default forwarding (DF), is the best effort forwarding behavior in existing routers. Such packets are forwarded in the order that they are received as soon as link capacity becomes available. If other higher-priority packets in other DiffServ classes are available for transmission, the latter are given preference over best effort default packets. Application traffic in the Internet that uses default forwarding is expected to be elastic in nature.

Expedited Forwarding PHB:

RFC 3246 defines the expedited forwarding (EF) PHB as a building block for low-loss, low-delay, and low- jitter end-to-end services through DiffServ domains. Therefore, unless the internet is grossly oversized to eliminate all queuing effects, care must be taken in handling traffic for EF PHB to ensure that queuing effects do not result in loss, delay, or jitter above a given threshold.

The EF PHB is designed to configure nodes so that the traffic aggregate has a well-defined minimum departure rate. The general concept outlined in RFC 3246 is this: The border nodes control the traffic aggregate to limit its characteristics (rate, burstiness) to some predefined level. Interior nodes must treat the incoming traffic in such a way that queuing effects do

not appear. In general terms, the requirement on interior nodes is that the aggregate's maximum arrival rate must be less than the aggregate's minimum departure rate.

Assured Forwarding PHB:

The assured forwarding (AF) PHB is designed to provide a service superior to best effort but one that does not require the reservation of resources within an Internet and does not require the use of detailed discrimination among flows from different users. The AF PHB is more complex than explicit allocation, but it is useful to first highlight the key elements of the explicit allocation scheme:

- Users are offered the choice of a number of classes of service for their traffic. Each class describes a different traffic profile in terms of an aggregate data rate and burstiness.
- Traffic from a user within a given class is monitored at a boundary node. Each packet in a traffic flow is marked out or in based on whether it does or does not exceed the traffic profile.
- Inside the network, there is no separation of traffic from different users or even traffic from different classes. Instead, all traffic is treated as a single pool of packets, with the only distinction being whether each packet has been marked in or out.
- When congestion occurs, the interior nodes implement a dropping scheme in which out packets are dropped before in packets.
- Different users will see different levels of service because they will have different quantities of in packets in the service queues.

7.6 SERVICE LEVEL AGREEMENTS

A service level agreement (SLA) is a contract between a network provider and a customer that defines specific aspects of the service that is to be provided. The definition is formal and typically defines quantitative thresholds that must be met. An SLA typically includes the following information:

- **A description of the nature of service to be provided**: A basic service would be IP-based network connectivity of enterprise locations plus access to the Internet. The service may include additional functions such as web hosting, maintenance of domain name servers, and operation and maintenance tasks.
- **The expected performance level of the service**: The SLA defines a number of metrics, such as delay, reliability, and availability, with numerical thresholds.
- **The process for monitoring and reporting the service level:** This describes how performance levels are measured and reported.

Figure 7.1 shows a typical configuration that lends itself to an SLA. In this case, a network service provider maintains an IP-based network. A customer has a number of private networks (for example, LANs) at various sites. Customer networks are connected to the provider via access routers at the access points.

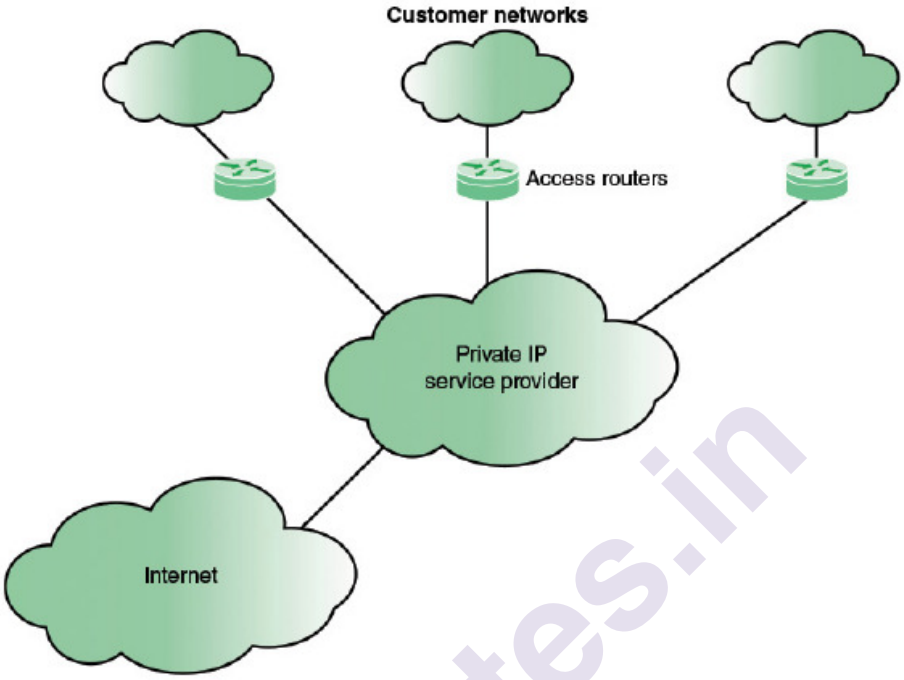


Figure 7.1 Typical Framework for Service Level Agreement

7.1 IP PERFORMANCE METRICS

The IP Performance Metrics Working Group (IPPM) is chartered by IETF to develop standard metrics that relate to the quality, performance, and reliability of Internet data delivery. Two trends dictate the need for such a standardized measurement scheme:

- The Internet has grown and continues to grow at a dramatic rate. Its topology is increasingly complex. As its capacity has grown, the load on the Internet has grown at an even faster rate. Similarly, private internets, such as corporate intranets and extranets, have exhibited similar growth in complexity, capacity, and load. The sheer scale of these networks makes it difficult to determine quality, performance, and reliability characteristics.
- The Internet serves a large and growing number of commercial and personal users across an expanding spectrum of applications. Similarly, private networks are growing in terms of user base and range of applications. Some of these applications are sensitive to particular QoS parameters, leading users to require accurate and understandable performance metrics.

A standardized and effective set of metrics enables users and service providers to have an accurate common understanding of the performance of the Internet and private internets. Measurement data is useful for a variety of purposes, including the following:

- Supporting capacity planning and troubleshooting of large complex internets.
- Encouraging competition by providing uniform comparison metrics across service providers.
- Supporting Internet research in such areas as protocol design, congestion control, and QoS.
- Verification of SLAs.

These metrics are defined in three stages:

1. **Singleton metric:** The most elementary, or atomic, quantity that can be measured for a given performance metric. For example, for a delay metric, a singleton metric is the delay experienced by a single packet.
2. **Sample metric:** A collection of singleton measurements taken during a given time period. For example, for a delay metric, a sample metric is the set of delay values for all the measurements taken during a one-hour period.
3. **Statistical metric:** A value derived from a given sample metric by computing some statistic of the values defined by the singleton metric on the sample. For example, the mean of all the one-way delay values on a sample might be defined as a statistical metric.

The measurement technique can be either active or passive.

- **Active techniques** require injecting packets into the network for the sole purpose of measurement. There are several drawbacks to this approach. The load on the network is increased. This, in turn, can affect the desired result.
- **Passive techniques** observe and extract metrics from existing traffic. This approach can expose the contents of Internet traffic to unintended recipients, creating security and privacy concerns. So far, the metrics defined by the IPPM working group are all active.

For the sample metrics, the simplest technique is to take measurements at fixed time intervals, known as periodic sampling. There are several problems with this approach. First, if the traffic on the network exhibits periodic behavior, with a period that is an integer multiple of the sampling period (or vice versa), correlation effects may result in inaccurate values.

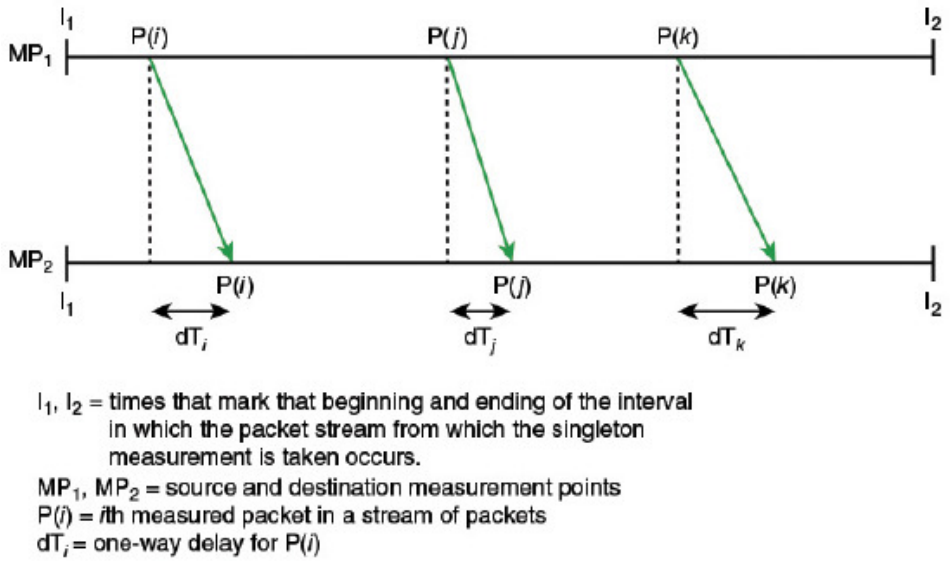


Figure 7.2 Model for Defining Packet Delay Variation

Figure 7.2 illustrates the packet delay variation metric. This metric is used to measure jitter, or variability, in the delay of packets traversing the network. The singleton metric is defined by selecting two packet measurements and measuring the difference in the two delays. The statistical measures make use of the absolute values of the delays.

7.2 OPENFLOW QOS SUPPORT

OpenFlow offers two tools for implementing QoS in data plane switches.

7.2.1 Queue Structures:

An OpenFlow switch provides limited QoS support through a simple queuing mechanism. One or more queues can be associated with a port. Queues support the ability to provide minimum data rate guarantees and maximum data rate limits. Queue configuration takes place outside the OpenFlow protocol, either through a command-line tool or through an external dedicated configuration protocol.

A data structure defines each queue. The data structure includes a unique identifier, port this queue is attached to, minimum data rate guaranteed, and maximum data rate. Counters associated with each queue capture the number of transmitted bytes and packets, number of packets dropped because of overrun, and the elapsed time the queue has been installed in the switch.

The OpenFlow Set-Queue action is used to map a flow entry to an already configured port. Thus, when an arriving packet matches a flow table entry, the packet is directed to a given queue on a given port.

7.2.2 Meters:

A meter is a switch element that can measure and control the rate of packets or bytes. Associated with each meter is a set of one or more bands. If the packet or byte rate exceeds a predefined threshold, the meter triggers the band. The band may drop the packet, in which case it is called a **rate limiter**. Other QoS and policing mechanisms can be designed using meter bands. Each meter is defined by an entry in the meter table for a switch. Each meter has a unique identifier. Meters are not attached to a queue or a port; rather, a meter can be invoked by an instruction from a flow table entry. Multiple flow entries can point to the same meter.

Figure 7.9 shows the structure of a meter table entry and how it is related to a flow table entry.

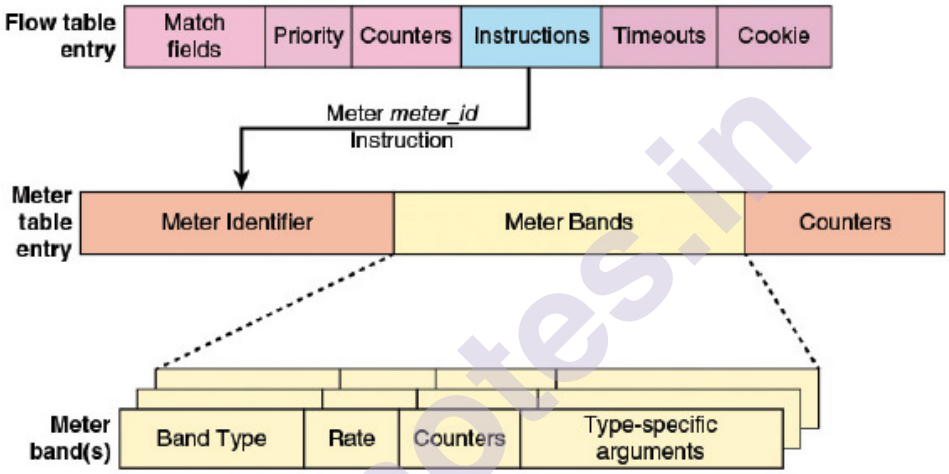


Figure 7.9 OpenFlow QoS-Related Formats

A flow table entry may include a meter instruction with ameter_id as an argument. Any packet that matches that flow entry is directed to the corresponding meter. Within the meter table, each entry consists of three main fields:

- **Meter identifier:** A 32-bit unsigned integer uniquely identifying the meter.
- **Meter bands:** An unordered list of one or more meter bands, where each meter band specifies the rate of the band and the way to process the packet.
- **Counters:** Updated when packets are processed by a meter. These are aggregate counters. That is, the counters count the total traffic of all flows, and do not break the traffic down by flow.

Each band has the following structure:

- **Band type:** drop **or** dscp remark**.**
- **Rate:** Used by the meter to select the meter band, defines the lowest rate at which the band can apply.

- **Counters:** Updated when packets are processed by a meter band.
- **Type specific arguments:** Some band types may have optional arguments. Currently, the only optional argument is for the ds cp remark band type, specifying the amount of drop in precedence.

The meter triggers a meter band if the packet rate or byte rate passing through the meter exceed a predefined threshold. A band of type drop drops packets when the band's rate is exceeded. This can be used to define a rate limiter band.

7.9BQoE: User Quality of Experience

WHY QOE?:

Before the advent of the public Internet, video content delivery was a monopoly of content publishers who delivered their products and services over closed video delivery systems built and managed by cable and satellite TV operators. The operators owned and operated the entire distribution chain as well as the video reception devices (set-top boxes) in the home. These closed networks and devices were under the full control of these operators and were designed, deployed, provisioned, and optimized specifically to deliver high-quality video to consumers.

Figure 7.10 shows an abstraction of the typical satellite TV end-to-end delivery chain. In practice, however, such content delivery and distribution chains are made up of very complex integrations of applications and systems.

As the illustration shows, the traffic scheduling system provides audio and video (A/V) content via the play-out system to be encoded and aggregated into a single MPEG transport stream (TS). Together with the program specific information (PSI), the transport stream is transmitted to the subscriber's set-top box (STB) via a satellite.

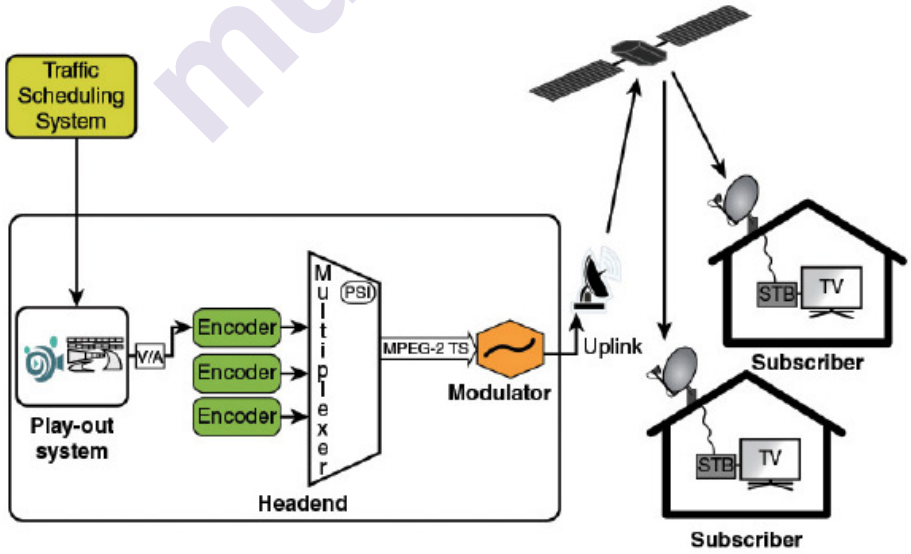


Figure 7.10 An Abstraction of a Content Distribution Network Using a Typical Satellite TV Distribution Network

Online Video Content Delivery:

Video delivery over the Internet takes a different approach. Because numerous subnetworks and devices that constitute the Internet are situated in varied geographical locations, video streams reach the user by traversing through uncharted territories, as illustrated in Figure 7.11. With this arrangement, the guaranteeing of a good network performance is often a very challenging task.

Internet service providers (ISPs) do not own the entire content distribution network, and the risk of quality degradations is high. The access network may consist of coax, copper, fiber, or wireless (fixed and mobile) technology. Issues such as packet delay, jitter, and loss may plague such networks.

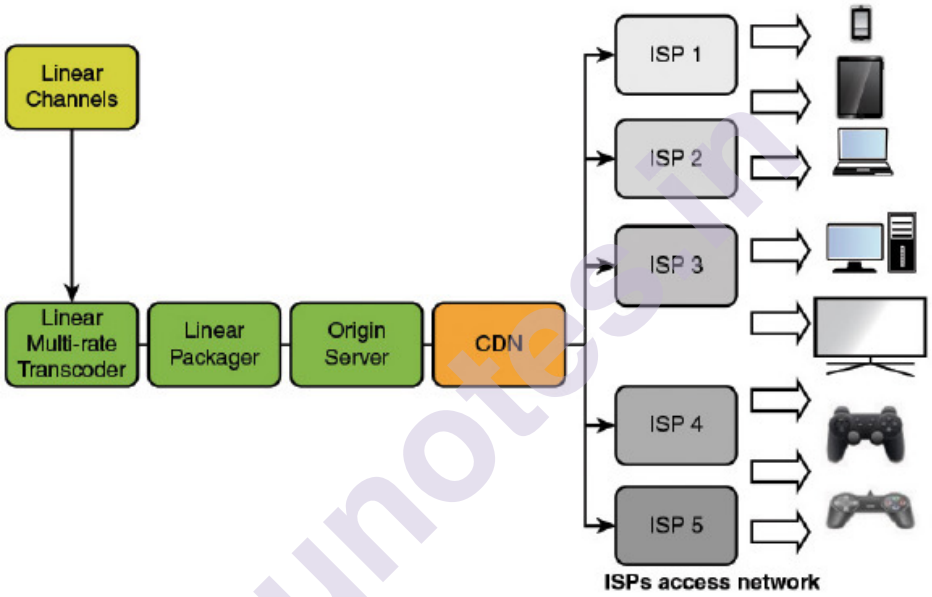


Figure 7.11 An Abstraction of a Content Distribution Network Using the Public Internet Distribution Network

The growth and expansion of the Internet over the past couple of decades has led to an equally huge growth in the availability of network-enabled video streaming services. Giant technological strides have also been made in the development of network access devices.

With the current popularity of these services, providers need to ensure that user experiences are comparable to what the users would consider to be their reference standards. Users' standards are often influenced by the typically high video quality experience with the older technology, that is, those offered by the cable and satellite TV operators. User expectations can also be influenced by capabilities that currently can only be adequately offered by broadcast TV. These capabilities include the following:

Trick mode functionalities, which are features of video streaming systems that mimic visual feedback given during fast-forward and rewind operations.

Contextual experiences across multiple screens, which includes the ability to pause viewing on one screen and switch to another, thus letting users take the video experience with them on the go.

The proliferation of different types of access devices further highlights the importance of QoE frameworks. As an illustration, the QoE for a user watching a news clip on a PDA will most likely differ from another user watching that same news clip on a 3G mobile phone. This is because the two terminals come with different display screens, bandwidth capabilities, frame rates, codecs, and processing power. Therefore, delivering multimedia content or services to these two terminal types, without carefully thinking about the users' quality expectations or requirements for these terminal types, might lead to service overprovisioning and network resource wastage.

Informally, QoE refers to the user perception of a particular service. QoE needs to be one of the central metrics employed during the design and management of networks, content delivery systems, and other engineering processes.

7.10 SERVICE FAILURES DUE TO INADEQUATE QOE CONSIDERATIONS

The stereoscopic 3D TV service is often cited as a prime example of a service that was a spectacular commercial failure because it had very poor QoE ratings.

In 2010, broadcasters such as Disney, Foxtel, BBC, and Sky began actively making 3D content delivery available as a service to their customers as a premium service experience. Indeed, each of these broadcasters rolled out their own dedicated 3D television channels. Within five years, all of them except Sky had to terminate their operations.
A number of factors contributed to the failure of these services.

- The first was the general unavailability of "wow video content" (that is, content that users are most likely to find exciting or take much interest in).
- The second was the need to wear special 3D glasses even when using these services in a home environment.
- Third, because broadcasters were initially in a rush to deploy the 3D TV technology, content was produced by inexperienced creators using inadequate systems and tools. This resulted in a great deal of poorly produced 3D content, which may have alienated the early subscribers.

7.11 QOE-RELATED STANDARDIZATION PROJECTS

Because the field of QoE has been growing rapidly, a number of projects have been initiated to address issues relating to best practices and standards. These projects have been aimed at preventing commercial. Table 7.1 summarizes the prominent ones amongst these project initiatives.

Organization	Mission	QoE-Related effort
QUALINET	A multidisciplinary consortium for QoE research	A common terminology for QoE framework
Eureka Celtic	A collaborative industry-driven European research in the area of telecommunications	Quality of Experience Estimators in Networks (QuEEN) agent to estimate QoE for generic services
International Telecommunication Union—Telecommunication Standardization Sector (ITU-T)	United Nations agency that produces recommendations with a view to standardizing telecommunications on a worldwide basis	QoE standardization IPTV QoE requirements
IEEE Standards Association (IEEE-SA)	A standards-setting body within IEEE, develops consensus standards through an open process that engages industry and brings together a broad stakeholder community	Standard for Network-Adaptive Quality of Experience (QoE)

Table 7.1 QoE Initiatives and Projects

7.12 DEFINITION OF QUALITY OF EXPERIENCE

Definition of Quality:

Quality is the resulting verdict produced by a user after he/she has carried a "comparison and judgment" process on an observable occurrence or event.

This process comprises the following key sequential steps:

- Perception of the event
- Reflection on the perception
- Description of the perception
- Evaluation and description of the result or outcome

Thus, quality is evaluated in terms of the degree to which the user's needs have been fulfilled within the context of the event. The result of this evaluation is usually referred to as the quality score (or rating) if it is presented with reference to a scale.

Definition of Experience:

Experience is an individual's description of a stream of perceptions, and his/her interpretation of one or multiple events. An experience might result from an encounter with a system, service, or an artifact.

Quality Formation Process

As shown in Figure 7.12, there are two distinct subprocess paths to the formation of a quality score: the perception path and the reference path.

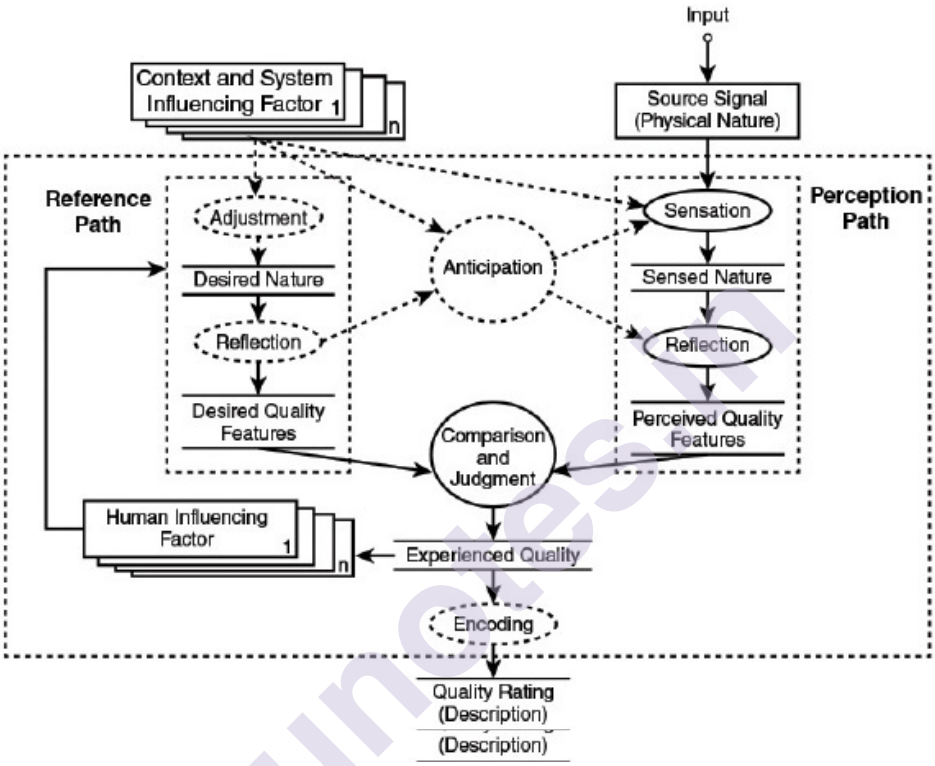


FIGURE 7.12 A Schematic Illustration of the Quality Formation Process from an Individual Point of View.

The reference path reflects the temporal and contextual nature of the quality formation process. This path is influenced by memories of former experienced qualities, as indicated by the arrow from experienced quality to the reference path.

The perception path is characterized by the physical input signal, which is to be assessed, reaching the sensory organs of the observer. This physical event is processed through low-level perceptual processes into a perceived feature within the constraints of the reference path. This perceived feature undergoes a reflection process, which interprets these sensory features through **cognitive** processing.

Definition of Quality of Experience:
Combining the concepts and definitions from the preceding sections, the definition of QoE that reflects broad industry and academic consensus is as follows:

Quality of experience (QoE) is the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility/enjoyment of the application or service in the light of the user's personality and current state.

7.13 QOE STRATEGIES IN PRACTICE

Key findings from QoE-related projects show that for many services, multiple QoS parameters contribute toward the overall user's perception of quality. This has resulted in the emergence of the concept of the QoE/QoS layered approach.

The QoE/QoS Layered Model:

The QoE/QoS layered approach does not ignore the QoS aspect of the network, but instead, user and service level perspectives are complementary, as shown in Figure 7.13.

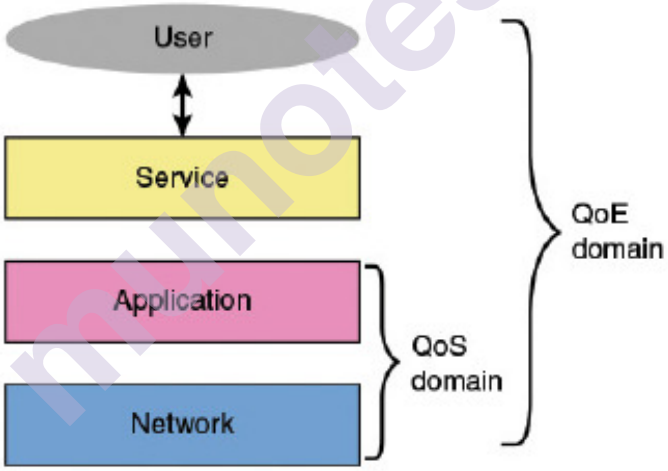


FIGURE 7.13 QoE/QoS Layered Model with the Domains of Interest for the Frameworks

The levels in the layered approach are as follows:

- **User:** The user interacts with the service. It is their degree of delight or annoyance from using the service that is to be measured. Being linked to human perception, QoE is hard to describe in a quantitative way, and it varies from person to person. The complexities of QoE at the user level stem from the differences between individual user characteristics, of which some might be time-varying, whereas others are of a relatively stable nature. The current practice in any QoE measurement is to identify and control for the relatively stable

characteristics of a user in a way that is satisfactory to at least a large proportion of the potential user group.

