COURSE BOOK



## Networks and Distributed Systems

DLMCSNDS01



Learning Objectives

##### Introduction 9



Computer networks are classiﬁed in different categories based on their size, architecture, and services. Despite all differences, a computer network is basically a collection of computing devices that communicate through the transmission of digital data. The digital data are con- veyed as digital signals that propagate through different types of physical media. The course book Networks and Distributed Systems will introduce the types of computer networks, con- cepts of digital transmission, basics of communication engineering, and various types of physical layers.

The data transmission between sending and receiving computers follows a sequence. The process is divided into several layers that provide different functionalities and services. These functionalities and services are standardized by layer protocols. You will be introduced to network layers, and services and protocols of each layer.

The internet is the largest type of computer network and includes more than ﬁve billion devices worldwide. You will learn about the Transmission Control Protocol/Internet Protocol reference model and protocol stack. As the internet grows in terms of services and number of users, the security concerns and vulnerabilities also increase proportionately. The security aspects of the internet will also be introduced.

Different types of distributed architectures of computer networks will be introduced. The algorithms and protocols for various types of distributed architectures will be discussed. Finally, a number of cutting-edge technologies are growing rapidly in the computer network- ing ﬁeld such as the Internet of Things (IoT), pervasive computing, mobile computing, ubiqui- tous computing, and distributed ledger technologies. You will be introduced with these con- temporary topics as well.

[www.iubh.de](http://www.iubh.de/)



# Unit 1

## Computer Networks

#### STUDY GOALS

On completion of this unit, you will be able to …

… explain the concepts of computer networking, types of computer networks, network topologies, and interconnections.

… understand the concepts of digital data transmission, communication engineering, and coding theory.

… apply the concepts of physical layer mediums and data transmission methods.

… estimate, analyze, and evaluate network performance.

DL-E-DLMCSNDS01-U01

1. Computer Networks

### Introduction

In our modern world, almost every aspect of life is moving toward dependency on com- puter networks (e.g., internet), including education, healthcare, security, daily transac- tions, banking, entertainment, and communication. There are various types of networks with different architectures, spatial scales, and organizational scopes and services. Regardless of the types and services, the underlying task of a computer network is to transmit data from one device to another, a process known as data communication. Data communication involves several steps, such as converting analog information to digital data, data encoding, compression, encryption, data to signal conversion, modu- lation, signal transmission, and propagation. The quality of service (QoS) of a computer network depends on the performance of data communication, which can be measured in terms of delay, throughput, and bit-error rate.

### Basic Concepts of Digital Data Transmission

A computer network is a collection of communicating computing devices. This commu- nication is done by transmitting digital data between sending and receiving devices through a series of intermediary connecting devices, and wired or wireless physical media. A network device is typically known as a node and the physical medium is called a link (Forouzan, 2013). The communication between nodes includes a series of steps, set of rules or mechanisms. These mechanisms are standardized by governing bodies and known as protocols, which give the following ﬁve key components of digital data transmission (Forouzan, 2013):

1. The “message” is the data or information to be conveyed between the communicat- ing devices.
2. The “sender” is the device that transmits the message.
3. The “receiver” is the device that receives the message.
4. The “link” is the physical medium of communication through which the message is conveyed from sender to receiver.
5. “Protocols” are standardized mechanisms that assist the communication.

###### Digital Data Transmission Steps

A computer is a digital device that stores data as a collection of binary numbers. A pair of communicating nodes basically exchanges a collection of zeros and ones. The data mostly propagate through the link in the form of electric or electromagnetic waves referred to as the signal. Because of this, there should be a mechanism to encode digi- tal data into an electric or electromagnetic waveform called the digital signal. This mechanism is known as line coding or digital baseband modulation. However, different nodes are required to transmit the signal at different frequency based on the type of the physical media. To do so, the digital signal is modulated using an analog signal

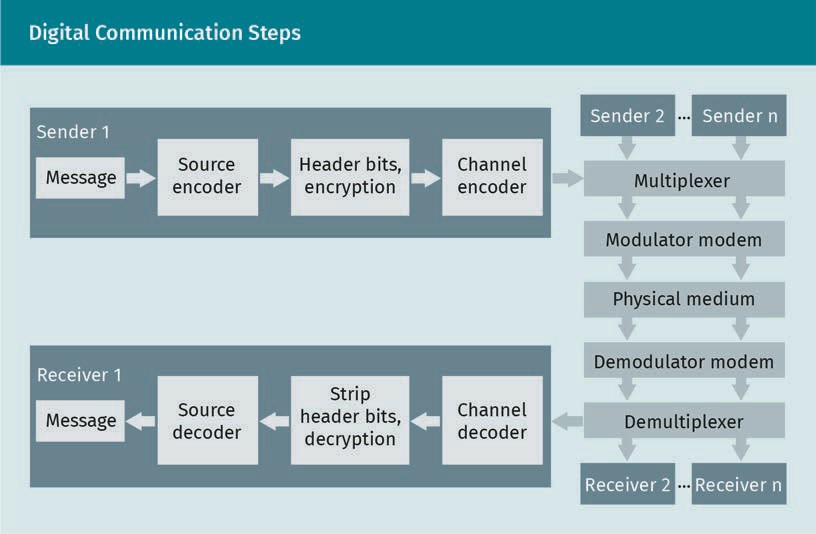
Computer Networks

with the desired frequency. This signal is known as the carrier. The device that performs modulation and demodulation is called the modem. The word “modem” is a short form of modulator-demodulator and applies to, for example, the digital subscriber line (DSL) modem or cable modem (Tanenbaum & Wetherall, 2014). DSL modems connect com- puters or routers with access to telephone lines to the broadband internet, while a cable modem provides broadband internet access to a network node through cable TV connections.

In a computer system, the message or information must be presented digitally. The conversion of the analog information to digital data is done through a technique called source coding (Comer, 2015). When a signal propagates through the physical medium, it experiences attenuation due to path loss, fading, or transmission loss. The signal can also become distorted due to noise, interference, multipath effect, or shadowing. Attenuation and distortion can ﬂip a zero to one or vice-versa, which is known as a bit- error (Kurose & Keith, 2017). Communicating nodes use channel encoding as an error detection and correction mechanism to ﬁnd and ﬁx bit-errors (Comer, 2015). The com- municating nodes include an encryption-decryption mechanism to secure the trans- mitted data from possible intruders.

The sending node requires some time to transmit data. This time is called transmission delay, and is proportional to the size of data (Kurose & Keith, 2017). The sender uses a mechanism called source encoding to compress the data before converting it into a signal (Comer, 2015). After data compression, the sending node needs to add in some “extra bits” alongside the actual data bits. These extra bits are called “header bits,” which contain the communicating parties' addresses, and the data bits are the “mes- sage.”

A network link can have multiple channels, meaning that a physical medium can carry transmissions from multiple senders simultaneously. Multiplexing is the mechanism of combining transmissions from multiple sources. Demultiplexing is the process of sepa- rating multiple transmission from a combined signal (Comer, 2015; Kurose & Keith, 2017).



###### Digital Communication Performance

The overall performance of a digital communication system or computer network is known as quality of service (QoS). The QoS largely depends on transmission rate, delay, channel capacity, throughput, bit error-rate (BER), among other factors, which are described as follows:

* “Transmission rate” is the number of bits per second a sender transmits or releases at the physical medium.
* “Delay” is the time to send a message from sender to the receiver.
* “Transmission delay” is the time a sender takes to transmit or release a bit.
* “Channel capacity” is the maximum number of bits a link can carry per second. In general, the sender transmits data at the rate equal to the channel capacity.
* “Link utilization” is the ratio of the number of bits a link is currently carrying and the channel capacity. Suppose a link capacity is two megabits per second (Mbps) and a sender is transmitting at one Mbps, then the link utilization is 50 percent. If the transmission rate is two Mbps, then the link utilization is 100 percent.
* “Throughput” is the number of bits per second successfully received by the receiver(s). For example, if the channel capacity is two Mbps and the transmission rate is one Mbps, then the throughput is one Mbps. If the transmission rate is

Computer Networks

increased to two Mbps, then the throughput is two Mbps. Now, if the transmission rate is further increased to three Mpbs, then the throughput is still two Mbps, because the channel is not able to handle more than two Mbps.

* “Bit error-rate” (BER) is the ratio of number of bit-errors and number of total trans- mitted bit. If the number of total transmitted bit is 10,000 and the number of bit- error is ten, then the BER is 0.001 or 10-3.

### Network Topologies and Interconnections

Before beginning construction, an engineer must plan the layout of the structure. Simi- larly, a network engineer needs to make a layout of the network before deploying the network links and nodes, i.e., devices. This layout is known as network topology. A net- work topology can be physical or logical. A physical topology is a physical mapping of the network in which the depicted locations of the network nodes are scaled in accord- ance with the physical measurements and properties of the network entities. In con- trast, a logical network is a logical mapping of the network that depicts how the net- work nodes are connected to each other without considering the actual physical locations of the node. A physical topology is important for designing, deploying, and analyzing a speciﬁc network, while a logical topology is important for studying and analyzing a generic network.

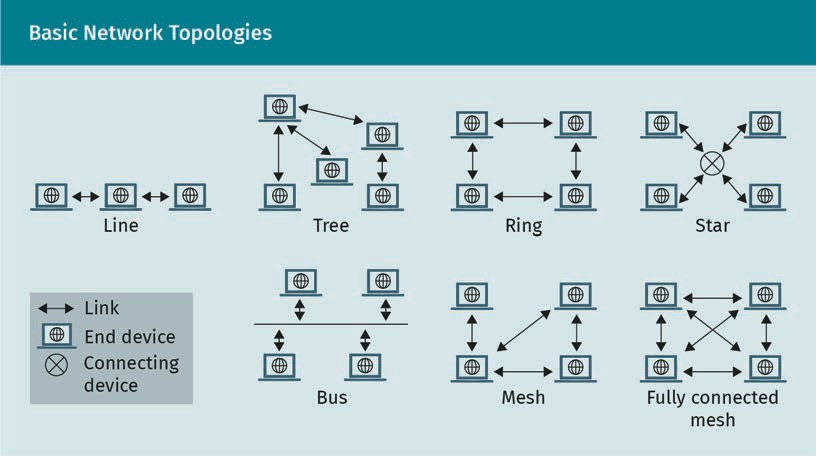
###### Network Topologies

The simplest network topology contains only a sender and a receiver. The topology becomes intricate as the number of network nodes increases. Networks with the same resources may each produce a different quality of service (QoS) due to the differences in topologies. Well-known network topologies are as follows:

* In a “bus” topology, all end nodes are connected to a common line or bus known as backbone.
* “Line nodes” of a line topology are connected to one after another in a serial man- ner. There are two end nodes in a line topology, each of them has only one connec- tion, whereas all other nodes in a line topology have two connections.
* A “ring” topology is similar to a line topology, with the exception that there are no end nodes in a ring topology.
* A “star” is the point at which all end nodes are connected to a common connecting node.
* A “mesh” occurs where the nodes are connected to each other in a random manner. A mesh network where all the nodes are connected to each other directly is known as fully connected mesh network.
* A “tree” combines a bus and star topology; thus, another name for a tree topology is star-bus topology.
* A “hybrid” topology is formed by combining other topologies. In practice, the hybrid topology is the most common topology for large networks.

Topology

Network topology is a logical or physical depiction of how the network nodes, i.e., devices, are connec- ted.



All basic topologies have their own advantages and disadvantages. A particular topol- ogy can be useful for a particular network based on the purpose of the network, QoS requirements, and resource availability.

Bus topology

The bus topology has the simplest architecture. It requires only one bus or cable to connect all end nodes and does not require any connecting device or node, meaning that the installation cost is inexpensive. Furthermore, installing an additional device is comparatively easy. The failure of a node does not hamper the connection between other nodes. However, because the backbone is shared by all nodes, a failure of the backbone will cause the entire network to collapse. The performance of nodes can also be affected by the addition of more nodes.

Line topology

A network with line topology consisting of n nodes requires n − 1 links. Unlike the bus network, data may travel through multiple repeaters, because nodes are connected through other nodes. The failure of a node breaks the network into two disconnected line networks. However, the failure of an end node does not impact the connection between other nodes. In a line topology, installing new devices is not quite as easy as in the bus topology.

Ring topology

Each node of a ring topology has two connections. If the end nodes of a line topology are connected to each other, then the line topology becomes a ring topology. Consid- ered from a different perspective, the failure of a node turns a ring topology into a line topology. The number of nodes and links are equal in a ring network. Similar to the line topology, data may travel through multiple connecting nodes. Ring topology allows the data to travel bidirectionally, meaning that if a single node is malfunctioning, other nodes can still be connected.

Computer Networks

Star topology

Each pair of communicating end devices has two links between them; the ﬁrst link con- nects the sender to the connecting device, and the second link connects the connect- ing device to the receiver. The failure of the connecting central node hampers the con- nection of all other nodes, thus breaking the entire network. Network extension is relatively easy for a star topology compared to the other types. All nodes of two sepa- rate star networks can be brought together simply by connecting the central connecting devices. A star network with n end nodes requires n connections or links.

Tree topology

In a tree topology, the nodes are called vertices, and links are known as edges. A tree network with n nodes requires n − 1 links. As with the start and bus topologies, net- work expansion is considered easy in a tree topology. The primary disadvantage of the tree topology is that the failure of a parent node may create multiple disconnected net- works.

Mesh topology

In this topology, multiple paths exist between two nodes, thus the failure of a node does not break the connection between other nodes. If there are n nodes, then a fully connected mesh network requires n n − 1 /2 links. A fully connected mesh network provides highest fall tolerance. However, installation costs can be high for a large amount of nodes n.

###### Interconnections

A computer network may consist of anywhere from a few devices (nodes) to a billion devices depending on network types. The network nodes are categorized into two types: end and connecting nodes. Application programs run on end nodes, for example, laptops, mobile phones, and servers. When two end nodes communicate, it means that the two application programs running on the two end nodes are communicating. Con- versely, connecting nodes are not capable of running application programs. Connecting nodes are designed to connect the end nodes within a network and can connect devi- ces from two different networks.

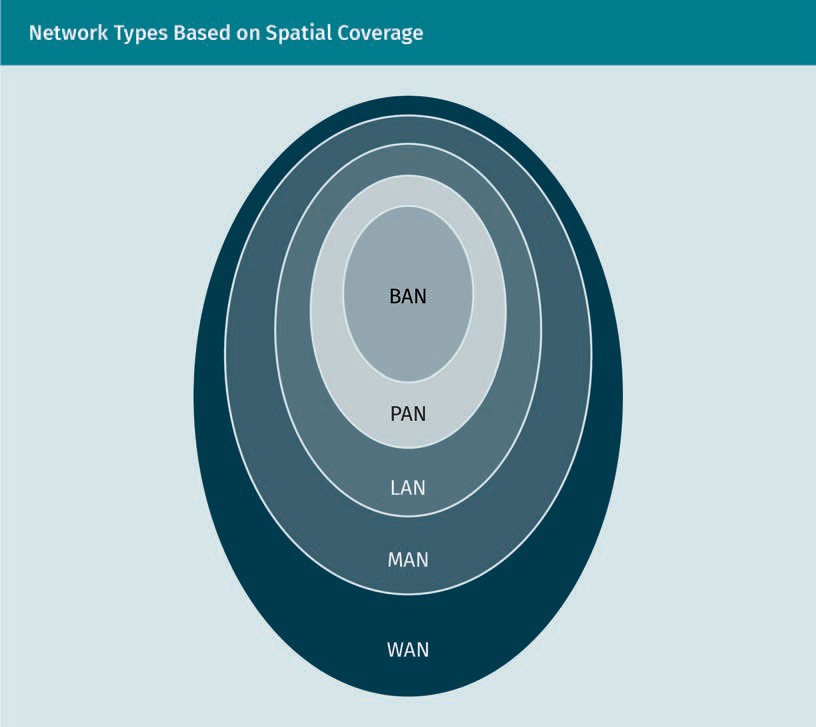
Network types

Based on the spatial size or coverage, networks are divided into many categories. Some of the basic categories are listed here:

* Body area network (BAN) nodes are located on the human body.
* Personal area network (PAN) covers individuals’ workplaces, typically a room.
* A local area network (LAN) generally covers a home or ofﬁce.
* The metropolitan area network (MAN) includes a city or a metropolitan area.
* Wide area network (WAN) provides the largest coverage, which includes the global area. The internet is the most common example of WAN.

Based on the service type, networks can be categorized in the following types:

* + Intranet is a private network that serves network facilities within an organization.
  + Extranet is an extension of intranet, whereby a limited number of authorized outsid- ers are allowed to access an organization’s private network. The outsiders are usu- ally the organization’s partners, suppliers, vendors, or service providers.
  + The internet is a global platform of interconnected networks. Nodes from different networks can communicate through the internet, creating a network of countless networks.



Connecting device The connecting device (i.e., router) between two net- works is known as a gateway (Kurose &

Keith, 2017).

Connecting devices

A connecting device has multiple input and output ports. When it receives a packet, the device transfers it to the relevant outgoing port in order to deliver it to the desired receiver. For this reason, connecting devices are also known as packet switches. There are several types of connecting devices, the most common of which are repeater, bridge, hub, switch, and router. A repeater is used to boost the energy level of the incoming signal. A bridge is a repeater with a ﬁlter for incoming packets. A hub allows multiple end nodes to be connected to a common network. Like a hub, a switch also connects multiple devices with the capability of ﬁltering incoming packets based on the media access control (MAC) addresses (Kurose & Keith, 2017).

Computer Networks

###### Delays

A data packet travelling through a connecting device can experience four types of delays (Newpaltz, n.d.). Precise estimation of delay is important to design network parameters in order to ensure the desired QoS.

Transmission delay

Transmission delay is the time required by a node to transmit a data packet. Using packet length L and the data transmission rate R, then the propagation delay can be calculated using the following formula:

Tt’ = L

R

Transmission delay for n-hop connection is nL/R.

Propagation delay

Once a packet is transmitted, it propagates through the physical medium toward the receiver. The propagation time is known as propagation delay, and is proportional to the distance between the sender and receiver. Using the distance ( and propagation speed ), the propagation delay can be expressed as follows:

Tp’ = (

)

n-hop

An n-hop connection has *(n-1)* connecting devices and *n* links.

For a wireless medium, the speed is equal to the speed of light.

Processing delay

Once the packet arrives at the incoming port of a connecting device, the packet is pro- cessed in order to forward it to the destination through the proper output port. This processing time is known as processing delay, and is dependent on the processing speed and packet size. Using the processing speed + and the packet size L, the for- mula for calculating processing delay is

Tp,’ = L

+

Queuing delay

When a packet arrives at the connecting device, the packet must wait at the buffer queue until all prior packets have been processed. This wait time at the buffer queue is known as the queuing delay. This is the most challenging delay to estimate, though the average queuing delay can be estimated using the formula below. Using the packet arrival rate at the buffer > and the service rate ., the estimated average queuing delay is

T = 1

/’

. − >

The total delay at the connecting node can be calculated using the following formula.

T = Tt’ + Tp’ + T,p’ + T/’

### Basics of Communication Engineering and Coding Theory

Computers are digital systems, and in a computer network, data are stored, presented, and transmitted digitally. However, the transmitted data propagate as an analog signal through the physical media. To understand the digital communication, it is important to understand the properties of digital and analog signals.

###### Types of Signals

A signal is a function that provides information about a physical phenomenon. In com- munication engineering, signal is a time varying voltage, current, or electromagnetic wave which conveys information. In general, the voltage (0) is the dependent variable of a signal and time (t) is the independent variable. Signals can be classiﬁed in differ- ent ways based on particular characteristics (Monolithic Power Systems (MPS), n.d.).

Continuous and discrete signals

Continuous and discrete signals are also known as continuous time and discrete time signals. In a continuous signal, the independent variable time (t) is measured continu- ously. If the signal is discrete, the independent variable time (t) takes only discrete val- ues.

Periodic and aperiodic signals

If the signal amplitude repeats after a certain period of time, it is referred to as a peri- odic signal. In contrast, the amplitude of an aperiodic signal does not repeat periodi- cally. A continuous periodic signal can be expressed as 0 t = 0 t + 1 · T and a discrete periodic signal can be expressed as 0 n = 0 n + 1 · 2 ; where 1 = 1, 2, 3, ….

Analog and digital signals

In an analog signal, the value of the dependent variable is continuous which means the signal amplitude can be any real number. In contrast, a digital signal amplitude can be only ﬁxed discrete values from a predeﬁned set. In general, discrete signals are dis- crete in both time and amplitude.

Composite signal, modulating signal, carrier, and modulated signal

A composite signal is formed by combining multiple sine waves. A modulating signal is a digital signal that represents the message or data to be transmitted. The modulating signal is combined with a pure sinusoid (i.e., sine wave) with the desired frequency to produce the modulated signal. The sinusoid is called the carrier signal or carrier.

Computer Networks

Even and odd signals

A signal 0 t is an even signal if it holds the property 0 t = 0 −t . In contrast, an odd signal shows the characteristic v(t) = -v(-t).

###### Coding Theory

In digital communication, coding can be used for several purposes such as data com- pression, encryption, error detection and correction, analog to digital signal conversion, and data transmission. The coding technique used to convert analog to digital signal is called source coding. Channel coding is a technique by which redundant bits are added to the data bits in order to detect and correct bit-errors. The coding technique used for data transmission, i.e., converting digital data to digital signal, is called line coding.

Source coding

Source coding converts a data or signal to bit stream. The simplest example of source coding is pulse code modulation (PCM). PCM converts an analog signal to digital bit stream through the following three steps:

1. “Sampling” reads the analog signal amplitude periodically. The idea is to convert the continuous-time signal into discrete-time signal.
2. “Quantizing” converts the sample values into integer numbers based on a prede- ﬁned set of integer values. The idea is to convert the analog amplitude values into discrete or digital values.
3. “Encoding” represents quantized values into binary numbers. If the quantization has four levels then the possible quantized values are zero, one, two, and three. The equivalent binary values are 00, 01, 10, and 11. If there are 2 quantization levels then it would take log2 2  bits to represent a quantized value.

###### Important Terminologies and Concepts

Bandwidth

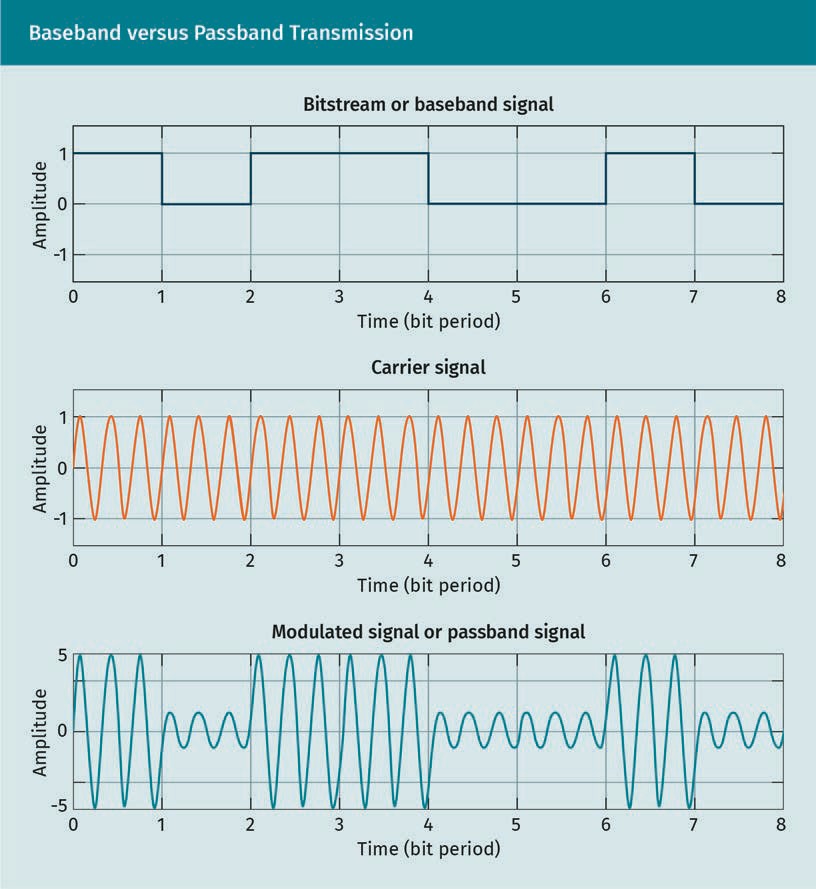
In digital communication, the transmitted signal is a composite signal, which is made of multiple sinusoids. The range of the frequencies of the sinusoids starting from the lowest to the highest is known as frequency bandwidth or bandwidth of the transmit- ted signal. The unit of the frequency bandwidth is hertz 34. In computer networking, bandwidth is the capacity of the transmission link to carry the maximum number of bits per second. Network bandwidth is also known as digital bandwidth.

Baseband and passband signals

Baseband is the bandwidth of the transmitted signal before it is modulated using a carrier. In other words, baseband is the bandwidth of the digital signal of the informa- tion signal. Passband is the bandwidth of the transmitted composite signal after analog modulation.

Baseband

This type of trans- mission is used in ethernet LANs. Wi-Fi uses passband transmission.



Noise

Noise is an unwanted signal that interferes with the desired signal. Noise is an indeter- ministic signal, meaning the amplitude of the noise signal is indeterministic for a given time. Additive white Gaussian noise (AWGN) is the simplest noise modeling for digital communication. White noise is a noise that has equal intensity for all frequencies. AWGN follows Gaussian distribution.

Shannon-Hartley theorem

According to the Shannon-Hartley theorem, if 5 is the passband bandwidth, then the channel capacity (6) in bits per second can be determined by the following equation:

6 = 5log2 1 + )/2 = 5log2 1 + )2R

Computer Networks

where ) is the signal power and 2 is the noise power. The term SNR is known as the signal-to-noise ratio, usually expressed in decibels ’5 (Tanenbaum & Wetherall, 2014).

Nyquist theorem

The Nyquist theorem indicates how frequently an analog signal should be sampled to convert it into digital signal. According to the Nyquist theorem, the sampling rate must be at least twice the maximum frequency of the composite signal (Tanenbaum & Wetherall, 2014).

Transmission modes

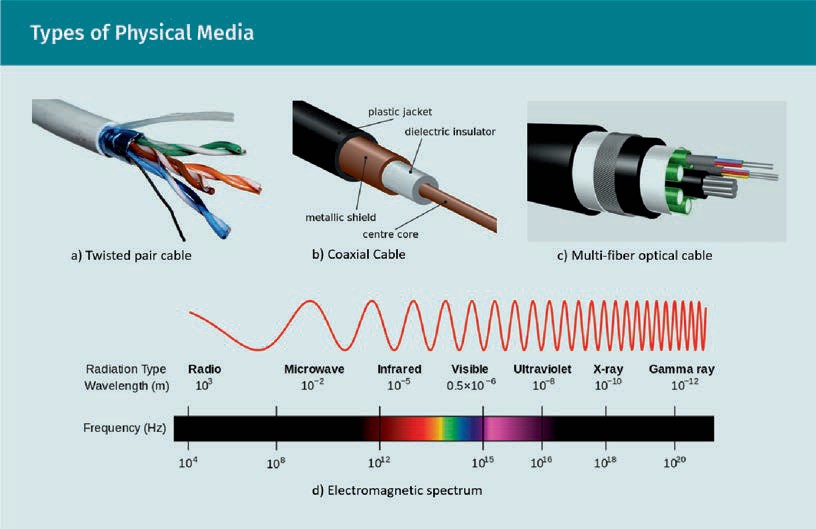
A system that supports simultaneous bidirectional data transmission is referred to as full-duplex. If it supports only unidirectional data transmission at any given time, then it is half-duplex. A simplex system supports data transmission in one direction only (Forouzan, 2013).

### The Physical Layer: Transmission Methods and Media

In a digital system, all data are presented using binary numbers, meaning that when two devices exchange data, they are essentially exchanging a collection of zeros and ones. Two communicating devices can be connected through a guided medium or through an unguided medium, known as a physical layer. When data propagate through the physical layer, they propagate in the form of energy. Based on the characteristics of the medium, the transmitted data can be presented in various ways. The quality of service (QoS) of the network is also directly impacted by the physical layer characteris- tics such as propagation speed, attenuation, and noise. The QoS includes delay, jitter, bit error rare, channel capacity, and throughput.

###### Physical Layer Types

When a packet travels from sender to the receiver, it may travel through a series of dif- ferent physical media. Physical layer media are categorized into two types: guided and unguided media. Guided media primarily refers to twisted pair copper cable, coaxial cable, and optical ﬁber. Unguided media commonly includes free space in which data propagates as electromagnetic waves of various ranges, such as radio wave, microwave, and infrared. Installing guided media is generally less expensive than using unguided media. An illustration of different types of physical media are presented below.



