CS 4453 Lecture Notes

INTRODUCTION TO

COMPUTER NETWORKS

Ruizhong Wei

Department of Computer Science
Lakehead University

Winter, 2015

Contents

Chapter 1

Introduction

1.1 An overview

Computer networks have been developed very fast and continues to grow explosively. Computer networks are now used in all aspects of businesses from advertising, planning, production, shipping, selling, billing to management etc. Social networks and other network services involve important parts of personal daily life. Computer networks are now almost everywhere.

Computer network consists of hardware and software. The companies need workers for designing, install, operating, managing computer networks. A very large part of software are involved or applied networks. Therefore, computer networks and internetworks are important part of computer science.

Now millions of devices are connected to the Internet, which are called hosts, end systems or nodes. Hosts are connected together by a network of communication links and packet switches. Most popular switches used now are routers and link-layer switches. Link-layer switches are typically used in access networks, while routers are typically used in the network core. Hosts access the Internet through Internet Service Providers (ISP). There are different ISPs such as local cable or telephone companies. ISPs provide a variety of types of network access to the hosts (DSL, high-speed local area network access, wireless access, etc). ISPs that provide access to hosts also need to interconnected. These lower-tier ISPs are interconnected through national and international upper-tier ISPs such as Level 3 Communications, AT&T, Sprint, NTT. A upper-tier ISP consists of high-speed routers interconnected

with high-speed fiber-optic links.

Computer networks and internet can trace back to the telephone network uses circuit switching to transmit information between two parties, which are popular in early 1960s. By 1972, the Advanced Research Projects Agency (ARPA) in United States developed ARPAnet that consists of about 15 nodes. The first host-to-host protocol between ARPAnet systems called network-control protocol (NCP) is published as RFC 001. During 1972 - 1980, several stand-alone packet-switching networks besides ARPAnet were developed. The early versions of TCP are also proposed, although that are quite different from today's TCP. In 1980, the internet was a research project that involved a few dozen sites. In 1983, the TCP/IP as new standard host protocol for ARPAnet published as RFC 801. In the late 1980s, important extensions were made to TCP to implement host-based congestion control. The DNS and 32-bit IP address (RFC 1034) were also developed. In the early 1980s the French launched the Minitel project, that was intended to bring data networking into everyone's home. The Minitel became a huge success in 1984 when the French government gave away a free Minitel terminal to each French household that wanted one. The Internet explosion happened in 1990s. Some important events includes the World Wide Web, the initial versions of HTML, HTTP, Web servers and web browsers, E-mail systems, instant messages, Peer-to-peer file sharing of MP3.

In current years, innovation in computer networking continues explosion. Faster routers and higher transmission speeds in both access networks and in network backbones are developed. Multimedia networking, Voice-over-IP (VoIP), high-speed public WiFi networks and medium-speed internet access via 3G and 4G cellular telephony networks, social networks, extensive private networks, cloud computing etc.

1.2 Key aspects of networking

Nowadays the computer networks are very complicated. To understand the computer networks, we need to understand some key aspects of them. Some basic things about the networking are as follows.

- Network applications and network programming

  Network services and other applications are implemented by software. In a programmer, you can write some network programming based on

your understanding of the interface of the network. For example, if you know how to use socket, then you can write some network applications without knowing the details of network. However, if you understand the underlying network mechanisms and technologies, then you can write faster, more reliable, better programs.

- Data communications

  Data communication refers to the low-level mechanisms and technologies used to send data across a physical communication medium. Although this topic is mostly belong to the Electrical Engineering, it provides a foundation of concepts on which the rest of networking is built.

- Packet switching and networking technologies

  Data are divided into small blocks, called packets, and includes some information about the sender and recipient. There are many packet switching technologies to meet various requirements for speed, distance and cost etc.

- Internetworking with TCP/IP

  Internet is a network of networks. Internetworking is substantially more powerful than a single networking technology because the approach permit new technologies to be incorporated at any time without requiring to replace old technologies.

- Network security

  Wide applications and services spreading in the public networks require methods to protect privacy and security of the networks.

- Additional networking concepts and technologies

  A large set of additional technologies and concepts of networking keep coming. For examples, multimedia networking, software defined networking (SDN), Internet of Things, etc.

We will discuss some of them in details later.