- **Service:** The service level provides a virtual level where the user's experience of the overall performance of the service can be measured. It is the interface where the user interacts with the service (for example, the visual display to the user). It is also where tolerance thresholds are measured. As an illustration, the QoE measures from the user perspective for streaming applications could be startup time, audio/visual quality, channel change delay, and buffering interruptions.
- **Application-level QoS (AQoS):** AQoS deals with the control of application-specific parameters such as content resolution, bit rate, frame rate, color depth, codec type, layering strategy, and sampling rate. The network capacity often dictates the bandwidth that will be allocated to a service for transmission. Because of this fixed underlying resource, some parameters at the application level are usually adjusted and controlled to achieve a desired quality level.
- **Network-level QoS (NQoS):** This level is concerned with the low-level network parameters such as service coverage, bandwidth, delay, throughput, and packet loss. There are a number of ways in which network-level QoS parameters impact QoE. One such way is via network delay, which impacts QoE especially for interactive services. For instance, the interactive nature of web browsing that requires multiple retrieval events within a certain window of time might be affected by delay variations of the network. Voice over IP (VoIP) services might have stringent response-time demands, whereas e-mail services might tolerate much longer delays.

Although the trade-offs between quality and network capacity may begin with application-level QoS because of network capacity considerations, an understanding of the user requirements at the service level (that is, in terms QoE measures) would enable a better choice of application-level QoS parameters to be mapped onto the network-level QoS parameters.

7.14 FACTORS INFLUENCING QOE

QoE must be studied and addressed by taking into account both technical and nontechnical factors. Many factors contribute to producing a good QoE. Here, the key factors are as follows:

- **User demographics:** The context of demographics herein refers to the relatively stable characteristics of a user that might have an indirect influence on perception, and intimately affects other technical factors to determine QoE. The grouping of users was based on demographic characteristics such as their attitudes toward adoption of new technologies, socio-demographic information, socioeconomic status,

and prior knowledge. Cultural background is another user demographic factor that might also have an influence on perception because of cultural attitude to quality.

- **Type of device:** Different device types possess different characteristics that may impact on QoE. An application designed to run on more than one device type, for example on a connected TV device such as Roku and on an iOS device such as an iPhone, may not deliver the same QoE on every device.
- **Content:** Content types can range from interactive content specifically curated according to personal interests, to content that is produced for linear TV transmission. Studies have suggested that people tend to watch video on-demand (VoD) content with a higher level of engagement than its competing alternative, linear TV. This may be because users will make an active decision to watch specific VoD content, and as a result, give their full attention to it.
- **Connection type:** The type of connection used to access the service influences users' expectations and their QoEs. Users have been found to have lower expectations when using 3G connections in contrast to a wire line connection even when the two connection types were identical in terms of their technical conditions. Users have also been found to lower their expectations considerably, and are more tolerant to visual impairments, on small devices.
- **Media (audio-visual) quality:** This is a significant factor affecting QoE, as it is the part of a service that is most noticeable by the user. The overall audio and video quality appears to be content dependent. For less- complex scenes (for example, head and shoulder content), audio quality is slightly more important than video quality. In contrast, for high-motion content, video quality tends to be significantly more important than audio quality.
- **Network:** Content delivery via the Internet is highly susceptible to the effects of delays, jitter, packet loss, and available bandwidth. Delay variation results in the user experiencing frame freeze and the lack of lip synchronization between what is heard (audio) and what is seen (video). Although video content can be delivered using a number of Internet protocols, not all of them are reliable. However, content delivery is guaranteed using TCP/IP. Nevertheless, bad network conditions degrade QoE because of increased rebuffering and increased interruptions in playback. Rebuffering interruptions in IP video playback is seen to be the worst degradation on user QoE and should be avoided at the cost of startup delay.
- **Usability:** Another QoE factor is the amount of effort that is required to use the service. The service design must render good quality without a great deal of technical input from the user.

- **Cost:** The long-established practice of judging quality by price implies that expectations are price dependent. If the tariff for a certain service quality is high, users may be highly sensitive to any quality degradations.

7.15 MEASUREMENTS OF QOE

QoE measurement techniques evolved through the adaptation and application of psychophysics methods during the early stages of television systems. We will see three QoE measurement methods:

1. Subjective Assessment
2. Objective Assessment
3. End-User Device Analytics

7.15.1 Subjective Assessment:

For subjective assessment of QoE, experiments are carefully designed to a high level of control (such as in a controlled laboratory, field tests, or crowdsourcing environments) so that the validity and reliability of the results can be trusted. It might be useful to consult expert advice during the initial design of the subjective experiment, because the topics of experimental design, experimental execution, and statistical analysis are complex. In general terms, a methodology to obtain subjective QoE data might consist of the following **phase**s:

- **Characterize the service:** The task at this stage is to choose the QoE measures that affect user experience the most. As an example, for a multimedia conferencing service, the quality of the voice takes precedence over the quality of video. Also, the video quality required for such applications does not demand a very high frame rate, provided that audio-to-video synchronization is maintained. Therefore, the resolution of individual frames can be considerably lower than the case of other video streaming services, especially when the size of the screen is small (such as a mobile phone). So, in multimedia conferencing, the QoE measures might be prioritized as voice quality, audio-video synchronization, and image quality.
- **Design and define test matrix:** Once the service has been characterized, the QoS factors that affect the QoE measures can be identified. For instance, the video quality in streaming services might be directly affected by network parameters such as bandwidth, packet loss, and encoding parameters such as frame rate, resolution, and codec. The capability of the rendering device will also play a significant role in terms of screen size and processing power. However, testing such a large combination of parameters may not be feasible.

- **Specify test equipment and materials:** Subjective tests should be designed to specify test equipment that will allow the test matrix to be enforced in a controlled fashion. For instance, to assess the correlation between NQoS parameters and the perceived QoE in a streaming application, at least a client device and a streaming server separated by an emulated network are needed.
- **Identify sample population:** A representative sample population is identified, possibly covering different classes of users categorized by the user demographics that are of interest to the experimenter. Depending on the target environment for the subjective test, at least 24 test subjects has been suggested as the ideal number for a controlled environment (for example, a laboratory) and at least 35 test subjects for a public environment. Fewer subjects may be used for pilot studies to indicate trending. The use of crowdsourcing in the context of subjective assessment is still nascent, but it has the potential to further increase the size of the sample population and could reduce the completion time of the subjective test.
- **Subjective methods:** Several subjective assessment methodologies exist within the industry recommendations. However, in most of them, the typical recommendation is for each test subject to be presented with the test conditions under scrutiny along with a set of rating scales that allows the correlation of the users' responses with the actual QoS test conditions being tested. There are several rating scales, depending on the design of the experiment.
- **Analysis of results:** When the test subjects have rated all QoS test conditions, a post-screening process might be applied to the data to remove any erroneous data from a test subject that appears to have voted randomly. Depending on the design of the experiment, a variety of statistical approaches could be used to analyze results. The simplest and the most common quantification method is the mean opinion score (MOS), which is the average of the opinions collected for a particular QoS test condition. The results from subjective assessment experiments are used to quantify QoE, and to model the impacts of QoS factors. However, they are time-consuming, expensive to carry out, and are not feasible for real-time in-service monitoring.

7.15.2 Objective Assessment:

For objective assessment of QoE, computational algorithms provide estimates of audio, video, and audiovisual quality as perceived by the user. Each objective model targets a specific service type. The goal of any objective model is to find the optimum fit that strongly correlates with data obtained from subjective experiments. A methodology to obtain objective QoE data might consist of the following phases:

- **Database of subjective data**: A starting point might be the collection of a group of subjective datasets as this could serve as benchmark for training and verifying the performance of the objective model. A typical example of one of these datasets might be the subjective QoE data generated from well-established subjective testing procedures.
- **Preparation of objective data**: The data preparation for the objective model might typically include a combination of the same QoS test conditions as found in the subjective datasets, as well as other complex QoS conditions. A variety of preprocessing procedures might be applied to the video data prior to training, and refinement of the algorithm.
- **Objective methods**: There are various algorithms in existence that can provide estimates of audio, video, and audiovisual quality as perceived by the user. Some algorithms are specific to a perceived quality artifact, while others can provide estimates for a wider scope of quality artifacts. Examples of the perceived artifacts might include blurring, blockiness, unnatural motion, pausing, skipping, rebuffering, and imperfect error concealment after transmission errors.
- **Verification of results**: After the objective algorithm has processed all QoS test conditions, the predicted values might benefit from a post-screening process to remove any outliers; this is the same concept applied to the subjective datasets. The predicted values from the objective algorithm might be in a different scale as compared to the subjective QoE datasets.
- **Validation of objective model**: The objective data analysis might be evaluated with respect to its prediction accuracy, consistency, and linearity by using a different subjective dataset. It is worth noting that the performance of the model might depend on the training datasets and the verification procedures. The Video Quality Experts Group (VQEG) validates the performance of objective perceptual models.

7.15.3 End-User Device Analytics:

End-user device analytics is yet another alternative method of QoE measurement. Real-time data such as the connection time, bytes sent, and average playback rate are collected by the video player application for each video viewing session and fed back to a server module where the data is pre-aggregated and then turned into actionable QoE measures. Some of the metrics reported for per-user and aggregate viewing sessions include startup delay, rebuffering delays, average bit rates, and the frequency of bit rate switches.

Operators may be inclined to associate viewer engagement levels with their QoE because good QoEs usually make viewers less likely to abandon a viewing session. The definition of viewer engagement may have different meanings for different operators and context. First of all,

operators might like to know which viewer engagement metrics affect QoE the most to guide the design of the delivery infrastructures. Second, they might also like to quickly identify and resolve service outages, and other quality issues. A minute of encoder glitch could replicate throughout the ISPs, and the various delivery infrastructures, and affect all their customers. Operators might like to know the scale of this impact, and how it affects users' engagement.

Finally, they would like to understand their customers' demographics (connection methods, type of device, bit rates of the consumed asset) within a demographic region so that resources can be strategically dimensioned.

7.16 APPLICATIONS OF QOE

The practical applications of QoE can be grouped into two areas based on the main usage.

1. **Service QoE monitoring:** Service monitoring allows the support teams (for example, service provider and network operator) to continually monitor the quality experienced by the end users of the service. A service alert message might be sent to the support teams when QoE falls below a certain threshold value, as this will allow the support teams to quickly identify and resolve service outages and other QoE issues.
2. **QoE-centric network management:** The ability to control and optimize the user experience when QoE degradation issues arise is the holy grail of QoE network management. Given the multidimensional aspect of the overall QoE (such as the network-level conditions of the subnetworks, application-level QoS, device capability, and user demographics), a typical challenge lies in providing actionable QoE information feedback to the network or service provider.

Two approaches in which QoE-centric network management can be exploited are as follows:

- In the first approach, a set of QoS measurement values together with the appropriate assumptions, are used in computing the expected QoE for a user.
- In the second approach, which is somewhat the opposite of the first, a target QoE for a user together with the appropriate assumptions is used to produce estimates of the required QoS values.

The first approach can be taken by a service provider, who can provide a range of QoS offerings with an outline of the QoE that the customer might reasonably expect.

The second approach can be taken by a customer who defines the required QoE, and then determines what level of service will meet that need.
Figure 7.14 illustrates a scenario where the user can make a selection from a range of services, including the required level of service (SLA). By contrast to the purely QoS-based management, the SLA here is not expressed in terms of raw network parameters. Instead, the user indicates a QoE target; it is the service provider that maps this QoE target together with the type of service selected, onto QoS demands.

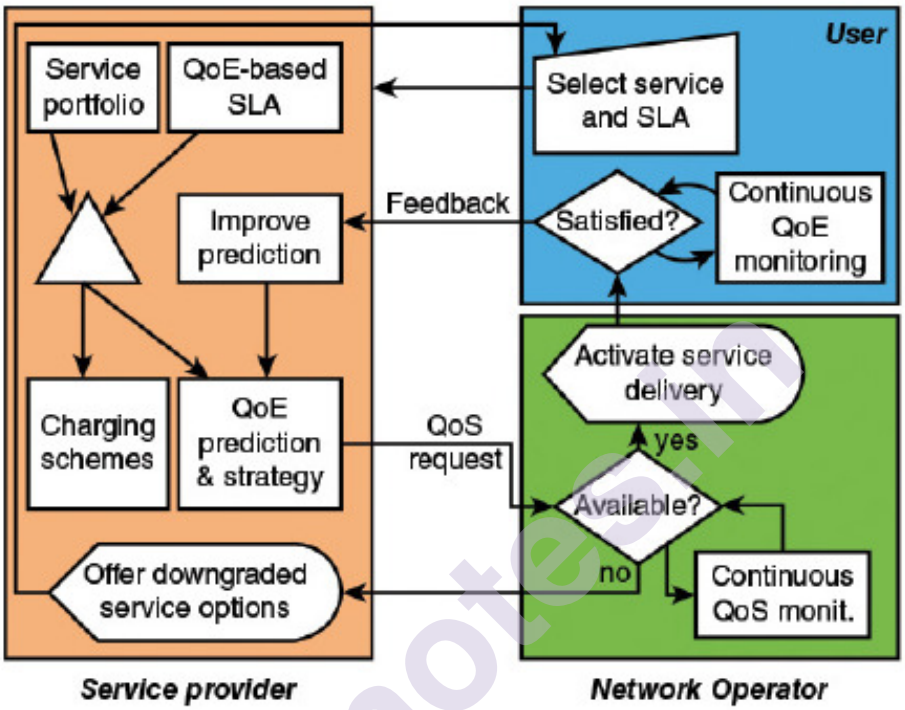


FIGURE 11.5 QoE-Centric Network Management

Figure 7.14 QoE-Centric Network Management

The service provider selects the appropriate quality prediction model and management strategy (for example, minimize network resource consumption) and forwards a QoS request to the operator. It is possible that the network cannot sustain the required level of QoS, making it impossible to deliver the requested QoE. This situation leads to a signal back to the user, prompting a reduced set of services/QoE values.

8

NETWORK DESIGN IMPLICATIONS OF QOS AND QOE

Unit Structure

8.0 OBJECTIVES

- Translate metrics from QoS to QoE domain.
- Select the appropriate QoE/QoS mapping model for a given operational situation.
- Deploy QoE-centric monitoring solutions over a given infrastructure.
- Deploy QoE-aware applications over QoE-centric infrastructure.

8.1 INTRODUCTION

QoE/QoS mapping model is a function that transforms metrics from QoS to QoE domains.

8.2 CLASSIFICATION OF QOE/QOS MAPPING MODELS

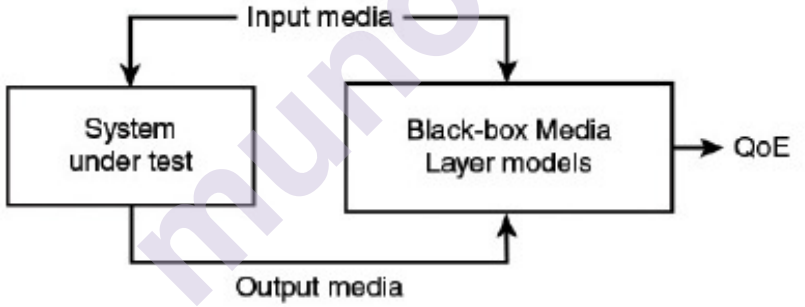
QoE/QoS mapping models can be classified according to their inputs into three categories:

1. Black-box media-based models
2. Glass-box parameter-based models
3. Gray-box parameter-based models

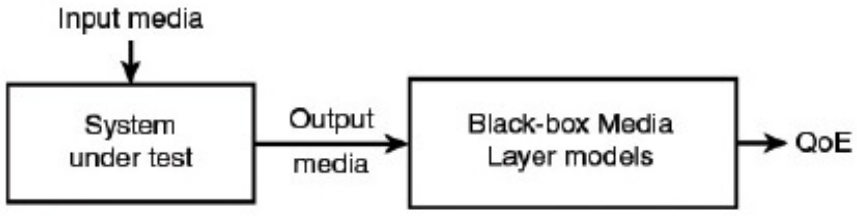
8.2.1 Black-Box Media-Based QoS/QoE Mapping Models:

Black-box media-based quality models rely on the analysis of media gathered at system entrance and exit. Hence, they account implicitly for the characteristics of examined media processing system. They are classified into two categories:

a-Double-sided or full-reference quality models: They use as inputs the clean stimulus and the corresponding degraded stimulus. They compare the clean and degraded stimulus in a *perceptual domain* that accounts for psychophysics capability of human sensory system. The perceptual domain is a transformation of traditional physical temporal and frequency domains performed according to characteristics of users perceptions. Basically, the larger the perceptual distance, the greater the degradation level. This model needs to align clean and degraded stimulus because the comparison is made on per-block basis. The stimulus alignment should be realized autonomously, that is, without adding extra control information describing stimulus structure.



(a): Double-sided or full-reference quality models.



(b): One-sided or no-reference mapping models.

Figure 8.1 Black-Box Media-Based QoS/QoE Mapping Models

B One-sided or no-reference quality models: They rely solely on the degraded stimulus to estimate the final QoE values. They parse the degraded stimulus to extract the observed distortions, which are dependent on the media type, for example, audio, image and video. As an example, artifacts extracted from audio stimulus include whistle, circuit noises, echoes, level saturation, clapping, interruptions, and pauses. The gathered distortions are adequately combined and transformed to compute the QoE values.

The main advantage of black-box quality models resides in their ability to measure QoE values using information gathered at the periphery of a given media processing system. Hence, they may be used in a generic fashion over different infrastructures and technologies. Moreover, it enables enhancing unconditionally quality models, that is, independently of technical and ethical constraint related to the measurement processes. Furthermore, black-box quality models may easily operate on either per-user or per-content basis.

The main shortcoming of black-box quality models resides in the requirements to access the final representation of stimulus, which is often inaccessible in practice for privacy reasons. Moreover, full-reference quality models use clean stimulus as inputs that is often unavailable or hardly accessible at the system output.

The full-reference black-box quality models are widely used for onsite benchmarking, diagnosis, and tuning of network equipment's, where clean stimulus is available. The black-box quality models are used offline for the evaluation of application-layer components, such as codec, packet loss concealment (PLC), and buffering schemes.

8.2.2 Glass-Box Parameter-Based QoS/QoE Mapping Models:

The glass-box parameter-based quality models quantify the QoE of a given service through the full characterization of the underlying transport network and edge devices. The set of considered characterization parameters and their combination rules are derived based on extensive subjective experiments and thorough statistical analysis. The glass-box parameter-based models may operate off line or on line according to the availability of characterization parameters at a given measurement instant. The characterization parameters include noise, packet loss, coding scheme, one-way delay, and delay jitter. The glass-box parameter-based models are generally less accurate and coarser than black-box media-based ones.

A well-known offline glass-box parameter-based model, named E-Model, has been defined by the ITU-T in Rec. G.101. The offline glass-box parameter-based quality models are suitable for planning purposes. They

enable a general overview pf QoE values of a voice transmission system at an early phase. However, for service monitoring and management, online models are needed. In such a case, the variable model parameters should be acquired at run time. This is especially suitable for IP-based services where control data, such as sequence number and time stamp, are included in each packet header. In such an environment, it is possible to extract static characterization parameters from signaling messages and variable ones from the received packets captured at the destination port. This means that parameters are acquired without acceding to the media content, which is preferable for privacy reasons.

8.2.3 Gray-Box QoS/QoE Mapping Models:

The gray-box quality models combine advantages of black- and glass-box mapping models. They sample basic characterization parameters at system output in addition to some control data describing the structure of clean stimulus. The control data may be sent in separate control packets or piggybacked inside transmitted media packets. Hence, perceptually important information about a given content can be considered by the quality models. Therefore, they can measure QoE value on per-content basis. Given its simplicity to deploy and its reasonable accuracy, this class of QoS/QoE mapping models is quickly proliferating.

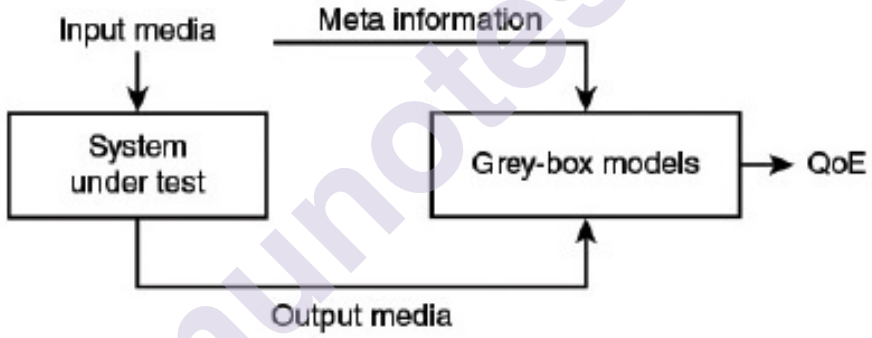


Figure 8.2 Gray-Box QoS/QoE Mapping Models

Tips for QoS/QoE Mapping Model Selection:

The following checklist of five items can aid in the selection of a QoS/QoE mapping model:

- Which types of operations am I considering?
- Which parameters do I have? Can I access the signals, the contents, the packet payload or the header?
- Do I expect specifications and usage conditions to use a given mapping model?
- How much precision do I need?
- Do I have all inputs available for selected mapping models?

8.3 IP-ORIENTED PARAMETER-BASED QOS/QOE MAPPING MODELS

The area of measuring QoE of IP networks and applications is still in its infancy. However, the popularity of multimedia and user-friendly IP-based services puts QoE at the center of interest of today's ecosystem. In contrast to legacy content-oriented telecom systems (for example, public switched telephone network, radio, and TV), IP networks carry clean media content from a server to a destination using a flow of media packets composed of a header and a payload. Therefore, parameters gathered at network layer, in addition to application layer, are easily accessible at run time on user devices. This enables measuring QoE at run time using online glass- or gray-box parameter-based quality models. The QoE over IP-based networks are time- varying in contrast to telecom networks, which are roughly time-invariant. This characteristic leads to considering instantaneous and overall QoE.

8.3.1 Network Layer QoE/QoS Mapping Models for Video Services:

The network layer QoS/QoE mapping models rely solely on NQoS metrics gathered from the TCP/IP stack except for the application layer (that is, transport, network, link, and physical layers). Ketyko et al. proposed the following parameter-based quality model for estimating video streaming quality in 3G environment:

$$\overline{QoE} = 8.49 - 0.02 \cdot AL - 0.01 \cdot VL - 1.12 \cdot AJ + 0.04 \cdot RSSI \qquad \textbf{(Eq. 1)}$$

where AL and VL refer respectively to audio and video packet loss rates, AJ and VJ represent respectively audio and video packet jitter (VJ), and RSSI is the received signal strength indicator.

Kim and Choi presented a two-stage QoE/QoS mapping model for IPTV over 3G networks. The first stage consists of combining a set of basic QoS parameters into one metric as follows:

$$QoS(L, U, J, D, B) = K\{W_L \cdot L + W_U \cdot U + W_J \cdot J + W_d \cdot D + W_b \cdot B\} \qquad \textbf{(Eq. 2)}$$

where L, U, J, D, and B refer, respectively, to packet loss, burst level, packet jitter, packet delay, and bandwidth. The constants K, W_I, W_u, W_J, W_d and W_b are predefined weighting coefficients, which depend on the type of the access network (that is, wired or wireless).

The second stage consists of computing QoE value as following:

$$QoE(QoS(X)) = Q_r\,(1 - QoS(X))^{QoS(X) \times A/R} \qquad \textbf{(Eq. 3)}$$

where, X is a vector of parameters {L, U, J, D, B} and Q_r is a scalar limiting the range of the IPTV QoE obtained as a function of the display size/resolution of the screen. The constant A expresses the subscribed service class and R is a constant reflecting the structure of the video frames.

8.3.2 Application Layer QoE/QoS Mapping Models for Video Services:

Besides NQoS parameters, application layer QoE/QoS mapping models use metrics gathered at application layers (AQoS). Moreover, they can account for the user behavior while interacting with a given video content Ma et al., the following parameter-based quality model is presented for video streaming application:

$$QoE = 4023 - 0.0672\,L_x - 0.742\,(N_{QS} + N_{RE}) - 0.106\,T_{mr} \quad \textbf{(Eq. 4)}$$

where Lx refers to the start-up latency, that is, the waiting time before playing a video sequence, N_{QS} is the number of quality switches that count the number of times the video bit rate is changed during a session, N_{RE} is the number of rebuffering events, and T_{MR} is the mean rebuffering time. Khan et al., estimate QoE of a generic streamed content video over wireless networks using MPEG4 codec:

$$QoE(FR, SBR, PER = \frac{a_1 + a_2 \mathrm{FR} + a_3.In(SBR)}{1 + a_4.per + a_5.(PER)^2} \quad \textbf{(Eq. 5)}$$

Where FR, SBR and PER refer, respectively, to the frame rate sampled at the application level, sent bit rate and packet error rate sampled at the network level. The co-efficient a_1 to a_5 are used to calibrate the quality model. This model has been updated to account for three types of video content: slight movement, gentlewalking, and rapid movement. The quality model is given by the following :

$$\text{QOE (FR,SBR,BLER,CT)} = a + \frac{b.e^{FR} + c.In\,(SBR) + CT.(D + e.In(SBR)}{1 + f.(BLER) + G.(BLER)^2} \quad \textbf{EQ. 6}$$

where a, b, c, d, e, f, g represents constants; CT is the content type of the video; and SBR and BLER refer respectively to the sent bit rate and the bit loss error rate.

A QoE /QoS mapping model for IPTV was developed by Kuipers et al, which accounts for the startup latency and zapping time. The quality model in given by the following

$$QoE = 3.5\, e^{-(0.15+0.19}N + 15 \quad \text{……. EQ 2.1}$$

where QoE is a one-dimension QoE component considering zapping behaviour, ZT is the zapping time expressed in seconds, and a and b are numeric constants that might be positive or negative.

8.4 ACTIONABLE QOE OVER IP-BASED NETWORKS

Actionable QoE refers to all techniques and mechanisms enabling to concretely measure and utilize QoE metrics. An actionable QoE solution strongly depends on the underlying system and services characteristics. Moreover, actionable QoE solution works over multiplane architectures that integrate data, control, and management planes. Basically, two solutions may be used to achieve actionable QoE:

1. System-oriented actionable QoE solution
2. Service-oriented actionable QoE solution

8.4.1 The System-Oriented Actionable QoE Solution:

The system-oriented actionable QoE solutions account for QoE measures within the delivery infrastructure. In such a condition, services are engineered while assuming that underlying system is perfect; that is, no degradations are inserted. Figure 2.3illustrates a nominal environment where system-oriented actionable QoE solution may be provided. As can be seen, actionable QoE solution requires

- A QoS measurement module that gathers basic **key performance indicators** (KPIs) from the underlying system,
- A QoE/QoS mapping model, and
- A resource management module of controlled devices.

Each service provider specifies a target QoE level that should be offered for its customers. The QoE/QoS mapping model should be selected in a way that guarantees

(a) the availability of quality model input parameters and

(b) conformity with service specifications and conditions.

A signaling procedure may be executed to do that. The management procedure may be executed either before starting a service or during its delivery. This should be realized using an autonomous decision system, including a policy that maps observed QoE measures to a course of actions executed by managed devices.