Twisted pair cable

Twisted pair cables are made of isolated copper wires, meaning the data are transmit- ted in the form of electrical signal or voltage wave. One wire transmits the voltage wave, while the other wire is used as a ground. The voltage difference between the two wires then represents data, i.e., either zero or one. Copper wires can be affected by external noise sources, such as heat or electromagnetic waves. If the noise source affects the wires unequally, then the voltage difference of the wires can output incor- rect data. To avoid this, the wires are twisted together in a spiral form, so that noise sources will affect them evenly. Twisted pair cables can be shielded or unshielded. Shielded twisted pair cables are covered with a metal layer to reduce electromagnetic noise. This type of cable is commonly used for LAN. In general, the data rate varies from 100 megabit per second (Mbps) to 1 gigabit per second (Gbps) with a bit error rate (BER) of 10-7 to 10-9. One downside to these cables is that they are relatively easy to tap (Solomon & Kim, 2021; Tanenbaum & Wetherall, 2014).

Coaxial cable

The construction of the coaxial cable differentiates it from the twisted pair cable. The core of a coaxial cable is essentially an insulated copper wire, which carries the signal. A metal foil is wrapped around the core, serving as the ground, and a metal sheath to protect from external noise. Coaxial cable provides comparatively higher frequency bandwidth; however, it also provides higher attenuation, which requires repeaters to boost up the signal energy for long distance communication. Coaxial cable is widely used by telephone, television, and internet connection providers. The data rate is up to 1 Gbps with a BER of 10-9 (Solomon & Kim, 2021; Tanenbaum & Wetherall, 2014).

Computer Networks

Optical ﬁber

Data propagates as optical signals in the optical ﬁber medium. The structure of optical ﬁber is similar to coaxial cable; instead of metal, it is made of ﬂexible glass. The core is coated to give it a lower refractive index, directing the light toward the core. Optical ﬁber is lightweight and offers very low attenuation and BER within the range of 10-11 to 10-13. It also provides a very high data rate of up to a few hundred Gbps. Furthermore, it is immune to electromagnetic noise and hard to tap. Optical ﬁber is suitable for long distance communication, which is why it is widely used as the backbone of internet and telephone networks. Some drawbacks of optical ﬁber are that it can be damaged easily, requires higher maintenance, and is expensive to install (Solomon & Kim, 2021; Tanenbaum & Wetherall, 2014).

Radio wave

Radio wave communications primarily use electromagnetic waves to transmit data ranging from three kilohertz (kHz) to 300 gigahertz (GHz). Waves with three kHz to one GHz frequencies are known as radio waves. One of the key characteristics of the radio wave is that it propagates through wireless media, i.e., free space, omnidirectionally. A radio wave with low frequency (i.e., three kHz to three megahertz (MHz)) is known as ground wave because, due to the diffraction, its propagation follows the contour of the ground or earth. The radio wave within three to 30 MHz range is known as skywave since it gets absorbed by the ground, but gets refracted by the ionosphere and comes back to the earth. This propagation mode is also known as skip propagation. The prop- agation mode of a radio wave with a high frequency (greater than 30 MHz) is known as line-of-sight communication, which means that any obstacle between the transmitter and receiver can highly attenuate or block the transmitted signal (Tanenbaum & Wetherall, 2014).

Microwaves

The electromagnetic waves within the frequency range of one GHz to 300 GHz are known as microwaves, and they follow the line-of-sight propagation. A collection of fre- quency ranges within microwave range are preserved for industrial, scientiﬁc, and med- ical (ISM) purposes. The most commonly used ISM band frequencies are 2.4 GHz and 5 GHz, which are also used for Wi-Fi. Bluetooth and near ﬁeld communication (NFC) tech- nologies also use ISM-band frequencies. Microwaves within the frequency range three GHz to 300 GHz are known as millimeter waves (mmWave) since the wavelengths varies from ten millimeter (mm) to one mm. The mmWave tend to be absorbed by the mois- ture, rain, and oxygen in the atmosphere, making it suitable for short distance commu- nication. However, short range communication signiﬁcantly improves the frequency reuse. Data transfer rate of the mmWave can be up to few Gbps (Tanenbaum & Wether- all, 2014).

Infrared

The frequency range of infrared waves varies from 300 GHz to 400 terahertz (THz). It cannot penetrate through the wall, making it suitable for secured short range line-of- sight communication (Tanenbaum & Wetherall, 2014). A common example of an infrared communication is a TV remote.

###### Contemporary Broadband Technologies

The digital subscriber line (DSL) and cable connections are currently the two most pop- ular wired broadband technologies. DSL allows for fast data transmission over a copper telephone line. DSL technology has a wide range of variations; however, the asymmetric digital subscriber line (ADSL) is most widely used. Cable broadband allows fast data transmission over coaxial copper TV cable. Fiber-to-the-home (FTTH) is another broad- band solution based on optical ﬁber, which also has many variations, the collection of which are referred to as ﬁber to the x (FTTX). Long term evolution (LTE) is a wireless sol- ution for broadband connection that allows fast data transmission over cellular phone infrastructure using radio waves. The following table compares the physical layer prop- erties of different broadband connection solutions (Kurose & Keith 2017; Oksman et al., 2016; Conformance Speciﬁcation Radio, 2011).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Broadband Technologies: Physical Layer Comparison | | | | |
|  | DSL | Cable | LTE | FTTH |
| Physical layer | Twisted pair copper wire | Coaxial cable, hybrid ﬁber coax | Radio wave 410—  2495 MHz | Optical ﬁber |
| Downlink bitrate | In general, 256 Kbps—  100 Mbps;  Maximum 1 Gbps | Up to  42.8 Mbps | Up to 100 Mbps | Average 20 Mbps up to 10 Gbps |

Summary

A computer network is a collection of communicating devices connected through links, including copper cable, optical ﬁber, and electromagnetic waves, and con- necting devices, such as routers, hubs, bridges, switches, and repeaters. A computer network is a digital system where data are presented and exchanged in digital form. Converting a piece of analog information into digital data is done through source coding in three stages: sampling, quantizing, and encoding. Encoding is the process of presenting data using binary bits which may involve compression and encryption techniques as well. The process of converting digital data into signal is known as digital modulation or line coding. Combining a signal with a high-frequency carrier wave is known as modulation. This combined signal is a composite signal that is made of multiple sinusoids. The bandwidth of the data-bearing digital signal is called baseband, and the bandwidth of the composite signal is called passband.

Computer Networks

According to Nyquist theorem, the sampling rate should be more than twice of the signal bandwidth. While signal propagates, it is affected by interfering signals or noise signals, which causes bit error. The capacity of a noisy channel can be calcu- lated using the Shannon-Hartley theorem. The quality of service of a network depends on various parameters such as bit-error rate, throughput, and delay. There are four main types of delays: propagation, transmission, queuing, and processing.



# Unit 2

## Communication Protocols

#### STUDY GOALS

On completion of this unit, you will be able to …

… analyze the transmitted and received packets and protocol

… apply the concepts of networking layers, services, and protocols.

… understand the basics of services, addressing, and protocols for datalink, network, and transport layers.

DL-E-DLMCSNDS01-U02

1. Communication Protocols

### Introduction

When someone intends to send information, for example, in the form of a letter, the primary intent of the sender is to convey the information to the receiver. In order to convey this information, a series of events must occur. First, the logical information is encoded into physical form by writing with ink on paper. The sender not only includes the information to be conveyed, but also the receiver's name, their own name, signa- ture, date, and location. Once the letter writing is done, the message cannot be sent directly, rather, it is folded and put in an envelope to protect it from damage and to secure the information from being read by other parties. Then, the sender and receiver addresses must be included on the outside of the envelope. Of course, this does not only contain the names, but includes a hierarchy of information needed for the deliv- ery (e.g., name, room or apartment number, house number, street, city, and country). The sender may need to include a stamp on the envelope, too.

All of this is only the ﬁrst step in the process of mailing a letter. The mail carriers and postal workers from both the side of the sender and receiver will also be involved, and all the post ofﬁces, mail carriers, truck drivers, pilots, captains, and any other person who handled the letter must maintain their own job protocols. The whole process of sending letter from sender to receiver can be divided into several steps or layers. Each layer deals with different services in order to forward the letter to its destination, and each of these services is maintained by a different job protocol.

A computer network or digital communication can be compared to sending a letter. The routers of a network can be compared to the post ofﬁces; network links can be consid- ered the roads and highways that connect the post ofﬁces; end devices can be consid- ered the houses and buildings; an email or editor can be compared to the paper and pen used to write the letter; folding the letter can be compared to data compression; and the envelope can be compared with the cryptography. Moreover, digital communi- cation also uses a hierarchy of addresses. Much like sending a letter, the whole process of sending data in a digital communication includes multiple steps or layers, each of which provides different services and follows various protocols.

### The ISO/OSI Reference Model

The International Organization of Standardization (ISO) has a standardized open sys- tem interconnection (OSI) model that indicates that computer networks communicate data through the following seven successive layers:

* Layer 7: Application
* Layer 6: Presentation
* Layer 5: Session
* Layer 4: Transport
* Layer 3: Network

Communication Protocols

* Layer 2: Datalink
* Layer 1: Physical

The application layer is the topmost layer of ISO/OSI model, while the physical layer makes up the bottom. When a sender sends a message, it is processed through all the layers starting from application layer to datalink layer, ﬁnally propagating through the physical layer. On the receiver side, the message travels through the datalink layer to application layer. These seven layers exist only on end devices; the connecting devices do not have the top four layers. Software applications are installed only on the end devices, not on the connecting devices. For example, routers have only lower three lay- ers (i.e., network, datalink, and physical), while switches have only two layers (i.e., data- link and physical).

###### Layer 7: Application Layer

The application layer deals with the software applications and application layer proto- cols. End users communicate through software applications, such as email applications, web browsers, audio, or video messengers. The applications installed on end devices communicate among them using various protocols depending on the types of the serv- ice. For example, hypertext transfer protocol (HTTP) is used for web services, and simple mail transfer protocol (SMTP) is used for email services. The primary service of the application layer is to establish the communication between end-users by connecting software applications installed on end-user devices and following application layer protocols. It also provides various services based on the application services, such as user-login, message upload and download, and ﬁle transfer.

###### Layer 6: Presentation Layer

On the sending side, the presentation layer receives the message from application layer and formats it into a desired data structure by using various encoding techniques. This encoding is also used to compress the data pass to the following layer below. On the receiving side, this layer receives data from session layer, then decompresses, decodes, and formats the data to present to the application layer. Loosely speaking, secure sockets layer (SSL), HTTP/HTML (agent), ﬁle transfer protocol (FTP), AppleTalk Fil- ing Protocol, and Telnet are some examples of presentation layer protocol (Knipp et al., 2002).

###### Layer 5: Session Layer

The session layer is responsible for establishing, maintaining, and ending communica- tion sessions between end-users. It also synchronizes the end-users for various modes of communication, such as simplex, half duplex, or duplex. For example, only node A can transmit to node B, but not vice versa. Half duplex supports bidirectional commu- nication, which means both A and B can transmit to each other, just not simultane- ously, while duplex communication supports simultaneous bidirectional communica-

tion (Forouzan, 2013). A common example of session layer service is when someone logs in to a server using a web browser, the session layer establishes connection, tracks session times, and closes the connection once the user logs out or the session time expires.

###### Layer 4: Transport Layer

This layer provides reliable data transfer services. On the sending side, the transport layer receives data from the layer above, breaks it into segments, and adds additional bits to ensure reliable data transfer. Reliable data transfer services include error detec- tion-correction mechanisms, ﬂow control, congestion control, and data loss detection and retransmission.

###### Layer 3: Network Layer

The network layer receives data from the transport layer and adds routing information to it. The two most important services of the network layer are packet forwarding and routing. Packet forwarding mainly deals with forwarding data to appropriate router out- put port. Routing service determines the best path between the end-users.

###### Layer 2: Data Link Layer

The data link layer connects the logical and physical forms of data. It receives data from the network layer which are presented as logical bits. This layer converts these logical bits to physical bits in the form of electric or electromagnetic signals and also provides media access control and synchronization between sender and receiver. More- over, like the transport layer, it provides error-detection and ﬂow control mechanisms.

###### Layer 1: Physical Layer

The physical layer is the connecting path or link between network nodes. In this layer, data travel in the form of electrical, electromagnetic, or optical signals. The most popu- lar physical links are coaxial cables, twisted pair cable, optical ﬁber, Wi-Fi, microwave signals, and radio signals.

###### Protocol Data Unit and Addressing

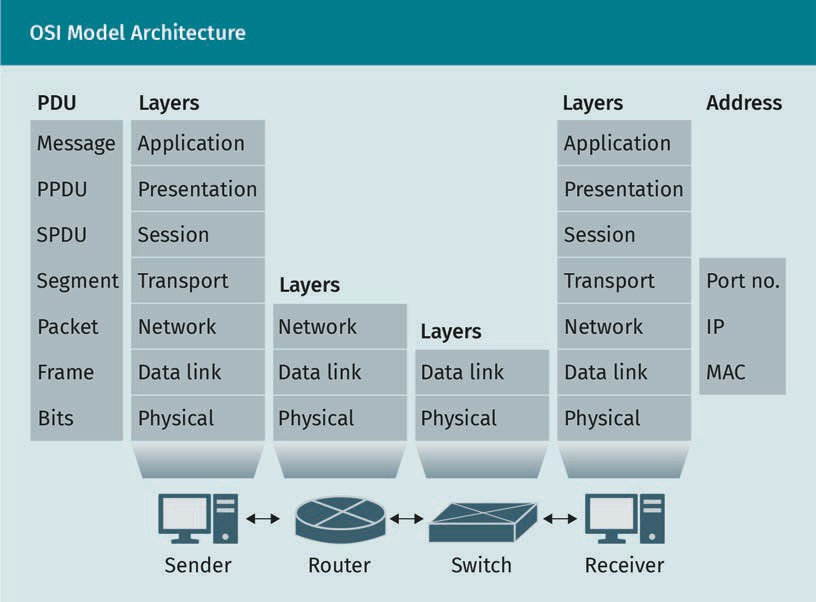
When a sender sends data to the receiver, the data are passed through all seven layers. Each layer adds some additional bits to the data received from the upper layer. The data unit of a layer is commonly known as protocol data unit (PDU). The PDU is known by different names at each layer. A PDU has two parts: header and payload. The data received from the upper layer is known as payload, and the additional bits added to the received data are known as header bits or header. The header bits include the serv-

Communication Protocols

ice information and the layer addresses of the sender and receiver. For example, a transport layer header includes the port numbers, a network layer includes IP addresses, and a datalink layer includes MAC addresses of the sender and receiver. These are explained in the following:

* A port number is 16-bit number which addresses an application program installed on sending or receiving end node. For example, port 80 is for Hypertext Transfer Protocol (HTTP) web services and 993 for secured mail transfer. For a list of transport layer port numbers, see (Internet Assigned Numbers Authority (IANA), n.d.).
* An Internet Protocol (IP) is 32-bit number which indicates an end device. More spe- ciﬁcally, it indicates the operating system of the end node. Each device such as lap- top, smart phone, or router, which is connected to the internet or local area network (LAN) obtains an IP address.
* A MAC address is 48-bit number that indicates the link layer device of a network node.
* The socket address is a combination of an IP address and a port number (Forouzan, 2013).

The following ﬁgure shows the ISO model architecture, layer names, PDU names of each layer, and the corresponding layer addresses.



### Data Link Layer: Standards and Technologies

The data link layer is the second layer, located below the network layer and above the physical layer. This layer is implemented in a network interface controller (NIC) card. NIC is a hardware that connects the data link layer to the physical layer. Two widely used NIC cards are ethernet and wireless Wi-Fi. In the layers above the data link layer, data are presented logically. However, in the physical layer, the data are presented in the form of electric or electromagnetic energy or waves. The data link layer connects the logical and the physical layers. The protocol data unit of this layer is called the frame. The frame header encapsulates network layer datagram or packet. The data link layer is divided into two sublayers: logical link control (LLC) and media access control (MAC). The LLC sublayer provides error detection and ﬂow control services, while the MAC layer provides MAC-based packet forwarding and multiple access control services.

###### MAC

The data link layer address, called MAC address, is a 48-bit or 6-byte hardware address, which means that each NIC card has a unique MAC address. The network layer address IP is a logical address which can be chosen or edited by the user, while the MAC is a physical address burnt into the NIC card, which a user is not allowed to edit. However, a device may have multiple NIC cards and, thereby, multiple MAC addresses. The use of virtual machines also allows multiple MAC addresses on a single device. A MAC address is presented in hexadecimal format and each byte is separated by a colon. The MAC address with all bits 1, i.e., FF:FF:FF:FF:FF:FF, is called a broadcast address. Address reso- lution protocol (ARP) is a network layer protocol which helps to ﬁnd the MAC address of a host for which the IP address is known. This procedure is documented in the Request for Comments 903 (RFC 903) (Finlayson et al., 1984).

###### Error Detection

The datalink layer calculates error detection bits (EDBs) based on the payload bit values. The EDBs are added in the frame header. The receiver side recalculates the EDBs. If the calculated EDBs are different than the sender’s EDB, then the bit error is detected. If the receiver’s EDBs are equal to the sender’s EDBs, then no bit error is detected. The data link layer uses a cyclic redundancy check (CRC) to detect bit errors. First, the sender chooses the number of bits in EDBs (represented with r). Then an r+1 bit number G*,* which is called generator, is chosen. The data D are then padded with r zeros. Let, D’ represent the zero padded D. Next, D’ is divided by G to calculate the remainder R. The R is represented in r*,* bits which is the desired EDBs. However, in order to make the process faster, the calculating remainder is made simpler by perfor- ming substructions using “exclusive or” (XOR) functions. For example, suppose D = 11001010, G = 1011, then R = remainder of (D/G).

Communication Protocols

1011 ) 11001010000 ( 11101

1011

1111010

1011

100010

1011

01110

1011

0101 so, R = 101

In practice, a data link layer frame uses 32-bit or 4B CRC bits. In general, a data link layer does not provide any error correction mechanism. Once a bit error is detected on the receiver side, the frame is simply dropped (Tanenbaum & Wetherall, 2014).

###### Multiple Access Protocols

An important service of the data link layer is that it allows multiple users to transmit over a single physical resource. There are numerous multiple access mechanisms, which can be broadly categorized into three types: resource partitioning, taking turns, and random access protocols. In resource partitioning protocols, the resource such as time or frequency is divided into small portions and divided among multiple users. In taking turns protocols, each user takes turns to use a common resource. These proto- cols are centralized, which means a central node maintains the resource distributions. Finally, random access protocols are decentralized, meaning each user has access to the shared resource in a random manner.

Random access protocols

Carrier sense multiple access (CSMA) protocol is a solution for random access. The operation of this protocol is similar to human conversation. A node listens before it talks. If a node senses that another node is transmitting, then it refrains from transmit- ting, waits for a random amount of time, then retransmits. However, CSMA may still suf- fer from collusions. If two nodes A and B sense at the same time that the other is not transmitting, then they may start to transmit simultaneously, and eventually resulting in collusion. This is known as a hidden node problem. Two improved versions of CSMA are as follows:

* Carrier sense multiple access with collision detection (CSMA/CD) implements the mechanism to detect collusion. It detects collision by measuring the signal-to-noise ratio. Once the collision is detected, it enters binary exponential backoff phase. It means if a node experienced collusion for m times to transmit a frame, then it will wait for the time to transmit 512×K bits. Here, K is a number randomly chosen from

the following range 0, 1, 2, 3, . . . . 27 − 1 . The efﬁciency of CSMA/CD is given by 1/ 1 + 5’p/’t , where ’p and ’t are propagation and transmission delays for a transmitted frame.

* + Carrier sense multiple access with collision avoidance (CSMA/CA) or CSMA/CD is a reactive solution to tackle hidden node problem. In particular, CSMA with collusion avoidance is a proactive solution to the hidden node problem. It is done through exchanging request to send (RTS) and clear to send (CTS) messages (Kurose & Keith, 2017; Comer, 2015).

###### Data Link Layer Standards

There are several data link layer standards depending on the characteristics of the physical layer. Ethernet is the most widely used datalink layer standard for wired LAN. Ethernet has multiple variations, such as 10BASE-T, 10BASE-2, 100BASE-T, 1000BASE-LX, 10GBASE-T, and 40GBASE-T, which are designed for 10 Mbps, 10 Mbps, 100 Mbps, 1 Gbps, 10 Gbps, and 40 Gbps LANs, respectively (Comer, 2015). “T” stands for twisted pair cable, and “BASE” stands for base band transmission. 10BASE-2 is standardized for thin coaxial cables and 1000BASE-LX is for optical ﬁber. The ethernet-based LANs are stand- ardized by IEEE 802.3. It implements CSMA/CD random access protocol and 4-byte CRC error detection technique. The second most widely used LAN is wireless Wi-Fi LAN oper- ates over 2.4 GHz or 5 GHz microwave frequency. Wi-Fi is standardized by IEEE 802.11. It implements CSMA/CA random access protocol. Data-over-cable service interface speci- ﬁcations (DOCSIS) is a standard protocol for hybrid ﬁber-coaxial (HFC), which combines optical ﬁber and coaxial cable. IEEE 802.15 standardizes the physical (PHY) and data link layers for wireless personal area network (WPAN). There are several versions of IEEE

802.15 for different types of WPAN. For example, 802.15.1 is for Bluetooth technology- based WPAN; 802.15.3 is for high rate WPAN; and 802.15.4 is for low rate WPAN (6LoWPAN). It is also considered as the standard for the Internet of Things (IoT). 802.15.5 and

802.15.6 standardize WMAN mesh network and wireless body area network, respectively (Comer, 2015).

###### Switching

Switches are Layer 2 devices, which means they only have two layers: physical and data link. Because of this, a switch is able to ﬁlter data based on the MAC address and for- ward it to the desired output port. Unlike a switch, a hub is a layer 1 device; hence, it is not able to ﬁlter and forward data based on the MAC address. A hub forwards incoming data to all output ports, which ﬂoods the network with unnecessary overheads. Switches overcome this shortcoming through MAC-based forwarding.

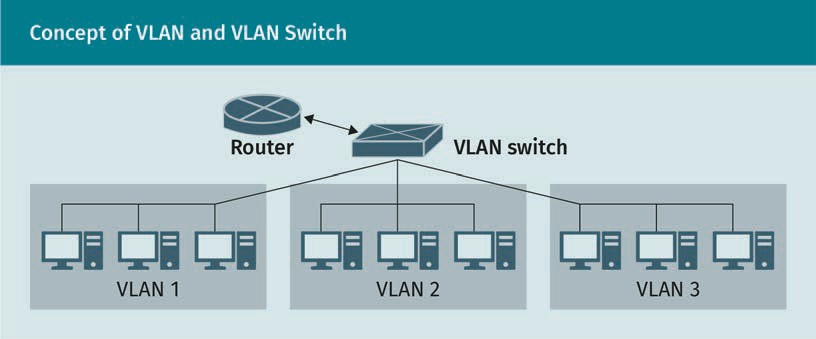
Communication Protocols

###### VLAN

In general, all end nodes of a LAN are connected to a common switch; hence, for N number of LANs there should be N number of switches. In addition, all nodes of a LAN reside in the same geographical area, such as same ofﬁce ﬂoor or building. A virtual LAN (VLAN) is a technology that allows the creation of multiple LANs in the same geo- graphical area using only one switch, called the VLAN switch.

Suppose there are 100 ports in a VLAN switch. The port numbers 2, 3, 6, 10, and 77 are

grouped to form LAN 1 and port numbers 11, 22, 41, 88, and 50 are grouped to form LAN 2 and so on. VLAN is standardized by IEEE 802.1Q protocol. It shims a 4 Byte VLAN tag within a traditional ethernet data frame (Tanenbaum & Wetherall, 2014; Solomon & Kim, 2021). VLAN is different from a virtual private network (VPN). Unlike VLAN, the nodes of a VPN are not necessarily connected to a common switch.

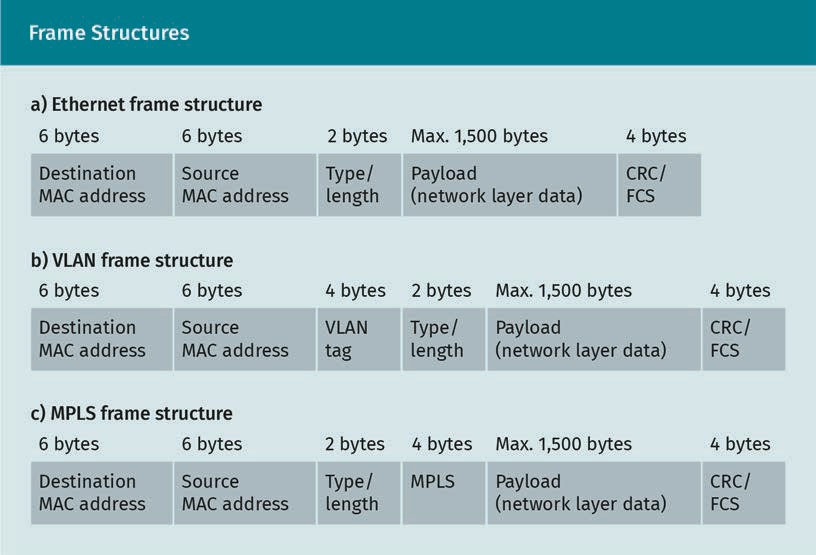


###### MPLS

Multiprotocol label switching (MPLS) is a 2.5 layer protocol that allows the forwarding of data through routers based on an MPLS header instead of an IP address. This techni- que makes the forwarding process faster by avoiding reading the IP header and looking up routing tables. First, a selected routed path is chosen between the source and desti- nation. In a traditional connectionless network, when a packet travels through a path, each router reads the header bits and IP addresses, then applies the routing algorithm, looks up routing tables, then ﬁnds the next hop. This entire process imposes signiﬁcant processing delays. In the case of MPLS, ﬁrst a path is chosen between source and desti- nation. Then the network layer data is encapsulated using an MPLS header, which uses a short label instead of long IP addresses. The MPLS label then tells the router what the next hop is. MPLS is standardized in RFC 3031, 6178, and 6790 (Rosen et al., 2001).

Virtual private net- work (VPN)

A VPN overlays a pri- vate network on a public network, allowing remote access to a private network.



Exercise

Wireshark is the post popular open-source packet sniffer and protocol analyzer. It allows to capture real time packets that are being sent or received by a computer. It provides graphical interface to read a packet’s header ﬁeld and payload information in a layered order. It is highly recommended to download and install Wireshark and start capturing packets to investigate what is inside them. A massive number of captured sample packets are available for free on Wireshark’s ofﬁcial website. To download sam- ple packets (.pcap or .cap ﬁles) for 802.11 Wi-Fi, 802.1Q VALN, MPLS, and ARP, see the Wireshark Foundation website (n.d.). Investigate the structure of the packet header ﬁeld and values.

### The Network Layer: Addressing and Routing

The network layer is located between the datalink and transport layers and is imple- mented in operating systems (OS). The network layer address is called the internet pro- tocol (IP) address, which indicates a network node. More speciﬁcally, it indicates the OS of a network node. A computer may have multiple OSs running on it, including virtual machines, each of which may have different IPs. Network layers are implemented in both end nodes and routers. The two major services of network layer are packet for- warding and routing.

Communication Protocols

###### Network Layer Addressing

There are two types of IP addresses: Internet Protocol version four (IPv4) and Internet Protocol version six (IPv6). IPv4 is the traditional network layer addressing scheme. An IPv4 address has four octets or 32-bits. A IPv4 address is written using the format 8 . b . c . ’, where each octet is separated by a period. Each end device of a network has a unique IP addressed, and each port of a router has a unique IP address.

Subnet mask

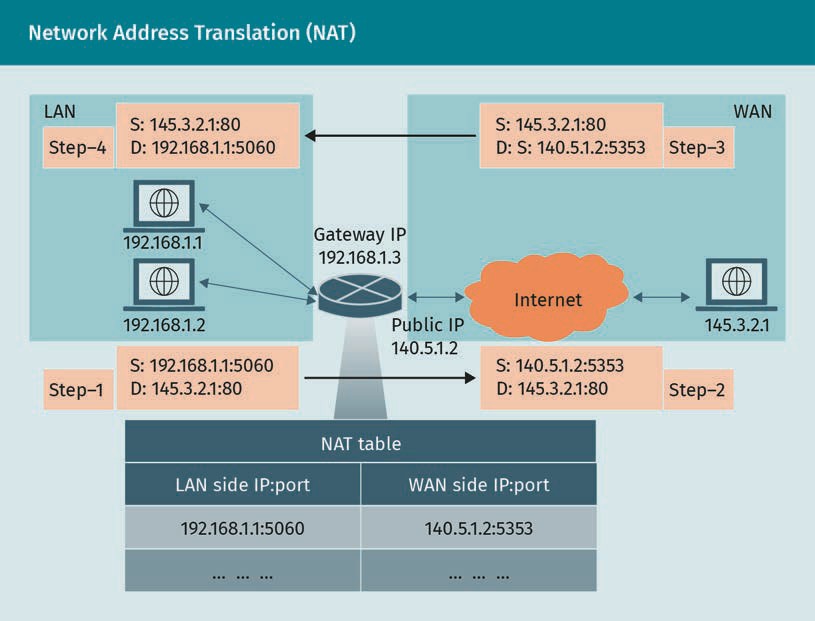
All the end nodes or hosts that are connected to the internet through a common router port maintain a common IP pattern. This common pattern is known as the subnet mask. The network that connects to the internet through a common router port is called the subnet. The IP address of a subnet host is usually denoted by the following

classless interdomain routing (CIDR) notation 8 . b . c . ’/e, where e indicates that the subnet mask contains e bits, allows for 2(32-e) hosts in a network. To be more practical, the total number of hosts in a subnet is 2(32-e)-2. For example, 192.168.0.4/23 indicates that the leftmost 23 bits of this IP address make up the subnet mask, and the subnet can have maximum 29-2 = 510 hosts. The internet service provider (ISP) assigns a sub- net mask to limit the number of users in a network. The subnet mask is also used to ﬁlter incoming and outgoing packets of a network.

Network address translation

Network address translation (NAT) is a remapping technique used to hide a local host IP from the global network or internet by combining the port number with IP address. It is done by NAT router, which maintains the remapping table. The following ﬁgure shows a NAT scenario in which the router connects a LAN inside port 192.168.1.3 and connects the LAN to the internet or WAN through the outside port 140.5.1.2. The four steps involved in this process are as follows:

1. The sending host with socket 192.168.1.1:5060 sends a message to the receiving host with socket 145.3.2.1:80. Here, 192.168.1.1 and 5060 are the local IP and port number of the sending host.
2. The NAT router translates the local IP and port to global (or public) IP and port. The global IP of the sender is 140.5.1.2 which is the WAN side port of the router. The port 5353 is chosen arbitrarily by the router to indicate the socket 192.168.1.1:5060.
3. A message is sent by the source socket 145.3.2.1:80, destined to the public socket 140.5.1.2:5353.
4. The NAT table translates the global address to local address 140.5.1.2:5353.



IPv4

Traditionally, IPv4 addresses are divi- ded into ﬁve classes: A, B, C, D, and E. The ﬁrst octet range for class A is 1—126. It ranges between 128

—191 for class B, and 192—223 for C. Class D is reserved for multicasting, and E is reserved for future

use.

IPv4

IPv4 is a 32-bit address which means it can identify a total of 232 (approximately 4.3 bil- lion) global IP addresses. According to the Cisco Annual Internet Report, the number of total internet users will be 5.3 billion by 2023. By that time, the total number of connec- ted devices will be 29.3 billion (Cisco, 2020). IPv4 is not capable of serving this mon- strous number of devices. For this reason, NAT scheme is used to preserve the global IP addresses. Using NAT, an ISP can use only one global IP to serve an entire subnet or LAN. Another advantage of NAT is that it hides local IPs from the internet, adding an extra layer of security to the local network. A further advantage is if the LAN chooses a new ISP, it still can use the same IP addresses for the LAN devices.

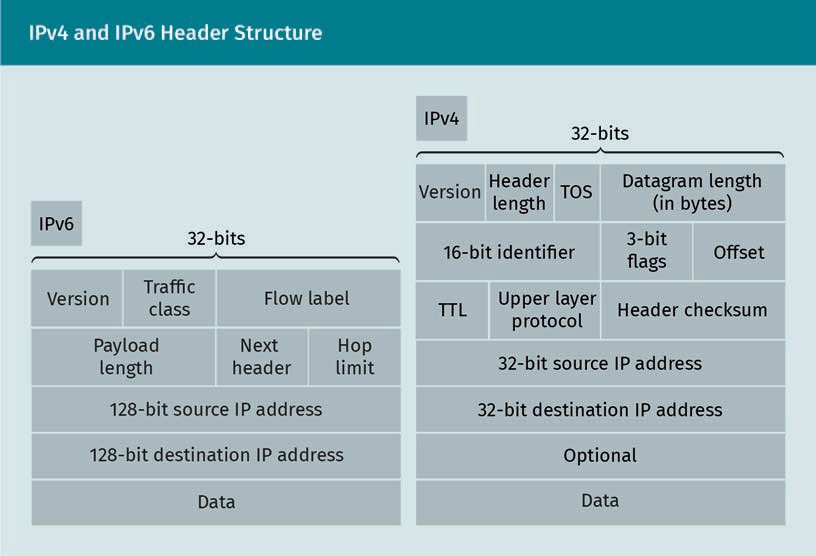
NAT introduces some disadvantages too. Firstly, NAT hides local IP addresses, which can be considered a security issue. Secondly, the translation process imposes an additional delay and layer of complexity. An important issue with NAT is that it violates the OSI layering architecture. Because a router is a network layer device, and the transport layer is absent, the router should not deal with a port number, which is a transport layer address. The capacity issue of IPv4 could be easily resolved by IPv6, bringing into question the necessity of NAT.

Communication Protocols

IPv6

IPv6 is a 128-bit network layer address which is introduced to resolve the capacity limi- tation of IPv4. IPv6 can represent up to 2128 = 3.4×1038 unique IP addresses, which is an astronomical number. IPv6 also simpliﬁes the network layer datagram header in order to reduce the processing delay. Additionally, it allows the ﬁltering and forwarding of data based on trafﬁc ﬂow.

The conversion from IPv4 to IPv6 is a slow process. As of April 2021, more that 30 per- cent of Google users use IPv6 (Google, n.d.). It is notable that the Internet Corporation for Assigned Names and Numbers (ICANN) maintains and distributes the IP address through the Internet Assigned Numbers Authority (IANA). The backward compatibility of IPv6 with IPv4 is acheived through tunneling, which is the technique of encapsulating entire IPv6 datagrams using IPv4 headers.



In the ﬁgure above, the time to live (TTL or hop limit) indicates the time or hop count after which a packet expires. The type of service (TOS) or trafﬁc class indicates the serv- ice type of a packet based on whether a packet is prioritized for queue, ﬁltering, or for- warding.

DHCP

When a new device comes to a network, it initially does not have an IP address. The device requests an IP address from the server, which the server provides. This process of achieving a new IP address is done by the dynamic host conﬁguration protocol (DHCP), and involves the following four steps:

* 1. Discovery. The client broadcasts a DHCPDISCOVERY message with source IP 0.0.0.0, port 68, and destination IP 255.255.255.255, port 67.
  2. Offer. The server replies with DHCPOFFER message with destination IP 255.255.255.255 and destination port 68. This message also contains the offered IP address for the client.
  3. Request. The client replies with a DHCPREQUEST message with source IP 0.0.0.0, source port 68, destination IP 255.255.255.255, and destination port 67. It also includes the requested IP address, which is the same IP address offered by the server in DHCPOFFER message.
  4. Acknowledge. The DHCP server acknowledges the request message by sending DHCPACK message with destination IP 255.255.255.255, and port 68.

###### Network Layer Services

The primary services that a network layer provides are packet forwarding and routing. Packet forwarding is a per router local problem, whereas routing considers the overall scale of the network, including all routers and links. Packet forwarding means forward- ing a packet from the router’s input port to the proper output port. Routing is the proc- ess of selecting the suitable route or path between sender and receiver. Each router maintains a forwarding table, which is derived from the routing algorithm. There are different types of routing protocols, which are used to serve different purposes.

###### Types of Routing

Routing is the process of selecting a suitable path between sending and receiving nodes. A “path” is a series of links, and a “suitable path” usually refers to the least-cost path. Cost can be a function of a number of variables, such as distance, delay, through- put, and energy. The cost is inversely related to the throughput and proﬁt. There are various types of routing techniques, some of which are introduced here.

Dynamic versus static routing

In the case of dynamic routing, the routing table is updated if there is any change in the network topology or link cost. Conversely, a static routing does not update the rout- ing table frequently, rather, the network administrator decides when to change the routing table.

Centralized versus decentralized routing

In the case of centralized routing, a central entity collects routing information from each router, then it performs the routing algorithm, and the updated table is forwarded to each router by the central entity. In a decentralized routing algorithm, the routing algorithm is performed in each router, which update their own routing tables.

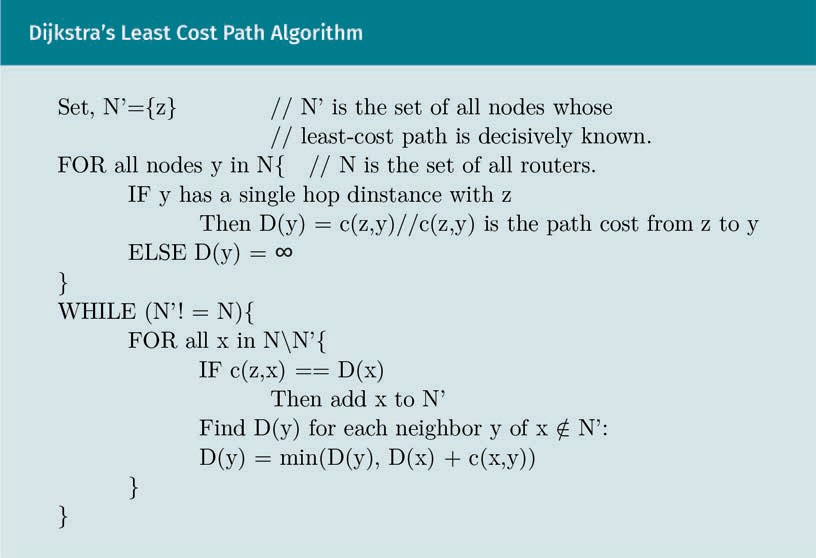
Communication Protocols

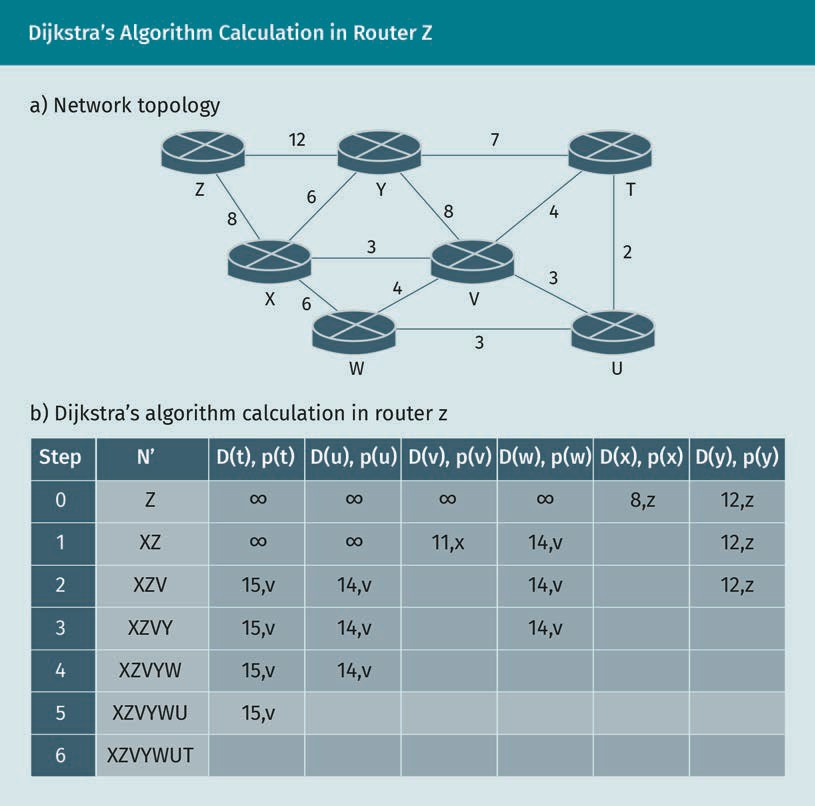
###### Routing Algorithms

A routing algorithm states each step of calculation to ﬁnd the suitable path. There are two types of routing algorithms: link state and distance vector.

Link state routing algorithm

In a link state routing algorithm, all routers share routing information with each other by broadcasting. Eventually, each router receives the routing information about the entire topology. Each router runs the same routing algorithm, calculates a suitable path, and updates the forwarding table. Dijkstra’s algorithm to ﬁnd the least cost path is the most widely used link state routing algorithm. It’s an iterative algorithm. If z is the source node, D(y) is the cost of the least-cost path from z to y, p(y) is the previous node of y within the least-cost path, and N is the set of all routers, then the Dijkstra’s least cost path algorithm can be stated as shown below.





Distance vector routing algorithm

Unlike the link state, the routers do not broadcast routing information; however, each router shares the routing information with the neighbor routers, i.e., the routers within a single hop distance. The routing information includes the number of links and asso- ciated link costs. Distance vector routing is an iterative, asynchronous, and distributed algorithm. The least cost path between 4 and < is calculated using the Bellman-Ford equation, given below:

’4 < = 71n= c z, x + dx y

Each router updates its table if a change in topology or link cost is noticed. Then, it for- wards the updated information to its neighbors. The neighbors also update their tables based on new information. If there is any change noticed in the table, then they for- ward the information to neighbors. This process continues until all routers have ach- ieved stable routing tables (Kurose & Keith, 2017).

Communication Protocols

Routing protocols

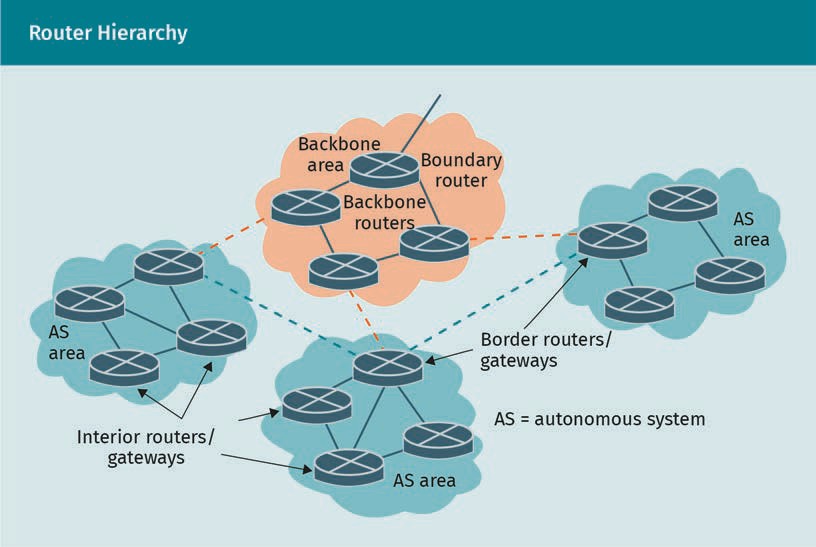
Routing protocols deﬁne how the message transfer and information exchange will take place between routers to run the routing algorithm. The set of routers within an auton- omous system (AS) or area is called routing domain. The routing protocol that deals with exchanging information among the routers within a domain is called intra-domain routing or intra-AS routing protocol. It is also known as the interior gateway protocol (IGP).

Each domain has a single IGP protocol, and different domains can have different IGP protocols. The router (or gateway) that connects the intra-domain routers with another domain is called border router. Routing protocol that deals with exchanging informa- tion among border routers is known as inter-domain routing protocol or exterior gate- way protocol (EGP) (Kurose & Keith, 2017). There are various types of routing protocols. Some widely used routing protocols are introduced below.

OSPF

Open shortest path ﬁrst (OSPF) is an IGP or intra-domain routing protocol that ﬁnds the shortest path between sender and receiver. It is based on the link state routing algorithm, i.e., Dijkstra’s algorithm. Each router maintains a routing database and rout- ing table. OSPF is a dynamic algorithm; once it senses any topological changes, it immediately recalculates the routing table and broadcast routing information to other routers. OSPF also provides message authentication, multi equal-cost paths, and multi- casting. OSPF is standardized in RFC 2328, 5340, 5709, 7474, 6549, and 6860 (Kurose &

Keith, 2017).



RIP

Routing information protocol (RIP) is a traditional IGP protocol. It is based on the dis- tance vector routing algorithm, which uses the Bellman-Ford equation to calculate path cost. RIP can handle a maximum 15-hop path. In RIP, hop count 16 indicates that the destination router is not reachable. RIP is rarely used nowadays due to its limitations, but it could be still useful for small networks, since it has fewer overhead bits. RIP also supports cryptographic authentication. RIP is standardized in RFC 2453, 4822 (Kurose & Keith, 2017).

BGP

Border gateway protocol (BGP) is an EGP or inter-domain routing protocol. BGP ﬁnds a suitable route between two nodes that reside in two different autonomous systems or domains. BGP exchanges information based on TCP transport layer protocol with the port number 179. There are two types of TCP connections for BGP: external BGP (EBGP) and internal BGP (IBGP). In EBGP, communicating routers reside in two different domains. IBGP sets up connections between the routers residing in the same domain. A border router uses EBGP to exchange information with another border router in a dif- ferent domain. A border router also uses IBGP to exchange information with the inte- rior routers within same domain. BGP protocol is documented in RFC 4271, 6286, 6608, 6793, 7606, 7607, 7705, 8212, and 8654 (Kurose & Keith, 2017).

ICMP

Internet Control Message Protocol (ICMP) is implemented in network layers of routers to exchange control information. ICMP uses IP headers; however, the datagram does not carry payload from transport layer. Instead, the IP header encapsulates the control message. The ﬁrst octet of the header is called ﬁeld and the second octet is known as code. The combination of ﬁeld and code number indicates a speciﬁc control message or error notiﬁcation. For example, “ﬁeld = 3, code = 1” indicates that the host is unreachable, and “ﬁeld = 3, code = 3” indicates the port is unreachable. ICMP protocol is standardized in RFC 792 and 6918 (Kurose & Keith, 2017).

###### Software Deﬁned Network

A software deﬁned setwork (SDN) is a network management technology where the rout- ing is done by a central entity. This central entity is actually a distributed system of multiple servers. SDN is mainly based on IPv6, which allows ﬂow-based forwarding. OpenFlow is a popular protocol for SDN (Benzekki et al., 2016; Kurose & Keith, 2017). The Open Network Operating System (ONOS) is a platform for SDN technology (Berde et al., 2014).

Exercise

Download sample packets (.pcap or .cap ﬁles) for ICMP, OSPF, and RIP from (Wireshark Foundation, n.d.). Now investigate the packet header ﬁelds and layer addresses. In addition, see what types of messages are being exchanged for a particular protocol, in what order, and what the message formats are.

Communication Protocols

### The Transport Layer: Reliability and Flow Control

The software that implements the transport layer services is known as a transport layer entity. Transport layer entities are mainly system programs implemented in the operat- ing system kernel. Also, a transport layer entity can be a part of an application program. The transport layer establishes a logical connection between sender-receiver applica- tion programs. It also provides reliable data transfer services between the sender and receiver. These reliable services include ﬂow control, congestion control, multiplexing, order packet delivery, packet loss detection and retransmission, connection establish- ment, and closing.

On the sending side, the transport layer receives message from the software applica- tion, then breaks the message into equal sized pieces called segments. Then, the port numbers of the sender and receiver applications are added to each segment header to identify the sender-receiver applications. The header also includes the service bits. The segment is then forwarded to the network layer. On the receiver side, the transport layer receives segments from the network layer and reads the port number from the segment header to identify the target application program. The subsequent segments are then merged and pushed to the application layer.

The transport layer services can be connection-oriented or connectionless. The most familiar connection-oriented protocol is the transmission control protocol (TCP). TCP provides reliable data transfer services. However, the most popular connectionless pro- tocol is the user datagram protocol (UDP). While UDP provides best effort services, it does not guarantee reliable data transfer. Some other popular transport layer protocols are stream control transmission protocol (SCTP), resource reservation protocol (RSVP), datagram congestion control protocol (DCCP), and QUIC (a spelling variant of “quick”) (Papastergiou et al., 2016; Polese et al., 2019).

###### User Datagram Protocol (UDP)

UDP is a connectionless transport layer protocol and does not guarantee reliable data transfer services, which means it does not provide in-order packet delivery, ﬂow con- trol, congestion control, error detection, packet loss detection, or retransmission serv- ices. UDP is useful for particular applications where achieving reliable data transfer is not assured, or implementing reliable data transfer is not essential. For example, it is useful for real-time services, voice over internet protocol (VoIP), data streaming, multi- casting, and broadcasting. Transaction based or query-response based protocols also use UDP, for example, domain name system (DNS) and dynamic host conﬁguration pro- tocol (DHCP).

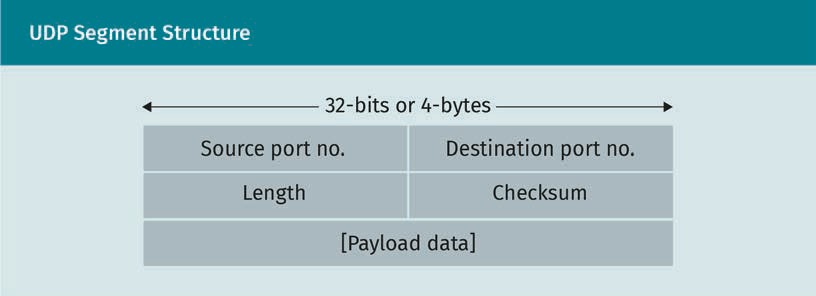
UDP segment structure

UDP protocol is standardized in RFC 768 (Postel, 1980). The DUP datagram structure is simple. The header size is 8 bytes which includes the following ﬁelds:

Flow

A ﬂow or trafﬁc ﬂow is a stream of pack- ets that move between the source and destination, and which can be distin- guished from other packets based on various attributes (or a combination of attributes), such as IP address, MAC, and port number.

* The “source” port includes 16-bit port number of the sending side application. This ﬁeld is optional.
* The “destination” port includes 16-bit destination port address.
* The “length” is the size of the datagram in bytes which includes both header and payload bits. The minimum value of length ﬁeld could be 8 bytes, meaning that the datagram includes only the header bits.
* The “checksum” is a 16-bit number that is used for error detection. This is known as internet checksum.



Internet checksum

To calculate checksum, the data are represented as 16-bit chunks. If the number of bits is not a multiple of 16, then extra zeros are added at the beginning of the data bits to make the data size a multiple of 16. This technique is known as zero padding. The 16-bit chunks are then added. If the addition produces a carry bit, then the carry bit is added to the sum. Finally, taking one's complement of the result produces the internet check- sum. For example, consider the following 4-byte data:

10001001 11000000 01101100 01111000

First, the 4-byte data has to be divided into two 16-bit words. Add the 16-bit words to produce the sum 11110110 00111000. Taking one's complement of the sum produces the checksum 00001001 11000111.

While calculating the checksum of a segment, both the header and payload bits are considered, but the checksum ﬁeld values are set to zero. When the receiver side receives the segment, it calculates the checksum of the received segment. If the calcu- lated checksum value is equal to the value of the checksum ﬁled, that means there is no bit error.

###### Transmission Control Protocol (TCP)

TCP is a connection-oriented, reliable data transfer protocol standardized in RFC 793, 879, 6528, 6691 (Information Sciences Institute, 1981). TCP is called “connection oriented” because it establishes a connection between sender-receiver pair before transferring the data or segment. After transmission of the segment, the connection is closed for-

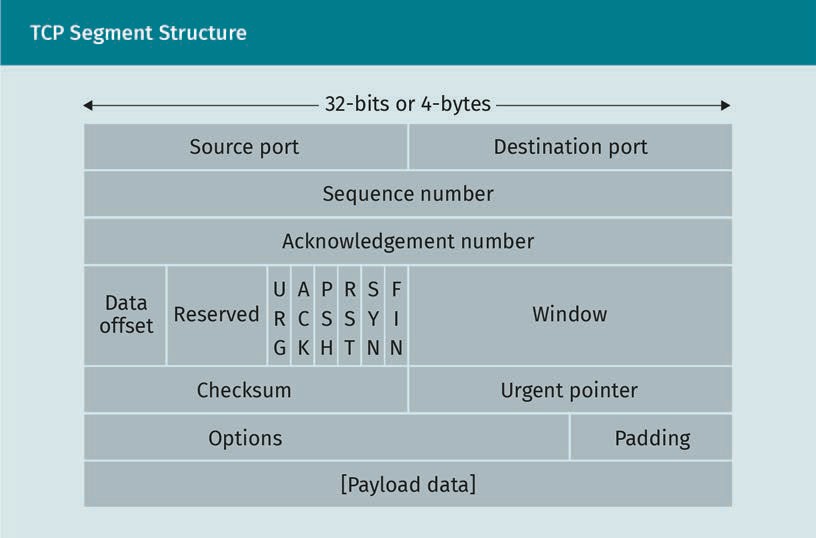
Communication Protocols

mally by both parties. The maximum TCP segment size (MSS) is 536 octets by default. TCP is considered reliable because it provides services to ensure error-free, in-order data delivery. These services include error detection, loss detection, retransmission, in- order delivery, ﬂow control, and congestion control.

TCP segment structure

TCP segment structure is deﬁned in RFC 793 (Information Sciences Institute, 1981). The segment header size is usually 40 bytes. The header includes the following ﬁelds:

* The “source port” ﬁeld contains a 16-bit port address of sender application.
* The “destination port” ﬁeld contains a 16-bit port address of receiving application.
* The “sequence number” ﬁeld contains a 32-bit number. It holds the sequence num- ber of the ﬁrst data octet of the segment. If the SYN bit is 1 then this ﬁeld value is equal to the ISN (initial sequence number) and ISN+1 indicates the ﬁrst data octet.
* The “acknowledgement number” ﬁeld contains a 32-bit number. This ﬁeld value is equal to the sequence number of the segment that the sender expects to receive next.
* The “data offset” ﬁeld indicates where the payload data starts in the segment.
* The “reserved” ﬁeld is a 6-bit ﬁeld which is reserved for future use.
* The “control ﬂags” ﬁeld contains six 1-bit ﬂags. URG ﬂag indicates urgent data; ACK indicates acknowledgement; PSH indicates to push the segment immediately to the next layer; RST indicates to reset connection; SYN indicates connection request; and FIN indicates to end data transmission.
* The 16-bit value “window” ﬁeld indicates the size of the receiver buffer, i.e., the num- ber of bytes this segment sender is expecting to receive.
* The “checksum” ﬁeld contains a 16-bit internet checksum value.
* The “urgent pointer” ﬁeld. If the URG is set to 1, then this 16-bit value indicates the sequence number of the octet next to the urgent data.
* The “options” ﬁeld may hold the maximum segment size (MSS) value. If this ﬁled does not exist, then segment can be of any size.
* The “padding” ﬁeld indicates the number of zero padding bits added to make the segment size multiple of 32.



TCP connection: Establishment and termination

In some cultures, when two parties participate in a transaction, they will ﬁrst greet each other by shaking hands before the actual transaction takes place. They end their busi- ness by saying farewell. Similarly, TCP exchanges initial messages to establish a connec- tion between the sender-receiver pair before the actual data transmission. After suc- cessful data transmission, the two parties exchanges some messages to end the communication.

The software program that initiates the connection is known as the client process, and the other party that accepts the connection invitation is called the host process. The step of establishing connection is known as handshaking. TCP takes three steps to establish a connection which is called three-way handshake:

* + 1. The client sends a TCP header with )> 2 = 1, A6K = 0, and )e/Aence 2B . = 0 (or a random number).
    2. The host replies with a TCP header where )> 2 = 1, A6K = 1,

AcknBDle’ge7ent 2B . = 1 (i.e., received sequence 2B . + 1), and

)e/Aence 2B . = 0.

* + 1. The client replies with a TCP he8’e, D1th )> 2 = 0, A6K = 1,

)e/Aence 2B . = 1, and AcknBDle’ge7ent 2B . = 1.

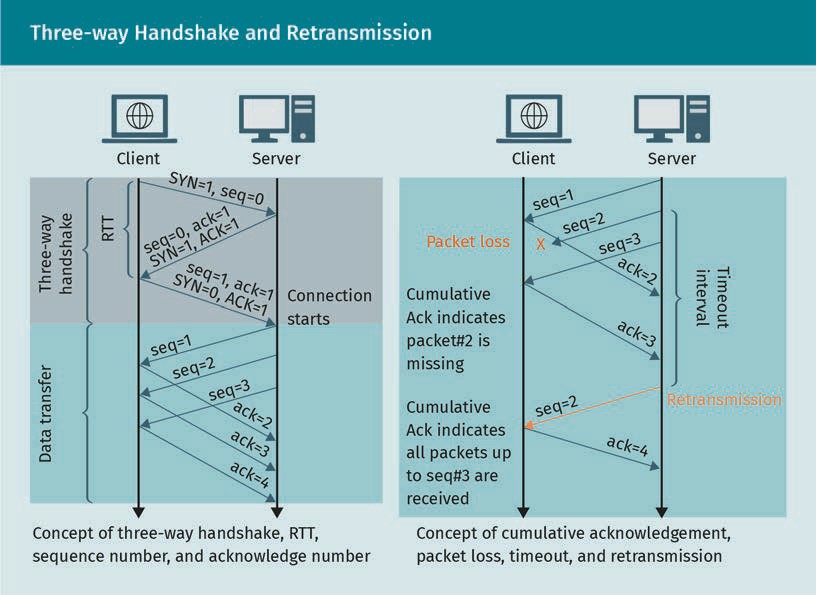
The connection termination is completed using the following four steps:

1. The client sends GI2 = 1 and )e/Aence 2B . = X. It means the client has ﬁnished sending data, but it can still receive data.
2. The host sends A6K = 1 and AcknBDle’ge7ent 2B . = X + 1.