This operational mode applies well for software-defined networking (SDN) where the network paths are managed by an SDN controller. In such a case, the measured QoE values are reported to the SDN controller, which uses them to define the behavior of SDN switches.

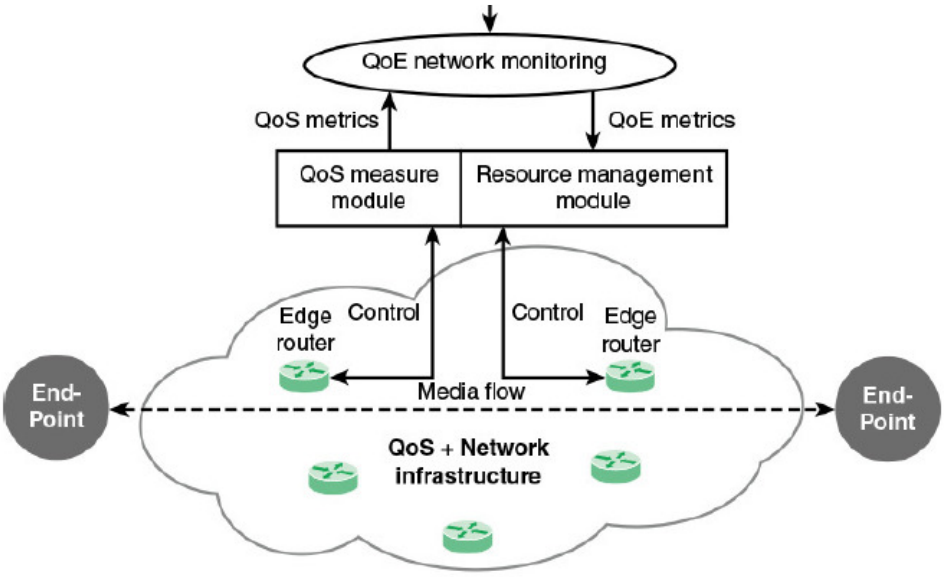


Figure 8.3 A Nominal Environment for Providing QoE-Centric Services

8.4.2 The Service-Oriented Actionable QoE Solution:

The service-oriented actionable QoE solutions account for QoE values measured at endpoints and service level. In such a situation, services are engineered to deal with the underlying system flaws to reach a specified QoE level. The services may change their behavior as a function of the current context and condition. The measurement module of KPI is installed on endpoints. The QoE/QoS mapping models may be deployed either on endpoints or specialized devices. The measured QoE values are sent to endpoints to configure different application modules at sender, proxy, and receiver entities.

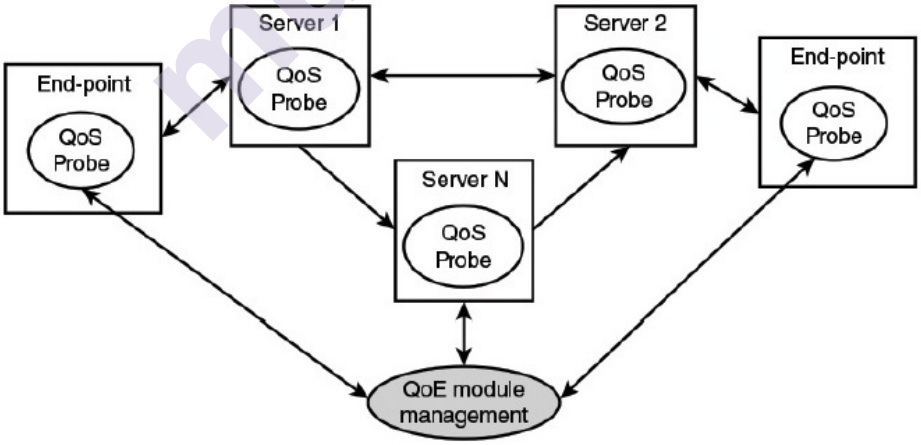


Figure 8.4 Service-Aware QoE Development Scheme

The service-oriented actionable QoE solution involves multiple advantages.

- Per-service, per-user, and per-content QoE monitoring and management solutions are performed to provide a given QoE level.
- It provides more adaptation possibility because it precisely discerns capability and the role of each service component.
- It reduces the communication overhead and balances computing loads.
- It enables component-level granularity treatment of QoE in addition to stream- and packet-level granularities.

8.5 QoE VERSUS QoS SERVICE MONITORING

8.5.1 MONITORING and its Classification:

Monitoring is a strategic function that should be supported by today's IT system. It returns indicators and provides clues regarding the system performance and its workload. Moreover, it enables detecting system dysfunction and defects as well as underperforming devices and applications so that the best course of actions may be undertaken. The monitoring solutions of current IT systems may be classified into the following four categories:

- **Network monitoring:** Provides measures about performance of paths and links used to deliver media units. They are collected at packet processing devices (router and switch) and may operate on per-flow or per- packet bases. The path characterization metrics, such as throughput, packet loss, reorder and duplication, delay, and jitter, are calculated using atomic metrics extracted from packet header, such as sequence number and time stamps.
- **Infrastructure monitoring:** Provides measures about devices performance and resources state, such as memory, CPU, IO, load, and so on.
- **Platform monitoring:** Provides performance indicators about the computing center where back-end servers are running. They may work over a virtualized infrastructure where business application logics are deployed using virtual machines.
- **Service monitoring:** Provides measures about services performance. The metrics are dependent on each application, and may be realized from technical or perceptual perspectives.

Typically, the monitoring solution in a distributed system involves a variety of **probes** that measure the performance of a given element participating in the service delivery chain. It includes a reliable and scalable manager that remotely configures probe behavior, especially in terms of the frequency and content of measurement reports.

The monitoring solution should provide communication facility between the manager and the managed devices. The interaction between the manager and managed entities is conventionally realized using the connectionless User Datagram Protocol (UDP) through a couple of reserved ports.

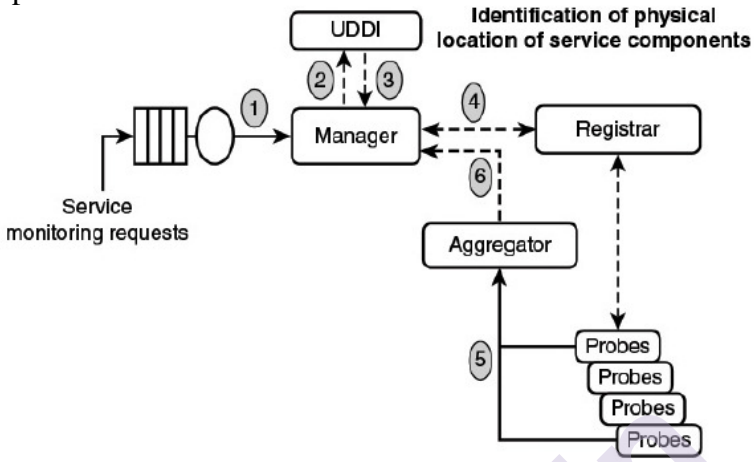


FIGURE 12.6 A Baseline and Generic Monitoring Solution

Figure 8.5 A Baseline and Generic Monitoring Solution

Figure 8.5 presents a typical configuration of an on-demand monitoring solution. The manager receives monitoring requests from customers expressed using specific syntax. Upon the receipt of a new monitoring request, the manager inquires with a Universal Description, Discovery, and Integration (UDDI) directory to get more information about the monitored service, such as location and properties. The monitoring solution, including a set of probes, is deployed over a given infrastructure. They register themselves automatically once a monitored component is activated in a preconfigured registrar. The registrar keeps traces and features of all active probes. They should be configured off line by the administrator to report metrics according to a given behavior during a service. The metrics generated by the probes may be aggregated and processed before sending them to the manager that performs data analytic procedure.

8.5.2 QoS Monitoring Solutions:

The emerging QoS monitoring solutions are basically developed for data centers and clouds where virtualization technology is supported. Figure 8.6 shows a network- and infrastructure-level monitoring solution built for cloud-based IPTV service. The audiovisual content servers are placed on a cloud. The traffic sent from the content servers to IPTV devices is permanently monitored through a set of Vprobes deployed across the network. A Vprobe is an open-ended investigatory tool that is used in the cloud environment to inspect, record, and compute the state of the hypervisor as well as each virtual machine running service business logics.

The flows of video packets are parsed at different measurement points. The information collected by Vprobes is used next to reconstruct service-level detailed records (SDRs). Each record contains the most relevant information of the complete session between an origin (server) and a destination (user). The critical parameters of the messages associated with an IPTV session are stored inside the SDRs.

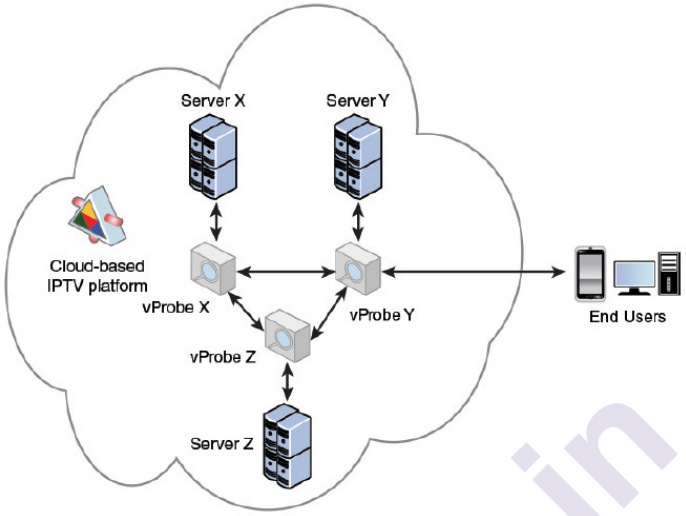


Figure 8.6 vProbes Approach in Cloud-Based IPTV Network

8.5.3 QoE Monitoring Solutions:

The QoE monitoring solution strongly depends on QoE/QoS mapping models. The diagrams in Figure 2.1 show four configurations that can be used to monitor at run time the QoE values of IP-based video streaming services. The configurations differ in term of the measurement and mapping model locations. Each configuration is denoted using XY expression, where X refers to the measurement location and Y refers to the quality model location. They may take one of these values: N for network, C for client, and B for both.

A. Static operation mode (NN): Both the measurements of KPIs and QoE are performed inside the network. The QoE/QoS mapping model is installed on a device listening to the service delivery path. The quality model uses collected KPIs, prior knowledge about video coding schemes, and endpoint characteristics. The characteristics of endpoints can be acquired either by polling them or by inspecting exchanged SDP (Session Description Protocol) messages. The QoE measurement points may include an endpoint emulator enabling a realistic reconstruction of received streams.

B. Nonembedded dynamic operation (BN): The measurement of KPIs is performed at both the network and the client, whereas QoE values are measured inside the network. The quality model uses gathered KPIs, prior knowledge about coding schemes, and information about the client obtained using a customized signaling protocols.

C. Nonembedded distributed operation (CN): The measurement of KPIs is performed at the client side, and these are sent periodically to the QoE/QoS mapping model located inside the network.

D. Operation embedded (CC): The measurement of KPIs and QoE is performed at the client side. The QoS/QoE mapping model is embedded inside the client. The measured QoE metrics may be reported to a centralized monitoring entity.

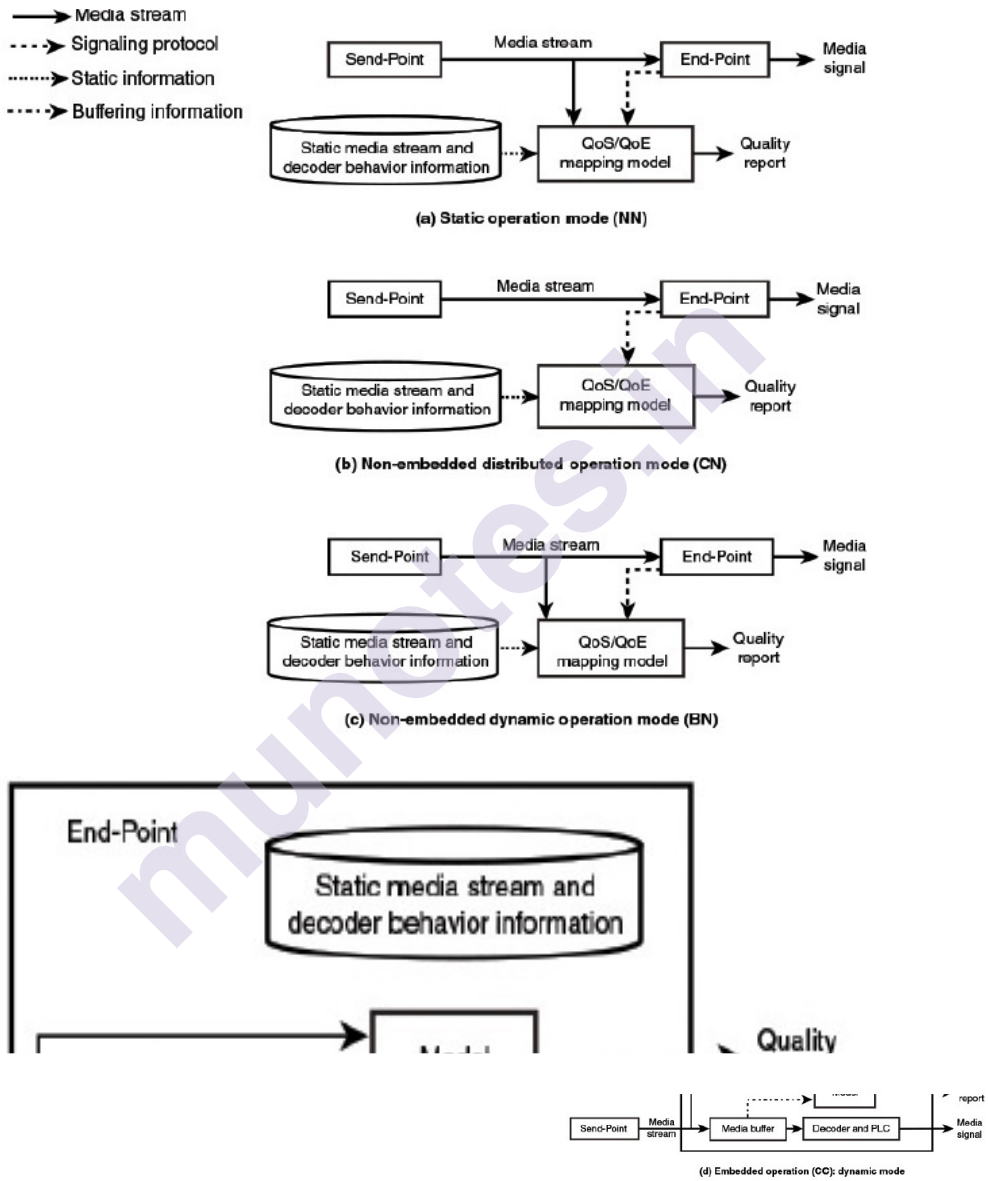


Figure 8.7 The Operational Working Modes of Quality Models in Networks

The architecture of QoE-agent is based on a layered definition of APIs that enable convenient grouping of different factors that influence QoE. The six layers are defined as follows:

- **Resource:** Composed of dimensions representing the characteristics and performance of the technical system(s) and network resources used to deliver the service. Examples of such factors include network QoS in terms of delay, jitter, loss, error rate, and throughput. Furthermore, system resources such as server processing capabilities and end user device capabilities are included.
- **Application:** Composed of dimensions representing application/service configuration factors. Examples of such factors include media encoding, resolution, sample rate, frame rate, buffer sizes, SNR, etc.
- **Interface:** Represents the physical equipment and interface through which the user is interacting with the application (type of device, screen size, mouse, etc.).
- **Context:** Related to the physical context (e. g. geographical aspects, ambient light and noise, time of the day), the usage context (e.g. mobility/no-mobility or stress/no-stress), and the economic context (e.g. the cost that a user is paying for a service).
- **Human:** Represents all factors related to the perceptual characteristics of users (e.g. sensitivity to audiovisual stimulus, perception of durations, etc.).
- **User:** Users' factors that are not represented in the Human layer. These factors encompass all aspects of humans as users of services or applications (e.g., history and social characteristics, motivation, expectation, and level of expertise).

8.6 QOE-BASED NETWORK AND SERVICE MANAGEMENT

The quantified QoE values may be considered in networks and services management. This enables getting an optimal trade-off that maximizes QoE and minimizes consumption of resources. The major challenge resides in the translation of QoE metrics into a course of actions that definitely enhance encountered QoE and reduce resources consumptions.

8.6.1 QoE-Based Management of VoIP Calls:

The management of Voice over IP (VoIP) based on QoE has been extensively investigated in the literature. The goal is to maintain a constant QoE level during a whole packet voice session transmitted over time-varying quality IP networks. Typically, QoE measurement probes following one glass-box parameter-based model are installed on VoIP endpoints. They collect at run time atomic KPIs, which are transformed and given as inputs to a QoE/QoS mapping model. After a new measure of QoE values is received, a QoS controller adjusts the reconfigurable

network parameters within a delivery path, such as queuing allocation and congestion thresholds.

8.6.2 QoE-Based Host-Centric Vertical Handover:

Mobile consumers over next-generation networks could be served at one moment by several overlapping heterogeneous wireless networks. The network selection/switching procedure can be performed either at the start or during the service. An internetwork hard handover occurs when users switch from one network to another because of specific reasons related to both consumers and providers. A handover could be managed in network- or host-centric way. In a host-centric approach, however, end nodes can perform a handover when quality of service becomes unsteady and unsatisfactory.

Figure 8.2 illustrates a likely envisaged scenario where the client could be served either by WiMAX or Wi-Fi systems. Appropriate equipment should be deployed and configured, such as outdoor and indoor units, server, router, and Wi-Fi and WiMAX access points to enable network handover. Throughout a vocal call, the client may switch from WiMAX system to Wi-Fi system, and vice versa.

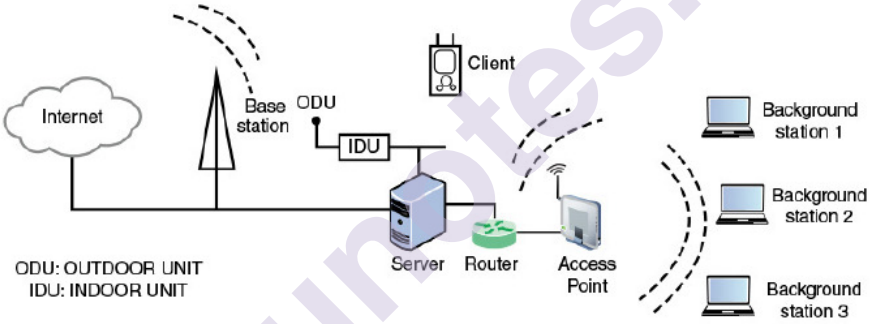


Figure 8.8 Network Selection Wi-Fi and WiMAX Based on Client and Link Quality

8.6.3 QoE-Based Network-Centric Vertical Handover:

The goal is to perform a handover between overlapping WLAN and GSM networks. This allows, on one hand, relatively exploiting the high capacity of a WLAN, and on the other hand, reducing the GSM network load and cost. Figure 8.9 shows a scenario where a mobile subscriber initiates a voice call to a landline PSTN subscriber using a WLAN as a last-wireless hop. Next, when the QoE of the voice call goes below a given critical threshold because of mobility or congestion, a handover is performed. In such a case, the mobile subscriber is linked to the landline subscriber using the GSM infrastructure. The hands-free terminal is equipped with two wireless card interfaces to allow connection to WLAN and GSM networks. The mobile terminal sends adequate "quality reports" to a PBX that analyzes received feedbacks. Once an unsatisfied score is detected, the PBX instructs the mobile terminal to perform a handover. To do that

seamlessly, a voice channel is opened using GSM infrastructure between the mobile terminal and PBX, which is responsible to relay received voice information toward the fixed subscriber.

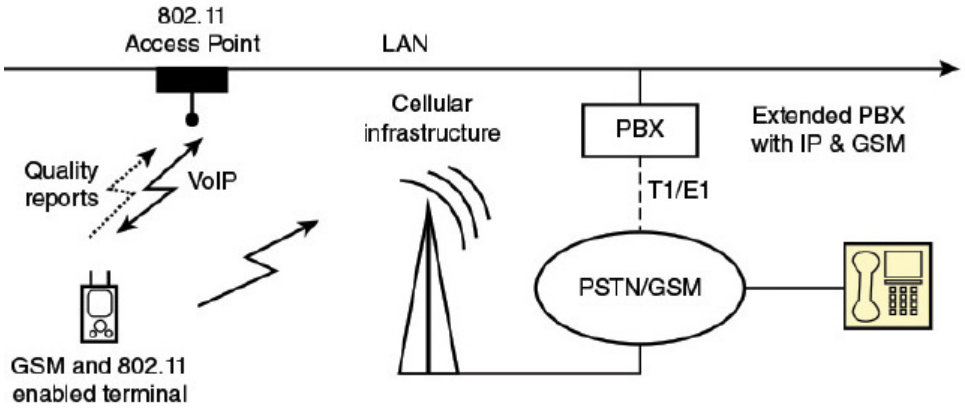


Figure 8.9 Handover Scenario Between WLAN and GSM Networks

UNIT V

9

MODERN NETWORK ARCHITECTURE: CLOUDS AND FOGCLOUD COMPUTING

Unit Structure

9.0 OBJECTIVES

The chapter begins with a definition of basic concepts, and then covers cloud services, deployment models, and architecture. The chapter then discusses the relationship between cloud computing and software-defined networking (SDN) and network functions virtualization (NFV).

After studying this chapter, you should be able to:

- Present an overview of cloud computing concepts.
- List a4nd define the principal cloud services.
- List and define the cloud deployment models.
- Compare and contrast the NIST and ITU-T cloud computing reference architectures.

- Discuss the relevance of SDN and NFV to cloud computing.

9.1 BASIC CONCEPTS

There is an increasingly prominent trend in many organizations to move a substantial portion or even all information technology (IT) operations to an Internet-connected infrastructure known as enterprise cloud computing. At the same time, individual users of PCs and mobile devices are relying more and more on cloud computing services to backup data, synch devices, and share. NIST defines cloud computing, in NIST SP-800-145, *The NIST Definition of Cloud Computing*, as follows:

Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (for example, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Basically, cloud computing provides economies of scale, professional network management, and professional security management. These features can be attractive to companies large and small, government agencies, and individual PC and mobile users. The individual or company only needs to pay for the storage capacity and services they use. The user, be it company or individual, does not have the hassle of setting up a database system, acquiring the hardware they need, doing maintenance, and backing up the data—all these are part of the cloud service.

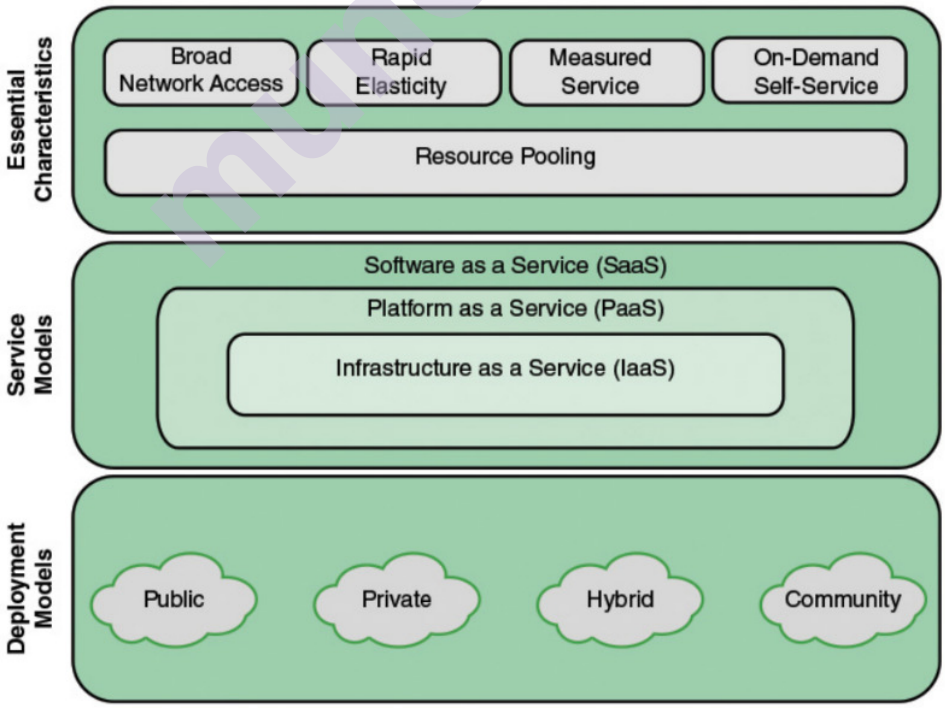


Figure 9.1: Cloud Computing Elements

Cloud networking:

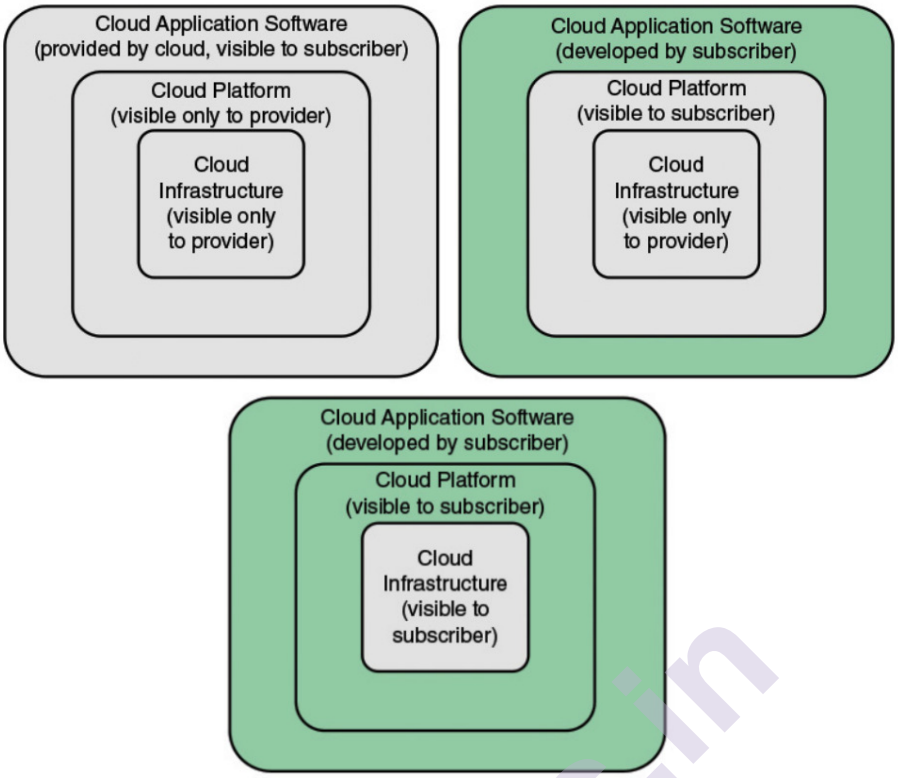
refers to the networks and network management functionality that must be in place to enable cloud computing. Most cloud computing solutions rely on the Internet, but that is only a piece of the networking infrastructure. One example of cloud networking is the provisioning of high performance / high reliability networking between the provider and subscriber. In this case, some or all of the traffic between an enterprise and the cloud bypasses the Internet and uses dedicated private network facilities owned or leased by the cloud service provider. More generally, cloud networking refers to the collection of network capabilities required to access a cloud, including making use of specialized services over the Internet, linking enterprise data centers to a cloud, and using firewalls and other network security devices at critical points to enforce access security policies. We can think of **cloud storage** as a subset of cloud computing. In essence, cloud storage consists of database storage and database applications hosted remotely on cloud servers. Cloud storage enables small businesses and individual users to take advantage of data storage that scales with their needs and to take advantage of a variety of database applications without having to buy, maintain, and manage the storage assets.

9.2 CLOUD SERVICES

As defined by NIST, there are three cloud service models:

- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)

These can be viewed as nested service alternatives (see Figure 9.2).



(a) SaaS (b) PaaS

(c)IaaS

Figure 9.2: Cloud Service Models

9.2.1 Software as a Service:

As the name implies, a **SaaS** cloud provides service to customers in the form of software, specifically application software running on and accessible in the cloud. SaaS follows the familiar model of web services, in this case applied to cloud resources. SaaS enables the customer to use the cloud provider's applications running on the provider's cloud infrastructure. The applications are accessible from various client devices through a simple interface such as a web browser. The use of SaaS avoids the complexity of software installation, maintenance, upgrades, and patches. Examples of services at this level are Google Gmail, Microsoft 365, Salesforce, Citrix GoToMeeting, and Cisco WebEx.

Common subscribers to SaaS are organizations that want to provide their employees with access to typical office productivity software, such as document management and e-mail. Individuals also commonly use the SaaS model to acquire cloud resources. Typically, subscribers use specific applications on demand. The cloud provider also usually offers data-related features such as automatic backup and data sharing between subscribers.

The following list, derived from an ongoing industry survey by OpenCrowd (http://cloudtaxonomy.opencrowd.com/taxonomy), describes example SaaS services. The numbers in parentheses refer to the number of vendors currently offering each service.

- **Billing:** (3)**:** Application services to manage customer billing based on usage and subscriptions to products and services.
- **Collaboration:** (18)**:** Platforms providing tools that allow users to collaborate in workgroups, within enterprises, and across enterprises.
- **Content management:** (7)**:** Services for managing the production and access to content for Web-based applications.
- **Customer relationship management:** (13)**:** Platforms for CRM application that range from call center applications to sales force automation.
- **Document management:** (6)**:** Platforms of managing documents, document production workflows, and providing workspaces for groups or enterprises to find and access documents.
- **Education:** (4)**:** Provides online services to Educators and Educational institutions.
- **Enterprise resource planning:** (8)**:** ERP is an integrated computer-based system used to manage internal and external resources, including tangible assets, financial resources, materials, and human resources.
- **Financials:** (11)**:** Applications for managing financial processes for companies that range from expense processing and invoicing to tax management.
- **Healthcare:** (10)**:** Services for improving and managing people's health and healthcare management.
- **Human resources:** (10)**:** Software for managing human resources functions within companies.
- **IT services management:** (5)**:** Software that helps enterprises manage IT services delivery to services consumers and manage performance improvement.
- **Personal productivity:** (5)**:** Software that business users use on a daily basis in the normal course of business. The typical suite includes applications for word processing, spreadsheets, and presentations.
- **Project management:** (12)**:** Software packages for managing projects. Features of packages may specialize the offering for specific types of projects such as software development, construction, and so on.
- **Sales:** (7)**:** Applications that are specifically designed for sales functions such as pricing, commission tracking, and so on.

- **Security:** (10)**:** Hosted products for security services such as malware and virus scanning, single sign-on, and so on.
- **Social networks:** (4)**:** Platforms for creating and customizing social networking applications.

9.2.2 Platform as a Service:

A **PaaS** cloud provides service to customers in the form of a platform on which the customer's applications can run. PaaS enables the customer to deploy onto the cloud infrastructure customer-created or -acquired applications. A PaaS cloud provides useful software building blocks, plus a number of development tools, such as programming language tools, runtime environments, and other tools that assist in deploying new applications. In effect, PaaS is an operating system in the cloud. PaaS is useful for an organization that wants to develop new or tailored applications while paying for the needed computing resources only as needed and only for as long as needed. AppEngine, Engine Yard, Heroku, Microsoft Azure, Force.com, and Apache Stratos are examples of PaaS.

A group of capabilities offered via cloud computing in which the cloud service customer can deploy, manage, and run customer-created or customer-acquired applications using one or more programming languages and one or more execution environments supported by the cloud service provider. The following list describes example PaaS services. The numbers in parentheses refer to the number of vendors currently offering each service:

- **Big data as a service** (19)**:** These are cloud-based services for the analysis of large or complex data sets that require high scalability.
- **Business intelligence** (18)**:** Platforms for the creation of business intelligence applications such as dashboards, reporting systems, and big data analysis.
- **Database** (18)**:** These services offer scalable database systems ranging from relational database solutions to massively scalable non-SQL datastores.
- **Development and testing** (18)**:** These platforms are only for the development and testing cycles of application development, which expand and contract as needed.
- **General purpose** (22)**:** Platforms suited for general-purpose application development. These services provide a database, a web application runtime environment, and typically support web services for integration.
- **Integration** (14)**:** Services for integrating applications ranging from cloud-to-cloud integration to custom application integration.

9.2.3 Infrastructure as a Service:

A group of capabilities offered via cloud computing in which the cloud service customer can provision and use processing, storage, or networking resources. Typically, customers are able to self-provision this infrastructure, using a web-based graphical user interface that serves as an IT operations management console for the overall environment. API access to the infrastructure may also be offered as an option. Examples of IaaS are Amazon Elastic Compute Cloud (Amazon EC2), Microsoft Windows Azure, Google Compute Engine (GCE), and Rackspace.

With **IaaS**, the customer has access to the resources of the underlying cloud infrastructure. IaaS provides virtual machines and other abstracted hardware and operating systems. IaaS offers the customer processing, storage, networks, and other fundamental computing resources so that the customer can deploy and run arbitrary software, which can include operating systems and applications. IaaS enables customers to combine basic computing services, such as number crunching and data storage, to build highly adaptable computer systems.

The following list describes example IaaS services. The numbers in parentheses refer to the number of vendors currently offering each service:

- **Backup and recovery** (14)**:** Platforms providing services to backup and recover file systems and raw data stores on servers and desktop systems.
- **Cloud broker** (7)**:** Tools that manage services on more than one cloud infrastructure platform. Some tools support private-public cloud configurations.
- **Compute** (31)**:** Provides server resources for running cloud-based systems that can be dynamically provisioned and configured as needed.
- **Content delivery networks** (2)**:** CDNs store content and files to improve the performance and cost of delivering content for web-based systems.
- **Services management** (7)**:** Services that manage cloud infrastructure platforms. These tools often provide features that cloud providers do not provide or specialize in managing certain application technologies.
- **Storage** (12)**:** Provides massively scalable storage capacity that can be used for applications, backups, archiving, file storage, and more.

Figure 9.3 compares the functions implemented by the cloud service provider for the three principal cloud service models and the traditional IT services provided by companies on their own premises.

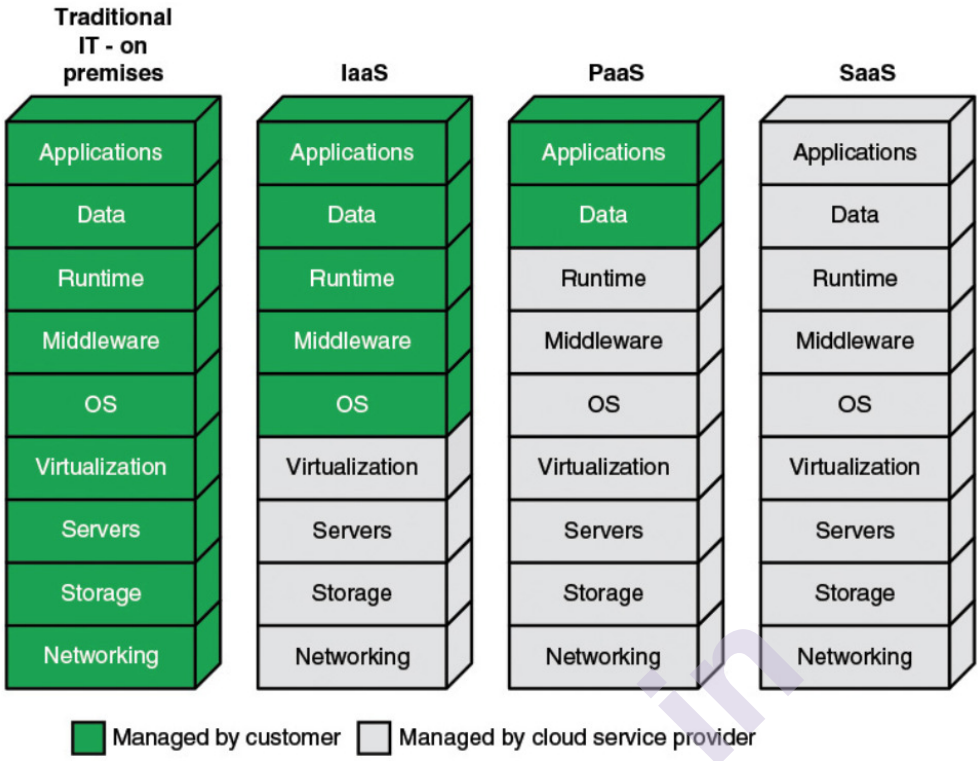


Figure 9.3: Separation of Responsibilities based on Cloud Service Models

9.2.4 Other Cloud Services:

A number of other cloud services have been proposed, with some available as vendor offerings. A useful list of these additional services is provided by ITU-T Y.3500 (*Cloud Computing — Overview and Vocabulary*, August 2014), which includes the following cloud service categories:

- **Communications as a Service (CaaS):** The integration of real-time interaction and collaboration services to optimize business processes. This service provides a unified interface and consistent user experience across multiple devices. Examples of services included are video teleconferencing, web conferencing, instant messaging, and voice over IP.
- **Compute as a Service (CompaaS):** The provision and use of processing resources needed to deploy and run software. CompaaS may be thought of as a simplified IaaS, with the focus on providing compute capacity.
- **Data Storage as a Service (DSaaS):** The provision and use of data storage and related capabilities. DSaaS describes a storage model where the client leases storage space from a third-party provider. Data is transferred from the client to the service provider via the Internet, and the client then accesses the stored data using software provided by the storage provider. The software is used to perform common tasks related to storage, such as data backups and data transfers.

- **Network as a Service (NaaS):** Transport connectivity services / intercloud network connectivity services. NaaS involves the optimization of resource allocations by considering network and computing resources as a unified whole. NaaS can include flexible and extended virtual private network (VPN), bandwidth on demand, custom routing, multicast protocols, security firewall, intrusion detection and prevention, wide-area network (WAN), content monitoring and filtering, and antivirus.

Y.3500 distinguishes between cloud capabilities and cloud services. The three capabilities types are application, platform, and infrastructure, corresponding to the basic service types of SaaS, PaaS, and IaaS. A cloud service category can include capabilities from one or more cloud capability types. Y.3500 also lists examples of emerging cloud service categories:

- **Database as a Service:** Database functionalities on demand where the installation and maintenance of the databases are performed by the cloud service provider.
- **Desktop as a Service:** The ability to build, configure, manage, store, execute, and deliver user desktop functions remotely. In essence, Desktop as a Service offloads common desktop apps plus data from the user's desktop or laptop computer into the cloud. Designed to provide a reliable, consistent experience for the remote use of programs, applications, processes, and files.
- **Email as a Service:** A complete e-mail service, including related support services such as storage, receipt, transmission, backup, and recovery of e-mail.
- **Identity as a Service:** Identity and access management (IAM) that can be extended and centralized into existing operating environments. This includes provisioning, directory management, and the operation of a single sign-on service.
- **Management as a Service:** Includes application management, asset and change management, capacity management, problem management (service desk), project portfolio management, service catalogue, and service level management.
- **Security as a Service:** The integration of a suite of security services with the existing operating environment by the cloud service provider. This may include authentication, antivirus, antimalware/spyware, intrusion detection, and security event management, among others.

9.2.5 XaaS:

XaaS is the latest development in the provisioning of cloud services. The acronym has three generally accepted interpretations, all of which mean pretty much the same thing:

- **Anything as a Service:** Where *anything* refers to any service other than the three traditional services.
- **Everything as a Service:** Although this version is sometimes spelled out, it is somewhat misleading, because no vendor offers every possible cloud service. This version is meant to suggest that the cloud service provider is providing a wide range of service offerings.
- **X as a Service:** Where *X* can represent any possible cloud service option.

XaaS providers go beyond the traditional "big three" services in three ways.

- Some providers package together SaaS, PaaS, and IaaS so that the customer can do one-stop shopping for the basic cloud services that enterprises are coming to rely on.
- XaaS providers can increasingly displace a wider range of services that IT departments typically offer internal customers. This strategy reduces the burden on the IT department to acquire, maintain, patch, and upgrade a variety of common applications and services.
- The XaaS model typically involves an ongoing relationship between customer and provider, in which there are regular status updates and a genuine two-way, real-time exchange of information. In effect, this is a managed service offering, enabling the customer to commit to only the amount of service needed at any time, and to expand both the amount and types of service as the customers' needs evolve and as the offerings available expand.

XaaS is becoming increasingly attractive to customers because it offers these benefits:

- **Total costs are controlled and lowered:** By outsourcing the maximum range of IT services to a qualified expert partner, an enterprise sees both immediate and long-term cost reductions. Capital expenditures are drastically reduced because of the need to acquire far less hardware and software locally. Operating expenses are lower because the resources used are tailored to immediate needs and change only as needs change.
- **Risks are lowered:** XaaS providers offer agreed service levels. This eliminates the risks of cost overruns so common with internal projects. The use of a single provider for a wide range of services provides a single point of contact for resolving problems.
- **Innovation is accelerated:** IT departments constantly run the risk of installing new hardware and software only to find that later versions that are more capable, less expensive, or both are available by the time installation is complete. With XaaS, the latest offerings are more

quickly available. Further, providers can react quickly to customer feedback.

9.3 CLOUD DEPLOYMENT MODELS

An increasingly prominent trend in many organizations is to move a substantial portion or even all information technology (IT) operations to enterprise cloud computing. The organization is faced with a range of choices as to cloud ownership and management. This section looks at the four most prominent deployment models for cloud computing.

9.3.1 Public Cloud:

A public cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services. The cloud provider is responsible both for the cloud infrastructure and for the control of data and operations within the cloud. A public cloud may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud service provider.

In a public cloud model, all major components are outside the enterprise firewall, located in a multi-tenant infrastructure. Applications and storage are made available over the Internet via secured IP, and, can be free or offered at a pay-per-usage fee. This type of cloud supplies easy-to-use consumer-type services, such as: Amazon and Google on-demand web applications or capacity, Yahoo! Mail, and Facebook or LinkedIn social media providing free storage for photographs. Although public clouds are inexpensive and scale to meet needs, they typically provide no or lower service level agreements (SLAs) and may not offer the guarantees against data loss or corruption found with private or hybrid cloud offerings. Public IaaS clouds do not necessarily provide for restrictions and compliance with privacy laws, which remain the responsibility of the subscriber or corporate end user. In many public clouds, the focus is on the consumer and small and medium-size businesses where pay-per-use pricing is available. The major advantage of the public cloud is cost. A subscribing organization pays only for the services and resources it needs and can adjust these as needed. Further, the subscriber has greatly reduced management overhead. The principal concern is security; however, a number of public cloud providers have demonstrated strong security controls, and, in fact, such providers may have more resources and expertise to devote to security that would be available in a private cloud.

9.3.2 Private Cloud:

A private cloud is implemented within the internal IT environment of the organization. The organization may choose to manage the cloud in

house or contract the management function to a third party. In addition, the cloud servers and storage devices may exist on premises or off premises.

Private clouds can deliver IaaS internally to employees or business units through an intranet or the Internet via a virtual private network (VPN), as well as software (applications) or storage as services to its branch offices. In both cases, private clouds are a way to leverage existing infrastructure, and deliver and chargeback for bundled or complete services from the privacy of the organization's network. Examples of services delivered through the private cloud include database on demand, e-mail on demand, and storage on demand.

A key motivation for opting for a private cloud is security. A private cloud infrastructure offers tighter controls over the geographic location of data storage and other aspects of security. Other benefits include easy resource sharing and rapid deployment to organizational entities.

9.3.3 Community Cloud:

A community cloud shares characteristics of private and public clouds. Like a private cloud, a community cloud has restricted access. Like a public cloud, the cloud resources are shared among a number of independent organizations. The organizations that share the community cloud have similar requirements and, typically, a need to exchange data with each other. One example of an industry that is using the community cloud concept is the healthcare industry. A community cloud can be implemented to comply with government privacy and other regulations. The community participants can exchange data in a controlled fashion.

The cloud infrastructure may be managed by the participating organizations or a third party and may exist on premises or off premises. In this deployment model, the costs are spread over fewer users than a public cloud (but more than a private cloud), so only some of the cost savings potential of cloud computing are realized.

9.3.4 Hybrid Cloud:

The hybrid cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (for example, cloud bursting for load balancing between clouds). With a hybrid cloud solution, sensitive information can be placed in a private area of the cloud, and less sensitive data can take advantage of the benefits of the public cloud.

A hybrid public/private cloud solution can be particularly attractive for smaller businesses. Many applications for which security concerns are less can be offloaded at considerable cost savings without committing the organization to moving more sensitive data and applications to the public cloud.

Table 9.2 Relative Strengths and Weaknesses of Cloud Deployment Models

	Private	**Community**	**Public**	**Hybrid**
Scalability	Limited	Limited	Very High	Very High
Security	Most Secure Option	Very Secure	Moderately Secure	Very Secure
Performance	Very Good	Very Good	Low to Medium	Good
Reliability	Very High	Very High	Medium	Medium to High
Cost	High	Medium	Low	Medium

9.4 CLOUD ARCHITECTURE

9.4.1 NIST Cloud Computing Reference Architecture:

NIST SP 500-292, *NIST Cloud Computing Reference Architecture*, September 2011, establishes a reference architecture, described as follows: The NIST cloud computing reference architecture focuses on the requirements of "what" cloud services provide, not a "how to" design solution and implementation. The reference architecture is intended to facilitate the understanding of the operational intricacies in cloud computing. It does not represent the system architecture of a specific cloud computing system; instead, it is a tool for describing, discussing, and developing a system-specific architecture using a common framework of reference.

NIST developed the reference architecture with the following objectives in mind:

- To illustrate and understand the various cloud services in the context of an overall cloud computing conceptual model.
- To provide a technical reference for consumers to understand, discuss, categorize, and compare cloud services.
- To facilitate the analysis of candidate standards for security, interoperability, and portability and reference implementations.

The reference architecture depicted in Figure 6.4 defines five major actors in terms of the roles and responsibilities, as defined in the list that follows.

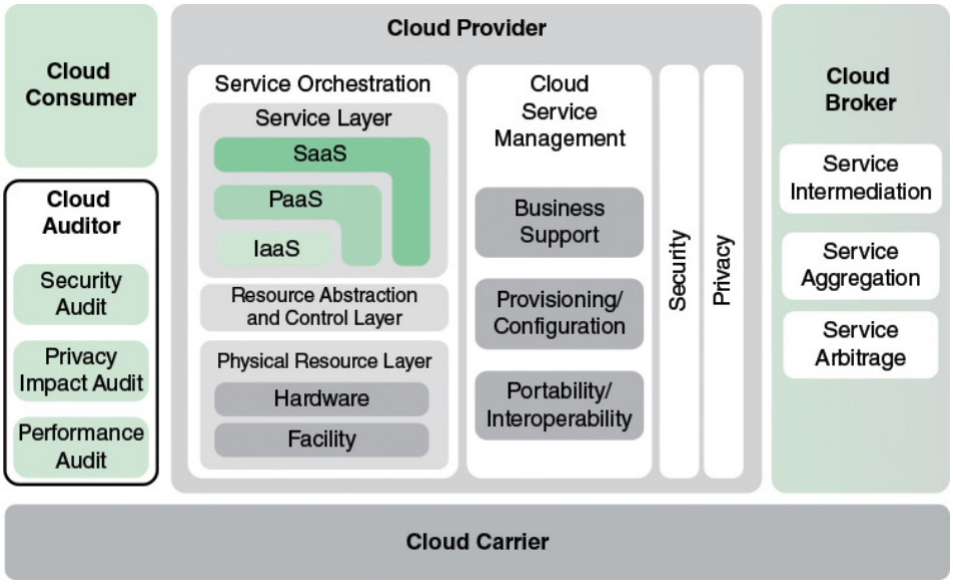


Figure 9.4: NIST Cloud Computing Reference Architecture

- **Cloud consumer**: A person or organization that maintains a business relationship with and uses services from cloud providers.
- **Cloud provider (CP)**: A person, organization, or entity responsible for making a service available to interested parties.
- **Cloud auditor**: A party that can conduct independent assessment of cloud services, information system operations, performance, and security of the cloud implementation.
- **Cloud broker**: An entity that manages the use, performance and delivery of cloud services and negotiates relationships between CPs and cloud consumers.
- **Cloud carrier**: An intermediary that provides connectivity and transport of cloud services from CPs to cloud consumers.

A **cloud provider** can provide one or more of the cloud services to meet IT and business requirements of **cloud consumers**. For each of the three service models (SaaS, PaaS, IaaS), the CP provides the storage and processing facilities needed to support that service model, together with a cloud interface for cloud service consumers. For SaaS, the CP deploys, configures, maintains, and updates the operation of the software applications on a cloud infrastructure so that the services are provisioned at the expected service levels to cloud consumers. The consumers of SaaS can be organizations that provide their members with access to software applications, end users who directly use software applications, or software application administrators who configure applications for end users.

For PaaS, the CP manages the computing infrastructure for the platform and runs the cloud software that provides the components of the

platform, such as runtime software execution stack, databases, and other middleware components. Cloud consumers of PaaS can employ the tools and execution resources provided by CPs to develop, test, deploy, and manage the applications hosted in a cloud environment.

For IaaS, the CP acquires the physical computing resources underlying the service, including the servers, networks, storage, and hosting infrastructure. The IaaS cloud consumer in turn uses these computing resources, such as a virtual computer, for their fundamental computing needs.

The **cloud carrier** is a networking facility that provides connectivity and transport of cloud services between cloud consumers and CPs. Typically, a CP will set up SLAs with a cloud carrier to provide services consistent with the level of SLAs offered to cloud consumers and may require the cloud carrier to provide dedicated and secure connections between cloud consumers and CPs.

A **cloud broker** is useful when cloud services are too complex for a cloud consumer to easily manage. Three areas of support can be offered by a cloud broker:

- **Service intermediation:** These are value-added services, such as identity management, performance reporting, and enhanced security.
- **Service aggregation:** The broker combines multiple cloud services to meet consumer needs not specifically addressed by a single CP, or to optimize performance or minimize cost.
- **Service arbitrage**: This is similar to service aggregation except that the services being aggregated are not fixed. Service arbitrage means a broker has the flexibility to choose services from multiple agencies. The cloud broker, for example, can use a credit-scoring service to measure and select an agency with the best score.

A **cloud auditor** can evaluate the services provided by a CP in terms of security controls, privacy impact, performance, and so on. The auditor is an independent entity that can assure that the CP conforms to a set of standards.

The figure Below (Figure 9.5) illustrates the interactions between the actors. A cloud consumer may request cloud services from a cloud provider directly or via a cloud broker. A cloud auditor conducts independent audits and may contact the others to collect necessary information. This figure shows that cloud networking issues in fact involve three separate types of networks. For a cloud producer, the network architecture is that of a typical large data center, which consists of racks of high-performance servers and storage devices, interconnected with highspeed top-of-rack Ethernet switches. The concerns in this context focus on virtual machine placement and movement, load balancing, and

availability issues. The enterprise network is likely to have a quite different architecture, typically including a number of LANs, servers, workstations, PCs, and mobile devices, with a broad range of network performance, security, and management issues. The concern of both producer and consumer with respect to the cloud carrier, which is shared with many users, is the ability to create virtual networks, with appropriate SLA and security guarantees.

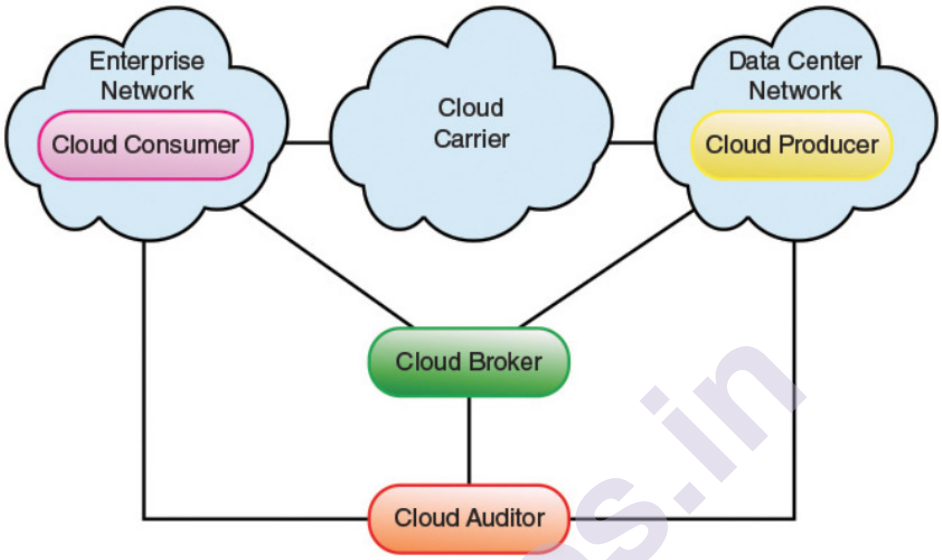


Figure 9.5: Interactions in Cloud Computing between Actors

Service orchestration: refers to the composition of system components to support the cloud provider activities in arrangement, coordination, and management of computing resources to provide cloud services to cloud consumers. Orchestration is shown as a three-layer architecture (see figure 9.4). Examples of resource abstraction components include software elements such as hypervisors, virtual machines, virtual data storage, and other computing resource abstractions.

Cloud service management: includes all the service-related functions necessary for the management and operation of those services required by or proposed to cloud consumers. It covers three main areas:

- **Business support:** This consists of business-related services dealing with customers, such as accounting, billing, reporting, and auditing.
- **Provisioning/configuration:** This includes automated tools for rapid deployment of cloud systems for consumers, adjusting configuration and resource assignment, and monitoring and reporting on resource usage.
- **Portability/interoperability:** Consumers are interested in cloud offering that support data and system portability and service interoperability. This is particularly useful in a hybrid cloud environment, in which the consumer may want to change the allocation of data and applications between on-premises and off-premises sites.

Security and **privacy** are concerns that encompass all layers and elements of the cloud provider's architecture.

9.4.2 ITU-T Cloud Computing Reference Architecture:

ITU-T Cloud Computing Architecture (published in ITU-T Y.3502, *Cloud Computing Architecture*, August 2014) is somewhat broader in scope than the NIST architecture and views the architecture as a layered functional architecture.