Communication Protocols

1. The host sends GI2 = 1 and )e/Aence 2B . = > . The host ﬁnishes sending data.
2. The client sends A6K = 1 and AcknBDle’ge7ent 2B . = > + 1. The host closes the connection after receiving this acknowledgement. The client does not close the connection immediately, but waits for twice the length of the timeout period before terminating the connection.

The ﬁgure below, titled “Three-way Handshake and Retransmission,” illustrates the con- cept of a three-way handshake.



Loss detection and recovery

After sending a packet with )e/Aence 2B . = X, the client waits to receive the reply from the host with AcknBDle’ge7ent 2B . = X + 1. It waits until the timeout interval. If the reply is not received within the timeout interval, then the packet is detected as lost, and the client retransmits it to recover the loss. The retransmission timeout inter- val is calculated with the following formula.

t17eBAt = est178te’RTT + 4 · ’e0RT T

est178te’RTT = 1 − K × est178te’RTT + K · s87pleRT T ’e0RT T = 1 − /3 × ’e0RTT + /3 est178te’RTT − s87pleRTT

Round trip time (RTT) is the time interval between sending a packet and receiving cor- responding acknowledgement. RTT is also known as round trip delay. The RTT of the most recent packet is known as sampleRTT.

Flow control

A received packet waits in a receiver buffer until the previous packet is processed. When packets are received out of order, the received packets also wait in the buffer until the missing packet is received. If the sending side transmits too fast compared to the receiver’s processing speed, then the receiver buffer will overﬂow, and newly arrived packets will be dropped or lost. The TCP ﬂow control mechanism restricts the sender from transmitting at a higher rate in order to protect the receiver buffer from overﬂow. It is done by setting the “Window” ﬁeld of the TCP header equal to the free buffer size. This window is also known as the receive window. If a client sets the receive window to 1000 bytes, then the server will not send more than 1000 bytes of data with- out acknowledgement. Suppose a server sent ﬁve sets of 1000 bytes consecutively. Now, the server has to pause transmission and wait until it receives all acknowledge- ments for those 5000 bytes. After receiving acknowledgement, the server will be able to send next 5000 bytes.

Congestion control

The client and server can be connected through multiples links and routers. In general, a router is connected with multiple links, thus it can receive data from multiple sources simultaneously. If the data arrival rate in a router is higher than its service rate, i.e., processing capacity, then the router buffer may overﬂow, causing packet drop or packet loss. The phenomenon of pushing too many data through a network link is called net- work congestion. TCP detects network congestion in two ways: retransmission timeout and triple-duplicate acknowledgement. Once the congestion is detected, the conges- tion window size is reduced to avoid congestion. The two most common congestion control protocols are TCP Tahoe and TCP Reno. However, there are more than thirty var- iants of TCP congestion control protocols available, though only a few of them are used in practice. Before discussing TCP Tahoe and Reno, the following concepts should be understood (Abed et al., 2011):

* + “Congestion window” is the number of octets or bytes a sender sends consecutively without receiving acknowledgement.
  + “Cumulative acknowledgement” TCP uses cumulative acknowledgement scheme. The acknowledgement of n indicates that the receiver received all packets up to n − 1 and it is waiting to receive the packet n.
  + “Duplicate acknowledgement” means if packet n + 1 is received before the packet n (out of order), then the receiver will send the acknowledgement n again. So, the sender will receive duplicate acknowledgement of n. If packet n + 2 is also received out of order before n, then the receiver will send the acknowledgement n for one more time. If the packet n + 3 is also received out of order, at this point the sender will receive triple-duplicate acknowledgement of n.
  + “Fast retransmission” means once the sender receives a triple duplicate acknowl- edgement, it retransmits the packet n immediately without waiting for the timeout. This scheme is known as fast retransmission. Fast retransmission indicates mild congestion in the network.
  + “Explicit congestion notiﬁcation (ECN)” is an additional feature of IP and TCP proto- cols, which allows end-to-end notiﬁcation of congestion.

Communication Protocols

TCP Tahoe and Reno

TCP Tahoe and Reno deal with congestion in two phases: congestion avoidance and fast recovery. Tahoe and Reno show no difference in congestion avoidance phase, but they act differently in fast recovery phase.

Congestion avoidance

This phase is also known as TCP slow start. Initially, the congestion window size 6W2( is set to 1 maximum segment size (MSS). For each successful transmission, the 6W2( is increased to power of two. Thus, after the ﬁrst successful transmission, it is 6W2( = 2; after a second successful transmission, 6W2( = 4, and so on. However, once 6W2( hits the predeﬁned threshold of segment size threshold, then the 6W2( stops increasing exponentially and instead increases linearly. More speciﬁcally, 6W2( increases by 1 MSS after each successful transmission.

Fast recovery

TCP enters in fast recovery phase once a retransmission is triggered. If the retransmis- sion is triggered, then TCP re-enters congestion avoidance phase. It sets ssth,eshBl’ = 6W2(/2, and 6W2( = 1. If the retransmission is triggered due to tri- ple duplicate acknowledgement, then TCP Reno reacts differently. It sets ssth,eshBl’ = 6W2(/2, and 6W2( = 6W2( − 3.

Exercise

Dowload and sample captured packets from Wireshark (Wireshark Foundation n.d.). Apply the acquired knowledge to investigate the TCP and UDP header structure and information.

Summary

Explicit congestion notiﬁcation (ECN) This is an extension added to IP, which indicates two ECN bits in the IP header and allows conges- tion to be recognized without any packets being dropped. If a router experiences congestion, i.e., it drops packets due to buffer overﬂow, it sets the ECN bit of the header to notify the end-nodes that congestion is detec- ted.

There are seven layers of computer networking: application, presentation, session, transport, network, datalink, and physical. Each layer provides various services and addressing. Port no., IP, and MAC are the addresses for transport, network, and data link layers, respectively. The application layer provides an application interface for users. The presentation layer encodes application layer data for compression and encryption purposes. The session layer maintains sender-receiver session connec- tion, and the transport layer provides reliable data transfer, error detection, ﬂow control, and congestion control services. The two most popular transport layer pro- tocols are TCP and UDP. TCP is a connection-oriented, reliable protocol, whereas UDP is an unreliable and connectionless protocol. This means UDP does not pro- vide ﬂow control, congestion control, packet loss detection, retransmission, or in- order delivery services. The two main services provided by the network layer are packet forwarding and routing. Two widely used routing algorithms are link state and distance vector routing. Link state routing is based on Dijkstra’s least cost path algorithm, whereas distance vector routing is based on the Bellman-Ford equation. Some other common network layer protocols are OSPF, RIP, and BGP. OSPF is based on link-state routing, and RIP is based on distance vector routing algorithms, and

both are predominantly intra-domain routing protocols. BGP is an inter-domain routing protocol. ICMP protocol is used to exchange control messages between routers.



# Unit 3

## The Internet Protocol Suite

#### STUDY GOALS

On completion of this unit, you will be able to …

… describe the history of the internet and the world wide web.

… understand the concepts of the transmission control protocol/internet protocol (TCP/IP) reference model and protocol stack.

… apply the knowledge of internet protocols and services to analyze data packets.

… analyze the security aspects of computer networking and internet.

DL-E-DLMCSNDS01-U03

1. The Internet Protocol Suite

### Introduction

The internet is the most widely used computer network. It is also the largest computer network in terms of the total number of connected devices, number of users, spatial area, and organizational scope. The term “internet” refers to “inter network,” which means the internet is a network of many interconnected networks (Kurose & Keith, 2017). It comprises mostly all types of networks regardless of the spatial (i.e., WAN, MAN, LAN, or PAN) and organizational scopes (i.e., intranet or extranet), network topologies, and network devices, and provides all types of network services, such as World Wide Web (WWW), email, ﬁle sharing, and audio and video streaming. Instead of the open system interconnection model from the International Organization for Standardization (OSI/ISO), the internet uses the transmission control protocol/internet protocol (TCP/IP) suites to establish and maintain communication between connected devices. This unit ﬁrst introduces the history of the internet and WWW; the concept, protocols, and services of TCP/IP protocol stack; and security issues related to internet communi- cation.

### History of the Internet and the World Wide Web

The idea of the internet originated in the early 1960s with the invention of the packet switching network (Cohen-Almagor, 2013). Gradually, it has developed through several phases and the protocol standard called TCP/IP was introduced in the 1980s. The prolif- eration of modern internet started after the invention of the WWW in the 1990s (Forou- zan, 2013).

###### 1960—1970: Development of Packet Switching (ARPANET)

Circuit switching In circuit switching technology, the com- municating nodes are connected through a dedicated link (Comer, 2015).

Before the genesis of the internet, telegraph and telephone networks were considered the cutting-edge technology for communicating between distant devices. Telegraph was used to exchange textual messages, and the telephone was used for audio communica- tion. Both of these technologies were based on circuit switching technology and used to exchange data at a constant rate. In 1961, the Massachusetts Institute of Technology (MIT) developed the idea of packet switching, which allowed for data transmission with variable bit rate (Cohen-Almagor, 2013). In 1964, the Rand Institute implemented packet switching for military networks to ensure secured audio transmission.

In 1965, the British National Physical laboratory (NPL) introduced the idea of the “packet” (Cohen-Almagor, 2013). The invention of the packet and packet switching ini- tialized the founding of the internet. The ﬁrst packet switched network was implemen- ted in 1967 by the Advanced Research Projects Agency (ARPA) of the United States Department of Defense to create a small computer network called ARPANET, which stands for advanced research projects agency network (Kurose & Keith, 2017; Leiner et al., 2009). The idea of ARPANET was to connect computers from different manufacturers

The Internet Protocol Suite

to a specialized computer called the interface message processor (IMP). In 1969, ARPA- NET was able to connect four nodes via interface message processors (IMPs) located at four different universities the University of California, Los Angeles, University of Califor- nia, Santa Barbara, Stanford Research Institute, and the University of Utah (Forouzan, 2013; Leiner et al., 2009).

###### 1970—1980: Development of TCP (ALOHANet, PRNET, and SATNET)

In 1972, ARPANET was able to connect ﬁfteen nodes. The ﬁrst protocol used to commu- nicate between the end nodes is called the network control protocol (NCP). The initial ARPANET was a single network. Later, in 1972, ARPANET initiated the internetting project to connect nodes from different networks with different properties, such as variant transmission rates or packet sizes (Kurose & Keith, 2017; Cohen-Almagor, 2013). This project introduced the idea of a gateway, which is a device used to connect two differ- ent networks. The same research group introduced the idea of transmission control protocol (TCP) in 1973 (Forouzan, 2013). Along wiht ARPANET, other packet switching net- works were introduced in mid-1970s, including the microwave-based ALOHANet devel- oped by the University of Hawaii, the Packet Radio Network (PRNET) from the Defense Advanced Research Projects Agency (DARPA), and the Atlantic Packet Satellite Network (SATNET) from BBN Technologies and the Advanced Research Projects Agency (Forou- zan, 2013; Leiner et al., 2009).

###### 1980—1990: Growth of TCP/IP (BITNET, CSNET, MILNET, NSFNET, and ANSNET)

After the development of other technologies beyond ARPANET, TCP was divided into two parts: transmission control protocol (TCP) and internet protocol (IP) (Forouzan, 2013). The combination of TCP and IP was named the TCP/IP protocol stack. The purpose of IP is to deal with packet routing; the purpose of TCP is to deal with other higher-level services, such as packet structuring, data segmentation and reassembling, and error detection. In 1981, the University of California (UC) Berkeley included TCP/IP with a UNIX operating system. In the same year, the National Science Foundation (NSF) developed the computer science network (CSNET) to connect universities that had no access to ARPANET, due to their lack of collaboration with the U.S. Department of Defense (DoD) (Forouzan, 2013; Kurose & Keith, 2017; Cohen, 2013). The “because it’s there network,” also known as “because it’s time network” (BITNET), was also developed in this year by Ira Fuchs to provide email and ﬁle sharing services to universities (Kurose & Keith, 2017; Leiner et al., 2009).

In 1983, the military network (MILNET) was developed for military services, while ARPA- NET was left for nonmilitary usage. In the same year, TCP/IP replaced NCP as the ofﬁcial protocol for ARPANET. In 1986, NSF developed NSFNET to connect the supercomputer centers with a 1.5 Mbps transmission rate (Kurose & Keith, 2017). In 1990, ARPANET was abolished and NSFNET was adopted. Due to excessive trafﬁc demand, NSFNET was replaced by the Advanced Network Services Network (ANSNet), which is a product of a nonproﬁt organization called the Advanced Network Services (ANS), formed by three companies: IBM, Merit, and Verizon (Forouzan, 2013; Kurose & Keith, 2017).

Packet switching

In a packet switching network, there is no dedicated path between sender and receiver. Packets from the same source can take dif- ferent paths to reach to the destination.

###### The 1990s: Development of the World Wide Web

The 1990s witnessed a massive development in internet applications and services, in which the invention of the World Wide Web played a key role. The WWW was invented by the European Organization for Nuclear Research (CERN) between 1989 and 1991 (Cohen-Almagor, 2013). CERN developed four fundamental elements of the WWW: hyper- text markup language (HTML), hypertext transmission protocol (HTTP), the web browser, and the web server. By 1993, there were around two hundred web servers across the world (Forouzan, 2013; Cohen-Almagor, 2013).

###### Who Controls the Internet? Standards and Administrations

There is no single authority that owns or rules the internet. However, there are various groups and organizations that maintain and regulate the internet’s infrastructure, standards, protocols, and other aspects. Some of these organizations are listed below:

* The Internet Society (ISOC) provides the leadership supports for internet standardi- zation through IAB, IETF, IRTF, and IANA (Forouzan, 2013).
* The Internet Architecture Board (IAB) is the technical advisor of ISOC. It monitors the development of TCP/IP protocol suits and RFCs through IETF and IRTF (Forouzan, 2013).
* The Internet Engineering Task Force (IETF) is responsible for developing and prepar- ing protocol standard drafts known as request for comments (RFCs). IETF is managed by Internet Engineering Steering Group (IESG) (Forouzan, 2013, Kurose & Keith, 2017).
* The Internet Research Task Force (IETF) deals with long-term research topics related to the internet. IRTF is managed by the Internet Research Steering Group (IRSG) (For- ouzan, 2013).
* The Internet Assigned Numbers Authority (IANA) maintains IP address allocation, domain name systems, and media types. It is managed by Public Technical Identiﬁ- ers (PTI), which is afﬁliated with Internet Corporation for Assigned Names and Num- bers (ICANN) (Forouzan, 2013, Kurose & Keith, 2017).

### TCP/IP Reference Model and Protocol Stack

The transmission control protocol/internet protocol is also known as TCP/IP protocol stack, TCP/IP protocol suite, or simply TCP/IP. This protocol suite is basically a set of standardized protocols arranged in hierarchical layers. Each upper layer protocol is supported by the lower layer protocols. Additionally, each protocol is compatible with its corresponding upper and lower layer protocols. TCP/IP is the standard protocol suite for the internet. Traditionally, this protocol suite is divided into four layers: appli- cation, transport, internet, and link layer. Modern internet protocols are divided into ﬁve layers: application, transport, internet or network, link or data link, and physical layers (Forouzan, 2013). The transferred data unit between the sender and receiver is

The Internet Protocol Suite

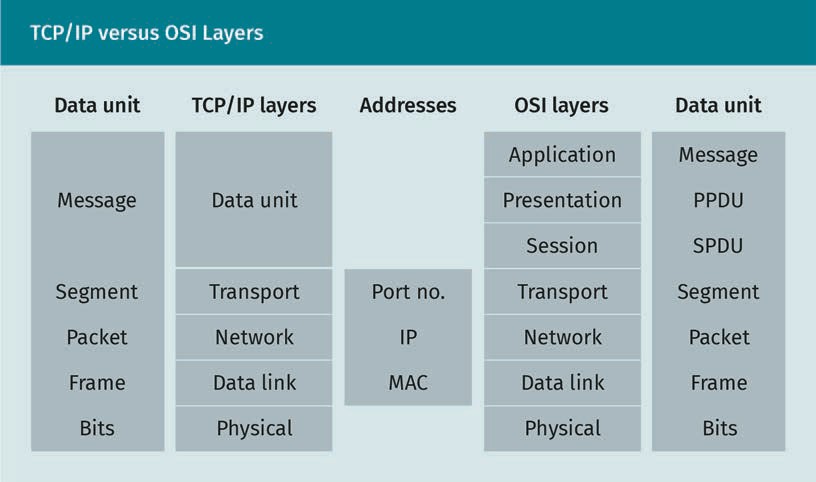
known by different names at different layers. For example, message, segment, packet, and frame are the names for the ﬁrst four layers when dealing with transferred data units.

###### TCP/IP Layers

Unlike the OSI model, TCP/IP does not include presentation and session layers. The services of these two layers are merged into application and transport layers. Descrip- tions of each layer of TCP/IP protocol suite are given below, following a top-down approach.

Layer 5: Application layer

The application layer sits on top of the protocol stack as the ﬁfth layer of the TCP/IP protocol suite. This layer is responsible for process-to-process communication. It establishes logical connections between the application programs or processes instal- led on sender and receiver nodes, such as browsers, email managers, and messengers. The application layer is available only on end nodes, and is absent on the connecting nodes, such as the router, switch, bridge, and hub. The transmitted or received data at the application layer is called a message. There are a number of application layers available for various internet services. Some frequently used application layer proto- cols are hypertext transfer protocol (HTTP) for web applications, simple mail transfer protocol (SMTP) for email forwarding, post ofﬁce protocol (POP) and internet message access protocol (IMAP) for email access, domain name system (DNS) for ﬁnding network layer address, ﬁle transfer protocol (FTP) for sharing ﬁles, simple network management protocol (SNMP) for managing network, TELNET for remote access, and secure shell (SSH) for secured remote access. (Forouzan, 2013).



Layer 4: Transport layer

The next layer is the transport layer, which maintains logical connections between end- to-end system processes. Like layer 5, the transport layer is also not available on con- necting nodes. On the side of the sender, the transport layer receives the message from application layer and breaks it into equally sized, smaller pieces, which it encapsulates with transport layer header bits. The encapsulated pieces are called segments. A seg- ment is the transferred data unit in the transport layer. The header bits contain the transport layer addresses of the source and destination, called source and destination port numbers. The purpose of the port number is to identify the desired application program on a device. The main service of a transport layer protocol is reliable packet delivery, which is done through multiple services, such as error detection, in-order packet delivery, packet loss detection and retransmission, or congestion detection and congestion control. Two of the most important transport layer protocols are the trans- mission control protocol (TCP) and user datagram protocol (UDP). TCP is used for the services that requires reliable data transfer, e.g., email or ﬁle transfer, whereas UDP is used for the services that do not guarantee reliable data transfer, such as video streaming or real-time services. Stream control transmission protocol (SCTP) is a trans- port layer protocol that is aimed at multimedia services (Forouzan, 2013; Kurose & Keith, 2017).

Layer 3: Network layer

The third layer of the TCP/IP protocol stack is called the internet layer or network layer. This layer is responsible for maintaining communication between system programs between sender and receiver. The network layer encapsulates segments with network layer header bits and forms them into datagrams, simply known as packets. A packet is the transferred data unit at the network layer level. A network layer address is called an internet protocol (IP) address. The purpose of the IP address is to identify a device or operating system, or, more speciﬁcally, the networking system program. A router is considered a network layer device, as its highest layer is the network layer. The two main services of network layer protocols are packet forwarding and routing. IP is the main protocol of this layer, which is responsible for packet formatting and routing. Some other important network layer protocols are internet control message protocol (ICMP), dynamic host conﬁguration protocol (DHCP), address resolution protocol (ARP), and internet group management protocol (IGMP) (Forouzan, 2013; Kurose & Keith, 2017).

Frame The frames are the transferred data units at the link

layer.

Layer 2: Link layer

The link layer is also known as the data link layer. On the sender’s side, the data link layer protocols receive network layer packets and encapsulate with link layer header bits to form a frame. This layer maintains a logical connection between the network interface cards (NICs) of sender and receiver nodes. An NIC is a hardware device or cir- cuit that connects a node to the network or the internet. A node can connect to a net- work through several ways, such as Wi-Fi, twisted pair cable based ethernet, optical ﬁber, satellite radio link, or cellular radio link. If a node is connected to the network in multiple ways, then each type of link requires a separate NIC. The services provided by link layer protocols include frame formation, error detection, media access control, and ﬂow control. Some important link layer protocols are ethernet, IEEE 802, and point-to- point protocol (PPP) (Forouzan, 2013; Kurose & Keith, 2017).

The Internet Protocol Suite

Layer 1: Physical layer

The physical layer represents the physical connection or link between the sender and receiver. The physical link can be in the form of copper wire, coaxial cable, optical ﬁber, or electromagnetic wave. The physical layer carries the bits of a frame as electric, elec- tromagnetic, or optical signals (Forouzan, 2013, Kurose & Keith, 2017).

Exercise

Download and open sample captured packets (.pcap or .cap ﬁles) for HTTP and TCP protocol from Wireshark’s website (Wireshark foundation, n.d.). Now, investigate the transferred data unit at each protocol layer. Find the packet sequence number, number of layers, header bit lengths, and addresses at different layers. Also, investigate what information you can extract from different layers.

### Examples of Internet Protocols and Services

TCP/IP protocol stack is a collection of protocols for the internet. These protocols are divided into layers, and there are multiple in each layer to provide various services. The most common protocols of TCP/IP suite are discussed in this section.

###### Application Layer Protocols

Hypertext transfer protocol (HTTP)

HTTP deﬁnes how a client-server program will be coded to retrieve web-based objects from a server. The transfer of an object from server to client is done through request and response messages, which are deﬁned by HTTP. HTTP operates over port no. 80 (Solomon & Kim, 2021). The client sends a request message to the server that contains the port no. 80, and the uniform resource locator (URL) or web address of the desired object, such as HTML ﬁle, text or document ﬁle, image, audio, or video (Comer, 2015, 2018; Kurose & Keith, 2017). The server replies with the response message that contains the requested object. HTTP connection can be either persistent or nonpersistent (Kur- ose & Keith, 2017).

File transfer protocol (FTP)

FTP is used to transmit ﬁles from server to client. FTP is a better solution to transfer larger ﬁles compared to HTTP. In addition, two systems may have different ﬁle naming convention, different directory system, or different ways of formatting data, all which are issues that FTP considers (Comer, 2015; Comer, 2018). FTP operates on TCP port num- ber 21 (Solomon & Kim, 2021).

Simple mail transfer protocol (SMTP)

SMTP is used to forward emails from a client to an email server or from one email server to another. SMTP operates on TCP port number 25 (Kurose & Keith, 2017).

Post ofﬁce protocol version 3 (POP3)

POP3 is used to retrieve email to the receiver’s computer or system from the email server. First, the client establishes a TCP connection on port 110, then they send the username and password for the authentication process (Kurose & Keith, 2017; Tanen- baum & Wetherall, 2014). After passing through the authentication process, the client should be able to see the received email list. POP3 allows the client to keep or delete an email after retrieving it.

Internet message access protocol version 4 (IMAP4)

Similarly to POP3, IMAP4 is also used for retrieving email from server. However, unlike POP3, IMAP4 allows the user to organize emails in directories or folders. It also allows for a search option. IMAP4 uses port 143 or 993 for a secured TCP connection (Tanen- baum & Wetherall, 2014).

Domain name system (DNS)

DNS protocol is used to retrieve IP address from hostname. DNS maintains a distrib- uted database through a hierarchy of DBS servers. DNS operates the user datagram protocol (UDP) port number 53 (Kurose & Keith, 2017). In general, DNS is used by appli- cation layer protocols, such as HTTP and SMTP, to translate a hostname to an IP address.

Simple network management protocol (SNMP)

SNMP is an application layer protocol to manage and monitor networks and devices. The protocol is employed on several servers, known as manager stations, to control and monitor other devices, known as agents. SNMP accomplishes its task through two subprotocols: structure of management information (SMI) and management informa- tion base (MIB) (Kurose & Keith, 2017).

###### Transport Layer Protocols

Transmission control protocol (TCP)

TCP is used for reliable data transfer, meaning it provides the following services: error detection, ordered delivery, ﬂow control, congestion detection and control, packet loss, and retransmission. Error detection is done through internet checksum. TCP uses a buf- fer on the receiver side to ensure order and delivery. Flow control is done by keeping the transmission window lower than the free buffer space in receiver’s system. In TCP, the receiver sends an acknowledgement each time a packet is received. The sender side uses a timer to track delayed acknowledgement. If the acknowledgement time exceeds the “timeout” period, then the packet is detected as a lost packet. Retransmis- sion is done to recover the lost packet. There are several variants of TCP used to handle the network congestion. TCP Tahoe and TCP Reno are the two most common variants (Kurose & Keith, 2017).

User datagram protocol (UDP)

TCP ensures ordered delivery and retransmission due to lost packets, causing TCP to incur undesired delays for real-time services. UDP is a transport layer protocol designed for real-time services or services where the bit rate is guaranteed. UDP is a

The Internet Protocol Suite

connectionless protocol, which means the receiver does not acknowledge packet deliv- ery. Consequently, UDP does not provide ﬂow control, packet loss detection, retrans- mission, ﬂow control, and congestion control services; however, UDP does provide error detection using a checksum mechanism (Kurose & Keith, 2017).

Real-time transport protocol (RTP) and stream control transmission protocol (SCTP) RTP and SCTP are two types of transport layer protocols that are used for real-time multimedia services. Though these are transport layer protocols, the RTP and SCTP datagrams are encapsulated by UDP header bits (Tanenbaum & Wetherall, 2014). Unlike applications, payload RTP does not have any speciﬁc port number. The port number for RTP is chosen arbitrarily, but it is always an even number; the following odd number is used as the port number for companion protocol SCTP. RTP supports only one type of data; in the case of multiple data types, SCTP is used as companion protocol.

###### Network Layer Protocols

Internet protocol (IP)

IP is the network layer protocol of TCP/IP protocol suite. IP has two versions: IPv4 and IPv6. IPv4 uses a 32-bit network layer address, while IPv6 uses a 128-bit network layer address (Kurose & Keith, 2017). In the near future, IPv4 will not be able to accommodate rapidly changing internet devices, which is why IPv6 is gradually replacing IPv4. How- ever, this replacement process could take a few decades, and currently a majority of internet devices are still using IPv4. Besides IP, the TCP/IP network layer provides few supplementary protocols, such as dynamic host conﬁguration protocol (DHCP), internet control message protocol (ICMP), internet group management protocol (IGMP), and address resolution protocol (ARP) (Tanenbaum & Wetherall, 2014).

When a new device comes to a network, it initially does not possess any IP address. The IP address can be setup by manually or automatically. DHCP is the protocol to setup IP address automatically to a new device to a network. First, the new device or client broadcasts a message called discovery message, where the source and destination IPs are set to 0.0.0.0 and 255.255.255.255, respectively. Source and destination port numbers are set to 68 and 67, respectively (Kurose & Keith, 2017). Then, the DHCP server replies with an offer message that includes an IP address for the client.

ARP works below the IP protocol. The purpose of this protocol is to translate network layer address to link layer address, meaning it retrieves the MAC address of a device based on its IP address. The sender broadcasts the destination’s IP address, which all nodes receive; however, only the destination node matches the IP address and returns the MAC address to the sender. All other nodes discard the message (Forouzan, 2013). ARP is deﬁned in RFC 826, 5227, 5494 (Arkko & Pignataro, 2009).

ICMP is an integral part of IP and provides error reports about datagram processing. For example, if a datagram is not delivered, if a gateway does not have enough buffer space to process the datagram, or if the gateway has a better route to redirect the data, these could be reprted. (Forouzan, 2013). ICMP is deﬁned in RFC 792, 777, 760 (Postel, 1981).

IGMP provides group membership reporting for multicasting services. The IP host uses this protocol to report the immediate multicast router about multicast group member- ship (Forouzan, 2013). Like ICMP, it is also a part of IP protocol. Nowadays, an updated version of IGMP called IGMPv2 is used, which is deﬁned in RFC 2236, 3376 (Cain et al., 2002).

Interior gateway protocols (IGPs)

Each internet service provider (ISP) can be considered as a domain or autonomous sys- tem (AS). The routing protocol that operates inside a domain is called intradomain routing protocol or interior routing protocol, such as routing information protocol (RIP), open shortest path ﬁrst (OSPF), or border guard protocol (BGP). Different domains may have different IGPs. RIP is based on distance vector routing algorithm, OSPF is based on link-state routing algorithm, and BGP is based on path-vector routing algorithm (Forou- zan, 2013). IGP is deﬁned in RFC 3906 (Shen & Smit, 2004).

Exterior gateway protocol (EGP)

An EGP is used to perform interdomain routing to ﬁnd a path between nodes residing in different domains or ISPs (Forouza, 2013). EGP is deﬁned in RFC 827, 904 (Mills, 1984).

Protocol independent multicast (PIM)

PIM is a “one-to-many” and “many-to-many” multicast protocol. The key characteristic of this protocol is that it creates the routing table based on the information or routing table provided by the unicast protocols, such as distance vector or link state protocol (Forouzan, 2013). PIM is deﬁned in RFC 7761, 8736 (Venaas & Retana, 2020).