The ITU-T document defines three actors:

- **Cloud service customer or user:** A party that is in a business relationship for the purpose of using cloud services. The business relationship is with a cloud service provider or a cloud service partner. Key activities for a cloud service customer include, but are not limited to, using cloud services, performing business administration, and administering use of cloud services.
- **Cloud service provider:** A party that makes cloud services available. The cloud service provider focuses on activities necessary to provide a cloud service and activities necessary to ensure its delivery to the cloud service customer as well as cloud service maintenance. The cloud service provider includes an extensive set of activities (for example, provide service, deploy and monitor service, manage business plan, provide audit data) as well as numerous sub-roles (for example, business manager, service manager, network provider, security and risk manager).
- **Cloud service partner:** A party, which is engaged in support of, or auxiliary to, activities of either the cloud service provider or the cloud service customer, or both. A cloud service partner's activities vary depending on the type of partner and their relationship with the cloud service provider and the cloud service customer. Examples of cloud service partners include cloud auditor and cloud service broker.

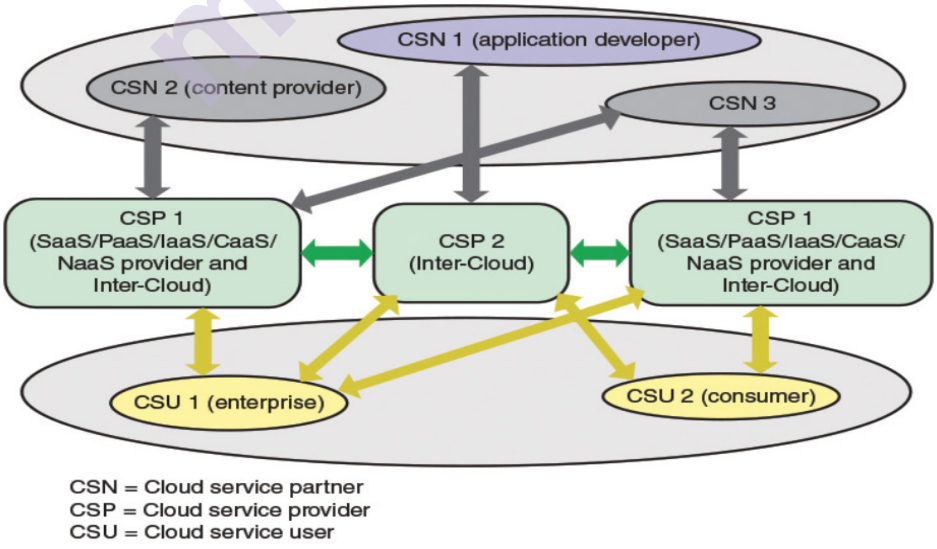


Figure 9.6: Actors with their possible roles in a Cloud Ecosystem

Figure above (figure 9.6) depicts the actors with some of their possible roles in a cloud ecosystem. Figure below (figure 9.7 shows the four-layer ITU-T cloud computing reference architecture.

The user layer is the user interface through which a cloud service customer interacts with a cloud service provider and with cloud services, performs customer related administrative activities, and monitors cloud services. It can also offer the output of cloud services to another resource layer instance. When the cloud receives service requests, it orchestrates its own resources and/or other clouds' resources and provides back cloud services through the user layer. The user layer is where the CSU resides.

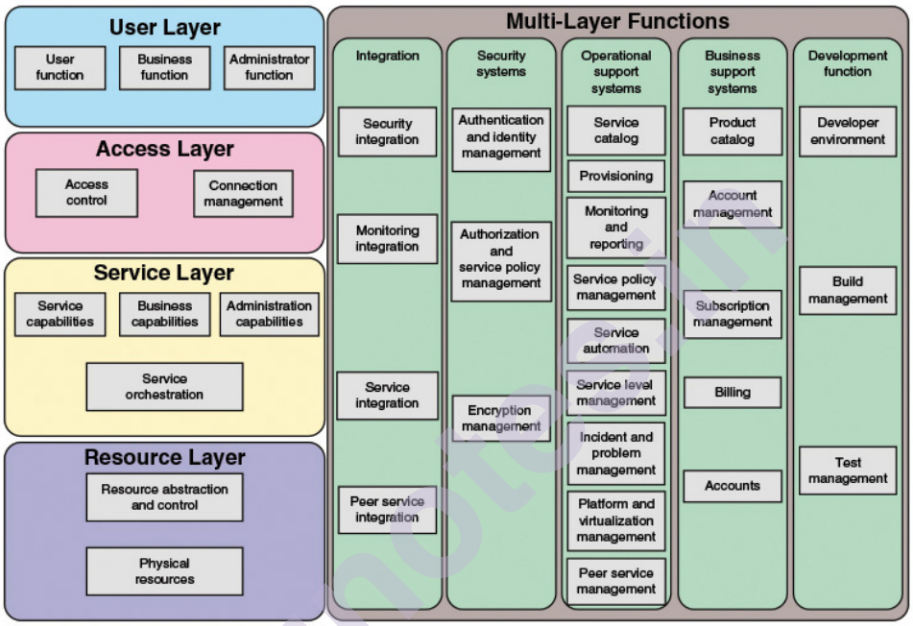


Figure 9.7: ITU-T Cloud Computing Reference Architecture

The access layer provides a common interface for both manual and automated access to the capabilities available in the services layer. These capabilities include both the capabilities of the services and also the administration and business capabilities. The access layer accepts user/partner/other provider cloud service consumption requests using cloud application programming interfaces (APIs) to access the provider's services and resources. The access layer is responsible for presenting cloud service capabilities over one or more access mechanisms — for example, as a set of web pages accessed via a browser, or as a set of web services that can be accessed programmatically. The access layer also deals with security and QoS.

The service layer contains the implementation of the services provided by a cloud service provider (for example, SaaS, PaaS, IaaS). The service layer contains and controls the software components that implement the services (but not the underlying hypervisors, host operating

systems, device drivers, and so on), and arranges to offer the cloud services to users via the access layer.

The resource layer consists of physical resources available to the provider and the appropriate abstraction and control mechanisms. For example, hypervisor software can provide virtual network, virtual storage, and virtual machine capabilities. It also houses the cloud core transport network functionality that is required to provide underlying network connectivity between the provider and users.

The multilayer functions include a series of functional components that interact with functional components of the four other layers to provide supporting capabilities. It includes five categories of functional components:

- **Integration:** Responsible for connecting functional components in the architecture to create a unified architecture. The integration functional components provide message routing and message exchange mechanisms within the cloud architecture and its functional components as well as with external functional components.
- **Security systems:** Responsible for applying security related controls to mitigate the security threats in cloud computing environments. The security systems functional components encompass all the security facilities required to support cloud services.
- **Operational support systems (OSS):** Encompass the set of operational related management capabilities that are required to manage and control the cloud services offered to customers. OSS is also involved in system monitoring, including the use of alarms and events.
- **Business support systems (BSS):** Encompass the set of business-related management capabilities dealing with customers and supporting processes, such as billing and accounts.
- **Development function:** Supports the cloud computing activities of the cloud service developer. This includes support of the development/composition of service implementations, build management and test management.

9.5 SDN AND NFV

Cloud computing predates software-defined networking (SDN) and network functions virtualization (NFV). While cloud computing can be, and has been, deployed and managed without SDN and NFV, both of these technologies are compelling for both private cloud operators and public cloud service providers.

In simplified and generalized terms, what SDN offers is centralized command and control of network resources and traffic patterns. A single

central controller, or a few distributed cooperating controllers, can configure and manage virtual networks and provide QoS and security services. This relieves network management of the need to individually configure and program each networking device.

What NFV offers is automated provisioning of devices. NFV virtualizes network devices, such as switches and firewalls, as well as compute and storage devices, and provides tools for scaling out and automatically deploying devices as needed. Therefore, each project or cloud customer does not require separate equipment or reprogramming of existing equipment. Relevant devices can be centrally deployed via a hypervisor management platform and configured with rules and policies.

9.5.1 Service Provider Perspective:

A large cloud service provider will deal with thousands of customers, with dynamic needs for capacity, both in terms of traffic-carrying capacity and in terms of compute and storage resources. The provider needs to be able to rapidly manage the entire network to handle traffic bottlenecks, manage numerous traffic flows with differing QoS requirements, and deal with outages and other problems. All of this must be done in a secure manner. SDN can provide the needed overall view of the entire network and secure, centralized management of the network. The provider needs to be able to deploy and scale in/out and up/down virtual switches, servers, and storage rapidly and transparently for the customer. NFV provides the automated tools for managing this process.

9.5.2 Private Cloud Perspective:

Large and medium-size enterprises see a number of advantages to moving much of their network-based operations to a private cloud or a hybrid cloud. Their customers are end users, IT managers, and developers. Individual departments may have substantial, dynamic IT resource needs. The enterprise typically will need to develop one or multiple server farms / data centers. As the overall resource demand grows, the ability to deploy and manage all of the equipment becomes more challenging. In addition, there are security requirements, such as firewalls, and antivirus deployments. Further complicating the scenario is the need for load balancing as projects grow and consume more resources, thus the need for rapid scalability and provisioning of devices becomes more pronounced. The need for automated provisioning of virtual networking equipment almost becomes a requirement, and with all the new virtual devices (especially in conjunction with the existing physical devices), centralized command and control becomes a must. SDN and NFV provide the enterprise with the tools to successfully develop and manage private cloud resources for internal use.

9.5.3 ITU-T Cloud Computing Functional Reference Architecture:

For our discussion of the relationship between cloud networking and NFV, it is instructive to look at an earlier version of this architecture, defined in *ITUT Focus Group on Cloud Computing Technical Report, Part 2: Functional Requirements and Reference Architecture*, February 2012 and shown in Figure 9.8. This architecture has the same four-layer structure as that of Y.3502, but provides more detail of the lowest layer, called the resources and network layer.

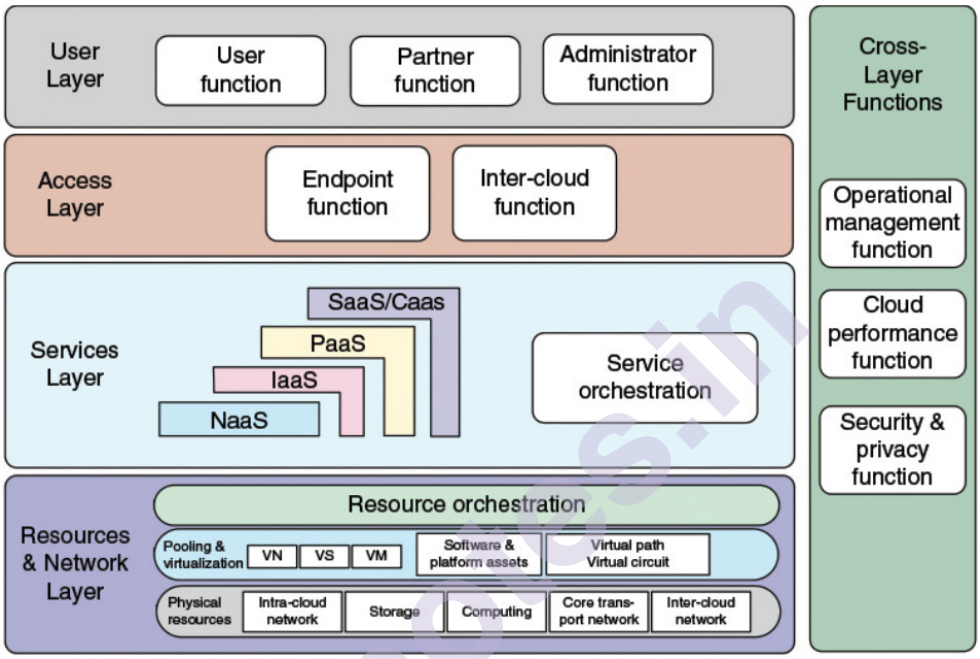


Figure 9.8: ITU-T Cloud Computing Functional Reference Architecture

A comparison of the resources and network layer of the ITU-T architecture to the NFV architectural framework suggests that the resources and network layer can be implemented using the network functions virtualization infrastructure (NFVI) for the lower two sublayers and virtualized infrastructure manager (VIM) for the resource orchestration sublayer. Thus, the general-purpose tools, often in the form of open software, plus commercial off-the-shelf physical resources, enable the cloud provider to effectively deploy and manage cloud services and resources. It should also be an effective strategy to map many of the upper layer functions in the cloud architecture to either virtual network functions or SDN control and application layer functions. Thus, both NFV and SDN contribute to the deployment of cloud services.

Similar reasoning applies to the NIST reference architecture. The service orchestration component consists of three layers: physical resource, resource abstraction and control, and service layers. The lower

two layers correspond quite well to the NFVI portion of the NFV architecture.

9.6 SUMMARY

- Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (for example, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.
- Cloud networking refers to the networks and network management functionality that must be in place to enable cloud computing.
- As defined by NIST, there are three cloud service models:
- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)
- A SaaS cloud provides service to customers in the form of software, specifically application software running on and accessible in the cloud.
- A PaaS cloud provides service to customers in the form of a platform on which the customer's applications can run.
- IaaS provides virtual machines and other abstracted hardware and operating systems. IaaS offers the customer processing, storage, networks, and other fundamental computing resources so that the customer can deploy and run arbitrary software, which can include operating systems and applications.
- A useful list of these additional services is provided by ITU-T Y.3500 (Cloud Computing — Overview and Vocabulary, August 2014), which includes the following cloud service categories:
- Communications as a Service (CaaS)
- Compute as a Service (CompaaS)
- Data Storage as a Service (DSaaS)
- Network as a Service (NaaS)
- NIST SP 500-292, NIST Cloud Computing Reference Architecture, September 2011, establishes a reference architecture, described as follows: The NIST cloud computing reference architecture focuses on the requirements of "what" cloud services provide, not a "how to" design solution and implementation.
- ITU-T Cloud Computing Architecture (published in ITU-T Y.3502,

Cloud Computing Architecture, August 2014) is somewhat broader in scope than the NIST architecture and views the architecture as a layered functional architecture.

9.7 UNIT END QUESTIONS

1. What is cloud computing and cloud networking?
2. Explain the cloud service models as given by NIST.
3. Differentiate between Traditional IT on-premise service and cloud service models with the help of a suitable diagram.
4. List and explain the various cloud service categories.
5. Write a short note on XaaS.
6. Explain the various cloud deployment models.
7. Compare the various cloud deployment models in terms of their strengths and weaknesses.
8. Write a short note on the NIST Cloud Computing Reference Architecture.
9. Write a short note on the ITU-T Cloud Computing Reference Architecture.
10. What is SDN and NFV and how is it related to Cloud Computing?

9.8 BIBLIOGRAPHY, REFERENCES AND FURTHER READING

- Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings, Addison Wesley Professional
- SDN and NFV Simplified A Visual Guide to Understanding Software Defined Networks and Network Function Virtualization by Jim Doherty, Pearson Education
- Network Functions Virtualization (NFV) with a Touch of SDN by Rajendra Chayapathi, Syed Farrukh Hassan, Addison Wesley
- CCIE and CCDE Evolving Technologies Study Guide by Brad Dodgeworth, Jason Gooley, Ramiro Garza Rios, Pearson Education
- ITU-T Y.3502, Cloud Computing Architecture, August 2014
- NIST SP 500-292, NIST Cloud Computing Reference Architecture, September 2011
- ITU-T Y.3500, Cloud Computing Architecture, August 2014

*****

10

MODERN NETWORK ARCHITECTURE: CLOUDS AND FOG THE INTERNET OF THINGS

Unit Structure

10.0 OBJECTIVES

This chapter provides a detailed look at IoT, providing coverage of the principal components of IoT-enabled things, and examines IoT reference architectures and looks at a three IoT implementations, one open source and two commercial.

After studying this chapter, you should be able to:

- Explain the scope of the Internet of Things.
- List and discuss the five principal components of IoT-enabled things.
- Compare and contrast the ITU-T and IoT World Forum IoT reference models.

- Describe the open source IoTivity IoT implementation.
- Describe the commercial ioBridge IoT implementation.

10.1 THE IOT ERA BEGINS

The future Internet will involve large numbers of objects that use standard communications architectures to provide services to end users. It is envisioned that tens of billions of such devices will be interconnected in a few years. This will provide new interactions between the physical world and computing, digital content, analysis, applications, and services. This resulting networking paradigm is being called the Internet of Things (IoT). This will provide unprecedented opportunities for users, manufacturers, and service providers in a wide variety of sectors.

Technology development is occurring in many areas. Not surprisingly, wireless networking research is being conducted and actually has been conducted for quite a while now, but under previous titles such as mobile computing, pervasive computing, wireless sensor networks, and cyber-physical systems. Many proposals and products have been developed for low power protocols, security and privacy, addressing, low-cost radios, energy efficient schemes for long battery life, and reliability for networks of unreliable and intermittently sleeping nodes. These wireless developments are crucial for the growth of IoT.

Many have provided a vision for the IoT. In a 2014 paper, in the *Internet of Things Journal*, the author suggests personal benefits such as digitizing daily life activities, patches of bionic skin to communicate with surrounding smart spaces for improved comfort, health, and safety, and smart watches and body nodes that optimize access to city services. Citywide benefits could include efficient, delay-free transportation with no traffic lights. Smart buildings could not only control energy and security, but also support health and wellness activities. In the same ways people have been provided new ways of accessing the world through smartphones, the IoT will create a new paradigm in the ways we have continuous access to needed information and services. Regardless of the level of positivity in one's view of the IoT or predictions about how soon this will be realized, it is certainly exciting to consider this future.

10.2 THE SCOPE OF THE INTERNET OF THINGS

ITU-T Y.2060, *Overview of the Internet of Things*, June 2012, provides the following definitions that suggest the scope of IoT:

- **Internet of Things (IoT):** A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

- **Thing:** With regard to the IoT, this is an object of the physical world (physical things) or the information world (virtual things), which is capable of being identified and integrated into communication networks.
- **Device:** With regard to the IoT, this is a piece of equipment with the mandatory capabilities of communication and the optional capabilities of sensing, actuation, data capture, data storage, and data processing.

Y.2060 characterizes the IoT as adding the dimension "Any THING communication" to the information and communication technologies which already provide "any TIME" and "any PLACE" communication (see Figure 10.1).

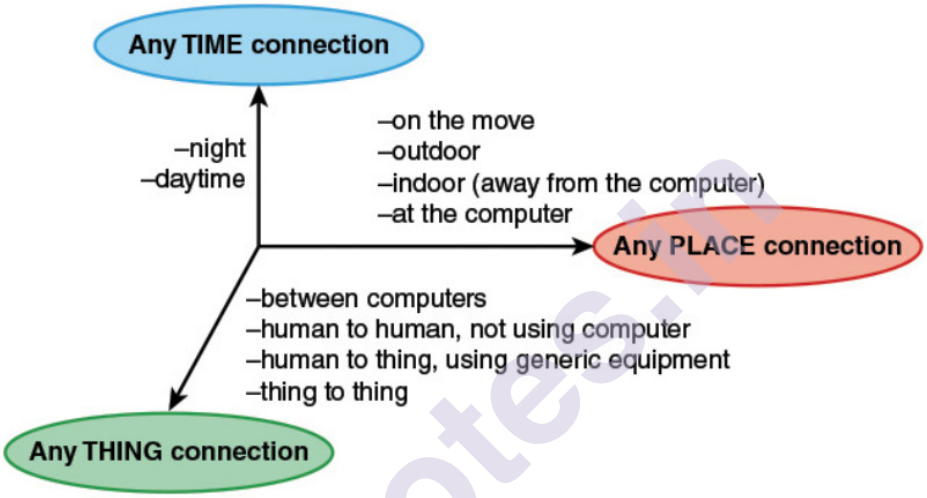


Figure 10.1: New Dimension introduced in IoT

In *Designing the Internet of Things*, the author condenses the elements of the IoT into a simple equation: Physical objects + Controllers, Sensors, Actuators + Internet = IoT

This equation neatly captures the essence of the Internet of Things. An instance of the IoT consists of a collection of physical objects, each of which:

- Contains a microcontroller that provides intelligence,
- Contains a sensor that measures some physical parameter/actuator that acts on some physical parameter,
- Provides a means of communicating via the Internet or some other network.

One item not covered in the equation, and referred to in the Y.2060 definition, is a means of identification of an individual thing, usually referred to as a *tag*.

Table 10.1, based on a graphic from Beecham Research, gives an idea of the scope of IoT.

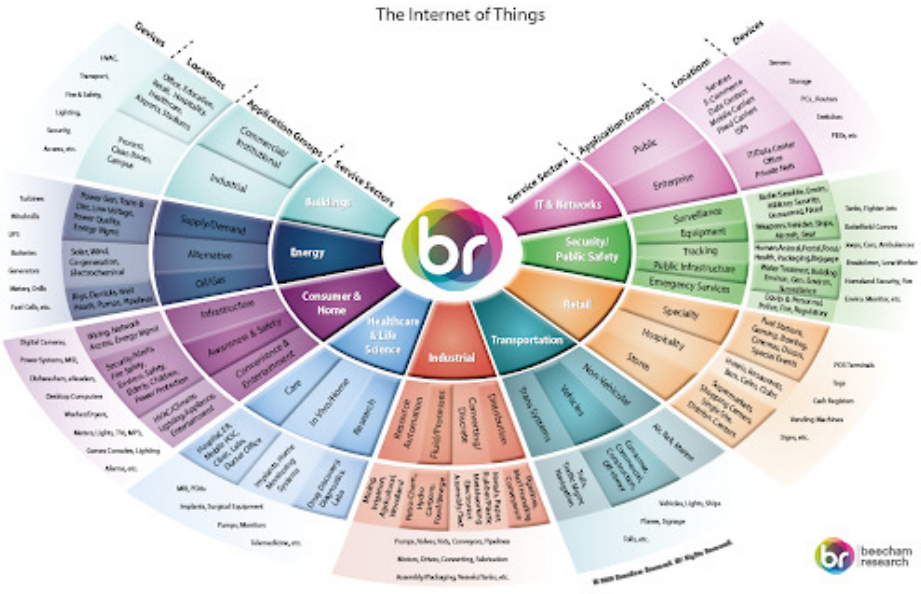


Figure 10.2: Graphic from Beecham Research about scope of IoT

Table 10.1: The Internet of Things (Source: Beecham Research)

Service Sectors	Application Groups	Locations	Device Examples
IT and networks	Public	Services, e-commerce, data centres, mobile carriers, fixed carriers	Servers, storage, PCs, routers, switches, PBXs
		ISPs	
	Enterprise	IT/data centre, office, private nets	
Security/public safety	Surveillance equipment, tracking	Radar/satellite, military security, unmanned, weapons, vehicles, ships, aircraft, gear	Tanks, fighter jets, battlefield communications, jeeps
	Public Infrastructure	Human, animal, postal, food/health, packaging, baggage, water treatment, building environmental, general environmental	Cars, breakdown lane workers, homeland security, fire, environmental monitor

	Emergency Services	Equipment and personnel, police, fire, regulatory	Ambulances, public security vehicles
Retail	Specialty	Fuel stations, gaming, bowling, cinema, discos, special events	POS terminals, tags, cash registers, vending machines, signs
	Hospitality	Hotels, restaurants, bars, cafes, clubs	
	Stores	Supermarkets, shopping centres, single site, distribution centre	
Transportation	Nonvehicular	Air, rail, marine	Vehicles, lights, ships, signage, tolls
	Vehicles	Consumer, commercial, construction, off-road	
	Transportation systems	Tolls, traffic management, navigation	
Industrial	Distribution	Pipelines, material handling, conveyance	Pumps, valves, vats, conveyers, pipelines, motors, drives, converting, fabrication, assembly/packing, vessels, tanks
	Converting, discrete	Metals, paper, rubber, plastic, metalworking, electronics assembly, test	
	Fluid/processes	Petro-chemical, hydrocarbon, food, beverage	
	Resource automation	Mining, irrigation, agricultural, woodland	
Healthcare and Life Sciences	Care	Hospital, ER, mobile PoC, clinic, labs,	MRIs, PDAs, implants, surgical equipment,

		doctor office	pumps, monitors, telemedicine
	In-vivo, home	Implants, home monitoring systems	
	Research	Drug discovery, diagnostics, labs	
Consumer and home	Infrastructure	Wiring, network access, energy management	Digital camera, power systems, dishwashers, eReaders, desktop computers, washer/dryer, meters, lights, TVs, MP#, games console, lighting, alarms
	Awareness and safety	Security/alert, fire safety, environmental safety, elderly, children, power protection	
	Convenience and entertainment	HVAC/climate, lightning, appliance, entertainment	
Energy	Supply/Demand	Power generation, transport and distribution, low voltage, power quality, energy management	Turbines, windmills, uninterruptible power supply (UPS), batteries, generators, meters, drills, fuel cells
	Alternative	Solar, wind, co-generation, electro-chemical	
	Oil/Gas	Rigs, derricks, well heads, pumps, pipelines	
Buildings	Commercial, institutional	Office, education, retail, hospitality, healthcare, airports, stadiums	HVAC, transport, fire and safety, lighting, security, access
	Industrial	Process, clean room, campus	

10.3 COMPONENTS OF IOT-ENABLED THINGS

The key ingredients of an IoT-enabled thing are sensors, actuators, a microcontroller, a means of communication (transceiver), and a means of identification (radio-frequency identification [RFID]). A means of communication is an essential ingredient; otherwise, the device cannot participate in a network. Nearly all IoT-enabled things have some sort of computing capability, no matter how rudimentary. And a device may have one or more of the other ingredients.

10.3.1 Sensors:

A **sensor** measures some parameter of a physical, chemical, or biological entity and delivers an electronic signal proportional to the observed characteristic, either in the form of an analog voltage level or a digital signal. In both cases, the sensor output is typically input to a microcontroller or other management element. A sensor is a device that converts a physical, biological, or chemical parameter into an electrical signal. A sensor may take the initiative in sending sensor data to the controller, either periodically or when a defined threshold is crossed; this is the active mode. Alternatively, or in addition, the sensor may operate in the passive mode, providing data when requested by the controller.

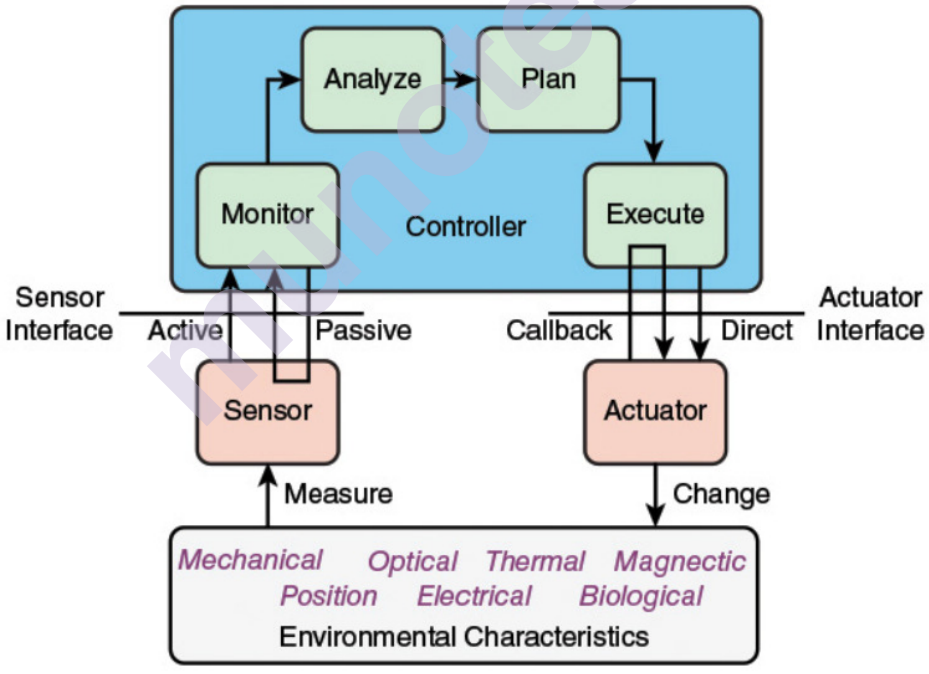


Figure 10.3 Interfaces for Sensors and Actuators

Types of Sensors:

The variety of sensors used in IoT deployments is huge. Sensors may be extremely tiny, using nanotechnology, or quite substantial, such as a surveillance camera. Sensors may be deployed individually or in very

small numbers on the one hand, or in large numbers on the other. Table 10.2 lists various types of sensors, with examples of each type.

Table 10.2 Types of Sensors

Category	What It Does	Device Examples
Position measuring devices	Designed to detect and respond to changes in angular position or in linear position of the device	Potentiometer, linear position sensor, hall effect position sensor, magnetoresistive angular, encoders (quadrature, incremental rotary, absolute rotary, optical)
Proximity, motion sensors	Designed to detect and respond to movement outside of the component but within the range of the sensor	Ultrasonic proximity, optical reflective, optical slotted, PIR (passive infrared), inductive proximity, capacitive proximity, reed switch, tactile switch
Inertial devices	Designed to detect and respond to changes in the physical movement of the sensor	Accelerometer, potentiometer, inclinometer, gyroscope, vibration sensor/switch, tilt sensor, Piezo shock sensor, LVDT/RVDT
Pressure/force	Designed to detect a force being exerted against it	IC barometer, strain gauge, pressure potentiometer, LVDT, silicon transducer, Piezoresistive sensor, capacitive transducer
Optical devices	Designed to detect the presence of light or a change in the amount of light on the sensor	LDR, photodiodes, phototransistors, photo interrupters, reflective sensors, IrDA transceiver, solar cells, LTV (light voltage) sensors
Image, camera devices	Designed to detect and change a viewable image into a digital signal	CMOS image sensor
Magnetic devices	Designed to detect and respond to the presence of a magnetic field	Hall effect sensor, magnetic switch, linear compass IC, Reed sensor
Media devices	Designed to detect and respond to the presence or the amount of a physical substance on the sensor	Gas, smoke, humidity, moisture, dust, float level, fluid flow
Current and voltage devices	Designed to detect and respond to changes in the flow of electricity in a wire or circuit	Hall effect current sensor, DC current sensor, AC current sensor, voltage transducer
Temperature	Designed to detect the amount of heat using different techniques and in different mediums	Thermistor NTC, thermistor PTC, resistance temp detectors (RTD)s, thermocouple, thermopile, digital IC, analog IC, infrared thermometer/pyrometer
Specialized	Designed to provide detection, measurement, or response in specialized situations, which also may include multiple functions	Audio Microphone, Geiger-Müller tube, chemical

10.3.2 Actuators:

An actuator is a device that accepts an electrical signal and converts it into a physical, chemical, or biological action. An **actuator** receives an electronic signal from a controller and responds by interacting with its environment to produce an effect on some parameter of a physical, chemical, or biological entity. In the direct mode of operation, the controller sends a signal that activates the actuator. In callback mode, the actuator responds to the controller to report completion or a problem, and requests further instructions.

Actuators are generally classified as follows:

- **Hydraulic:** Hydraulic actuators consist of a cylinder or fluid motor that utilizes hydraulic power to facilitate mechanical process. The mechanical motion gives an output in terms of linear, rotary, or oscillatory motion.
- **Pneumatic:** Pneumatic actuators work on the same concept as hydraulic actuators except compressed gas is used instead of liquid. Energy, in the form of compressed gas, is converted into linear or rotary motion, depending on the type of actuator.
- **Electric:** Electric actuators are devices powered by motors that convert electrical energy to mechanical torque.
- **Mechanical:** Function through converting rotary motion to linear motion. Devices such as gears, rails, pulley, chain, and others are used to help convert the motion.

10.3.3 Microcontrollers:

The "smart" in a smart device is provided by a deeply embedded microcontroller. An embedded system is any device that includes a computer chip, but that is not a general-purpose workstation, desktop, or laptop computer. The term **embedded system** refers to the use of electronics and software within a product that has a specific function or set of functions, as opposed to a general-purpose computer, such as a laptop or desktop system. Types of devices with embedded systems are almost too numerous to list. Examples include cell phones, digital cameras, video cameras, calculators, microwave ovens, home security systems, washing machines, lighting systems, thermostats, printers, various automotive systems (for example, transmission control, cruise control, fuel injection, anti-lock brakes, and suspension systems), tennis rackets, toothbrushes, and numerous types of sensors and actuators in automated systems.

Often, embedded systems are tightly coupled to their environment. This can give rise to real-time constraints imposed by the need to interact with the environment. Constraints, such as required speeds of motion,

required precision of measurement, and required time durations, dictate the timing of software operations. If multiple activities must be managed simultaneously, this imposes more complex real-time constraints.

Application processors are defined by the processor's ability to execute complex operating systems, such as Linux, Android, and Chrome. Thus, the application processor is general-purpose in nature. A good example of the use of an embedded application processor is the smartphone. The embedded system is designed to support numerous apps and perform a wide variety of functions. Most embedded systems employ a **dedicated processor**, which, as the name implies, is dedicated to one or a small number of specific tasks required by the host device. Because such an embedded system is dedicated to a specific task or tasks, the processor and associated components can be engineered to reduce size and cost.

Early **microprocessor** chips included registers, an arithmetic/logic unit (ALU), and some sort of control unit or instruction processing logic. As transistor density increased, it became possible to increase the complexity of the instruction set architecture, and ultimately to add memory and more than one processor. Contemporary microprocessor chips include multiple processors, called cores, and a substantial amount of cache memory.

Microcontroller is a single chip that contains the processor, non-volatile memory for the program (ROM or flash), volatile memory for input and output (RAM), a clock and an I/O control unit. Also called a *computer on a chip*. A **microcontroller** chip makes a substantially different use of the logic space available. Figure 7.4 shows in general terms the elements typically found on a microcontroller chip. As shown, a microcontroller is a single chip that contains the core, nonvolatile memory for the program (ROM), volatile memory for input and output (RAM), a clock, and an I/O control unit. The processor portion of the microcontroller has a much lower silicon area than other microprocessors and much higher energy efficiency.

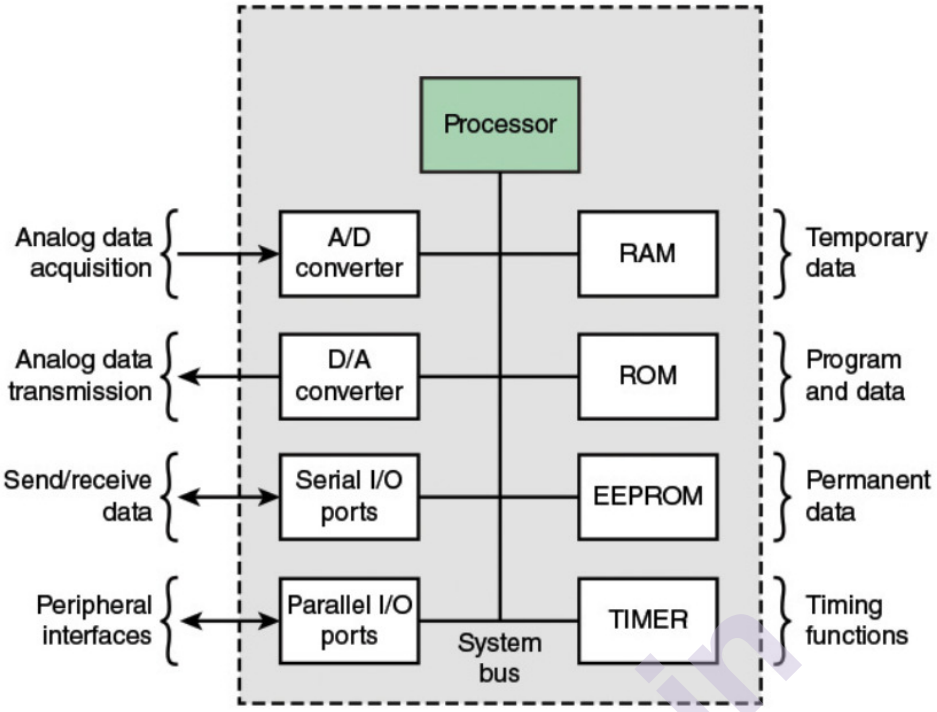


Figure 10.4 Typical Microcontroller Chip Elements

Microcontrollers come in a range of physical sizes and processing power. Processors range from 4-bit to 32-bit architectures. Microcontrollers tend to be much slower than microprocessors, typically operating in the megahertz (MHz) range rather than the gigahertz (GHz) speeds of microprocessors. Another typical feature of a microcontroller is that it does not provide for human interaction. The microcontroller is programmed for a specific task, embedded in its device, and executes as and when required.

10.3.4 Transceivers:

A transceiver is a device that is capable of both transmitting and receiving information. A **transceiver** contains the electronics needed to transmit and receive data. Most IoT devices contain a wireless transceiver, capable of communication using Wi-Fi, ZigBee, or some other wireless scheme. Figure 10.5 is a simplified block diagram showing the basic elements of a transceiver.

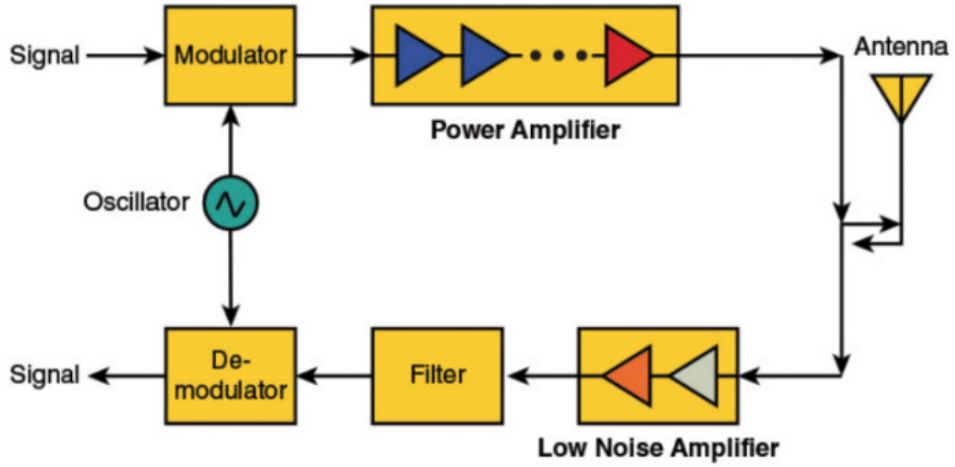


Figure 10.5 Simplified Transceiver Block Diagram

10.3.5 RFID:

A RFID is a data collection technology that uses electronic tags attached to items to allow the items to be identified and tracked by a remote system. The tag consists of an RFID chip attached to an antenna. **Radiofrequency identification (RFID)** technology, which uses radio waves to identify items, is increasingly becoming an enabling technology for IoT. The main elements of an RFID system are tags and readers. RFID tags are small programmable devices used for object, animal and human tracking. They come in a variety of shapes, sizes, functionalities, and costs. RFID readers acquire and sometimes rewrite information stored on RFID tags that come within operating range (a few inches up to several feet). Readers are usually connected to a computer system that records and formats the acquired information for further uses.

The range of applications of RFID is wide and ever expanding. Four major categories of application are tracking and identification, payment and stored-value systems, access control, and anticounterfeiting.

The most widespread use of RFID is for tracking and identification. Early use of RFID was for large high-value items such as train cars and shipping containers. As the price has dropped and the technology improved, this application has expanded dramatically. For example, millions of pets have implanted RFID devices allowing lost animals to be identified and returned to their owner. Another example: tracking and managing the billions of consumer items and components that flow through supply changes is a formidable task and there has been widespread adoption of RFID tags to simplify the task.

Another key area is payment and stored value systems. Electronic toll systems on highways are one example. Another is the use of electronic key “fobs” for payment at retail stores and entertainment venues.

Access control is another widespread application area. RFID proximity cards control building access at many companies and universities. Ski resorts and other leisure venues are also heavy users of this technology.

RFID also is effective as an anti-counterfeiting tool. Casinos use RFID tags on chips to prevent the use of counterfeit chips. The prescription drug industry uses RFID tags to cope with the counterfeit drug market. The tags are used to ensure the pedigree of drugs as they move through the supply chain and also to detect theft.

Figure 10.6 shows the key elements of an RFID system. Primary wireless communication is between a tag and a reader. The reader retrieves identification information and, depending on the application, other information about the tagged item. The reader then communicates this to a computer system which includes an RFID-related database and RFID-related applications.

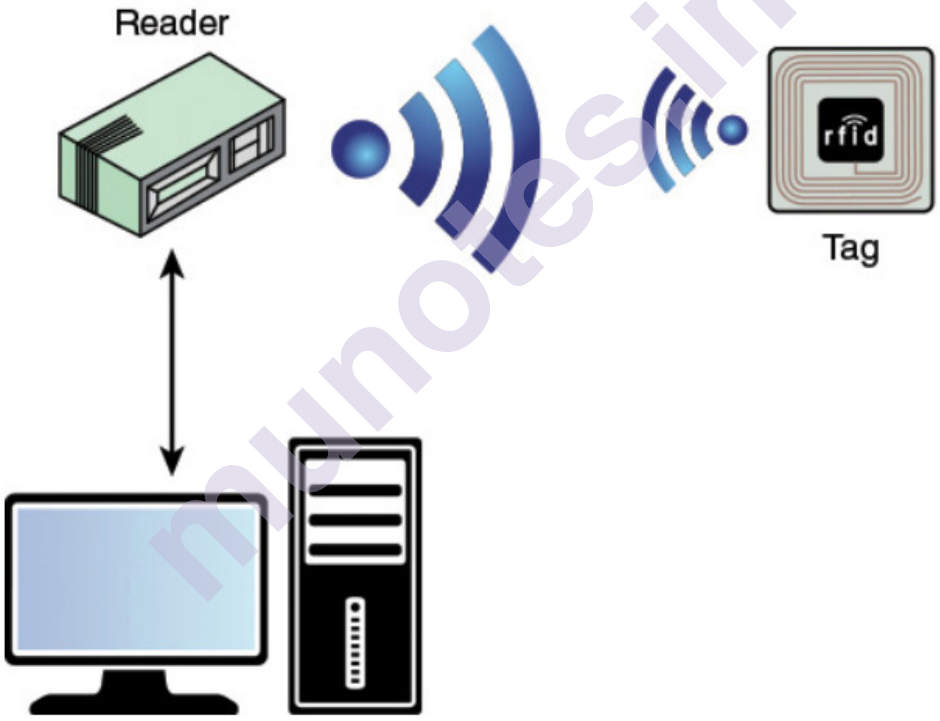


Figure 10.6 Elements of an RFID System

Figure 10.7 shows the two key components of a tag. The antenna is a metallic path in the tag whose layout depends on the size and shape of the tag and the operating frequency. Attached to the antenna is a simple microchip with very limited processing and nonvolatile storage.

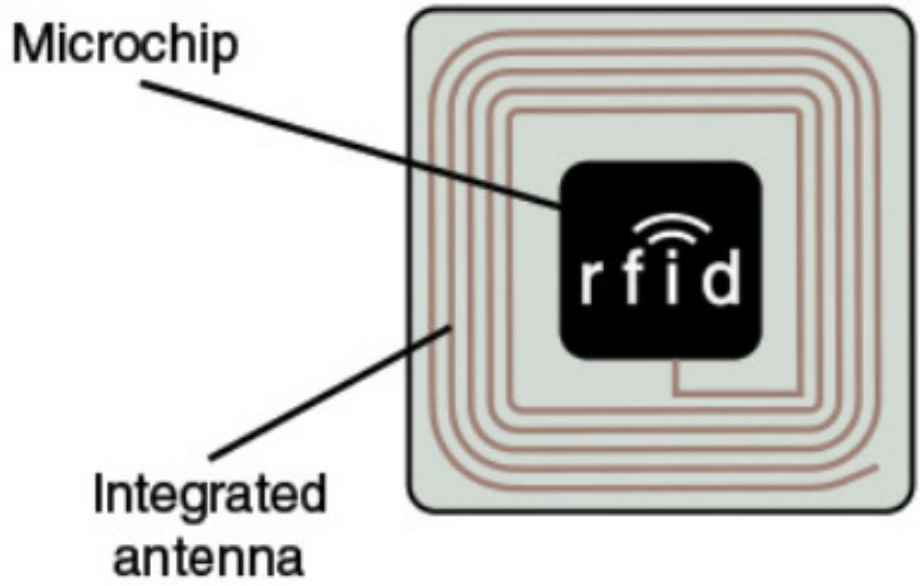


Figure 7.7 RFID Tag

RFID tags are classified as active, semi-passive, or passive. Active RFID tags produce their own signal from a battery, whereas passive RFID tags obtain their power from an RF signal impinging on the tag. Semi-passive tags do have a battery but behave like passive tags. Active tags are considerably more expensive than passive tags and typically are physically larger.

RFID readers communicate with tags through an RF channel. The reader may obtain simple identification information or a more complex set of parameters. The dialogue is often a simple ping and response but may involve a more complex multiple exchange of information. There is a wide variety of different readers in terms of functionality and basic operating style. In general, there are three categories of readers:

- **Fixed:** Fixed readers create portals for automated reading of tags as they pass by. Common applications are to read tags as the associated items enter a room, pass through warehouse dock doors, or travel on a conveyor line.
- **Mobile:** Mobile readers are hand-held devices with an RFID antenna and reader and some computing capability. They are made for manually reading tags on the move. They are useful for inventory applications.
- **Desktop:** This type of reader is typically attached to a PC or point-of-sale terminal and provides easy input.

10.4 IOT ARCHITECTURE

Given the complexity of IoT, it is useful to have an architecture that specifies the main elements and their interrelationship. An IoT architecture can have the following benefits:

- It provides the IT or network manager with a useful checklist with which to evaluate the functionality and completeness of vendor offerings.
- It provides guidance to developers as to which functions are needed in an IoT and how these functions work together.
- It can serve as a framework for standardization, promoting interoperability and cost reduction.

We begin this section with an overview of the IoT architecture developed by ITU-T. We then look at one developed by IoT World Forum. The latter architecture, developed by an industry group, offers a useful alternative framework for understanding the scope and functionality of IoT.

10.4.1 ITU-T IoT Reference Model:

The ITU-T IoT reference model is defined in Y.2060, *Overview of the Internet of Things*, June 2012. Unlike most of the other IoT reference models and architectural models in the literature, the ITU-T model goes into detail about the actual physical components of the IoT ecosystem. This is a useful treatment because it makes visible the elements in the IoT ecosystem that must be interconnected, integrated, managed, and made available to applications. This detailed specification of the ecosystem drives the requirements for the IoT capability. An important insight provided by the model is that the IoT is in fact not a network of physical things. Rather, it is a network of devices that interact with physical things, together with application platforms, such as computers, tablets, and smartphones, that interact with these devices.

The unique aspect of IoT, compared to other network systems, of course, is the presence of a number of physical things and devices other than computing or data processing devices. Figure 7.8, adapted from one in Y.2060, shows the types of devices in the ITU-T model. The model views an IoT as functioning as a network of devices that are tightly coupled with things. Sensors and actuators interact with physical things in the environment. Data capturing devices read data from/write data to physical things via interaction with a data carrying device or a data carrier attached or associated in some way with a physical object.

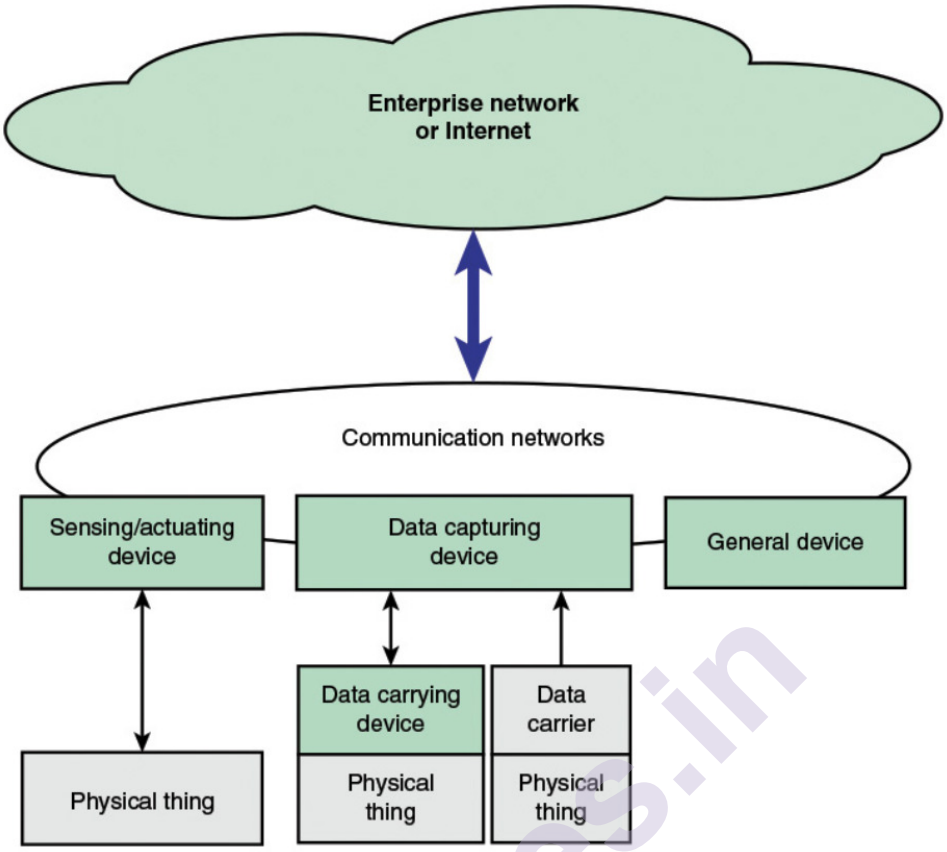


Figure 10.8: Types of Devices and their Relationship with Physical Things

Y.2060 notes that technologies used for interaction between data-capturing devices and data-carrying devices or data carriers include radio frequency, infrared, optical, and galvanic driving. Examples of each include the following:

- **Radio frequency:** An RFID tag is an example.
- **Infrared:** Infrared badges are in use in military, hospital, and other settings where the location and movement of personnel needs to be tracked. Examples include infrared reflective patches used by the military and battery-operated badges that emit identifying information. Remote control devices used in the home or other settings to control electronic devices can also easily be incorporated into an IoT.
- **Optical:** Bar codes and QR codes are examples of identifying data carriers that can be read optically.
- **Galvanic driving:** An example of this is implanted medical devices that use the conductive properties of the body. In implant-to-surface communication, galvanic coupling is used to send signals from an implanted device to electrodes on the skin. This scheme uses very little power and reduces the size and complexity of the implanted device.

Figure 10.9 provides an overview of the elements of interest in IoT. The various ways that physical devices can be connected are shown on the left side of the figure. It is assumed that one or multiple networks support communication among the devices.

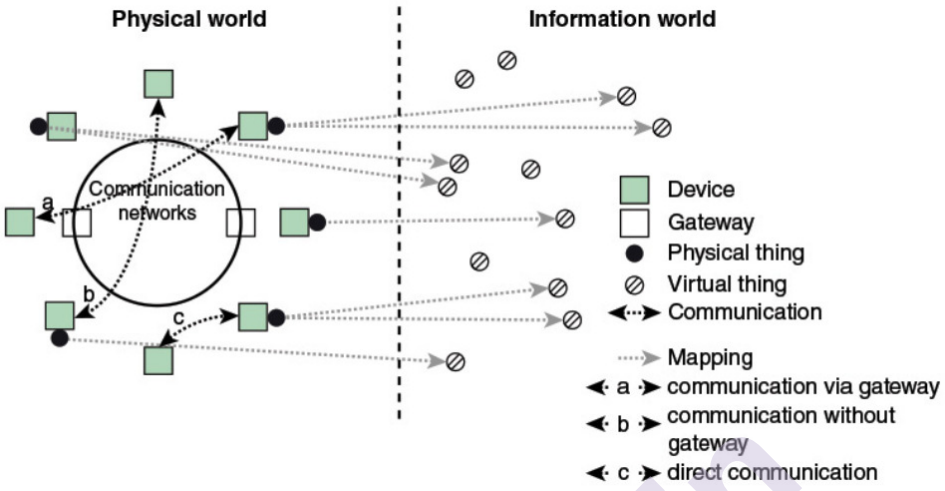


Figure 10.9: Technical Overview of the IoT (Y.2060)

Figure 10.9 introduces one additional IoT-related device: the gateway. At a minimum, a gateway functions as a protocol translator. Gateways address one of the greatest challenges in designing for IoT, which is connectivity, both among devices and between devices and the Internet or enterprise network. Smart devices support a wide variety of wireless and wired transmission technologies and networking protocols. Further, these devices typically have limited processing capability. Y.2067, *Common Requirements and Capabilities of a Gateway for Internet of Things Applications*, June 2014, lays out the requirements for IoT gateways, which generally fall into three categories:

- The gateway supports a variety of device access technologies, enabling devices to communicate with each other and across an Internet or enterprise network with IoT applications. The access schemes could include, for example, ZigBee, Bluetooth, and Wi-Fi.
- The gateway supports the necessary networking technologies for both local and wide-area networking. These could include Ethernet and Wi-Fi on the premises, and cellular, Ethernet, digital subscriber line (DSL), and cable access to the Internet and wide-area enterprise networks.
- The gateway supports interaction with application, network management, and security functions.

The first two requirements involve protocol translation between different network technologies and protocol suites. The third requirement is generally referred to as an **IoT agent** function. In essence, the IoT agent

provides higher-level functionality on behalf of IoT devices, such as organizing/summarizing data from multiple devices to pass on to IoT applications, implementing security protocols and functions, and interacting with network management systems.

The Reference Model:

Figure 7.10 depicts the ITU-T IoT reference model, which consists of four layers as well as management capabilities and security capabilities that apply across layers. We have so far been considering the device layer. In terms of communications functionality, the device layer includes, roughly, the OSI physical and data link layers. We now look at the other layers.