###### Link Layer Protocols

Reverse address resolution protocol (RARP)

Unlike in ARP, where an unknown MAC address is searched based on a known IP address, in RARP, an unknown IP address is searched based on a known MAC address (Forouzan, 2013). RARP is deﬁned in RFC 903 (Finlayson et al., 1984). An updated version of RARP, called dynamic RARP (DRARP) is deﬁned in RFC 1931 (Brownell, 1996).

Neighbor discovery protocol (NDP)

This link layer protocol of TCP/IP protocol suite utilizes IPv6. NDP is used to gather net- work information, such as discovering presence of nodes, ﬁnding routers, and ﬁnding reachability information of the routes to the neighbor nodes. NDP is deﬁned in RFC 4861, 5942, 6980, 7048, 7527, 7559, 8028, 8319, 8425 (Department of Defence High Perform- ance Computing Modernization (DoD HPC), n.d.).

Exercise

Download and open sample captured packets (.pcap or .cap ﬁles) for HTTP, SMTP, DNS, TCP, UDP, SCTP, DHCP, and ARP protocols from Wireshark (Wireshark Foundation, n.d.). Investigate the packets of each protocol, ﬁnd types of messages exchanged for each protocol between sending and receiving nodes, and ﬁnd the addresses and contents of each message.

The Internet Protocol Suite

### Security Aspects of Communication on the Internet

In the modern world, almost all aspects of life are dependent on the internet, including healthcare, ﬁnances, and education. At the same time, various aspects of life are made vulnerable due to the security issues of the internet. Ensuring secure communication is crucial. Securing communication means ensuring conﬁdentiality, integrity, and availa- bility. The combination of these terms is referred to as the CIA triad (Stallings, 2017).

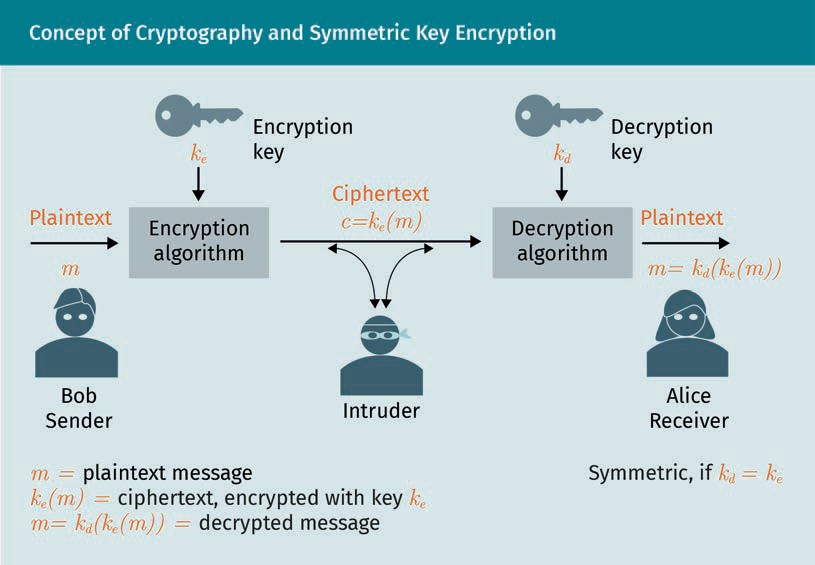
* Conﬁdentiality refers to preventing disclosure of data of information to unauthor- ized users.
* Integrity refers to preventing the editing or corruption of data by unauthorized users.
* Availability implies that the authorized user should be able to access the resources on demand. An unauthorized user should not be able to access the resources.

The key to internet security is cryptography. There are two main types of cryptography techniques: symmetric and asymmetric (also called public key) cryptography. There are various types of cryptography techniques categorized under these two types. In this section, we will focus on the most contemporary and widely used cryptography techni- ques.

A cryptography scheme is made up of an encryption and a decryption algorithm. Each algorithm takes two inputs and produces an output. The two relevant inputs for

encryption algorithms are plaintext or message 7 and key ke , and the output is a ciphertext or encrypted text, c = ke 7 . The relevant inputs for a decryption algo- rithm are a ciphertext c and key k’ , and the output is the decrypted text ’ = k’ c = k’ ke 7 . If the encrypted text was decrypted correctly, then the decrypted text is equal to the plaintext (Stallings, 2017).

The following ﬁgure illustrates the idea of cryptography technique. The ﬁgure shows the sender, Bob, sending a message 7 to the receiver, Alice, in a secured manner, in order to protect it from intruders.



###### Symmetric Key Cryptography

In symmetric key cryptography, both the encryption and decryption algorithms use the same key, i.e., ke = k’ = k. The simplest example of a symmetric key cryptogra- phy is Caesar cipher cryptography (CCC). The key to the encryption algorithm of CCC var- ies from zero to 25, i.e., k ∈ 0, 1, 2, …, 25 (Stallings, 2017). Using the key k, the encrypted text is calculated by replacing any letter K of the plaintext with K + k th

from the alphabet. For example, if the key k = 1, and the plaintext 7 = c8t,

then the encrypted text c = k 7 = K + k 7B’ 26 = ’bA. If k = 2, the encrypted text or ciphertext c = k 7 = K + k 7B’ 26 = ec0. The decrypted text is calculated by the formula ’ = K − k 7B’ 26*.* However, CCC is a primitive example of symmetric key cryptography and it is too easy to crack the encryp- ted code.

There are many other examples of symmetric key cryptography; however, most of these are easy to decode or crack using modern day computers. The advanced encryption standard (AES) is the most frequently used symmetric key algorithm. Data encryption standard (DES) is another popular algorithm; however, it is not overly difﬁcult to crack DES code using powerful computers, hence an improved version, called 3DES, is used instead of DES. AES, DES, and 3DES are known block ciphers, because each of these techniques breaks the plaintext into ﬁxed size blocks before performing encryption on each block. Each of these techniques uses multiple steps called rounds to accomplish encryption, decryption, and generate a key. The following table shows the differences among the characteristics of AES, DES, and 3DES algorithms (Stallings, 2017). One limita- tion of symmetric key encryption is that if an intruder can capture the key, they can

The Internet Protocol Suite

decipher all ciphertext. Therefore, the key has to be shared between sender and receiver in a secure way. The sharing of keys between sender and receiver is done through asymmetric key encryption or public key encryption.

|  |  |  |  |
| --- | --- | --- | --- |
| Difference between AES, DES, and 3DES Parameters | | | |
|  | AES | DES | 3DES |
| Key size (bits) | 128/192/256 | 56 | 168 |
| Plaintext block size (bits) | 128 | 64 | 64 |
| Number of rounds | 10/12/14 | 16 | 48 |

###### Asymmetric Key or Public Key Cryptography

In an asymmetric key cryptography technique, the key for the encryption algorithm is different from the key for the decryption algorithm ke ≠ k’. The encryption algo- rithm requires a pair of keys, known as the sender’s public key (k4+A) and private key (ks+ ,). The decryption algorithm also requires two keys, known as the receiver’s public key (k,+ A) and private key (k,+ ,). The public keys can be known by any users; however, a private key is known to only the owner of the key. The ciphertext is found by encrypting the plaintext (7) with the receiver’s public key (k,+ A), hence, c = k,+ A 7 *.* The decryption is done using the receiver’s private key, hence, ’ = k,+ , c = k,+, k,+A 7 = 7. The most popular asymmetric key cryp- tography is the RSA algorithm, named after the Ramir, Shivest, and Adleman, who ﬁrst

publicly described the algoritm in 1978. Some other popular asymmetric key cryptogra- phies are the elliptic curve and the Difﬁe-Hellman algorithm (Stallings, 2017; Kurose & Keith, 2017).

The RSA algorithm

The RSA algorithm has the following three stages (Stallings, 2017):

1. Key generation. In order to generate public-private key pair, the algorithm chooses two prime numbers p and / given than p ≠ /. Then, calculate n = p/, calcu- late cp n = p − 1 / − 1 , select integer e given that cp n and e are coprime, and 1 < e < cp n . Calculate ’ = e−1 7B’ cp n . Now, the public key kpA = e, n *, and* private key kp, = ’, n
2. Encryption. The ciphertext is calculated by c = 7e 7B’ n
3. Decryption. The decrypted text is calculated by ’ = c’ 7B’ n

Suppose 7 = A, k,+A = 7, 187 , and k,+, = 23, 187 . To calculate the ciphertext ﬁrst, character A must be converted to a decimal value. The decimal value assigned to A by the American Standard Code for Information Exchange (ASCII) is 65

(Carnegie Mellon University (CMU), n.d.). So, the ciphertext,

c = 657 7B’ 187 = 142. Decrypted text,

’ = 14223 7B’ 187 = 65 = A.

Authentication using public key cryptography

Besides encrypting symmetric key, another use of public key cryptography is user authentication. Once a receiver receives a message, it needs to ensure that the mes- sage was sent by the actual user. This can be done by encrypting the message m with the private key of the sender (ks+ ,) and attaching it to the encrypted text. The receiver decrypts it by using the sender’s public key (ks+ A). Only the sender can have the private key, so, the message can only be decrypted correctly if it is sent by the authentic user. However, encrypting the entire message is a time-consuming process; a faster solution to this is applying a hash function (Stallings, 2017; Kurose & Keith, 2017).

Hash function

The main idea of a hash function is to produce a ﬁxed length output irrespective of the input length. A 128-bit hash function produces a 128-bit output for a given input, but the input length can vary. The output can vary depending on different inputs, but the output length will be ﬁxed at 128 bits. A good hash function has the following proper- ties:

* For a message 7, an n-bit hash function will produce the output h 7 of n-bit length.
* It should be computationally infeasible to ﬁnd the inverse of the hash function

h−1 h 7 = 7. This property is called preimage resistance.

* It should be computationally infeasible to produce a message 7’ for which the hash value h 7’ = h 7 . This property is known as second preimage resistance.

Message digest 5 (MD5), secure hash algorithm-1 (SHA-1), and SHA-2 are frequently used hash algorithms.

Digital signature

The receiver needs to make sure that the received message is from the authentic user, and that the received message is not edited or corrupted. This is achieved through a digital signature, which has two purposes: ensuring authentication and integrity. In a digital signature, ﬁrst the hash value of the message h 7 is calculated. Then, h 7 is encrypted with the sender’s private key to produce the digital signature

s = ks+ , h 7 . The message m is encrypted with the receiver’s public key to ﬁnd the ciphertext c = k,+ A 7 for conﬁdentiality. The signature is then attached or merged with the ciphertext s c = ks+ , 7 k,+A 7 and sent to the receiver. The receiver uses the sender’s public key to decrypt the received hash value ks+A ks+, h 7 . The receiver uses its private key to decrypt the received message ’ = k,+ , k,+A 7 . Then the receiver calculates the hash value of the decrypted message h ’ = h k,+, k,+A 7 . If h ’ = k,+A ks+, h 7 or h k,+, k,+A 7 = ks+A ks+, h 7 then the sender is authentic, and the integrity is preserved (Stallings, 2017; Kurose & Keith, 2017).

The Internet Protocol Suite

###### Securing TCP/IP Layers

TCP/IP protocol stack does not include a security layer. However, an additional security layer is included with TCP/IP and various security protocols are used at each layer to ensure their security. Some of the most commonly used protocols for the various layers are discussed below.

Application layer security

There are many security protocols at the application layer that are designed to secure various applications. Some of the application layer security protocols are given below:

* hypertext transfer protocol secure (HTTPS). HTTP does not provide any security measure, so HTTPS offers a secure version of this protocol. HTTPS supports an encryption mechanism to protect data from eavesdropping and man-in-the-middle attacks. HTTPS is essentially HTTP that works on the secure sockets layer/transport layer security (SSL/TLS). HTTPS may use port 443 instead of 80 (Tanenbaum & Wetherall, 2014).
* ﬁle transfer protocol secure (FTPS). This is an extension of the ﬁle transfer protocol (FTP), which, like HTTPS, also works on SSL/TLS. It is deﬁned in RFC 4217 (Ford-Hutch- inson, 2005).
* secure shell (SSH). This protocol provides secure remote access and works on port 22 (Tanenbaum & Wetherall, 2014).
* pretty good privacy (PGP). This protocol is used for secure authentication, conﬁden- tiality, and digital signature (Stallings, 2017).

Transport layer security

Transport layer and application layer security is achieved by introducing a security layer between the application and transport layer, called the secure sockets layer (SSL). The updated version of SSL is known as transport layer security (TLS). TLS primarily uses AES for conﬁdentiality, Difﬁe-Hellman (elliptic curve mode supported) or RSA for key exchange, and secure hash algorithms (SHA) for hashing. TLS/SSL supports a range of encryption algorithms called cipher suite (Stallings, 2017). The client-server pair agrees upon the cipher suite during the TLS handshake, i.e., during connection.

Network layer security

Network layer security is achieved through internet protocol security (IPsec). IPsec works with both TCP and UDP transport protocols. IPsec protects and encapsulates the entire packet presented to the IP layer, including all upper layer header bits (Forouzan, 2014).

OWASP

OWASP stands for Open Web Application Security Project (OWASP). It is an online plat- form for opensource articles, documentation, software tools, and technologies related to the web application security.

Exercise

* 1. Download and open sample captured packets (.pcap or .cap ﬁles) for HTTPS, and HTTP from Wireshark’s website (Wireshark Foundation, n.d.). Compare the packets and note the differences.
  2. Download sample captured packets (.pcap or .cap ﬁles) for TLS protocol from the same site. Find the types of supported encryption algorithms and the encryption key.

Summary

The internet is the most complicated, widespread, and utilized computer network. The protocol stack for the internet is known as TCP/IP protocol suites. The commu- nication protocols for internet are divided into ﬁve layers—application, transport, network or IP, data link, and physical layers. Unlike the OSI layering model, TCP/IP does not include presentation and session layers. The services of these two layers are merged into the application and transport layers. Some popular application layer protocols are HTTP for web service, SMTP for forwarding email, POP3 and IMAP for retrieving email from servers, and DNS for translating host names to IP address. The most popular transport layer protocols are TCP for reliable transmission and UDP for unreliable transmission, such as real time transmission. The network layer protocol is IP or IPv4. There are multiple supplementary protocols for IPv4 such as DHCP, ICMP, IGMP, and APR. Some commonly-used link layer protocols are RARP and NDP. TLS protocol is used to secure the transport layer. IPsec is used to secure net- work layer communication. Some application layer security protocols are HTTPS, SSH, PGP, and FTPS.



# Unit 4

## Architectures of Distributed Systems

#### STUDY GOALS

On completion of this unit, you will be able to …

… understand the concept of distributed system architectures, such as client-server architecture and service-oriented architecture.

… understand the concept of edge computing, cloud computing, and peer-to-peer computing.

… understand the concept of web- and micro- services.

… analyze various types of distributed systems.

… design and implement distributed systems.

DL-E-DLMCSNDS01-U04

1. Architectures of Distributed Systems

### Introduction

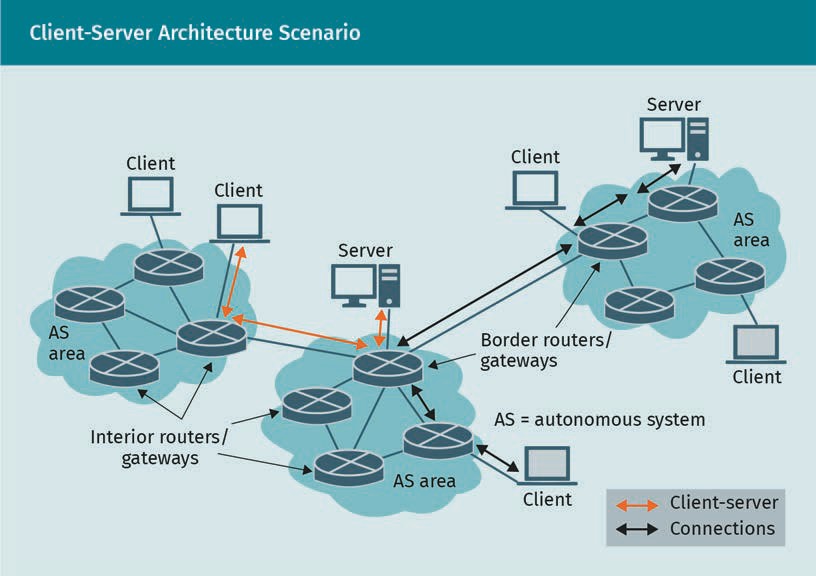
The distributed system has been deﬁned in various ways in different sources. Simply put, a distributed system runs multiple programs or processes on multiple nodes to accomplish a common goal or service. A computer network is a good example of a dis- tributed system, especially the internet. For example, consider a web service where a computer runs a client application, such as a browser, to send requests and receive web services from another computer, which runs a server process such as Apache HTTP web server. The client-server based distributed architecture is most prevalent in the internet. Another common distributed architecture is peer-to-peer (P2P) where a group of computers use a common application or process to share ﬁles among them. The advancement and increased complexity of internet services introduced several advanced variants of client-server architectures, such as service-oriented architectures, edge computing, cloud computing, fog computing, ubiquitous computing, and pervasive computing. This unit sheds light on different types of distributed systems, more speciﬁ- cally, client-server architecture, service-oriented architectures, edge and cloud comput- ing, and peer-to-peer computing.

### Client-Server Architectures

Client-server architecture is the most widely used distributed architecture for computer networks. The terms “client” and “server” refer to software programs. However, in gen- eral, the term “server” is used to refer to a computer with powerful processing speed, memory, and storage capacity, that runs the server program. The client, then, refers to the computer that runs the client program, and is the process that requests and receives service, while the server is the process that provides service to clients (Comer, 2015). The server and client are identiﬁed by a pair of identiﬁers or addresses (the com- puter’s address and the application’s address), i.e., IP address and port number, which are known as a socket. An IP identiﬁes a particular computer or operating system, and a port number identiﬁes a particular process on that computer. In a client-server archi- tecture, the server is always active, but the client is not necessarily always active. The server always listens to receive requests from the client, which starts the connection by sending the connection request to the server through the socket (Kurose & Keith, 2017). The server receives the connection request and accepts it by sending an acknowledge- ment. Once the connection is established, the client sends the request for the service or object, such as a ﬁle or data. The server transfers the requested object to the client and then terminates the connection. The client also ends the connection after receiv- ing the object (Kurose & Keith, 2017; Tanenbaum & Wetherall, 2014). Some widely used client-server architectures are web services, email services, and the domain name sys- tem (DNS) (Forouzan, 2013; Tanenbaum & Wetherall, 2014).

The following ﬁgure illustrates a client-server architecture scenario. The orange and black connections show two separate client-server pairs.

Architectures of Distributed Systems



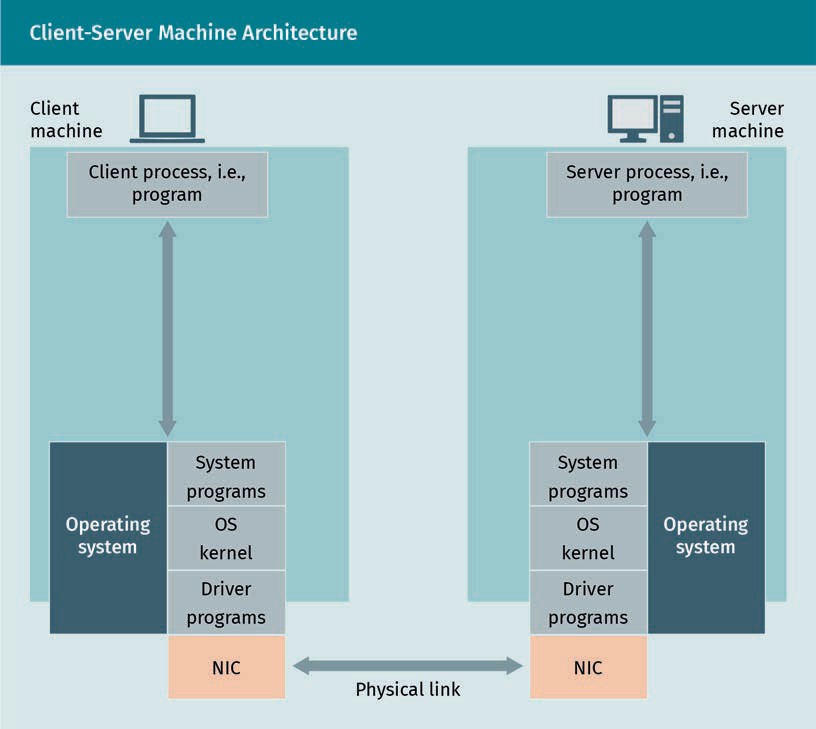
###### Client-Server TCP/IP Layers

Client-server architecture provides internet services, therefore, it follows the TCP/IP protocol stack. For example, a web service is based on client-server architecture, so it follows the TCP/IP layers. The application layer entity for the client is implemented as the client application, such as the browser; the application layer entity for the server is implemented as the web server application, such as Apache server. Client-server archi- tecture works for both connectionless and connection-oriented services, meaning the transport layer protocol can be TCP or UDP (Kurose & Keith, 2017). The network layer protocols can be based on either IPv4 or IPv6. The transport and network layer proto- cols are implemented as system programs of the operating system. The software part of the link layer is implemented as the system programs and drivers, and the hardware part is the network interface card (NIC).

The following ﬁgure depicts the architecture of client and server machines. It shows that the client or server process (i.e., application program) is installed on top of the operating system (OS); this is the application layer part of the client or server. Some system programs are implemented in the OS that deal with the transport and network layer activities of the client or server. In addition, some driver programs are implemen- ted in the OS, which deals with the software part of the datalink layer of the client or server. Finally, the network interface card (NIC) implements the hardware part of the datalink layer of the client or server.

Apache server This is an open-

source HTTP server for modern operat- ing systems includ- ing UNIX and Win- dows.



###### Client-Server Interaction

The client-server interaction is implemented through a series of operations, as shown in the above ﬁgure. The operations are implemented as functions, i.e., system calls. The operations are (Comer, 2015; Kurose & Keith, 2017; Tanenbaum & Wetherall, 2014)

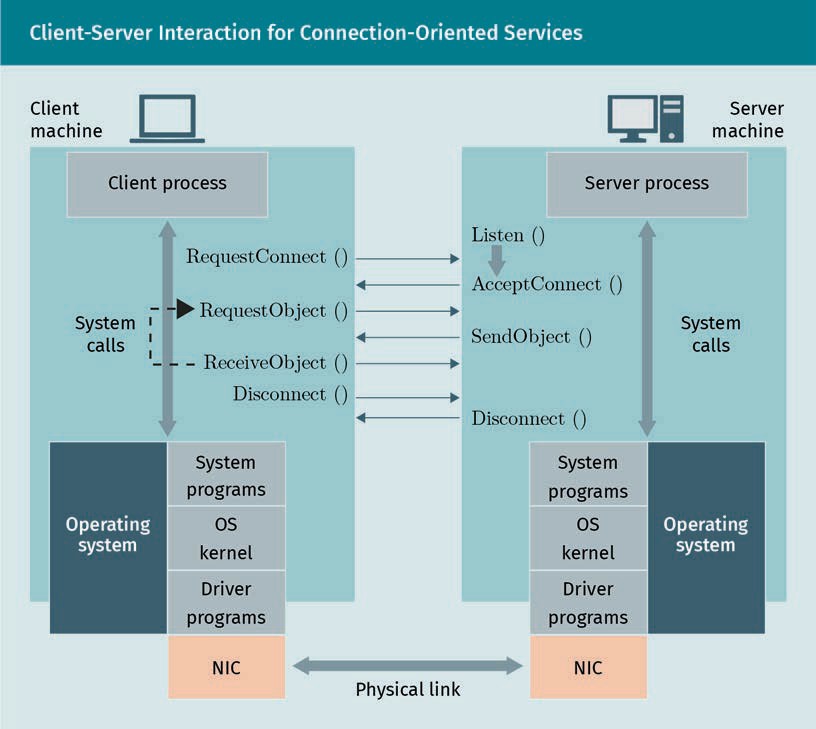
* wait-and-listen. This operation takes place on the server side only. The server machine and process are always active and waiting passively to listen to requests from a client. This is done by implementing a function or system call called, e.g., Listen().
* connection request. The client sends a connection request to the server which is done through a system call, e.g., RequestConnect(). The RequestConnect() func- tion takes sockets from client and server as arguments, which basically contain the IPs of the client and server machines and port numbers of the client and server processes. The server IP identiﬁes the server machine and the server port number identiﬁes the server process. For web services, the port number is 80. Also, for web services, the client initially identiﬁes by the server machine by the host name. The DNS protocol retrieves the IP address of the server based on the host name.

Architectures of Distributed Systems

* connection accept. Once the server listens to the connection request, it identiﬁes the requested application by its port number. If the service is available and the cli- ent socket is valid, then the server calls the system call (e.g., AcceptConnect()) to accept the connection request and notify the client.
* object request. When the connection is established between server and client, the desired service, i.e., the object ﬁle or data, is requested by calling another function such as RequestObject(). The object identiﬁer is passed as an argument to the system call. For a web service, the object identiﬁer is the URL of the target ﬁle.
* object send. Once the object request is received, the server checks the availability of the object. If the object is not available, then it sends an error message; otherwise, the server sends the requested object. This operation is also implemented through a system call, such as SendObject(). This function also returns the data size.
* receive object. The object reception at the client side is also done through a system call, such as ReceiveObject(). This method returns the occupied or free buffer space size.
* disconnect. Once the client receives the entire ﬁle, it may request the next object through RequestObject(). If no further object is required, then it calls the Discon- nect() system call to notify the server that it is waiting to end the connection. How- ever, it does not disconnect immediately, but waits for a certain period of time before disconnecting. Once the server receives the disconnect() notiﬁcation, it ter- minates the connection from the server side by calling Disconnect().

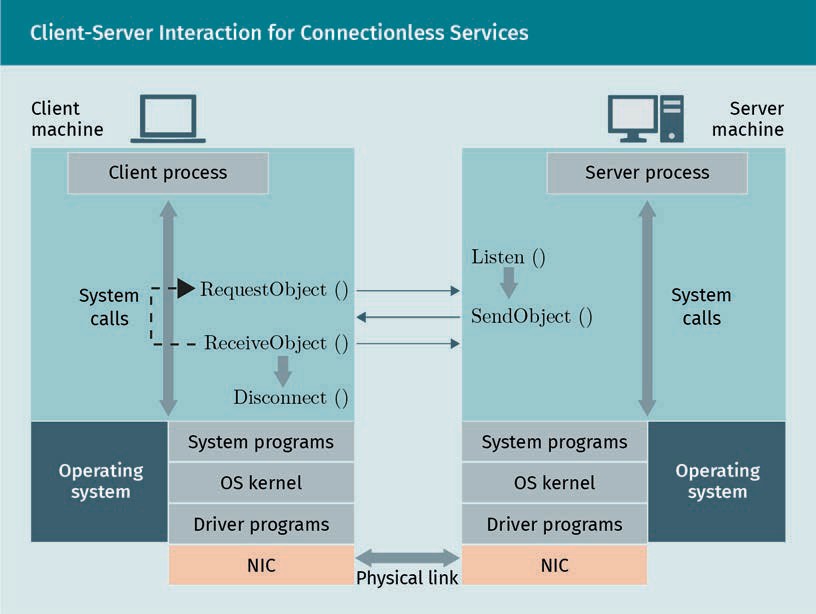
In case of a connectionless service, the connection request and connection accept operations are absent. Also, the client does not send the disconnect notiﬁcation. It simply terminates the connection without notifying the server. On the server side, the connection ends as soon as all of the data have been transmitted. The following ﬁgure shows the client-server interaction for connectionless services (Kurose & Keith, 2017).

The following ﬁgure shows the steps of the TCP based connection-oriented client- server interaction between the client and server machines.



The following ﬁgure shows the client-server interaction for UDP based connectionless services.

Architectures of Distributed Systems



Exercise

Open the Wireshark software and ﬁnd the capturing packet (Wireshark Foundation, n.d.). Then open a browser and surf to a website and spend a short amount of time (about half a minute) opening various pages on that website. Now, ﬁlter out the pack- ets to show only those packets exchanged between your computer and the website you were surﬁng through. Figure out which packets are responsible for setting up the server-client connection. You should ﬁnd the “Client Hello” and “Server Hello” packets. Then ﬁnd the HTTP request message, i.e., Get method. Identify the HTTP response mes- sage and investigate all of the above-mentioned messages.

### Service-Oriented Architectures, Web- and Micro- Services

Service-oriented architecture (SOA) allows the server to be compatible with various applications developed by different programming languages. The primary goal of the SOA is to provide a structure that allows loosely coupled software components, in order to adapt the existing software with newly developed features, while minimizing the expense (Gabhart & Bhattacharya, 2008).

###### Beneﬁts of Service-Oriented Architechtures (SOA)

IT technology and the software industry change rapidly, which is why software infra- structure should be ﬂexible to new features. However, the adaptability cost should be low. SOA was developed as a solution to these two key requirements. SOA organizes the services in layers; this layering abstraction allows an organization to leverage its exist- ing resources to wrap up as a service to integrate new features and technologies. This approach avoids rebuilding the service from scratch; therefore, it adapts the new fea- tures faster, management of new features become easier, and it lowers the cost of implementation. Some beneﬁts of SOA are as follows (Endrei et al., 2004):

* + leveraging existing assets
  + easier integration of new features on top of existing assets
  + easier management
  + faster adaptability with new features, services, or technology
  + enabling reuse of existing assets
  + reduced implementation cost

###### Evolution of SOA Standards

To improve the inter-software communication, which is known as distributed comput- ing, several standards were developed in the 1970s, such as distributed computing environment/remote procedure calls (DCE/RPC) and common object request broker architecture (CORBA) (Gabhart & Bhattacharya, 2008). DCE/RPC is a function-oriented solution, while CORBA is object-oriented. These standards allow communication between applications, regardless of the type of programming language or operating system, a feature which is known as interoperability. Achieving interoperability is a pri- mary goal of distributed architecture.

These standards were developed and maintained by a group of vendors (Endrei et al., 2004; Gabhart & Bhattacharya, 2008). These standards were complicated and the lack of communication or agreement among the vendors made it difﬁcult to carry on the development of these standards, which has made it difﬁcult to follow and implement by the other vendors. Later, Microsoft developed the simple object access protocol (SOAP), which is signiﬁcantly simpler than DCE/RPC and CORBA. Also, SOAP is main- tained by a single vendor only. However, an advanced distributed model requires many additional features, such as encryption, authentication, security, service management, and transaction management. A great advantage of SOAP is that it allows new features to be integrated on top of the existing features. In SOA, standards are layered, as shown in the following ﬁgure (Gabhart & Bhattacharya, 2008).

|  |  |
| --- | --- |
| SOA Service Layers and Standards | |
| Service layers | Standard examples |

Architectures of Distributed Systems

|  |  |
| --- | --- |
| SOA Service Layers and Standards | |
| Service management | WS Distributed Management |
| Business process orchestration | WSBPEL, WS-BPEL Extension |
| Security Authentication Authorization Encryption | Web Services Security WS Federation Language  WS-Security Kerberos Binding |
| Transaction management | Web Services Coordination |
| Guaranteed message delivery | Web Services Reliable Messaging |
| Advanced messaging Asynchronous notiﬁcation Attaching ﬁles to messages | Web Services Notiﬁcation Web Services Addressing |
| Basic messaging SOAP | SOAP 1.2, HTTP, etc. |

###### Web Services

Web services architecture is popular because it is distributed in nature and capable of supporting heterogeneous applications via the internet. Heterogenous, in this case, means it is independent of a programming language, operating system, or hardware speciﬁcation, and provides loose coupling between consumer and provider. It is also based on open standards, such as eXtensible markup language (XML), SOAP, universal description, discovery and integration (UDDI), and the web services description lan- guage (WSDL) (Tanenbaum & Wetherall, 2014). Using open standards provides a wide range of interoperability, which means solutions from various vendors are compatible with each other. It also allows us to develop web services without having knowledge about consumer solutions and vice versa. Some key characteristics of web services are as follows (Endrei et al., 2004):

* self-contained. A basic client application developed by XML or HTTP should be sufﬁ- cient, no additional application would be needed. On the server side, the basic web server application and a servlet engine would be sufﬁcient.
* self-describing. Client and server applications only need the knowledge about the format and content of the request and response message.
* modular. Web services are deployed over the web using J2EE, CORBA, DCOM, and other technologies to ensure heterogeneity.
  + independent and interoperable language. Client-server applications are compatible, irrespective of the programming language.
  + based on open standards. A selection of these standards includes XML, SOAP, UDDI, and WSDL.
  + composable. A basic web service can be used as the basis to aggregate complex services.
  + dynamic. Web services should be dynamic.

###### Microservices

Microservice WireMock is a mock server that can be used for testing microservices.

A microservice is an emerging distributed software architecture. The microservice architecture develops a software application as a collection of small independent serv- ices, where each service executes its own independent process. The term “small” is not strictly deﬁned. The processes are heterogenous, which means they can be developed using different languages, and they may have their own data structure and or their own separate centralized management (Baresi & Garigga, 2020).

Microservice architecture is opposite to the monolithic architecture, in which an appli- cation is written using a large programming code (Namiot & Sneps-Sneppe, 2014). The monolithic architecture has various shortcomings, such as slowing down productivity, slow adaptability with new services, difﬁculties with integrating new features, and it prevents developers and vendors from working independently. The lack of scalability and interoperability are further shortcomings of monolithic architecture. Microservice architecture aims to avoid the above-mentioned shortcomings (Dragoni et al., 2017).

Service mesh architectures

Service mesh is an infrastructure layer that provides service to service communication for microservices. This communication is done through a proxy (Khatri & Khatri, 2020). Modern software for distributed computing is broken into small components as serv- ices, so the applications become a network of microservices. Each service runs a dedi- cated business function. To execute a service, it may share data from other services. A service may become overloaded by requests from other services. A service mesh pro- vides the infrastructure to pass requests from one service to another. The service mesh is designed into the application software as an array of network proxies. Once a user requests a webpage, the request goes to a proxy server. If the requested object is unavailable to the proxy, then the request is forwarded to the server. The server sends the object to the proxy server, which forwards it to the user. The next time a user requests the object, it will be sent directly from the web proxy. A service mesh routes the requests between microservices through proxies. Since the microservices run alongside, they are referred to as sidecars (Sharma & Singh, 2020). Without the sidecar concept, each microservice is required to be programmed separately to conduct serv- ice-to-service communication.

Exercise

Docker is a popular open-source tool for providing services within virtualized contain- ers. This enables the easy developing, shipping, and running of software applications. Due to virtualization techniques, Docker allows the user to separate application pro-

Architectures of Distributed Systems

grams from the infrastructure. It also allows the user to manage the infrastructure in the same way one can manage applications. First, browse through Docker’s ofﬁcial web- site to learn more about it. Download and install the Docker system by following the instructions given on the website. Finally, run a test application by following the tuto- rial on Docker’s website (Docker, n.d.).

### Edge and Cloud Computing

Cloud computing and edge computing are advanced distributed computing architec- tures. They are developed based on client-server architecture and are considered advanced service-oriented architectures. The main idea of cloud computing is to pro- vide on-demand IT resources over the internet. “Cloud” refers to a remote storage or computing platform provided by a third party, which is remotely accessible over the internet. Basically, clouds are remote database centers that provide various services over the internet. Although, logically, the cloud is a centralized infrastructure, practi- cally, it is distributed over a group or cluster of servers called data centers (Ward & Barker, 2013). A cloud can provide various types of services to multiple clients. It can provide a client with larger storage, fast and complex computing, or processing plat- forms without deploying any hardware or software resources in the existing machine. Typically, cloud servers are located at the network core, and clients are located at the network edge. Hence, cloud might not be suitable for real-time data processing and analysis due to the long-distance communication between the cloud and clients, i.e., edge devices. This shortcoming can be resolved by placing some data centers close to the edge devices. The time sensitive data processing can be done by the edge data centers, while the time insensitive data processing can be done at remote clouds or data centers at the network core. This enhanced architecture or cloud computing is known as edge computing (Khan et al., 2019).

###### Properties of Clouds

Cloud and cloud computing are described in various ways across sources. The following are key characteristics of cloud computing systems (Ward & Barker, 2013):

* on-demand. A cloud computing system should be able to provide the recourse of services to an authorized user on an on-demand basis.
* broad access. The cloud should be accessible from multiple locations and multiple platforms, such as desktop PCs, laptops, smartphones, Windows OS, MAC OS, and Android OS.
* resource pooling. The cloud provider should be able to pool resources in a shared manner among customers or consumers of various hardware or software speciﬁca- tions.
* rapid elasticity. The provider should be able to provide scalable services to the con- sumers based on their demand.
* measured service. The cloud provider should be able to monitor and meter the services or resources provided to each consumer.

###### Clouds Services

Cloud computing systems can provide various services. These services are typically categorized in the following ways (Ward & Barker, 2013; Rimal et al., 2009):

* + Software-as-a-service (SaaS) refers to end-user applications, which are deployed and maintained by the service provider. The end user does not need to worry about the infrastructure, platform, or underlying maintenance of the applications or soft- ware service. They simply consume the ready product, i.e., the software in which they are interested. For example, this could be the web-based email service from a spe- ciﬁc company. The consumer can access the email on various browsers, different operating systems, and using multiple devices without knowing the underlying infra- structure of the email web-service.
  + Platform-as-a-service (PaaS) provides remote access to the desired platforms, such as operating systems or hardware. It allows the user to install applications on the preferred platform. For example, Kali Linux is a popular operating system that comes with many software and digital forensics tools for testing network security. However, deploying this operating system and tools could be unsafe for the organization’s IT security, or it could be expensive to deploy. A number of cloud providers offer

cloud-based virtual machines and operating systems. The provider offers a wide range of machine speciﬁcations and operating systems.

* + Infrastructure-as-a-service (IaaS) provides the most ﬂexible cloud services. It pro- vides basic networking facilities, hardware virtualization, and storage services.

###### Types of Clouds

Typically, clouds can be divided into three types: private, public, and hybrid (Rimal et al., 2009). The private cloud provides IT resources over the internet and is limited to the conﬁnes of a single organization. The cloud can be deployed and maintained within the organization or by a third party. The public cloud provides cloud services to multiple organizations. Hence, the resources are shared among multiple consumers. The security concerns related to public clouds are higher compared to private clouds. The hybrid cloud infrastructure is a combination of both public and private cloud infrastructures, meaning that some resources and services are dedicated to a single organization, while some other resources and services are shared among multiple organizations.

###### Edge Computing

Edge computing is the distributed computing architecture of a few popular modern technologies, such as the Internet of Things (IoT), 5G, and vehicle-to-vehicle (V2V) com- munication. The main goal of edge computing is to provide low latency while support- ing mobility and location awareness to the real time application. In general, the end nodes, especially the mobile nodes, have lower capacity in terms of processing speed, memory, and storage capacity (Khan et al., 2019). In comparison, core devices, such clouds and data centers, have a higher capacity for processing speed, memory, and

Architectures of Distributed Systems

storage. For this reason, the end device processes light computations and the cloud processes computationally expensive data. However, this architecture is not suitable to serve real time or delay sensitive applications (Khan et al., 2019). This is because if the delay-sensitive data is processed at mobile end devices, it would impose large latency due to the limited processing and memory capacity of the device.

Processing delay-sensitive data is also not a suitable solution because the cloud is located at the network core and mobile devices are located at the network edge. This means that the distance between the cloud and end device may be too large, imposing unwanted latency due to propagation delay, queuing delay, and processing delay at multiple hops. Edge computing provides the solution to this dilemma by moving the cloud and data center to the edge of the network. The distance between the network edge and end device is comparatively much smaller. Thus, it will impose less latency, which makes the edge computing suitable for delay-sensitive applications and mobile nodes (Khan et al., 2019; Liu et al., 2019).

Edge computing characteristics

The key characteristics of edge computing are (Khan et al., 2019)

* proximity. One of the key functions of edge computing is to ensure cloud services at end user proximity. Edge computing brings the cloud computing services at the net- work edge by densely deployed smaller data centers or clouds.
* mobility support and location awareness. Edge computing allows real-time comput- ing for mobile nodes. The mobility support is done though Locator ID Separation Protocol (LISP), which uses two numbers—one is to identify the network locator, and another is to identify the node. These numbers can be IP addresses, MAC addresses, or GPS coordinates.
* low latency. Bringing clouds to the proximity of the end user reduces the physical distance between the cloud servers and end device. It also reduces the number of hops between them, which leads to reduced propagation delay, per router process- ing delay, and queuing delay. Thus, the overall latency is reduced.
* context-awareness. Network load, user location, and other the real-time network information is used to provide context-aware services to the end nodes. Context information are used to improve user satisfaction, i.e., quality of service (QoS).
* heterogeneity. Edge computing should be able to support various software applica- tions, platforms, operating systems, hardware speciﬁcations, and infrastructures developed by various vendors.

Variants of edge computing

Much research has been done in recent years on edge computing. Based on the infra- structure and type of services, several variants of edge computing have been devel- oped. Some of the key variants of edge computing are discussed below (Khan et al., 2019):

* + Mobile edge computing is developed to serve the mobile end points. It is also known as multi-access edge computing (MEC). This technology deploys clouds at the base stations of cellular networks.
  + Fog computing is an edge computing cloud can be connected with numerous sen- sors, known as fog. These sensors may generate and forward data repeatedly. This huge amount of data generated by fog sensors may overload the edge cloud, which is why the data generated by the sensor are ﬁrst processed by the co-located com- puting devices. Fog forwards processed data to the cloud instead of the raw data, which improves the processing performance of the edge cloud.
  + Cloudlet is a small-scale cloud data center deployed in between mobile end devices and cloud at the core. The main goal of cloudlet is to minimize latency for mobile applications.

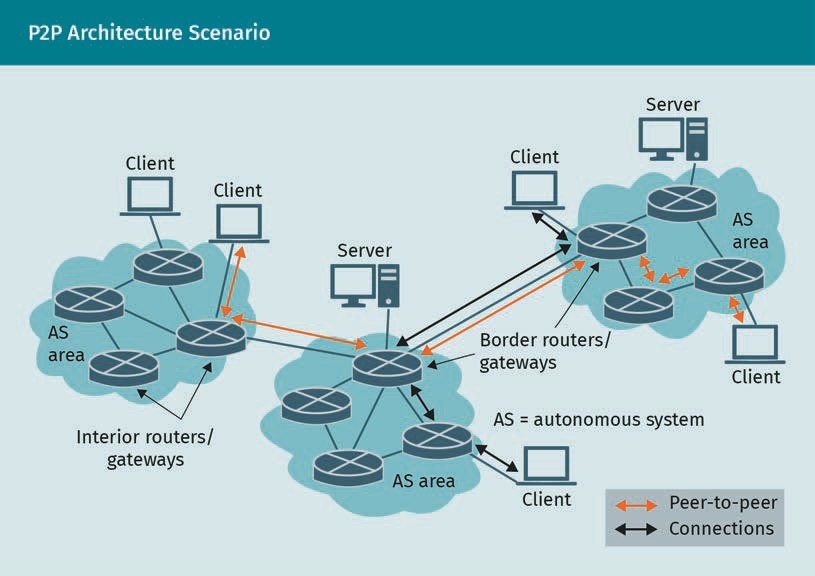
In recent years, a number of companies have developed and launched cloud and edge computing platforms, tools, and products. Some popular edge computing products are Cloudpath, pCloud, ParaDrop, SpanEdge, AWS IoT Greengrass, Azure IoT Edge, and Cloud IoT Edge. A detailed survey of edge computing tools is provided by Liu et al. (2019).

### Peer-to-Peer Computing

In the client-server architecture, the server is always active, while the client can be active or inactive as they please. Also, the client only requests an object, e.g., ﬁles or data, and the server provides the requested object. In P2P architecture, there is no dedicated server, and no node or peer is always active. Nodes, i.e., peers, may connect through multiple connecting nodes without any help from a server (Kurose & Keith, 2017). Each peer runs a common P2P application and each of those contributes to the network by sharing objects or ﬁles. A peer can search an object in the application user interface (API). The API shows the available objects from various sources, i.e., peers. The peer can download the object from any of the available sources. BitTorrent and µ are some popular P2P ﬁle sharing applications. Skype and other voice over IP (VOIP) serv- ices also operate over P2P architecture (Tanenbaum & Wetherall, 2014).

The following ﬁgure shows a peer-to-peer architecture scenario. The orange and black lines show two separate peer-to-peer connections.

Architectures of Distributed Systems



###### BitTorrent

BitTorrent is a P2P protocol for ﬁle distribution. Each peer shares contents to the net- work. They can download ﬁles from each other, and likewise, each peer can upload ﬁles to other peers. A peer contains an index called “torrent” that lists all available peers and their contents, as well as its own location. Initially, the index is empty, but each peer uploads the metadata information of the shared ﬁles to the torrent, and eventu- ally the index contains the whole list of the shared ﬁles from all peers. Torrent does not contain the entire content of the ﬁles, but the metadata information of them. A peer of the P2P network tracks all active peers, which is known as a tracker. Active peers refers to all nodes that are uploading or downloading data. The set of these nodes is called swarm (Tanenbaum & Wetherall, 2014). The same ﬁle can be shared by multiple peers, known as seeds (Tanenbaum & Wetherall, 2014). The tracker provides the information about seed locations. To maintain better service, each node should share ﬁles and upload data. Some nodes are only interested in downloading, and stay away from uploading or stop uploading as soon as it has downloaded the desired ﬁles. These nodes are called leeches. To avoid leeching, the leeches are choked, i.e., if a peer does not upload, then other peers avoid sharing with the leech (Tanenbaum & Wether- all, 2014).

If there is only one tracker, the system will be centralized and it will suffer from issues like overloading or single point failure. Distributed Hash Tables (DHT) is a P2P protocol that allows multiple trackers to decentralize the network services (Tanenbaum & Wetherall, 2014).

###### P2P versus Client-Server Architecture Performance

Suppose there are N nodes. The upload capacity of the nodes is A1, A2, A3, …, An − 1, An , the download capacity of the nodes is ’1, ’2, ’3, …, ’n − 1 , ’n , and the minimum download capacity is ’71n. Suppose all nodes are required to download a ﬁle with the ﬁle size F. The ﬁle can be distributed either using client-server architecture or P2P architecture (Kurose & Keith, 2017).

In case of client-server architecture, the ﬁle will be distributed by the server. Suppose the upload rate of the server is As. Hence, the time to upload the ﬁle by the server is G /As. In order to distribute the ﬁle to N clients, the ﬁle has to be uploaded N times. Hence the total upload time is 2G /As. The download time of the ﬁle by the 1th node is G /’1. The maximum download time is G /’71n. So, the ﬁle distribution time to all nodes is 78= 2G /As, G /’71n (Kurose & Keith, 2017).

In the case of P2P architecture, let us assume that, initially, the no node has the ﬁle. Hence, a node downloads the ﬁle from the server. So, the server upload time is G /As. Now, the rest of the nodes have two options: They can download the ﬁle from the server or from the ﬁrst node. Once the second node downloads the ﬁle, the rest of the nodes have three options to download the ﬁle: server, ﬁrst node, or second node. Hence, the ﬁle distribution time to all nodes for P2P architecture is

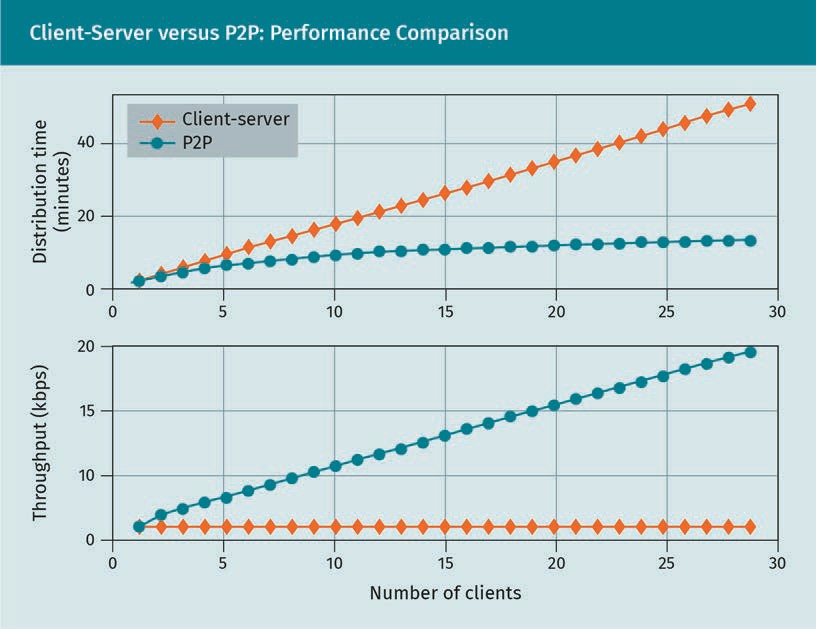
78= 2G /As, G /’71n, G / As + ∑n A1 (Kurose & Keith, 2017). The following

1 = 0

plot compares the ﬁle distribution time and throughput between client-server and P2P

architecture. The plot is generated based on the following parameters: ﬁle size of 5 Mb, client’s upload rate 5 kbps, server’s upload rate 50 kbps, client’s download rate of 50 kbps. The plot shows that if the number of nodes (or peers) increases, then the per- formance of the P2P model gets better compared to client-server model.

Architectures of Distributed Systems



###### Hybrid Architecture

Client-sever and P2P architectures are the most prevalent, but there are many hybrid architectures that combine both client-server and P2P models. A common example is messenger applications (Kurose & Keith, 2017). The messenger server maintains the list and locations of users. Once the users are connected, they exchange information based on P2P architecture.

Summary

Distributed computing is deﬁned in many ways across sources. The essence of dis- tributed computing is to execute multiple processes on various platforms and devi- ces in order to accomplish a common goal. These processes, platforms, and devices can be homogenous or heterogenous. The internet is a good example of distributed computing. Client-server architecture and P2P are the two most popular distributed architectures for the internet. In a client-server architecture, a user computer called the client requests for connection and services (i.e., objects, such as ﬁles or data) to the server computer. On the other hand, a P2P architecture does not need any server computer. All nodes communicate directly through connecting nodes without any help of a server computer. The node of a P2P network is called peer. Peers are

connected through a common application program through which all peers share, upload, and download objects. In a basic client-server architecture, shareable resources are encapsulated as objects. Service-oriented architecture (SOA) is an advanced client-server architecture where sharable resources are encapsulated as services in order to enhance the interoperability and heterogeneity of applications, platforms, and infrastructures. Microservice is an enhanced version of SOA where applications are implemented as a collection of loosely coupled services. Cloud computing is another variant of client-server architecture where a cloud (cluster of data centers) located at a network core provides storage, platform, application, and data processing services to an end user. An enhanced variant of cloud computing is edge computing, where a small-scale cloud is placed at the network edge to keep it closer, in order to ensure QoS for delay-sensitive services and user mobility.



# Unit 5

## Distributed Algorithms and Applications

#### STUDY GOALS

On completion of this unit, you will be able to …

… understand the concept of communication and synchronization of distributed systems.

… analyze the concept of transactions and data management.

… apply the aspects for distributed services and applications.

… design and implement distributed algorithms.

DL-E-DLMCSNDS01-U05

1. Distributed Algorithms and Applications

### Introduction

System call A computer pro- gramming term, which means requesting a func- tion or system pro- gram of an operating

system.

Unlike a centralized system, the programs of a distributed system are installed on dif- ferent geographical locations, machines, and operating systems. Hence, synchronizing the programs across different places and systems is crucial. In a centralized system, all programs are located on a single machine or operating system, meaning that synchro- nization is simply done by following the clock of the system. However, because of dif- ferences in locations and machines, the clock reading may differ from machine to machine in a distributed system. Also, inter-process communication is a big challenge in a distributed system because the callee and caller functions of a system call are located on different machines. A widely-used beneﬁt of a distributed system is that data or ﬁles can be replicated on multiple storages. However, it introduces a new chal- lenge of consistency, which means once a piece of data has been updated, all other replicas have to be updated accordingly. Hence, users or processes on various machines would read the same, i.e., consistent, information. Multiple processes of a distributed system may share a common data or memory. Therefore, distributed algo- rithms must be designed in a way that interrelated processes do not access the resource simultaneously in order to maintain the preferred order of execution, which means the algorithms should be designed to ensure that distributed processes are mutually exclusive. This unit aims to provide an insight to the topics of a distributed system, such as inter-process communication and synchronization, distributed algo- rithms for mutual exclusion, and data consistency of a distributed transaction. Addi- tionally, this unit sheds light on the security issues of a distributed system.

### Communication and Synchronization in Distributed Systems

In a centralized system, multiple programs (i.e., processes) are installed on a common system. The processes can communicate with each other through shared data and memory. Unlike a centralized system, the programs or processes of a distributed system are installed in different systems, meaning communication through shared memory is not possible. Communication generally takes place through exchanging messages between the processes. Modern distributed systems can be immensely large and com- plex, making it difﬁcult to implement inter-process communication through a simple message-passing technique. In this section, we will discuss the widely used communi- cation models for large-scale distributed systems. Furthermore, in a distributed system, processes can share resources. When a resource is shared among multiple processes, they should be synchronized in order to protect the resource from unwanted or acci- dental access. This section will also discuss various process-synchronization techni- ques for distributed systems.

Distributed Algorithms and Applications

###### Inter-Process Communication in Distributed Systems

Two widely used communication models for distributed systems are remote procedure call (RPC) and message-oriented middleware (MOM) (van Steen & Tanenbaum, 2017).

Remote procedure call (RPC)

In RPC, when a process on machine A calls a process on machine B, the caller process on machine A is suspended and the called process executes on machine B. Messages can be passed from the caller to the called process and vice-versa as function argu- ments and returns (Birrell & Nelson, 1984). Though the idea of RPC is simple and straightforward, the implementation is interactive, considering the different address spaces of the machines, speciﬁcation differences, data representation differences, availability of the machine and connection, and parameter passing challenges.

The RPC is known as a request-response protocol, like the HTTP client-server protocol. The client machine initiates the RPC call by sending a request message to the server machine. The request message carries the parameters that are to be passed to the called procedure. It is notable that the word “procedure” means subroutine, i.e., a piece of program. Some key terminologies of RPC are discussed below (van Steen & Tanen- baum, 2017):

* A “stub” is a piece of program that converts passed parameters during an RPC call. In a distributed system, the client and server procedures or functions are located in different machines. Hence, during an RPC call the passed parameters need to be converted. This conversion can be for various reasons. For example, the pointer value of one machine is different from another machine. Also, if the data formatting is different on a client-server machine, then the passed parameter has to be conver- ted.
* A “client stub” is the client-side stub. The client-stub encapsulates a message before transferring to the server.
* A “server stub” is the server-side stub. When the server receives a message, it passes through the server stub.
* “Marshalling” is the process of encapsulating a parameter. The purpose of marshal- ling is to represent data in a neutralized format. For example, some computers (e.g., Intel Pentium) write data in each byte from right to left, called little endian, while some other computers (e.g., older advanced RISC machines (ARM) processors) write left to right, called big endian. Hence, during RPC the data must be represented in a neutral format that is translateable to all types of machines. The goal of marshalling is to represent the parameter values in a neutral format.
* “Unmarshalling” is intended to convert data from a neutralized format to the desired format.

The six steps of RPC are as follows (van Steen & Tanenbaum, 2017):

Little endian

This is an order in which the least sig- niﬁcant value is stored in the lowest address space.

Big endian

An order in which the most signiﬁcant value is stored in the lowest address space.

* + 1. The client procedure calls the client-stub.
    2. The client-stub performs marshalling i.e., encapsulates the parameter to form the message or packet and calls the client's operating system (OS) program for message transmission.
    3. The client's OS (via link and physical layer) transmits the packet from the client’s machine to the server’s machine.
    4. The server’s OS passes the received packet to the server-stub.
    5. The server’s stub performs unmarshalling i.e., decapsulates the received packet.
    6. The server procedure calls the server-stub and repeats the above-mentioned steps in reverse order to respond to the client.

The RPC protocol is standardized in the Request for Comments (RFC) 5531 (Thurlow, 2009).

Message-oriented middleware (MOM)

The way of communication in RPC is synchronous in nature. Hence, the client process remains blocked until a response is received from the server side. Some modiﬁcations are made to make RPC less synchronous in nature. The concept of message-oriented middleware (MOM) is developed to provide asynchronous communication between the server and client in a distributed system.

Middleware is a software which lies between the OS and application software on both the client side and server side (Etzkorn, 2017). Middleware aids to deploy applications over a distributed system by encapsulating the complexities of distributed applica- tions, encapsulating heterogeneity of hardware and operating systems, providing uni- form interfaces, and providing common services. These services allow us to build port- able and interoperable applications.

Generally, MOM provides persistent synchronous communication where the implemen- tation of RPC is intricated. However, MOM supports both synchronous and asynchro- nous modes of communication. It also supports both persistent and nonpersistent modes of communication.

In a synchronous communication, the sender is blocked until the message is received by the receiver or even until a response is received by the sender. This means that the sender can send its second message only after the response from the ﬁrst message is received. In an asynchronous communication, the sender can send the second message immediately after sending the ﬁrst. The second message can be sent before receiving the response from the ﬁrst message, and the second message can even be sent before the ﬁrst message has been delivered to the receiver.

###### Synchronization in Distributed Systems

In a centralized system, synchronization among processes is done by asking the time to the OS. Each process resides in the same OS, thus implementing the synchronization is simple and straightforward. In the case of distributed systems, processes reside in dif-

Distributed Algorithms and Applications

ferent OSs. Thus, there is no idea of a global time. Some systems are faster or slower compared to others. This makes synchronization challenging to implement in a distrib- uted system.

The processes on a distributed system can be synchronized in two ways: using physical clocks or logical clocks (van Steen & Tanenbaum, 2017).