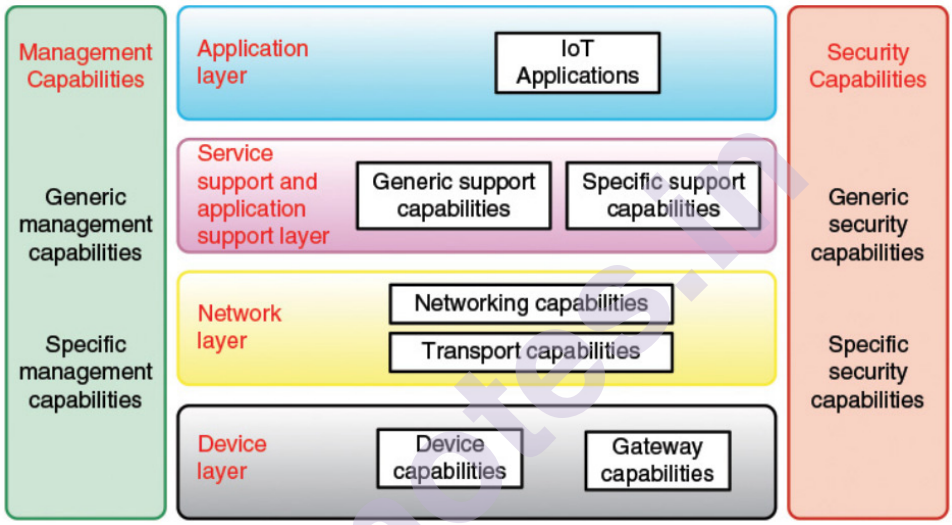


Figure 10.10: ITU-T Y.2060 IoT Reference Model

The **network layer** performs two basic functions. Networking capabilities refer to the interconnection of devices and gateways. Transport capabilities refer to the transport of IoT service and application specific information as well as IoT-related control and management information. Roughly, these correspond to OSI network and transport layers.

The **service support and application support layer** provides capabilities that are used by applications. Generic support capabilities can be used by many different applications. Examples include common data processing and database management capabilities. Specific support capabilities are those that cater for the requirements of a specific subset of IoT applications.

The **application layer** consists of all the applications that interact with IoT devices.

The **management capabilities layer** covers the traditional network-oriented management functions of fault, configuration, accounting, and performance management. Y.2060 lists the following as examples of generic support capabilities:

- **Device management:** Such as device discovery, authentication, remote device activation and deactivation, configuration, diagnostics, firmware/software updating, device working status management
- **Local network topology management:** Such as network configuration management
- **Traffic and congestion management:** Such as the detection of network overflow conditions and the implementation of resource reservation for time-critical/life-critical data flows

Specific management capabilities are tailored to specific classes of applications. An example is smart grid power transmission line monitoring.

The **security capabilities layer** includes generic security capabilities that are independent of applications. Y.2060 lists the following as examples of generic security capabilities:

- **Application layer:** Authorization, authentication, application data confidentiality and integrity protection, privacy protection, security audit, and antivirus
- **Network layer:** Authorization, authentication, user data and signalling data confidentiality, and signalling integrity protection
- **Device layer:** Authentication, authorization, device integrity validation, access control, data confidentiality, and integrity protection

Specific security capabilities relate to specific application requirements, such as mobile payment security requirements.

10.4.2 IoT World Forum Reference Model:

The IoT World Forum (IWF) is an industry-sponsored annual event that brings together representatives of business, government, and academia to promote the market adoption of IoT. The IoT World Forum Architecture Committee, made up of industry leaders including IBM, Intel, and Cisco, released an IoT reference model in October 2014. This model serves as a common framework to help the industry accelerate IoT deployments. The reference model is intended to foster collaboration and encourage the development of replicable deployment models.

This reference model is a useful complement to the ITU-T reference model. The ITU-T Y.206x series seems most concerned with

defining a framework to support development of standards for interaction with IoT devices. The IWF is concerned with the broader issue of developing the applications, middleware, and support functions for an enterprise based IoT.

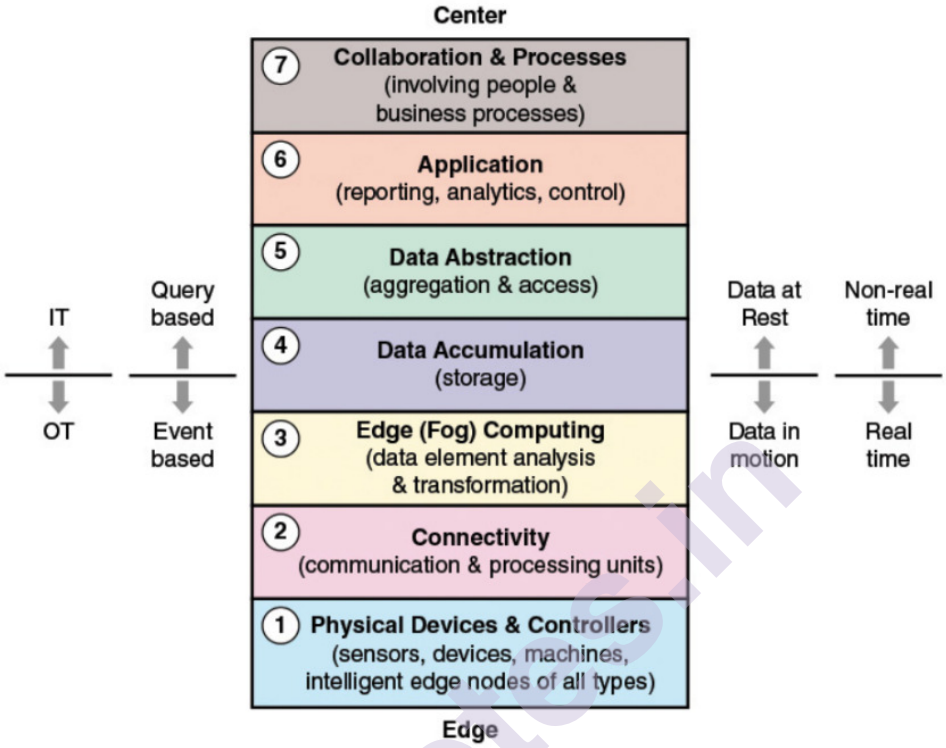


Figure 10.11: IoT World Forum Reference Model

Figure 10.11 depicts the seven-level model. The white paper in the IWF model issued by Cisco indicates that the model is designed to have the following characteristics:

- **Simplifies:** It helps break down complex systems so that each part is more understandable.
- **Clarifies:** It provides additional information to precisely identify levels of the IoT and to establish common terminology.
- **Identifies:** It identifies where specific types of processing is optimized across different parts of the system.
- **Standardizes:** It provides a first step in enabling vendors to create IoT products that work with each other.
- **Organizes:** It makes the IoT real and approachable, instead of simply conceptual.

Level 1 consists of physical devices and controllers that might control multiple devices. Level 1 of the IWF model corresponds approximately to the device level of the ITU-T model. As with the ITU-T

model, the elements at this level are not physical things as such, but rather devices that interact with physical things, such as sensors and actuators. Among the capabilities that devices may have are analog-to-digital and digital-to-analog conversion, data generation, and the ability to be queried/controlled remotely.

Level 2 of the IWF model corresponds approximately to the network level of the ITU-T model. The main difference is that the IWF model includes gateways in level 2 whereas the ITU-T model puts the gateway at level 1. From a logical point of view, this level enables communication between devices and communication between devices and the low-level processing that occurs at level 3. From a physical point of view, this level consists of networking devices, such as routers, switches, gateways, and firewalls that are used to construct local and wide-area networks and provide Internet connectivity. This level enables devices to communicate with one another and via the upper logical levels, with application platforms such as computers, remote control devices, and smartphones.

In many IoT deployments, massive amounts of data may be generated by a distributed network of sensors. Rather than store data permanently in central storage accessible to IoT applications, it is often desirable to do as much data processing close to the sensors as possible. Thus, the purpose of the edge computing level is to convert network data flows into information that is suitable for storage and higher-level processing. Processing elements at this level may deal with high volumes of data and perform data transformation operations, resulting in the storage of much lower volumes of data. The Cisco white paper on the IWF model lists the following examples of edge computing operations:

- **Evaluation:** Evaluating data for criteria as to whether it should be processed at a higher level
- **Formatting:** Reformatting data for consistent higher-level processing
- **Expanding/decoding:** Handling cryptic data with additional context (such as the origin)
- **Distillation/reduction:** Reducing/summarizing data to minimize the impact of data and traffic on the network and higher-level processing systems
- **Assessment:** Determining whether data represents a threshold or alert; this could include redirecting data to additional destinations

Processing at the edge computing level is sometimes referred to as **fog computing**. Fog computing and fog services are expected to be a distinguishing characteristic of the IoT. Figure 7.12 illustrates the concept. Fog computing represents an opposite trend in modern networking from cloud computing. With fog computing, massive numbers of individual

smart objects are interconnected with fog networking facilities that provide processing and storage resources close to the edge devices in an IoT. Fog computing addresses the challenges raised by the activity of thousands or millions of smart devices, including security, privacy, network capacity constraints, and latency requirements. The term *fog computing* is inspired by the fact that fog tends to hover low to the ground, whereas clouds are high in the sky.

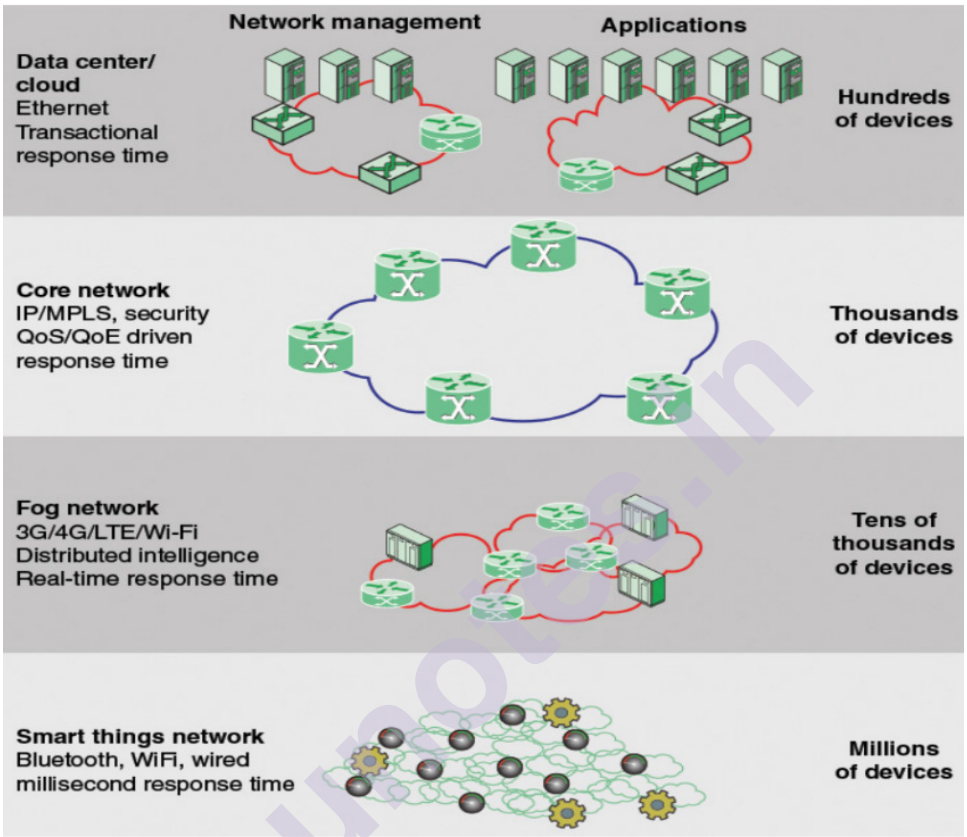


Figure 10.12: Fog Computing

The Data Accumulation Level is where data coming from the numerous devices, and filtered and processed by the edge computing level, is placed in storage that will be accessible by higher levels. This level marks a clear distinction in the design issues, requirements, and method of processing between lower-level (fog) computing and upper-level (typically cloud) computing. Data moving through a network is referred to as *data in motion*. The rate and organization of the data in motion is determined by the devices generating the data. Data generation is event driven, either periodically or by an event in the environment. To capture the data and deal with it in some fashion, it is necessary to respond in real time. By contrasts, most applications do not need to process data at network transfer speeds. As a practical matter, neither the cloud network nor the application platforms would be able to keep up with data volume generated by a huge number of IoT devices. Instead, applications deal with *data at rest*, which is data in some readily accessible storage facility.

Applications can access the data as needed, on a non-real-time basis. Thus, the upper levels operate on a query or transaction basis, whereas the lower three levels operate on an event basis. The Cisco white paper on the IWF model lists the following as operations performed at the data accumulation level:

- Converts data-in-motion to data-at-rest
- Converts format from network packets to database relational tables
- Achieves transition from event based to query based computing
- Dramatically reduces data through filtering and selective storing

The data accumulation level absorbs large quantities of data and places them in storage, with little or no tailoring to specific applications or groups of applications. A number of different types of data in varying formats and from heterogeneous processors may be coming up from the edge computing level for storage. The data abstraction level can aggregate and format this data in ways that make access by applications more manageable and efficient. Tasks involved could include the following:

- Combining data from multiple sources. This includes reconciling multiple data formats.
- Perform necessary conversions to provide consistent semantics of data across sources.
- Place formatted data in appropriate database. For example, high-volume repetitive data may go into a big data system such as Hadoop. Event data would be steered to a relational database management system, which provides faster query times and an appropriate interface for this type of data.
- Alerting higher-level applications that data is complete or had accumulated to a defined threshold.
- Consolidating data into one place (with ETL [extract, transform, load], ELT [extract, load, transform], or data replication) or providing access to multiple data stores through data virtualization.
- Protecting data with appropriate authentication and authorization.
- Normalizing or denormalizing and indexing data to provide fast application access.

The Application Level contains any type of application that uses IoT input or controls IoT devices. Generally, the applications interact with level 5 and the data at rest, and so do not have to operate at network speeds. Provision should be available for streamlined operation that allows applications to bypass intermediate layers and interact directly with Layer 3 or even Layer 2. The IWF model does not strictly define applications, considering this beyond the scope of IWT model discussion.

The **Collaboration and Processes Level** recognizes the fact that people must be able to communicate and collaborate to make an IoT useful. This may involve multiple applications and exchange of data and control information across the Internet or an enterprise network.

Figure 10.13, adapted from one in a Cisco presentation on the IWF model, pulls together the key concepts in the IWF model.

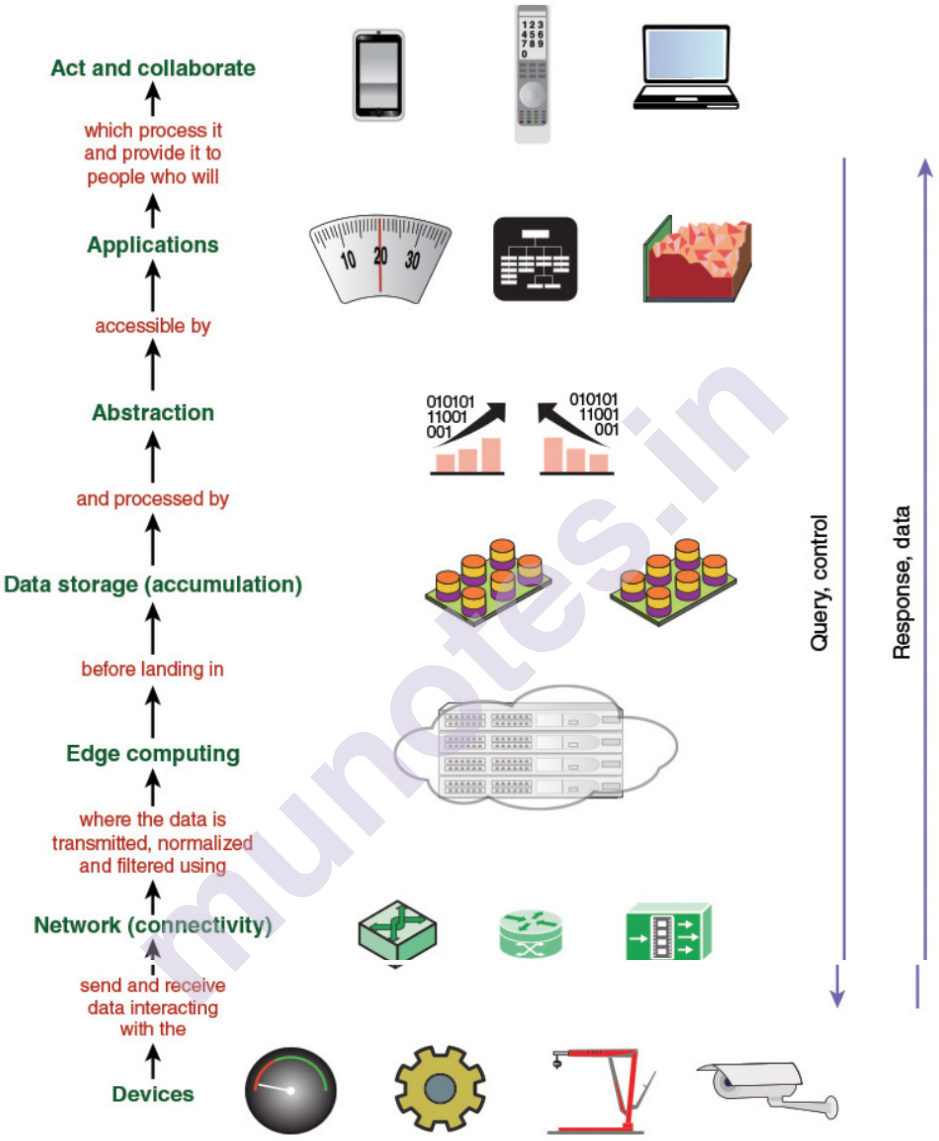


Figure 10.13: IoT World Forum Reference Model – Basic Premises

10.5 IOT IMPLEMENTATION

This section turns to the practical issue of deploying IoT devices and software, by looking at three implementation efforts. First, we examine an open-source software initiative, and then look at two vendor offerings.

10.5.1 IoTivity:

IoTivity is an open-source software initiative. Their objective is to provide a standard and open-source implementation so devices and services will be able to work together regardless of who makes them. Two organizations are playing a key role in the IoTivity project. The project is sponsored by the Open Interconnect Consortium (OIC). OIC is an industry consortium whose purpose is to promote an open-source implementation to improve interoperability between the billions of devices making up the IoT. To this end, OIC is working on developing standards and an overall framework that will establish a single solution covering interoperability across multiple vertical markets and use cases. The charter of the IoTivity project is to develop and maintain an open-source implementation compliant with OIC final specifications and which passes OIC certification testing. The IoTivity Project is hosted by the Linux Foundation, the nonprofit consortium dedicated to fostering the growth of Linux and collaborative development. As a Linux Foundation project, IoTivity is overseen by an independent steering group that will work with the OIC. Developers who want to get involved with the project can access RESTful-based application programming interfaces (APIs) and submit code for peer review through the project's server. It will be made available across a range of programming languages, operating systems, and hardware platforms.

Protocol Architecture:

The IoTivity software provides a number of general-purpose query/response functions to be implemented in IoT devices and in application platforms. IoTivity makes a distinction between a **constrained device** and an unconstrained device. Many devices in the IoT, particularly the smaller, more numerous devices, are resource constrained. The term constrained device refers to a device with limited volatile and nonvolatile memory, limited processing power, and a low data rate transceiver. The term *unconstrained device* simply refers to any device without severe resources constraints. Such devices might run a general-purpose operating system, such as iOS, Android, Linux, or Windows. Unconstrained devices would include IoT devices with a good amount of processing power and memory, and application platforms for IoT applications.

To accommodate constrained devices, the overall protocol architecture (see Figure 10.14) is implemented in both constrained and unconstrained devices. At the transport level, the software relies on User Datagram Protocol (UDP), which requires minimal processing power and memory, running on top of Internet Protocol (IP). Running on top of UDP is the Constrained Application Protocol (CoAP), which is a simplified query/response protocol designed for constrained devices. The IoTivity implementation uses libcoap, which is a C implementation of CoAP that can be used both on constrained and unconstrained devices.

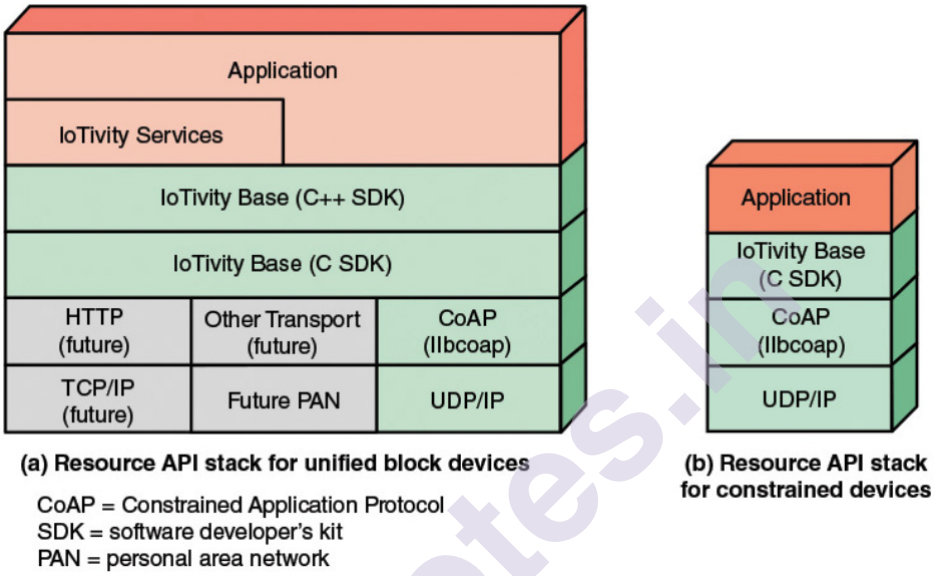


Figure 10.14 IoTivity Stack Blocks

The IoTivity base is a set of software development tools that support the creation of applications for communication between clients that host IoT applications and servers, which are IoT devices. The base is implemented in C, with additional tools inC++ for unconstrained devices. This software is a base for the development of open-source applications that will be part of the IoTivity package, in addition to proprietary, value-added applications developed by vendors.

Constrained Application Protocol:

CoAP is defined in RFC 7252, *The Constrained Application Protocol*, June 2014. The RFC describes CoAP as a specialized web transfer protocol for use with constrained nodes and constrained networks in the IoT. The protocol is designed for machine-to-machine (M2M) applications such as smart energy and building automation. CoAP provides a request/response interaction model between application endpoints, supports built-in discovery of services and resources, and includes key concepts of the web such as URIs and Internet media types. CoAP is designed to easily interface with HTTP for integration with the

web while meeting specialized requirements such as multicast support, very low overhead, and simplicity for constrained environments.

IoTivity Base Services:

The IoTivity Base is software that runs on top of the CoAP API. It presents a resource model to higher layers, consisting of clients and servers. A server hosts resources, which are of two kinds: entity and entity handler. An entity corresponds to an IoT thing, either an actuator or a sensor. An entity handler is an associated device, such as one that caches data from one or more sensors, or a proxy for gateway type protocol conversion. The IoTivity Base provides the following services to higher layers:

- **Resource registration:**
  This is used to register a resource for future access.
- **Resource and device discovery:**
  This operation returns identification information for all resources of a given type on the network service. The operation is sent via multicast to all services.
- **Querying resource (GET):**
  Get information from resource.
- **Setting a resource state (PUT):**
  This operation sets the value of a simple resource.
- **Observing resource state:**
  This operation fetches and registers as an observer for the value of a simple resource. Notifications are then provided to the client on an application-specific schedule.

The following example of querying a resource is from the IoTivity website. This example fetches the state from a light source in the following steps (see Figure 10.15):

1. The client application calls resource.get(...) to retrieve a representation from the resources.
2. The call is marshaled to the stack, which is either running in process or out of process (daemon).
3. The C API is called to dispatch the request. The call may look like the following: OCDoResource(OC_REST_GET, "//192.168.1.11/light/1, 0, 0, OC_CONFIRMABLE, callback);
4. Where CoAP is used as a transport, the lower stack will send a GET request to the target server.
5. On the server side, the OCProcess() function (message pump) receives and parses the request from the socket, then dispatches it to the correct entity handler based on the URI of the request.

6. Where the C++ API is used, the C++ entity handler parses the payload and marshals it to the client application depending on if the server stack is running in process or out of process (daemon).
7. The C++ SDK passes it up the C++ handler associated with the OCResource.
8. The handler returns the result code and representation to the SDK.
9. The SDK marshals the result code and representation to the C++ entity handler.
10. The entity handler returns the result code and representation to the CoAP protocol.
11. The CoAP protocol transports the results to the client device.
12. The results are returned the OCDoResource callback.
13. The results are returned to the C++ client application's syncResultCallback.

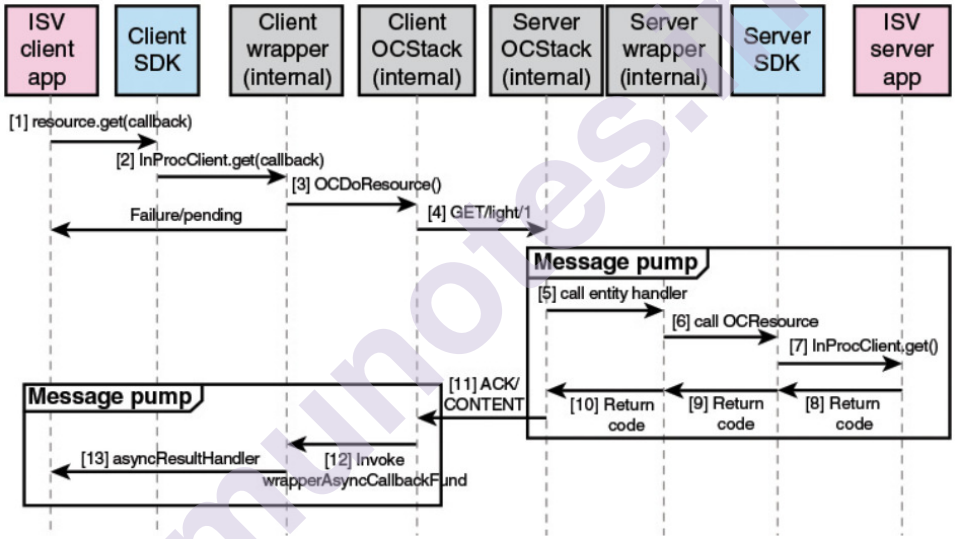


Figure 10.15: Sequence Diagram for Querying Resource State

IoTivity Services:

The IoTivity Base services provide a RESTful API for the basic functions. On top of this, the current release includes four applications referred to as IoTivity Services. IoTivity Services provide a common set of functionalities to application development. These primitive services are designed to provide easy, scalable access to applications and resources and are fully managed by themselves. The four services are as follows:

- **Protocol Plugin Manager:**

Makes IoTivity applications communicate with non-IoTivity devices by plugin protocol converters. It provides several reference

protocol plug-ins and plug-in manager APIs to start/stop plug-ins.

- **Soft Sensor Manager:**

Provides physical and virtual sensor data on IoTivity in a robust manner useful for application developers. It also provides a deployment and execution environment on IoTivity for higher level virtual sensors. Its functions include the following**:** collect physical sensor data; manipulate collected data by aggregating based on its own composition algorithms; providing data to applications; detect specific events and changes.

- **Things Manager:**

Creates groups, finds appropriate member things in the network, manages member presence, and makes group action easy. This service eases the task of applications by enabling them to deal with a group of things with single commands/responses.

- **Control Manager:**

provides framework and services to implement a controller, a controllee, and REST framework for a controller. It also provides APIs for application developers.

10.5.2 Cisco IoT System:

In 2015, Cisco introduced a suite of integrated and coordinated products known as the Cisco IoT System. The philosophy guiding product development is based on the following observations. Cisco estimates that 50 billion devices and objects will be connected to the Internet by 2020. Yet today, more than 99 percent of things in the physical world remain unconnected. To capitalize on the unprecedented opportunities presented by this wave of digitization, companies and cities are increasingly deploying IoT solutions.