Physical clock synchronization

Each computer has a physical clock or timer. It is basically a crystal that oscillates at a certain frequency based on the amount of voltage applied to it. Each oscillation is counted as a clock tick. However, not all timers provide the exact same time. This time difference is known as clock skew. To overcome the clock skews in a distributed system, an external global clock is needed. The external global clock provides what is known as universal coordinated time (UTC) (van Steen & Tanenbaum, 2017). At the beginning of each second, 40 shortwave radio stations around the world broadcast a short pulse to regulate the time. Also, there are multiple satellites that provide a UTC time-keeping service. The machines in a distributed system are synchronized with UTC. Several algo- rithms have been developed to resolve this synchronization problem, such as network time protocol (NTP), Berkeley algorithm, and the reference broadcast synchronization (RBS).

Network Time Protocol (NTP)

Network time protocol is a client-server model-based protocol for clock synchroniza- tion (van Steen & Tanenbaum, 2017). It operates on connectionless transport protocol, such as user datagram protocol (UDP) (Mills, 1991). In NTP, at time T0, a client requests a timestamp from the server. The server’s clock is synchronized with UTC. The server records the message receive time (T1), then responds to the client with a timestamp which contains the message receiver time (T1) and the message transmission time at server (T2). The client receives the server’s timestamp and records the receive time (T3). Based on this information, the client calculates the round trip-time and estimates the delay. From this estimation, the client minimizes the clock synchronization error (van Steen & Tanenbaum, 2017). The NTP is standardized in RFC 5905, 7822, and 8573 (Mills et al., 2010).

Berkeley algorithm

The server plays a passive role in NTP protocol. This means that in NTP, the server only receives requests from the client, and it responds to them accordingly. The Berkeley algorithm is an active protocol where the time server plays an active role. In this case, the time server is basically a system program that collects time from the clients repeat- edly after a certain period. Based on the received average time, the server suggests for the client to slow down or speed up to adjust the time (van Steen & Tanenbaum, 2017). This protocol is useful when the time server is not synchronized with UTC.

Reference broadcast synchronization (RBS)

RBS protocol is suitable for synchronizing time in wireless sensor networks (Elson et al., 2002). Like the Berkeley algorithm, a sender broadcasts a message considering that all receivers are at a ﬁrst hop distance. The receiver accepts the message and compares

the message construction time and message delivery time, and gives an estimate of the delay (van Steen & Tanenbaum, 2017). The delay contains the propagation time and the message processing time at the network interface card.

Logical clock synchronization

In a distributed system, not all nodes must necessarily agree on a time based on a physical clock. What is important is that the order of events that occur at various nodes be synchronized. This concept of synchronization excludes physical clocks and it is known as logical clock synchronization. Two widely-known logical clock synchronization protocols are Lamport’s logical clocks and vector clocks (van Steen & Tanenbaum, 2017).

Lamport’s logical clock

Lamport’s logical clock utilizes the term “happens before,” indicated by the arrow sym- bol “ ”. A 5 means event A occurs before event B (van Steen & Tanenbaum, 2017). Therefore, the logical time of A, 6 A will be smaller that logical time of 5, 6 5 *,* hence 6 A < 6 5 . Suppose event 6 occurs on another node, and 6 occurs after 5, then the time 6 6 > 6 5 > 6 A . Logical time always increases and never decreases (van Steen & Tanenbaum, 2017). If A 5 is true, then 6 5 = 6 A + n, with n indicating the number of events occurred between event A and B.

Vector clocks

Lamport’s logical clock does not account for causality. Suppose message 71 and 72 are received at node n1. Also, 72 arrived after 71, hence, 6 71 < 6 72 . This relation does not mean the event of receiving message 71 and 72 are related to each other. The message 71 and 72 can be sent by two different nodes that are independent from each other. The vector clocks algorithm considers causality, which means the clock time information also provides information about the relation between messages (van Steen & Tanenbaum, 2017). This is done simply by including some additional information with the clock time value. This information may include the process ID number or node ID number.

Exercise

Write a program to simulate the classic “producer-consumer” concurrency problem. The code should have two threads named “producer” and “consumer.” Assume that the pro- ducer and consumer share a common buffer. Consider that the buffer size is 2. The producer generates data and writes them into the buffer, while the consumer removes data from the buffer. Declare a variable called “counter.” The producer increases coun- ter value by one once it writes data in the buffer, and the consumer decreases the counter value by one each time it removes a piece of data from the buffer. While the counter value is 2, the producer is not allowed to write in the buffer. The consumer is not able to remove a data from the buffer while counter value is zero. Santiago (2020) provides a Python-based tutorial related to concurrency problems.

Distributed Algorithms and Applications

### Distributed Algorithms

The key to a distributed system is concurrency and collaboration among multiple pro- cesses running on various machines. This means that when multiple processes are not causally related, they could have simultaneous access to shared resources. However, if the processes are casually related, they should not have simultaneous access to any shared resource, but instead need to maintain the preferred order of execution. This is known as mutual exclusion (van Steen & Tanenbaum, 2017). Distributed algorithms should be designed in a way that the processes maintain mutual exclusion, i.e., do not have access to a shared resource simultaneously if causally related. The algorithms should also be designed in a way that avoids both starvation and deadlock.

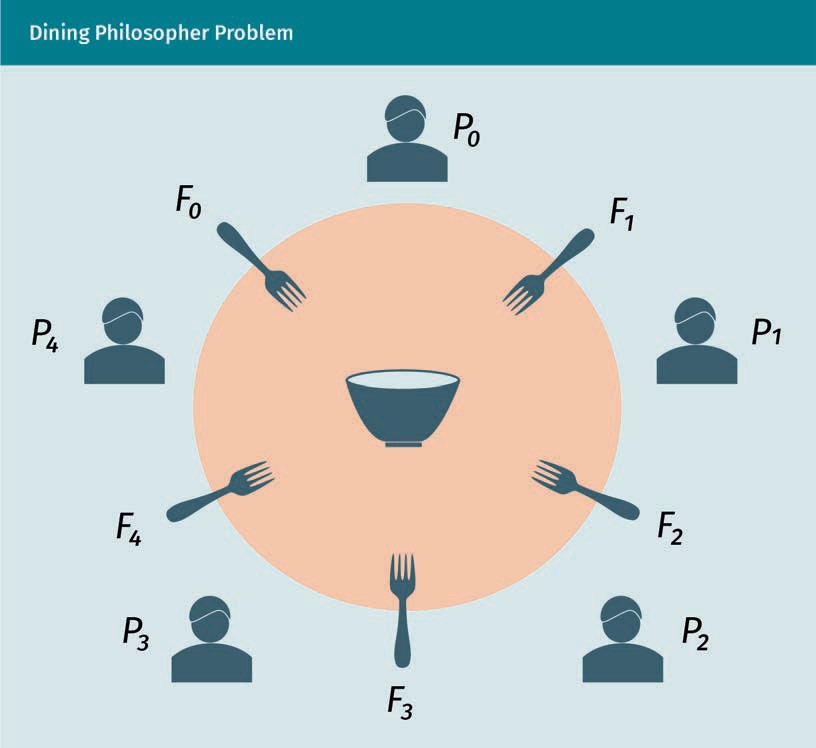
The dining philosophers problem is a classic illustration of deadlock and concurrency issues (Silberschatz et al., 2013). In this situation, a number of philosophers (represent- ing distributed processes) p0, p1, p2, …, pn − 1 are sitting around a dining table in a circular fashion in order to share food from a bowl in the center of the table. There are n forks (representing resources), which are distributed so that between any two philosophers, there is one fork. In the following ﬁgure, there are ﬁve philosophers and ﬁve forks. Now, if all philosophers need to pick up two forks in order to eat, then it would not be possible for them all to do this simultaneously. Only two philosophers could have two forks at any given time. It is possible that all philosophers can each have one fork simultaneously. However, in that case, all of them would be waiting for the next philosopher to release a fork since two are required to eat. Since they are all waiting for another fork, none of them is going to release their fork. This means all phi- losophers will be starving together, which will lead to a deadlock situation. To avoid starvation and deadlock, the solution should consider allowing only four philosophers to sit. A philosopher is able to pick forks only if both forks are available. An odd num- bered philosopher should pick the right fork ﬁrst, then left and vice versa (Silberschatz et al., 2013).

Starvation

This is a situation in which a process unfairly waits for a shared resource while other pro- cesses keep pro- gressing.

Deadlock

This is a situation in which causally rela- ted processes are forced to wait for each other, meaning that no process can progress.



Various types of algorithms for distributed systems have been developed over the years. Some commonly used distributed algorithms are discussed below.

###### Centralized Algorithm

In a centralized algorithm, a process acts as a scheduling coordinator. If a process wants to access to a shared resource, it ﬁrst sends a request to the coordinator proc- ess. If the resource is not occupied by other processes, then the coordinator sends a notiﬁcation granting permission (van Steen & Tanenbaum, 2017). A centralized algo- rithm is easier to implement and is more fair. The coordinator monitors to make sure no process is starving. However, the coordinator suffers from single point failure; if the coordinator is out of order, then the whole system starves. Additionally, if there are too many processes, then the coordinator suffers from request overloading.

Distributed Algorithms and Applications

###### Distributed Algorithms

Distributed algorithms were ﬁrst introduced by Ricart and Agrawala (1981). In a distrib- uted algorithm, if a process wants to access a resource, it sends a request message to all other processes. The message contains the process identiﬁcation (ID), the resource name, and a logical timestamp. If all other processes are not using the resource or the timestamps of all other processes are greater than the timestamp of the requesting process, then they respond with an okay (OK) message. If a process is already using the resource, then it keeps the request message in queue, and sends the okay message after it is done with the resource. If a process receives a request message and its time- stamp is earlier than the request message’s timestamp, then the process does not respond with an OK message. Instead, it utilizes the shared resource ﬁrst, and only then responds with the OK message. Distributed algorithms remove the single point of fail- ure shortcoming of the centralized algorithm. However, it still suffers from N point fail- ure. This means if there are N processes and any of them is out of order, then all other processes will not receive the OK response from the failed process, causing all other processes to starve. If there are too many processes, then there will also be too many broadcast messages, which will reduce the system throughput (Ricart & Agrawala, 1981).

###### Token Ring Algorithm

In this algorithm, the distributed processes create a logical ring in which each process knows the neighboring process (van Steen et al, 2017). Once the process p0 is done with

using a shared resource, it passes a token to the neighbor p1. If p1 is done using the resource, then it passes the token to p2. Whichever process possesses the token has the

right to access the resource. This algorithm is fair, and processes do not starve. How- ever, an issue could be that if a process is out of order, then there must be a mecha- nism to bypass the failed process to form a new ring. Also, the inclusion of a new proc- ess requires reforming the ring.

###### Decentralized Algorithm

Lin et al. (2004) proposed a fully distributed algorithm in which 2 replicas are created of a resource. A coordinator is assigned for each resource replica. Once a process requests to use a resource, the coordinators vote for the access permission. If the num- ber of votes for permission is 7 > 2 /2, then the process is allowed to access the resource.

### Transactions and Data Management (Consistency and Replication)

In a distributed system, the term transaction refers to operating on a set of data or ﬁle that is dispersed over multiple database or network nodes. A modern-day example of distributed transaction could be multiple users or processes working on a common ﬁle on cloud from different machines or network nodes. Data replication is another beneﬁt of distributed transaction. Nowadays, it is common practice to replicate data or ﬁles on cloud storage for backup purposes or to improve quality of service (QoS). Maintaining consistency is crucial for distributed transactions, which means once the ﬁle or data are edited or updated in one machine, they must be updated on all other machines too.

###### Distributed Transaction

A distributed transaction maintains the following properties: location transparency, replication transparency, concurrency transparency, and failure transparency (Traiger et al., 1982). The user can move and access the data from various locations and nodes. Multiple users or processes can also access the data from various locations and nodes simultaneously. This property is known as location transparency. The data can be repli- cated on various nodes, and the user only needs to update only one piece of data in order to update all of them. This property is called replication transparency. Multiple transactions may occur on a piece of distributed data by one or multiple process con- currently, a property which is known as concurrency transparency. According to the property of failure transparency, either all the transactions occur on a piece of data on various nodes, or none of them occur (Traiger et al., 1982).

###### Distributed Replication

Distributed replication is primarily necessary for two reasons: maintaining backups of data and improving quality of service by increasing throughput (van Steen et al, 2017). A server of a large data center receives too many requests from users, causing the server to overload, meaning it cannot satisfy the user demands. One widely practiced solution is replicating the contents to a server that is located closer to the users, thereby dis- tributing the server load to make it not overloaded anymore. The delay is also decreased due to the reduction in geographical distance and hop count. Distributed replication needs to deal with the following two challenges: placement of the replica- server and replica-content (van Steen & Tanenbaum, 2017).

Placement of replica-server

Placement of replica-server means ﬁnding the most suitable geographical location. One of the solutions, proposed by Qiu et al. (2001), has proposed that for each server node, the average distance from user nodes is to be calculated. The server node with smallest average should be chosen as the replica-server. However, they also suggested to consider the distance in terms of bandwidth rather than geographical distance.

Distributed Algorithms and Applications

Radoslavov et al. (2002) proposed a solution which ignores the distances between the clients and servers, rather it focusses on choosing the service area of the server and assumes that all users are uniformly distributed in that geographical region. Szymaniak et al. (2006) proposed an algorithm that ﬁnds the region where content is demanded by the greatest number of nodes. Once this has been identiﬁed, it ﬁnds the node which has the least inter-node latency.

Placement of replica-content

There can be three kinds of content replicas: permanent replicas, server-initiated repli- cas, and client-initiated replicas (van Steen & Tanenbaum, 2017). Permanent replicas are the initial replicas (i.e., initiated locally) and usually fewer in number. Permanent replicas are mainly the data storage for contents. Server-initiated replicas are created by a server process based on a request received by another server process. The pur- pose of the server-initiated replicas is to enhance system performance. Client-initiated replicas are created by a server process based on a request from a client process.

###### Consistency Management

Once a replica is edited, all other replicas must be updated in order to be consistent. There are two types of protocols to propagate updates to the other nodes: push and pull (van Steen & Tanenbaum, 2017). Push protocols are implemented in a server. Once there is a change in data, the push protocol forwards the update to all other nodes that contain replicas. Push protocols are also known as server-based protocols. In contrast, a pull protocol forwards update information only when an update is requested by a cli- ent. Pull protocols are also known as client-based protocols.

There are several models of consistency in distributed systems. Maintaining continuous consistency is crucial and challenging. There are three criteria used to characterize inconsistency: deviation in numerical values between replicas, deviation in staleness between replicas, and deviation in the ordering of update information (van Steen & Tanenbaum, 2017). Some important consistency models are discussed below.

Data-centric consistency

The data centric consistency model is a contract between a server process and data store. As long as the process operates following the rules, the memory performs cor- rectly (van Steen & Tanenbaum, 2017). Some data-centric consistency models are described here:

* Sequential consistency was introduced for multi-process systems with shared mem- ory (Lamport, 1979). A shared memory or data store is sequentially consistent if the read and write operations of multiple processes are performed sequentially by fol- lowing the preferred order speciﬁed in the program’s algorithm.
* Causal consistency is a modiﬁcation of sequential consistency. Two events are said to be causally related if the occurrence of one event depends upon another (van Steen et al, 2017). Suppose process p1 performs the event e1, which writes a piece of data x. Process p2 performs the event e2, which reads the piece of data *x*. If e2 is per- formed before e1 then the reading of x will be erroneous. Two events are called con-

Inter-node latency This is the time that elapses while data are transferred between two nodes.

current events if they are not causally related (van Steen & Tanenbaum, 2017). Causal consistency writes the causally related events in a speciﬁed order in a single machine or in multiple distributed machines. The order of concurrent events may differ on different machines.

* + Eventual consistency is relevant in a large-scale distributed system if data stores do not experience updates for a long time. If this occurs, then all stores eventually become consistent, i.e., they all contain the same data. This type of consistency is one of the key concepts in the basically availabile, soft-state, eventually consistent (BASE) model (Fox et al., 1997). Eventual consistency is applicable to a large-scale distributed system where databases are comparatively more tolerant of inconsis- tency. The domain name system (DNS) is an example that uses eventual consistency. DNS maintains a hierarchy of databases in which lower-level databases keep repli- cas of the top-level database. Any modiﬁcation in the top-level database does not propagate immediately to the lower-level databases. The top-level database sends an update to a low-level database once the low-level database requests an update. Over time, all databases become consistent eventually (van Steen & Tanenbaum, 2017).

Client-centric consistency

Unlike data-centric consistency, which focuses on maintaining global consistency, i.e., it makes replicas consistent for various users, client-centric consistency focuses on main- taining local or individual consistency. It means the replicas will be consistent on dis- tributed nodes for the individual client. Some popular client-centric consistency proto- cols are listed below (van Steen & Tanenbaum, 2017).

* + Monotonic read consistency ensures that once a process has seen the updated value of a data, it will never see an older value of that data. One example of this is an email database. Individual email mailboxes can be installed on multiple machines. Once a new email is included in the mailbox, all the machines do not necessarily update the mailbox immediately. The mailbox will only update when the user logs in.
  + Monotonic write consistency means if the client performs the write operation, then before updating the data ﬁrst, it will check if there has been any other write opera- tion prior to this performed on a different machine.
  + Read your writes consistency means the write operations will be completed before read operations, irrespective to which node has performed the writing operation.
  + Writes follow reads consistency implies any consecutive write operation by a client process on data will be performed on a replica that is updated with the value most recently read by the client process.

### Security Aspects for Distributed Services and Applications

In general, ensuring the security of a computer system means ensuring the conﬁden- tiality, integrity, and authentication triad (CIA-triad) (Stallings, 2017). However, ensuring the security of a distributed system is more challenging, compared to a centralized sys-

Distributed Algorithms and Applications

tem, due to the complex and distributed design, which makes the system more vulner- able. The three major security challenges of a distributed system are secure channel, access control, and security management, all of which depent on the CIA-triad (van Steen & Tanenbaum, 2017).

###### Secure Channel

Securing the channel means authenticating the communicating parties, i.e., sender and receiver, so that an attacker will not be able to pretend to be a valid sender or receiver. Having a secure channel also means the transferred message will not be accessed by an unauthorized user. So, only the authorized senders and receivers will have access to the message to read it and acquire the information. Another property of a secure chan- nel is that the message will not be edited or altered by an unauthorized user or system. A secure channel must guarantee conﬁdentiality, authentication, and integrity.

Conﬁdentiality is achieved through encrypting the message using symmetric-key encryption algorithm. Among symmetric key algorithms advanced encryption standard (AES) is most widely used (Stallings, 2017). However, an asymmetric key encryption algo- rithm is used to secure the exchange of symmetric key. The sender encrypts the sym- metric key using receiver’s public key (Stallings, 2017). Only the receiver has the private key so, no intruders are able to disclose the symmetric key. The most commonly used asymmetric key encryption algorithm is RSA (Stallings, 2017). The authentication of the sender and receiver is achieved through encrypting the message using asymmetric key encryption. However, in this case, the message is encrypted using the sender’s private key. The receiver can collect the public key from a trusted authority to decrypt the encrypted message. However, asymmetric key encryption-decryption is a time consum- ing process. For this reason, instead of encrypting the actual message ﬁrst, the hash code of the message is encrypted. This process is known as a digital signature (Stal- lings, 2017). Data integrity is achieved through a digital signature, which protects against data forgery. It ensures that content is not edited and ownership is authentic.

###### Access Control

Access control is also known as authorization (van Steen & Tanenbaum, 2017). Authori- zation ensures that only the authorized users, i.e., processes, are able to access to the software and hardware resources. Authorization comes after the authentication of the user or process. A common example of violation of authorization is a denial of service (DoS) attack or distributed denial of service (DDoS) attack. Access control can be imple- mented in the following ways:

* The access control list (ACL) is a database or list that included the name or ID of the authorized users or processes (van Steen & Tanenbaum, 2017).
* Access certiﬁcates are carried by each user and process. They list the rights of using various resources. One shortcoming of this process is that it requires a third-party authority to issue certiﬁcates. Thus, revoking a certiﬁcate may add additional delay (van Steen & Tanenbaum, 2017).
* Firewalls regulate or ﬁlter the access of a user based on the various properties of the user such as internet protocol (IP) address, media access control (MAC) address, and device name. A ﬁrewall can be implemented in between the machine and gate- way router. This type of ﬁrewall is known as packet-ﬁltering gateway. Firewalls can also be implemented at the application level. A proxy-gateway is a special type of application-level ﬁrewall which is implemented as the frontend of an application (van Steen & Tanenbaum, 2017).

###### Security Management

Security management involves the generation and distribution of public-private key pair of asymmetric key encryption and can be implemented in various ways (Stallings, 2017). The most prevalent solution is a third-party trusted authority that distributes public key and certiﬁcates for authentication and digital signature purposes.

###### Secure Mobile Code

Securing mobile code is an important issue in modern distributed systems. Nowadays, a distributed system transfers both data and source codes. Securing programming code is challenging because code can be malicious, such as malware, spyware, or ransom- ware. Malicious code can be downloaded with a piece of data or a ﬁle. Implementing sandbox is a popular solution for this issue. For example, every smartphone app runs in a sandbox. Sandbox executes the code in a restricted environment once installed and monitors activities. If any malicious activity is detected, sandbox halts the execu- tion of the code.

Summary

The inter-process communication of a distributed system is done through remote procedure call (RPC). RPC is a request-response protocol and synchronous in nature, thus the concept of message-oriented middleware (MOM) is developed to provide asynchronous communication. Middleware is a program that resides between system programs and application programs on both client and server sides. The synchronization in distributed system can be achieved by using either global physical clock or logical clock. The global physical clock is known as univer- sal coordinated time (UTC). Two widely known logical clock synchronization proto- cols are Lamport’s logical clocks and vector clocks.

Distributed Algorithms and Applications

A key advantage of a distributed system is the ability to replicate data on various stores. However, when there is a change in a store, the other stores should be noti- ﬁed to update the replicas in order to provide same information to all users, which is known as a consistency issue. Consistency can be managed from the server side (data-centric consistency), as well as the client side (client-centric consistency). A distributed system can be secured by ensuring conﬁdentiality, integrity, and authentication. Access control or authorization can be maintained by access control list (ACL), access certiﬁcate, and ﬁrewall.



# Unit 6

## From Distributed Systems to Ubiquitous Computing

#### STUDY GOALS

On completion of this unit, you will be able to …

… understand the concepts of decentralized and mobile computing.

… analyze the concepts of mobile computing protocols.

… apply the aspects for distributed ledger technology in Internet of Things (IoT).

… design and implement a decentralized application.

DL-E-DLMCSNDS01-U06

1. From Distributed Systems to Ubiquitous Computing

### Introduction

Services and decisions of a centralized computer network are controlled in a single node. In a distributed system, the computation is done parallelly in multiple nodes. A distributed system can be either centralized or decentralized. In a centralized distrib- uted system, computation is done by multiple nodes, but the decision or service is still controlled centrally by a single node. In a decentralized distributed system, the deci- sion or service is not controlled centrally. Distributed ledger technology is a decentral- ized distributed system where a ledger i.e., database is distributed over multiple nodes. Different nodes can access and update the ledger concurrently. This unit will shed light on distributed ledger system. This unit will also discuss ubiquitous computing and Internet of Things (IoT). Ubiquitous computing also known as pervasive computing aims to enable computing on every kind of devices at every location. These devices may include computers, laptops, smartphones, smart watches, wearables, home appliances, and cars. Casually speaking, ubiquitous computing aims to perform computing on vari- ous things. The IoT aims to connect these various things to the internet.

### Distributed Ledger Technology

A distributed ledger technology (DLT) is based on a distributed decentralized network where a database is deployed over multiple distributed machines and there is no cen- tral authority to maintain the database. DLT provides a platform for developing decen- tralized and distributed applications for registering, sharing, and synchronizing transac- tions on digital assets (Antal et al., 2021). Since DLT is a decentralized and distributed technology, it is primarily built on peer-to-peer (P2P) network architecture instead of client-server architecture. Properties of distributed ledger technology are

* decentralized. DLT is implemented based on decentralized architecture, meaning there is no central authority. This architecture protects from any single point failure, or single point security attack, and is a protection against tempering by single party (Zhu et al., 2019).
* distributed. The resources of DLT are distributed on multiple machines. These resources include both software and hardware resources (Zhu et al., 2019).
* immutable. The actions and transactions on a device can be traced and audited by a ledger (Chowdhury et al., 2019).
* irreversible. The ledger transaction activities are irreversible after a certain period of time.
* data consistency. DLT ensures consistency of ledger data stored in distributed stor- age (Chowdhury et al., 2019).
* data provenance. Each transaction is digitally signed by public key cryptography which ensures the authenticity of the data source (Chowdhury et al., 2019).

From Distributed Systems to Ubiquitous Computing

* distributed consensus. DLT follows consensus algorithms in order to come to an agreement about a piece of data (Zhu et al., 2019).
* accountable and transparent. Since DLT preserves authenticity, immutability, irrever- sibility, and provenance, it promotes the accountability and transparency of transac- tions (Zhu et al., 2019).

###### Types of Distributed Ledgers

There are two main types of distributed ledgers: public and private. (Chowdhury et al., 2019). A public ledger is transparent and open to the public, meaning anyone can update the ledger state through transactions. A public ledger can raise privacy con- cerns for certain data. Unlike the public ledger, in a private ledger, only authorized enti- ties are allowed to access and update the ledger state. This allows it to preserve data privacy (Chowdhury et al., 2019).

###### DLT Data Structure

There are two main data structures for a distributed ledger: blockchain and direct acy- clic graph (DAG). Blockchain structure forms a chain of blocks or link list with hashes. Each block contains all transactions took place recently, usually within last couple of seconds or minutes (Antal et al, 2021). Hash of all transactions in a block is generated and forms a Merkel Tree data structure. The root of the tree is pointed in the block. Blockchain is an append-only data structure. All new transactions are appended to the tree. Thus, blockchain structure conﬁrms an immutable log of entire history of transac- tion in that network.

Direct acyclic graph (DAG) is comparatively less popular data structure for distributed transactions. The graph starts with a beginning transaction that is approved directly or indirectly by the other transactions of the graph. Once a new transaction is submitted, it must validate two previous transactions of the graph that were not approved yet (Antal et al., 2021).

###### DLT Protocol Stack

Application layer

The application layer provides the software user interface, which can be developed by a third party on top of an underlying DLT architecture (Zhu et al., 2019).

Contract layer

Smart contracts are computer protocols that deﬁne a set of rules about state transi- tions and corresponding actions. Smart contracts are executed automatically on each node. Each contract has its own set of assets and states with a unique address. A smart contract can execute polynomial computational tasks. In other words, smart contracts provide a platform to build business applications such as cryptocurrencies (Zhu et al., 2019).

Consensus

This type of algo- rithm is used to ach- ieve agreement on a single data value in a distributed system.

Consensus layer

The consensus layer deals with distributed consensus in order to ensure the trustwor- thiness of a block. It also ensures that all nodes have consistent ledger copies. Due to network faults, delay, or malicious nodes, there might be disagreement between net- work nodes, and consensus layer aims to resolve these disputes (Farahani et al., 2021).

Network layer

The goal of the network layer is to provide distributed network architecture. It is done through forming a P2P network based on communication protocols like the transmis- sion control protocol/internet protocol (TCP/IP), for example.

Data layer

The data layer collects data from the lower layer through transactions and encapsu- lates it with data layer headers (Farahani et al., 2021). It also digitally signs the data for authentication and integrity purposes.

Device layer

This layer represents the hardware or physical devices such as sensors, actuators, and networking hardware, which can be connected both with wires and wirelessly (Farahani et al., 2021). The concept of the device layer is similar to that of the physical layer of the open systems interconnection (OSI) model or TCP/IP model.

###### DLT Platforms

Some of the most popular DLT platforms with their key features includeBitcoin, Ether- eum, Multichain, and the electro-optical system (EOS). Bitcoin is a DLT system that introduced the ﬁrst digital currency known as cryptocurrency. It is based on a P2P dis- tributed network. Moreover, it is a public DLT which means any user across the world is allowed to use the P2P network. It uses proof of work (PoW), also known as the Naka- moto consensus algorithm, to ensure the validity of each block. It improves the preci- sion accuracy of data stored in a distributed ledger (Chowdhury et al., 2019).

Ethereum, like Bitcoin, is also based on public DLT and its primary application is crypto- currency. It supports smart contract, which is deployed by transacting Ethereum crypto- currency called Ether. Ethereum also uses the proof of work (PoW) consensus algorithm (Chowdhury et al, 2019). Multichain is a public platform that allows the deployment of private distributed ledger applications. It was developed based on Bitcoin blockchain and allows the users to update block size and protocols, among other aspects. (Chowd- hury et al., 2019). The electro-optical system (EOS) was initially developed on the Ether- eum platform. The DLT platform uses a delegated proof of stake (DPoS) consensus algorithm. Details of the DPoS algorithm can be found in Luo et al. (2018). DPoS uses 21 validators known as block producers (Chowdhury et al., 2019). Some other well-known DLT platforms are Cardano, Fabric, Sawtooth, IOTA Multichain, Cobra, and Waltonchain (Chowdhury et al., 2019).

From Distributed Systems to Ubiquitous Computing

### Aspects of Mobile Computing

Mobile computing refers to a computing system that is based on a mobile network. The term mobile network means that the nodes of the network can move. Examples of mobile nodes are mobile phone devices, wearables like ﬁtness tracker or smart watches, laptops, and personal digital assistants (PDAs). A mobile computing system is developed on a wireless network and is mainly built on wireless communication proto- cols.

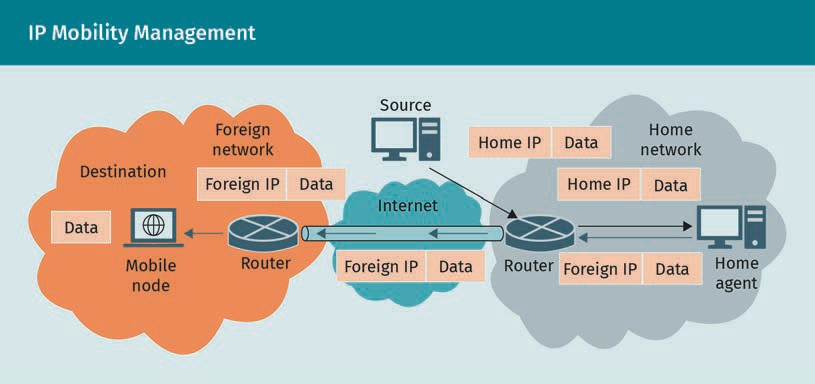
###### Mobile Computing and Mobile Networks

The key characteristic of a mobile computing system is that it is deployed on a mobile network. A network can be considered mobile if it shows any of the following criteria (Ahmed et al., 2010):

* user mobility. The user should be able to access to the same service from various locations.
* device mobility. The user should be access to a service from various end devices.
* network mobility. The entire network can be mobile for example mobile ad hoc or body sensor network. It could also mean that a user should be able to access to a service through different networks. For example, a user can connect to a real-time service through Wi-Fi while roaming through home, coffee shop, or a campus net- work.
* host mobility. The mobile device could be either a client or a server. If it is a server, then it should be able to serve a client while roaming through various geographical areas and networks.

###### Mobile IP Management

Mobile devices are continually roaming from network to network, depending on the mobility of its owner or carrier. As a result, the management of the IP address of a mobile node (known as IP mobility) is challenging. The solution to IP mobility is ach- ieved through implementing a home agent. In a network, the IP of a mobile node is always identiﬁed by its home IP address, even it moves to a new geographical location. However, while it is moving away from the home location, it is associated with a “care of” address, which identiﬁes the current location of the mobile node. Once a datagram is directed to a mobile node, the datagram is ﬁrst received by the home agent, which simply forwards the datagram to the mobile node. The protocol for managing IP mobi- lity is deﬁned in the internet protocol standards RFC 3344, 4721, 5944, and 6275 (Perkins, 2010).



###### Mobile Network Routing Protocols

The topology of a mobile network is not ﬁxed and continually changes over time. Thus, the ID or IP of a neighbor node, route between two nodes, or link cost between two nodes may change over time. Traditional routing algorithms or protocols perform poorly for mobile networks because they do not address this issue. Some of the com- mon routing protocols for mobile networks are provided below (Othman, 2007).

Ad hoc on-demand distance vector (AODV)

AODV is developed to resolve routing issues of mobile ad hoc network and is used by the mobile nodes of the network. It offers a small memory and processing overhead, low power consumption, and low network utilization. It also avoids the problems of classical distance vector protocols. AODV is deﬁned in RFC 3561 (Perkins et al., 2003).

Dynamic source routing protocol (DSR)

DSR is a simple routing protocol for mobile ad hoc network with multiple hops. Two major goals of this algorithm are route discovery and route maintenance to an arbitrary destination. This on-demand protocol allows multiple routes to a destination and allows the sender to choose a route. DSR guarantees loop-free routing and offers rapid recovery from a change in topology. DSR was primarily designed for a network with up to 200 mobile nodes. DSR protocol is deﬁned in RFC 4728 (Yih-Chun et al., 2007).

Optimized link state routing protocol (OLSR)

OLSR protocol is also developed for mobile ad hoc wireless networks. It is an optimized form of link state routing algorithm that selects some nodes as multipoint relays (MPRs). The purpose of these selected nodes is to broadcast messages to declare link state information. The use of MPRs reduces the message overhead compared to the classical link state algorithm. In the classical link state algorithm, each node retrans- mits each message once it has been received for the ﬁrst time. However, in OLSR, only

From Distributed Systems to Ubiquitous Computing

the selected nodes (i.e., MPRs) generate link state information, thereby reducing net- work ﬂooding. OLSR is suitable for a large scale mobile ad hoc network with high node density. OLSR is deﬁned in RFC 3626 (Clausen & Jacquet, 2003).

### Aspects of Pervasive Computing and the Internet of Things

Pervasive computing aims to make computing available from any geographical location and electronic device, at any time. End nodes of pervasive computing include not only desktop or laptop computers, but also cellphone devices, sensors, actuators, home appliances, and wearable devices. An emerging application of pervasive computing is the IoT. Pervasive computing is also referred to as ubiquitous computing.

###### Properties of Pervasive Computing

The key properties of pervasive computing are as follows (Poslad, 2011):

* context awareness. Network nodes have to be context aware in order to optimize performance.
* distribution. The system should be deployed over a distributed network.
* autonomy. Network nodes should be self-governing, meaning that they should be able to operate without human intervention.
* human-computer interactions (HCI). Human-computer interactions should be mini- mized and hidden.
* intelligence. The network nodes should be intelligent, i.e., artiﬁcial intelligence should be integrated.

###### Internet of Things

One of the biggest outcomes of ubiquitous or pervasive computing is the IoT. Tradi- tional internet connects only computers, while the IoT connects computers along with a wide variety of electronics, giving it its name. In recent years, several protocol stand- ards, hardware platforms, and technologies have been introduced for IoT technology.

IoT protocol standards

The Internet Engineering Task Force (IETF) formed the Constrained RESTful Environ- ments (CORE) research group, which develops protocol standards for constrained IP networks. A constrained IP network includes nodes with limited memory size, low power, and low throughput capacity. This type of network may experience compara- tively more packet loss and may include a large number of nodes (sensors and actua- tors) which may turn on or off periodically. IoT is an example of constrained IP network (Constrained RESTful Environments (CoRE), n.d.). Some IoT protocol standards are intro- duced below.

###### Constrained Application Protocol (CoAP)

CoAP is a web transfer protocol designed for a wireless network with constrained nodes i.e., low power, low memory, and low data transfer rate. These nodes may have 8-bit microcontrollers and small ROM and RAM. CoAP operates on request/response archi- tecture. It includes basic ideas of the web, for example, uniform resource identiﬁers (URIs) and media types. CoAP is standardized in 7252, 7959, 8613, and 8974 (Shelby et al., 2014).

###### IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN)

The low-power wireless personal area network (LoWPAN) comprises devices with low power, computation speed, memory, and bit rate. These devices follow properties given in IEEE 802.15.4-2003 standard. 6LoWPAN network assumptions, its problem statements and goals are standardized in RFC 4919 (Kushalnagar et al., 2007).

###### WebSocket

WebSocket is aimed at supporting browser-based applications that require bidirec- tional communication with servers that do not depend on opening multiple HTTP con- nections. WebSocket utilizes existing concepts, such as proxies, ﬁltering, and authenti- cation. WebSocket operates over port 80 and 443 to use the HTTP proxies. Websocket is deﬁned in RFC 6455 (Fette & Melnikov, 2011).

###### IoT Technologies

The development in system on chip (SoC) technology introduced a few low-power, low- cost, lightweight network nodes, which are being treated as the hardware platform of various IoT technologies. Some of the popular IoT technologies and relevant hardware platforms include radio frequency identiﬁcation (RFID) and low range (LoRa).

Transponders This is a network device that operates as both a transmitter and a responder.

Radio frequency identiﬁcation (RFID)

Radio frequency identiﬁcation (RFID) system uses an electromagnetic ﬁeld to identify or track objects, and record data (Jia et al., 2012). In an RFID setup, there are RFID tags (also called transponders) and RFID readers. An RFID tag is attached to the target object. In general, the RFID tag is passive and is powered by the radio wave released from the RFID reader. However, there are also active RFID tags which are powered by battery. RFID readers are transceivers, containing a radio frequency interface (RFI) mod- ule and a wireless switch.

LoRa

LoRa stands for low range. It is a spread spectrum modulation technique for low-power wide area network (LPWAN) technology. LoRa provides long-range communication using low-power network nodes. For example, using LoRa it is possible to communicate over

From Distributed Systems to Ubiquitous Computing

a distance of 160 meters using 86.5 megajoules of energy. LoRa operates on lower industrial, scientiﬁc, and medical (ISM) band frequencies (868 megahertz (MHz) and 433 MHz for the EU; 915 MHz and 433 MHz for the US). LoRaWAN is a MAC layer protocol that operates on a network with star topology (Bor et al., 2016).

Summary

A distributed ledger technology (DLT) is a decentralized distributed network infra- structure in which a ledger, i.e., database, is deployed over multiple distributed machines with no central authority to maintain the database. DLT provides a plat- form for developing decentralized and distributed applications for registering, shar- ing, and synchronizing transactions on digital assets. The key properties of DLT are: decentralized, distributed, immutability, irreversibility, data consistency, data prove- nance, distributed consensus, accountability, and transparency.

There are two types of distributed ledger: public and private. Two data structures that prevent DLT are blockchain and direct acyclic graph. DLT protocol stack con- tains following layers: application, contract, consensus, network, data, and device. Some popular DLT platforms are Bitcoin, Ethereum, Multichain, and EOS.

Mobile computing considers mobile computing nodes, such as user mobility, device mobility, network mobility, and host mobility. The IP of a mobile node is managed by implementing a home agent. ad hoc on-demand distance vector (AODV), dynamic source routing protocol (DSR), and optimized link state routing protocol (OLSR) are some well-known routing protocols for mobile network. One of the big- gest outcomes of ubiquitous or pervasive computing is Internet of Things (IoT). Tra- ditional internet connects computers, while the IoT also connects a wide variety of electronics, such as home appliances, handheld devices, sensors, switches, and actuators.

Constrained application protocol (CoAP), constrained RESTful environments (CoRE), and IPv6 over low-power wireless personal area networks (6LoWPAN) are some well-known protocols for IoT. There are various hardware and software platform for IoT. Some of the popular platforms are RFID, Zigbee, NodeMCU, and LoRa.



# Appendix 1

## List of References

List of References

Abed, G. A., Ismail, M., & Jumari, K. (2011). A survey on performance of congestion control mechanisms for standard TCP versions. *Australian Journal of Basic and Applied Scien- ces*, *5*(12), 1345—1352. <http://www.ajbasweb.com/old/ajbas/2011/Decem-> ber-2011/1345-1352.pdf

Ahmed, H., Talukder, A., & Yavagal, R. (2010). *Mobile computing: Technology, applications and service creation*. McGraw Hill.

Antal, C., Cioara, T., Anghel, I., Antal, M., & Salomie, I. (2021). Distributed ledger technol- ogy review and decentralized applications development guidelines. *Future Internet*, *13*(3), 62—94. https://doi.org/10.3390/ﬁ13030062

Arkko, J., & Pignataro, C. (2009, April). *IANA allocation guidelines for the address resolu- tion protocol (ARP)* (RFC 5494). Internet Engineering Task Force. https://data- tracker.ietf.org/doc/html/rfc5494

Baran Ivo. (2007, October 22). FTP cable3. *Wikimedia Commons*. https://commons.wiki- media.org/wiki/File:FTP\_cable3.jpg

Baresi, L., & Garriga, M. (2020). Microservices: The evolution and extinction of web serv- ices? In A. Bucchiarone, N. Dragoni, S. Dustdar, P. Lago, M. Mazzara, V. Rivera, & A. Sado- vykh (Eds.), *Microservices* (pp. 3—28). Springer. https://doi.org/ 10.1007/978-3-030-31646-4\_1

Benzekki, K., El Fergougui, A., & Elbelrhiti Elalaoui, A. (2016). Software-deﬁned network- ing (SDN): A survey. *Security and Communication Networks*, *9*(18), 5803—5833. https:// onlinelibrary.wiley.com/doi/epdf/10.1002/sec.1737

Berde, P., Gerola, M., Hart, J., Higuchi, Y., Kobayashi, M., Koide, T., Lantz, B., O’Connor, B., Radoslavov, P., Snow, W., & Parulkar, G. (2014, August 22). ONOS: Towards an open, dis- tributed SDN OS. *HotSDN ’14: Proceedings of the third workshop on hot topics in soft- ware deﬁned networking* (pp. 1—6). Association for Computing Machinery. https:// doi.org/10.1145/2620728.2620744

Birrell, A. D., & Nelson, B. J. (1984). Implementing remote procedure calls. *ACM Transac- tions on Computer Systems (TOCS)*, *2*(1), 39—59. https://web.eecs.umich.edu/~mosharaf/ Readings/RPC.pdf

Bor, M., Vidler, J. E., & Roedig, U. (2016). LoRa for the Internet of Things. *EWSN ’16 Pro- ceedings of the 2016 international conference on embedded wireless systems and net- works* (pp. 361—366). Junction Publishing.

Brownell, D. (1996). *Dynamic RARP extensions for automatic network address acquisi- tion.* Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/rfc1931

List of References

Cain, B., Deering, S., Kouvelas, I., Fenner, B., & Thyagarajan, A. (2002, October). *Internet group management protocol, version 3*. Internet Engineering Task Force. https://data- tracker.ietf.org/doc/html/rfc3376

Chowdhury, M. J. M., Ferdous, M. S., Biswas, K., Chowdhury, N., Kayes, A. S. M., Alazab, M., & Watters, P. (2019). A comparative analysis of distributed ledger technology platforms. *IEEE Access*, *7*, 167930—167943. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnum- ber=8902067

Cisco. (2020). *New Cisco annual internet report forecasts 5G to support more than 10% of global mobile connections by 2023*. The Newsroom. https://newsroom.cisco.com/ press-release-content?articleId=2055169#:~:text=By%202023%3A,of%20those%20will

%20be%20mobile

Clausen, T., & Jacquet, P. (2003). *Optimized link state routing protocol (OLSR)* (RFC 3626). RFC Editor. https://doi.org/10.17487/RFC3626

Carnegie Mellon University (CMU). (n.d.). *ASCII table*. Carnegie Mellon University. https:// [www.cs.cmu.edu/~pattis/15-1XX/common/handouts/ascii.html](http://www.cs.cmu.edu/~pattis/15-1XX/common/handouts/ascii.html)

Cohen-Almagor, R. (2013). *Moral, ethical, and social dilemmas in the age of technology: Theories and practice*. IGI Global. https://doi.org/10.4018/978-1-4666-2931-8.ch002

Comer, D. E. (2015). *Computer networks and internets* (6th ed., global ed.). Pearson Edu- cation.

Comer, D. E. (2018). *The internet book: Everything you need to know about computer net- working and how the internet works* (5th ed.). CRC Press.

Conformance Speciﬁcation Radio. (2011). *3rd generation partnership project; Technical speciﬁcation group radio access network; Evolved universal terrestrial radio access (E- UTRA); User equipment (UE) conformance speciﬁcation radio transmission and recep- tion*. European Telecommunication Standards Institute (ETSI).

Constrained RESTful Environments (CoRE) (n.d.). *Charter for working group*. Internet Engineering Task Force. https://datatracker.ietf.org/wg/core/about/

Docker. (n.d.). *Orientation and setup*. https://docs.docker.com/get-started/

Department of Defence High Performance Computing Modernization (DoD HPC). (n.d.). *Neighbor discovery protocol attacks*. U.S. Department of Defense High Performance Computing Modernization Program. https://[www.hpc.mil/program-areas/networking-](http://www.hpc.mil/program-areas/networking-) overview/2013-10-03-17-24-38/ipv6-knowledge-base-security/neighbor-discovery-proto- col-attacks

Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustaﬁn, R., & Saﬁna,

L. (2017). Microservices: Yesterday, today, and tomorrow. In M. Mazzara & B. Meyer (Eds.),

*Present and ulterior software engineering* (pp. 195—216). Springer.

Elson, J., Girod, L., & Estrin, D. (2002). Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, *36*(SI), 147—163. https:// doi.org/10.1145/844128.844143

Endrei, M., Ang, J., Arsanjani, A., Chua, S., Comte, P., Krogdahl, P., Luo, M., & Newling, T. (2004). *Patterns: Service-oriented architecture and web services*. IBM Corporation.

Etzkorn, L. H. (2017). *Introduction to middleware: Web services, object components, and cloud computing*. Chapman and Hall/CRC.

Farahani, B., Firouzi, F., & Lücking, M. (2021). The convergence of IoT and distributed ledger technologies (DLT): opportunities, challenges, and solutions. *Journal of Network and Computer Applications*, *177*, 102936. https://doi.org/10.1016/j.jnca.2020.102936

Fette, I., & Melnikov, A. (2011). *The Websocket protocol* (RFC 6455). RFC Editor. https:// datatracker.ietf.org/doc/html/rfc6455

Finlayson, R., Mann, T., Mogul, J., & Theimer, M. (1984, June). *A reverse address resolution protocol* (RFC 903). Internet Engineering Task Force. https://datatracker.ietf.org/doc/ html/rfc903

Ford-Hutchinson, P. (2005, October). *Securing FTP with TLS* (RFC 4217). Internet Engineer- ing Task Force. https://datatracker.ietf.org/doc/html/rfc4217

Forouzan, B. A. (2013). *Data Communications and networking* (5th ed.). McGraw-Hill.

Fox, A., Gribble, S. D., Chawathe, Y., Brewer, E. A., & Gauthier, P. (1997, October). Cluster- based scalable network services*. Proceedings of the sixteenth ACM symposium on oper- ating systems principles* (pp. 78—91). https://doi.org/10.1145/268998.266662

Gabhart, K., & Bhattacharya, B. (2008). *Service oriented architecture ﬁeld guide for exec- utives*. Wiley & Sons.

Google. (n.d.). *Statistics*. Google IPv6. https://[www.google.com/intl/en/ipv6/statis-](http://www.google.com/intl/en/ipv6/statis-) tics.html

Internet Assigned Numbers Authority (IANA). (n.d.). *Service name and transport protocol port number registry.* https://[www.iana.org/assignments/service-names-port-numbers/](http://www.iana.org/assignments/service-names-port-numbers/) service-names-port-numbers.xhtml?&page=1

Inductiveload. (2007, October 26). EM Spectrum properties edit. *Wikimedia Commons*. https://commons.wikimedia.org/wiki/File:EM\_Spectrum\_Properties\_edit.svg

Information Sciences Institute. (1981). *Transmission control protocol: DARPA internet program protocol speciﬁcation* (RFC 793)*.* Internet Engineering Task Force. https://data- tracker.ietf.org/doc/html/rfc793

List of References

Jia, X., Feng, Q., Fan, T., & Lei, Q. (2012, April). RFID technology and its applications in Internet of Things (IoT). *2012 2nd international conference on consumer electronics, communications and networks (CECNet)* (pp. 1282—1285). IEEE.

Khan, W. Z., Ahmed, E., Hakak, S., Yaqoob, I., & Ahmed, A. (2019). Edge computing: a sur- vey. *Future Generation Computer Systems*, *97*, 219—235. https://doi.org/10.1016/ j.future.2019.02.050

Khatri, A., & Khatri, V. (2020). *Mastering service mesh: Enhance, secure, and observe cloud-native applications with Istio, Linkerd, and Consul*. Packt.

Kurose, J., & Keith R. (2017). *Computer networking: A top-down approach* (7th ed., global edition). Pearson Education.

Kushalnagar, N., Intel Corp, Montenegro, G., Microsoft Corporation, & Schumacher, C. (2007). *IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals* (RFC 4919). RFC Editor. https://www.rfc-edi- tor.org/rfc/pdfrfc/rfc4919.txt.pdf

Lamport, L. (1979). How to make a multiprocessor computer that correctly executes mul- tiprocess program. *IEEE Transactions on Computers*, *C-28*(9), 690—691. https://doi.org/ 10.1109/TC.1979.1675439

Leiner, B. M., Cerf, V. G., Clark, D. D., Kahn, R. E., Kleinrock, L., Lynch, D. C., Postel, J., Rob- erts, L. G., & Wolff, S. (2009). A brief history of the Internet. *ACM SIGCOMM Computer Communication Review*, *39*(5), 22—31. https://doi.org/10.1145/1629607.1629613

Lin, S. D., Lian, Q., Chen, M., & Zhang, Z. (2004, February). A practical distributed mutual exclusion protocol in dynamic peer-to-peer systems. In G. M. Voelker & S. Shenker (Eds.), *International workshop on peer-to-peer systems* (pp. 11—21). Springer.

Liu, F., Tang, G., Li, Y., Cai, Z., Zhang, X., & Zhou, T. (2019). A survey on Edge computing systems and tools. *Proceedings of the IEEE, 107*(8), 1537—1562. https://doi.org/10.1109/ JPROC.2019.2920341

Luo, Y., Chen, Y., Chen, Q., & Liang, Q. (2018, November). A new election algorithm for DPos consensus mechanism in blockchain. *2018 7th international conference on digital home (ICDH)* (pp. 116—120). IEEE.

Mills, D. L. (1984). *Exterior gateway protocol formal speciﬁcation* (RFC 904). Internet Engi- neering Task Force. https://datatracker.ietf.org/doc/html/rfc904

Mills, D. L. (1991). Internet time synchronization: The network time protocol. *IEEE Trans- actions on communications*, *39*(10), 1482—1493.

Mills, D., Martin, J. (Ed.), Burbank, J., & Kasch, W. (2010). *Network time protocol version 4: Protocol and algorithms speciﬁcation* (RFC 5905). RFC Editor. https://[www.rfc-editor.org/](http://www.rfc-editor.org/) info/rfc5905

Monolithic Power Systems (MPS). (n.d.). *Analog signals vs. digital signals*. Monolithic Power Systems. https://[www.monolithicpower.com/en/analog-vs-digital-signal](http://www.monolithicpower.com/en/analog-vs-digital-signal)

Namiot, D., & Sneps-Sneppe, M. (2014). On micro-services architecture. *International Journal of Open Information Technologies*, *2*(9), 24—27. https://[www.researchgate.net/](http://www.researchgate.net/) publication/265292970\_On\_Micro-services\_Architecture

Newpaltz. (n.d.). *Notes on different types of delays*. <http://cs.newpaltz.edu/>

~easwaran/CN/Module2/delays.html

Oksman, V., Strobel, R., Wang, X., Wei, D., Verbin, R., Goodson, R., & Sorbara, M. (2016). The ITU-T’s new G.fast standard brings DSL into the gigabit era. *IEEE Communications Maga- zine*, *54*(3), 118—126. https://doi.org/10.1109/MCOM.2016.7432157

Othman, M. (2007). *Principles of mobile computing and communications*. CRC Press.

Papastergiou, G., Fairhurst, G., Ros, D., Brunstrom, A., Grinnemo, K. J., Hurtig, P., & Mangi- ante, S. (2016). De-ossifying the internet transport layer: A survey and future perspec- tives. *IEEE Communications Surveys & Tutorials*, *19*(1), 619—639. <http://dx.doi.org/> 10.1109/COMST.2016.2626780

Perkins, C. (2010). *IP mobility support for IPv4, revised* (RFC 5944). RFC Editor. https:// [www.rfc-editor.org/info/rfc5944](http://www.rfc-editor.org/info/rfc5944)

Perkins, C., Belding-Royer, E., & Das, S. (2003). *Ad hoc on-demand distance vector (AODV) routing* (RFC 3561). RFC Editor. https://[www.rfc-editor.org/info/rfc3561](http://www.rfc-editor.org/info/rfc3561)

Polese, M., Chiariotti, F., Bonetto, E., Rigotto, F., Zanella, A., & Zorzi, M. (2019). A survey on recent advances in transport layer protocols. *IEEE Communications Surveys & Tutorials*, *21*(4), 3584—3608. https://doi.org/10.1109/COMST.2019.2932905

Poslad, S. (2011). *Ubiquitous computing: smart devices, environments and interactions*. Wiley & Sons.

Postel, J. (1980). *User datagram protocol* (RFC 768). Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/rfc768

Postel, J. (1981, September). *Internet control message protocol: DARPA internet program protocol speciﬁcation* (RFC 792). Internet Engineering Task Force. https://data- tracker.ietf.org/doc/html/rfc792

Qiu, L., Padmanabhan, V. N., & Voelker, G. M. (2001, April). On the placement of web server replicas. *Proceedings IEEE INFOCOM 2001: Conference on computer communica- tions—Twentieth annual joint conference of the IEEE Computer and Communications Society*, *3*, 1587—1596. IEEE.

List of References

Radoslavov, P., Govindan, R., & Estrin, D. (2002). Topology-informed internet replica placement. *Computer Communications*, *25*(4), 384—392. https://doi.org/10.1016/ S0140-3664(01)00410-8

Ricart, G., & Agrawala, A. K. (1981). An optimal algorithm for mutual exclusion in com- puter networks. *Communications of the ACM*, *24*(1), 9—17. https://citeseerx.ist.psu.edu/ viewdoc/download?doi=10.1.1.507.9382&rep=rep1&type=pdf

Rimal, B. P., Choi, E., & Lumb, I. (2009, August 25—27). A taxonomy and survey of cloud computing systems. *2009 ﬁfth international joint conference on INC, IMS and IDC* (pp. 44

—51). IEEE. https://doi.org/10.1109/NCM.2009.218

Rosen, E., Vishwanathan, A., & Callon, R. (2001). *Multiprotocol label switching architec- ture* (RFC 3031)*.* Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/ rfc3031

Santiago, B. (2020). *Concurrency and parallelism for beginners*. Github. https:// github.com/santiagobasulto/pycon-concurrency-tutorial-2020

Sharma, R., & Singh, A. (2020). *Getting started with Istio service mesh: Manage microser- vices in Kubernetes*. Apress. https://doi.org/10.1007/978-1-4842-5458-5

Shelby, Z., Hartke, K., Bormann, C., & Frank, B. (2014). *The constrained application proto- col (CoAP)* (RFC 7252). RFC Editor. https://[www.rfc-editor.org/info/rfc7252](http://www.rfc-editor.org/info/rfc7252)

Shen, N., & Smit, H. (2004, October). *Calculating interior gateway protocol (IGP) routes over trafﬁc engineering tunnels* (RFC 3906). International Engineering Task Force. https://datatracker.ietf.org/doc/html/rfc3906

Silberschatz, A., Galvin, P. B., & Gagne, G. (2013). *Operating system concepts essentials*. Wiley.

Solomon, M., & Kim, D. (2021). *Fundamentals of communications and networking* (3rd ed.). Jones & Barlett Learning.

Srlefﬂer. (2009, June 15). Optical ﬁber cable. *Wikimedia Commons*. https:// commons.wikimedia.org/wiki/File:Optical\_ﬁber\_cable.jpg

Stallings, W. (2017). *Cryptography and network security: Principles and practice* (7th ed.). Pearson.

Szymaniak, M., Pierre, G., & van Steen, M. (2006). Latency-driven replica placement. *IPSJ Digital Courier*, *2*, 561—572. <http://www.globule.org/publi/LDRP_ipsj2006.pdf>

Tanenbaum, A. S., & Wetherall, D. J. (2014). *Computer networks: New international edition*

(5th ed.). Pearson.

Thurlow, R. (2009). *RFC 5531 on RPC: Remote procedure call protocol speciﬁcation ver- sion 2* (RFC 5531). RFC Editor. https://[www.rfc-editor.org/pipermail/rfc-dist/2009-May/](http://www.rfc-editor.org/pipermail/rfc-dist/2009-May/) 002287.html

Tkgd2007. (2008, May 31). Coaxial cable cutaway. *Wikimedia Commons*. https:// commons.wikimedia.org/wiki/File:Coaxial\_cable\_cutaway.svg

Traiger, I. L., Gray, J., Galtieri, C. A., & Lindsay, B. G. (1982). Transactions and consistency in distributed database systems. *ACM Transactions on Database Systems (TODS)*, *7*(3), 323

—342.

Van Steen, M., & Tanenbaum, A. S. (2017). *Distributed systems* (3rd ed.). Maarten van Steen.

Venaas, S., & Retana, A. (2020, February). *PIM message type space extension and reserved bits* (RFC 8736). Internet Engineering Task Force. https:// datatracker.ietf.org/doc/html/rfc8736

Ward, J. S., & Barker, A. (2013). *A cloud computing survey: Developments and future trends in infrastructure as a service computing*. arXiv. https://arxiv.org/abs/1306.1394v1

Wireshark Foundation. (n.d.). Wireshark wiki. *GitLab*. https://wiki.wireshark.org/Sample- Captures

Yih-Chun, H. U., Maltz, D. A., & Johnson, D. B. (2007). *The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4* (RFC 4728). RFC Editor. https://www.rfc-edi- tor.org/rfc/rfc4728.txt

Zhu, Q., Loke, S. W., Trujillo-Rasua, R., Jiang, F., & Xiang, Y. (2019). Applications of distrib- uted ledger technologies to the Internet of Things: A survey. *ACM Computing Surveys (CSUR)*, *52*(6), 1—34.