Cisco IoT System addresses the complexity of digitization with an infrastructure that is designed to manage large-scale systems of diverse endpoints and platforms, and the data deluge they create. The system consists of six critical technology pillars that, when combined together into an architecture, help reduce the complexities of digitization. Cisco also announced a number of IoT products within the six pillars and will continue to roll out new products as part of the Cisco IoT System. Figure 10.16 illustrates the six IoT system pillars as described in the list that follows.

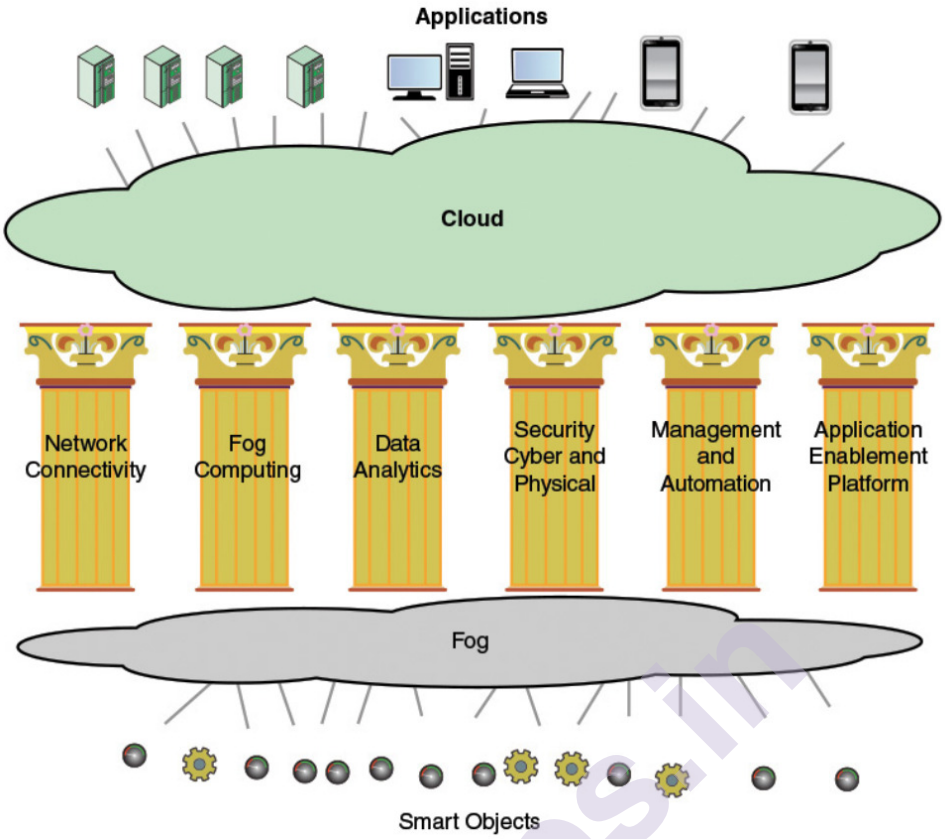


Figure 10.16: Cisco IoT System

- **Network connectivity:** Includes purpose-built routing, switching, and wireless products available in ruggedized and nonruggedized form factors.
- **Fog computing:** Provides Cisco's fog computing, or edge data processing platform, IOx.
- **Data analytics:** An optimized infrastructure to implement analytics and harness actionable data for both the Cisco Connected Analytics Portfolio and third-party analytics software.
- **Security:** Unifies cyber and physical security to deliver operational benefits and increase the protection of both physical and digital assets. Cisco's IP surveillance portfolio and network products with TrustSec security and cloud/cyber security products allow users to monitor, detect and respond to combined IT and operational technology (OT) attacks.
- **Management and automation:** Tools for managing endpoints and applications.
- **Application enablement platform:** A set of APIs for industries and cities, ecosystem partners and third-party vendors to design, develop, and deploy their own applications on the foundation of IoT System capabilities.

Figure 10.17 highlights the key elements of each pillar followed by an overview of each pillar.

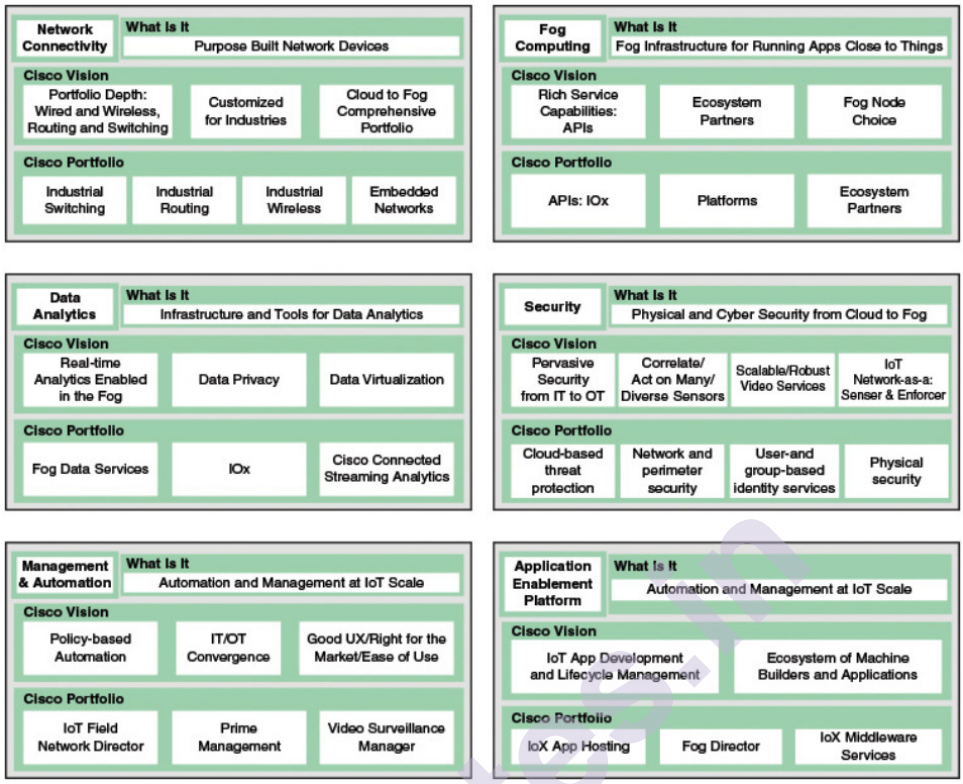


Figure 10.17 The Cisco IoT Pillars

Network Connectivity:

The network connectivity component of Cisco IoT System is a collection of network products for the edge of the network, to support connectivity of smart objects, gateways, and other edge computing devices. Many smart objects are deployed in harsh or demanding environments, such as factories, farms, and other outdoor environments. Typically, these devices communicate wirelessly with limited transmit/receive range. Therefore, edge networking devices need to meet a number of unique requirements, including the following:

- Supporting large numbers of end systems
- Operating in demanding and possibly remote environments
- Close proximity to supported IoT objects

The network connectivity component brings together a number of pre-existing and new products designed to support IoT. The product line includes reliable, scalable, high-performance networking solutions with a broad portfolio of routing, switching, and wireless products, available in ruggedized and nonruggedized form factors, as well as software only solutions that integrate into third-party devices. The product portfolio is organized into the following product categories:

- **Industrial switching:** A range of compact, ruggedized Ethernet switches that handle security, voice, and video traffic across industrial networks.
- **Industrial routing:** These products are certified to meet harsh environmental standards. They support a variety of communications interfaces, such as Ethernet, serial, cellular, WiMAX, and RF mesh.
- **Industrial wireless:** Designed for deployment in a variety of harsh or demanding environments. These products provide wireless access point functionality and implement Cisco VideoStream, which uses multicast encapsulated in unicast to improve multimedia applications.
- **Embedded networks:** Cisco Embedded Service switches are optimized for mobile and embedded networks that require switching capability in harsh environments.

Fog Computing:

The fog computing component of IoT System consists of software and hardware that extends IoT applications to the network edge, enabling data to be efficiently analyzed and managed where generated, thus reducing latency and bandwidth requirements. The goal of the fog computing component is to provide a platform for IoT-related apps to be deployed in routers, gateways, and other IoT devices. To host new and existing applications on fog nodes, Cisco provides a new software platform, called IOx, and an API for deploying applications on IOx. The IOx platform combines the Cisco IOS operating system and Linux (see Figure 10.18). Currently, IOx is implemented on Cisco routers.

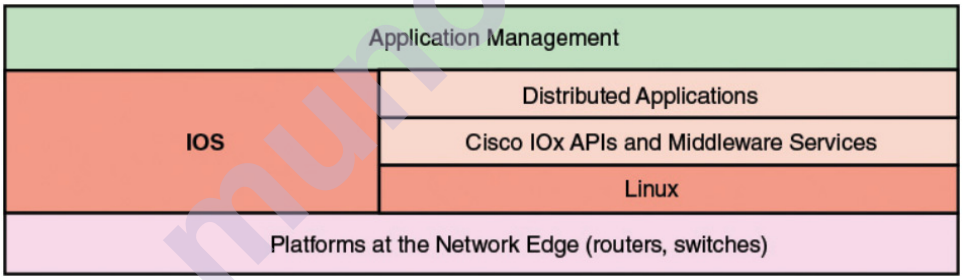


Figure 10.18: Cisco IOx

Cisco IOS (originally Internetwork Operating System) is software used on most Cisco Systems routers and current Cisco network switches. IOS is a package of routing, switching, internetworking, and telecommunications functions integrated into a multitasking operating system. This is not to be confused with Apple's iOS operating system that runs on iPhones and iPads. With IOS as a base, IOx combines the communication and computing resources that are required for IoT into a single platform for application enablement at the network edge. As Figure above shows, an IOx platform, such as a router, runs IOS and Linux in parallel, using the multitasking capability of the multicore processor. Linux is used as a base to support APIs and middleware services that

enable partner companies to implement fog applications on the IOx platform.

Data Analytics:

The data analytics component of IoT System consists of distributed network infrastructure elements and IoT-specific APIs that run business-specific software analytics packages throughout the network architecture — from the cloud to the fog — and that allow customers to feed IoT data intelligently into business analytics. The Cisco IoT analytics infrastructure includes the following:

- **Infrastructure for realtime analytics:** The integration of network, storage, and compute capabilities on select Cisco routers, switches, Unified Communications System (UCS) servers, and IP cameras allows analytics to run directly on fog nodes for real-time collection, storage, and analysis at the network edge.
- **Cloud to fog:** Cisco Fog Data Services includes APIs to apply business rules and control which data remains in the fog for real-time analytics and which is sent to the cloud for long-term storage and historical analysis.
- **Enterprise analytics integration:** Using IOx APIs, enterprises can run analytics on fog nodes for realtime intelligence. Fog Data Services allows IoT data exporting to the cloud. Integration of IoT data can increase operational efficiency, improve product quality, and lower costs.
- **Analytics for security:** Cisco IP cameras with storage and compute capabilities support video, audio, and data analytics at the network edge so enterprises gain real-time security intelligence, including event processing and classification.

Security:

The intent of the security component is to provide solutions from the cloud to the fog that address the full attack continuum—before, during, and after an attack. The component includes cloud-based threat protection, network and perimeter security, user- and group- based identity services, video analytics, and secure physical access. The security portfolio includes the following elements:

- **Cloud-based threat protection:** Provided by Cisco's Advanced Malware Protection (AMP) package. This is a broad spectrum of products that can be deployed on a variety of Cisco and third-party platforms. AMP products use big data analytics, a telemetry model, and global threat intelligence to help enable continuous malware detection and blocking, continuous analysis, and retrospective alerting.
- **Network and perimeter security:** Products include firewall and intrusion prevention systems.
- **User- and group- based identity services:** Products include an Identity Service Engine, which is a security policy management

platform that automates and enforces context-aware security access to network resources; and Cisco TrustSec technology, which uses software-defined segmentation to simplify the provisioning of network access, accelerate security operations, and consistently enforce policy anywhere in the network.

- **Physical security:** Cisco's physical security approach consists of hardware devices and software for security management. Products include video surveillance, IP camera technology, electronic access control, and incident response. Cisco physical security solutions can be integrated with other Cisco and partner technologies to provide a unified interface that delivers situational awareness and rapid, informed decisions.

Management and Automation:

The management and automation component is designed to provide simplified management of large IoT networks with support for multiple siloed functions, and to enable the convergence of OT data with the IT network. It includes the following elements:

- **IoT Field Network Director:** A software platform that provides a variety of tools for managing routers, switches, and endpoint devices. These tools include fault management, configuration management, accounting management, performance management, diagnostic and troubleshooting, and a northbound API for industry-specific applications.
- **Cisco Prime Management Portfolio:** A remote management and provisioning solution that provides visibility into the home network. The package discovers detailed information about all connected devices in the home and enables remote management.
- **Cisco Video Surveillance Manager:** Provides video, analytics and IoT sensor integration for providing physical security management.

Application Enablement Platform:

This component provides a platform for cloud-based app development and deployment from cloud to fog, simply and at scale. Also offers open APIs and app development environments for use by customers, partners, and third parties. It features the following elements:

- **Cisco IOx App Hosting:** With IOx capability, customers from all segments and solution providers across industries will be able to develop, manage, and run software applications directly on Cisco industrial networked devices, including hardened routers, switches, and IP video cameras.
- **Cisco Fog Director:** Allows central management of multiple applications running at the edge. This management platform gives administrators control of application settings and lifecycle, for easier access and visibility into large-scale IoT deployments.
- **Cisco IOx Middleware Services:** Middleware is the software "glue"

that helps programs and databases (which may be on different platforms) work together. Its most basic function is to enable communication between different pieces of software. This element provides tools necessary for IoT and cloud apps to communicate.

10.5.3 ioBridge:

IoBridge provides software, firmware, and web services designed to make it simple and cost-effective to Internet-enable devices and products for manufacturers, professionals and casual users. By providing all the components necessary to web-enable things, ioBridge's customers avoid the complexity and cost associated with piecing together solutions from multiple vendors. The ioBridge offering is essentially a turnkey solution for a broad range of IoT users.

ioBridge Platform:

IoBridge provides a complete end-to-end platform that is secure, private, and scalable for everything from do-it-yourself (DIY) home projects to commercial products and professional applications. ioBridge is both a hardware and cloud services provider. The IoT platform enables the user to create the control and monitoring applications using scalable Web technologies. ioBridge features end-to-end security, real-time I/O streaming to web and mobile apps, and easy-to-install and easy-to-use products. Figure 10.19 illustrates some of the major features of ioBridge's technology. The tight integration between the embedded devices and the cloud services enable many of the features shown in the diagram that are not possible with traditional web server technology. Note that the off-the-shelf ioBridge embedded modules also include web-programmable control or "rules and actions." This enables the ioBridge embedded module to control devices even when it is not connected to the ioBridge cloud server.

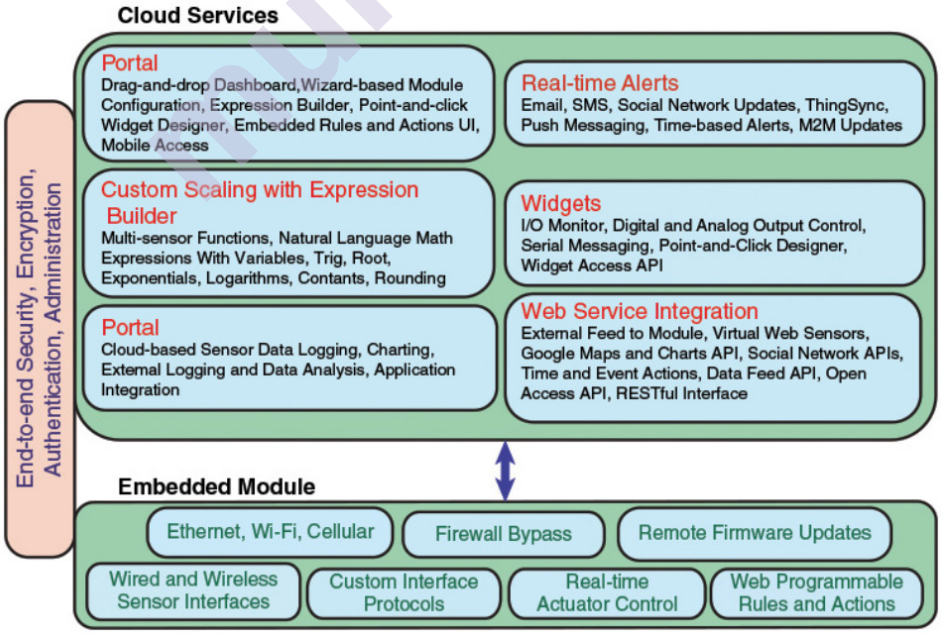


Figure 10.19: The ioBridge Internet of Things Platform

The major offerings on the device side are firmware, Iota modules, and gateways. Firmware is added where possible to devices to add the functionality to communicate with ioBridge services. Iotas are tiny-embedded firmware or hardware modules with either Ethernet or Wi-Fi network connectivity. Gateways are small devices that can act as protocol converters and bridges between IoT devices and ioBridge services. In essence, the IoT platform provides a seamless mashup of embedded devices with web services. IoBridge markets hardware boards, firmware, and software that can be installed in embedded devices together with apps that can run on platforms such as smartphones and tablets, as well as web services.

ThingSpeak:

ThingSpeak is an open source IoT platform developed by ioBridge. ThingSpeak enables the creation of sensor logging applications, location-tracking applications, and a social network of things with status updates. It offers the capabilities of real-time data collection, visualizing the collected data in the form of charts, the ability to create plug-ins and apps for collaborating with web services, social networks, and other APIs.

The basic element of ThingSpeak is a ThingSpeak channel, which is hosted on the ThingSpeak website. A channel stores data sent to ThingSpeak and consists of the following elements:

- **Eight fields for storing data of any type:** These can be used to store the data from a sensor or from an embedded device.
- **Three location fields:** Can be used to store the latitude, longitude and the elevation. These are very useful for tracking a moving device.
- **One status field:** A short message to describe the data stored in the channel.

IoBridge-enabled devices and platforms with ioBridge apps can communicate via a channel. A ThingSpeak channel can also connect with Twitter so that sensor updates and other data can be communicated via tweet. Note that ThingSpeak is not limited to ioBridge devices; it can work with any device that includes the software necessary to communicate via a ThingSpeak channel. A user begins by defining a channel on the ThingSpeak website. This is an easy interactive process that includes the following steps:

1. Create new channel with unique ID.
2. Specify whether the channel will be public (open to view by anyone) or private.
3. Create from one to eight fields, which can hold any type of data, giving each field a name.
4. Create API keys. A channel has one write API key. Any data communicated to the channel will only be written into one or more

fields if the data is accompanied by the API key. A channel may have multiple read API keys. If the channel is private, data can only be read by presenting the API key. A user can define an app to an API key to perform some sort of data processing or directing.

ThingSpeak provides apps that allow for an easier integration with web services, social networks, and other APIs. Some of the apps provided by ThingSpeak are the following:

- **ThingTweet:** Allows the user to post messages to twitter via ThingSpeak. In essence, this is a TwitterProxy which redirects your posts to Twitter.
- **ThingHTTP:** Allows the user to connect to web services and supports GET, PUT, POST, and DELETE methods of HTTP.
- **TweetControl:** Enables user to monitor Twitter feeds for a specific keyword and then process the request. Once the specific keyword is found in the Twitter feed, the user can then use ThingHTTP to connect to a different web service or execute a specific action.
- **React:** Sends a tweet or trigger a ThingHTTP request when the channel meets a certain condition.
- **TalkBack:** Queues up commands and then allows a device to act upon these queued commands.
- **TimeControl:** Can perform a ThingTweet, ThingHTTP, or a TalkBack at a specified time in the future. Can also be used to allow these actions to happen at a specified time throughout the week.

In addition to the listed apps, ThingSpeak allows users to create ThingSpeak applications as plug-ins using HTML, CSS, and JavaScript, which can be embedded inside a website or inside a ThingSpeak channel.

RealTime.io:

Another offering of ioBridge is RealTime.io. This technology is similar to, but more powerful and sophisticated than, ThingSpeak. RealTime.io is a cloud platform that enables any device to connect to cloud services and mobile phones to provide control, alerts, data analytics, customer insights, remote maintenance, and feature selection. The intent is that product manufacturers that leverage ioBridge's technology will be able to quickly and securely bring new connected home products to market while slashing their cost-per-connected device.

The RealTime.io App Builder allows the user to build web apps directly on the RealTime.io cloud platform. The user can write web applications based on HTML5, CSS, and JavaScript and create interactions with devices, social networks, external APIs, and ioBridge web services. There is an in-browser code editor, JavaScript library, app update tracking,

device manager, and single sign on with existing ioBridge user accounts. RealTime.io natively works with ioBridge Iota-based devices and firmware. RealTime.io has built-in template apps or custom apps. Template apps are prebuilt apps that the user can start with and then customize. Custom apps allow the user to upload their own files and images without any starter templates. Figure 10.20 shows the overall ioBridge environment.

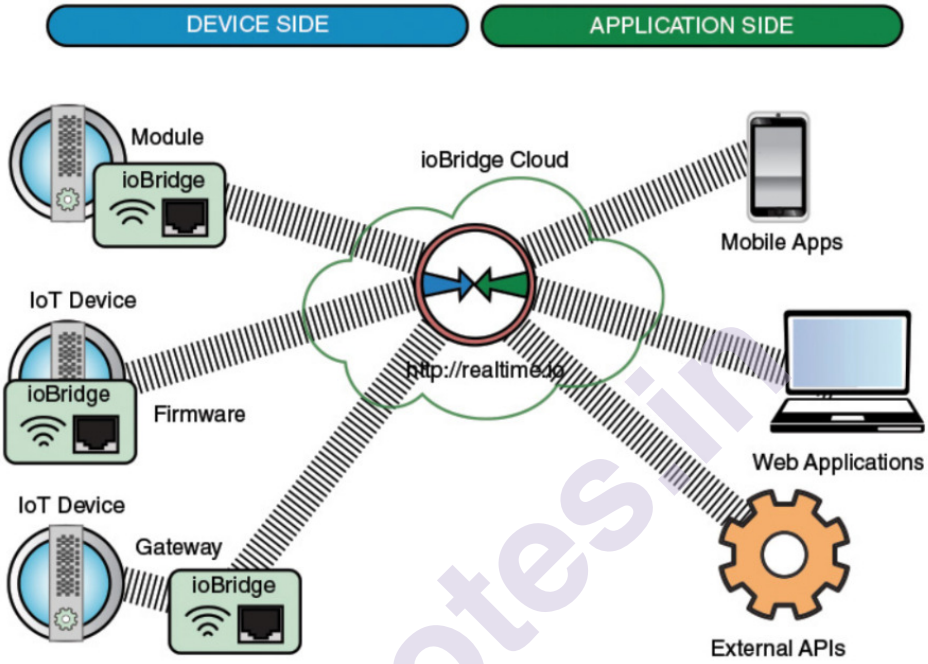


Figure 7.20: ioBridge environment

SUMMARY

- Y.2060 characterizes the IoT as adding the dimension "Any THING communication" to the information and communication technologies which already provide "any TIME" and "any PLACE" communication.
- In Designing the Internet of Things, the author condenses the elements of the IoT into a simple equation: Physical objects + Controllers, Sensors, Actuators + Internet = IoT
- The key ingredients of an IoT-enabled thing are sensors, actuators, a microcontroller, a means of communication (transceiver), and a means of identification (radio-frequency identification [RFID]).
- The ITU-T IoT reference model is defined in Y.2060, Overview of the Internet of Things, June 2012.
- Gateways address one of the greatest challenges in designing for IoT, which is connectivity, both among devices and between devices and the Internet or enterprise network.
- The IoT World Forum Architecture Committee, made up of industry

leaders including IBM, Intel, and Cisco, released an IoT reference model in October 2014. This model serves as a common framework to help the industry accelerate IoT deployments.

- IoTivity is an open-source software initiative. Their objective is to provide a standard and open-source implementation so devices and services will be able to work together regardless of who makes them.
- The term constrained device refers to a device with limited volatile and nonvolatile memory, limited processing power, and a low data rate transceiver. The term unconstrained device simply refers to any device without severe resources constraints.
- In 2015, Cisco introduced a suite of integrated and coordinated products known as the Cisco IoT System.
- Cisco IoT System addresses the complexity of digitization with an infrastructure that is designed to manage large-scale systems of diverse endpoints and platforms, and the data deluge they create.
- IoBridge provides software, firmware, and web services designed to make it simple and cost-effective to Internet-enable devices and products for manufacturers, professionals and casual users.
- ThingSpeak is an open source IoT platform developed by ioBridge. ThingSpeak enables the creation of sensor logging applications, location-tracking applications, and a social network of things with status updates.
- RealTime.io is a cloud platform that enables any device to connect to cloud services and mobile phones to provide control, alerts, data analytics, customer insights, remote maintenance, and feature selection.

10.7 UNIT END QUESTIONS

1. Explain the scope of the Internet of Things.
2. List and discuss the five principal components of IoT-enabled things.
3. Compare and contrast the ITU-T and IoT World Forum IoT reference models.
4. Describe the open source IoTivity IoT implementation.
5. Describe the commercial ioBridge IoT implementation.
6. Write a short note on:
   a. Sensors
   b. RFID
   c. ITU-T IoT Reference Model
   d. Fog Computing
   e. IoT World Forum Reference Model
   f. Cisco IoT System
   g. ThingSpeak

10.8 BIBLIOGRAPHY, REFERENCES AND FURTHER READING

- Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings, Addison Wesley Professional
- SDN and NFV Simplified A Visual Guide to Understanding Software Defined Networks and Network Function Virtualization by Jim Doherty, Pearson Education
- Network Functions Virtualization (NFV) with a Touch of SDN by Rajendra Chayapathi, Syed Farrukh Hassan, Addison Wesley
- CCIE and CCDE Evolving Technologies Study Guide by Brad Dodgeworth, Jason Gooley, Ramiro Garza Rios, Pearson Education
- Y.2060, Overview of the Internet of Things, June 2012.
- Y.2067, Common Requirements and Capabilities of a Gateway for Internet of Things Applications, June 2014
- The Internet of Things Reference Model. White paper, 2014. http://www.iotwf.com/.
- Building the Internet of Things. Presentation, 2014. http://www.iotwf.com/.
- Cisco IoT System: Deploy, Accelerate, Innovate. Cisco white paper, 2015.
- Ferguson, J., and Redish, A. "Wireless Communication with Implanted Medical Devices Using the Conductive Properties of the Body." Expert Review of Medical Devices, Vol. 6, No. 4, 2011. http://www.expert-reviews.com.
- Seghal, A., et al. "Management of Resource Constrained Devices in the Internet of Things." IEEE Communications Magazine, December 2012.
- Vaquero, L., and Rodero-Merino, L. "Finding Your Way in the Fog: Towards a Comprehensive Definition of Fog Computing." ACM SIGCOMM Computer Communication Review, October 2014.
- Krakowiak, S. Middleware Architecture with Patterns and Frameworks. 2009.http://sardes.inrialpes.fr/%7Ekrakowia/MW-Book/
- McEwen, A., and Cassimally, H. Designing the Internet of Things. New York: Wiley, 2013.
- Scherz, P., and Monk, S. Practical Electronics for Inventors. New York: McGraw-Hill, 2013.
- Stankovic, J. "Research Directions for the Internet of Things." Internet of Things Journal, Vol. 1, No. 1, 2014.

*****