Application	Layer 5
Transport	Layer 4
Internet	Layer 3
Network Interface	Layer 2
Physical	Layer 1

Figure 1.1: TCP/IP model of networks

1.3 Protocols, standards and models

Internet is a very complicated system which contains different networks and different nodes. We need some rules for everyone so that the multiple entities connecting to the internet must agree and follow. Otherwise, we will not be able to communicate each other. These agreements are called communication protocols, network protocols or protocols. A communication protocol specifies the details for one aspect of computer communication, including actions to be taken when errors or unexpected situations arise. A given protocol can specify low-level details, such as the voltage and signals to be used, or high-level items, such as the format of messages that application programs exchange.

Each protocol just do one thing for the networking. So the design of protocols needs to consider how to collaborate together to let the work done. The network designers organize the protocols (and the network hardware and software that implement the protocols) into layers. There are two standard layering models commonly used now. One is Internet protocols (TCP/IP) model and another is OSI (Open systems interconnection) model. These models are also referred as protocol stacks.

The Internet Protocol model consists of 5 layers as shown in Figure 1.1.

OSI model consists of 7 layers. We will discuss OSI model later. Each computer contains a set of layered protocols. The data flow from a sender to a receiver is as shown in Figure 1.2. When an application sends data, the data is divided as packets and outgoing packets pass down each layer of protocols. Then the packets transmitted across the underlying physical network. When a packet reaches the receiving computer, it passes up through each layers of protocols.

When a sending packet goes through a protocol, a protocol header will be added to wrap the packet. The receiving computer get the protocol information from these headers.

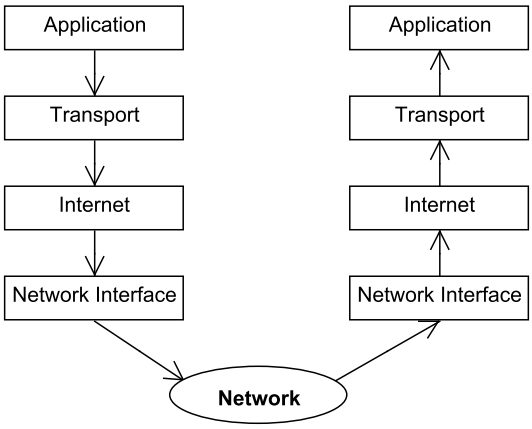

Figure 1.2: How data passing through layers

It is important that everyone should agree with the protocols. Therefore we need some standard for the protocols. Internet standards are developed by the Internet Engineering Task Force (IETF). The IETF standard documents are called requests for comments (RFCs). Other bodies also specify standards for network components, most notably for network links. The IEEE 802 LAN/MAN Standards Committee specifies the Ethernet and wireless WiFi standards.

1.4 Internet trends

The Internet has experienced exponential growth for over 25 years regarding to the number of computers on it. The Internet has been doubling in size every nine to fourteen months. As it grew, the Internet changed in two ways. First, the communication speeds increased dramatically. A backbone link in current Internet can carry about 200.000 times as many bits per second as a backbone link in the original Internet. Second, new applications arose the appealed to a broad cross section of society. The high-speed computation and communication technologies shifted the focus of Internet from resource sharing to general-purpose communication. Types of data across the Internet has been changed a lot. The data shifted from text, images, video clips to high-def videos. The audio data shifted from alert sounds, human voice,

audio clips to high-fidelity music.

New networking technologies and new Internet applications continue to emerge. Some examples:

- VoIP
- Network TV
- Digital cellular
- Wireless access
- Big data
- Social networks
- Sensor networks
- Online banking, shopping etc.
- Internet of things
- Cloud computing

Chapter 2

Introduction of the OSI Network Model

A computer network comprises two or more nodes, i.e., computers connected by underlying communication channels. The internetwork is constructed from nesting of networks.

Networks are usually complex. A modern network is designed, constructed and described in a layered structure. The layered architecture has many advantages. The communication only between the neighbored layers. Changes in one layer have no significant impact on the functioning or other layers.

The components in each layer communicate with those in the corresponding layer at the other end through a virtual linked connection, while the data actually go all the way from the upper layers to the lowest layer of one system, traveling through the physical channel, reaching the bottom layer of the other system, and then traveling from the bottom layer to the top layer of the other system.

ISO (International Standards Organization) uses a seven-layer open systems interconnect model (OSI) witch is show in Figure 2.1

The seven layers are as follows:

Application	Detailed application specific data being exchanged
Presentation	Conversions for representing data
Session	Management of connections between programs
Transport	Delivery of sequences of packets
Network	Format of individual data packets
Data link	Access to and control of transmission medium
Physical	Medium and signal format of raw bit transmission

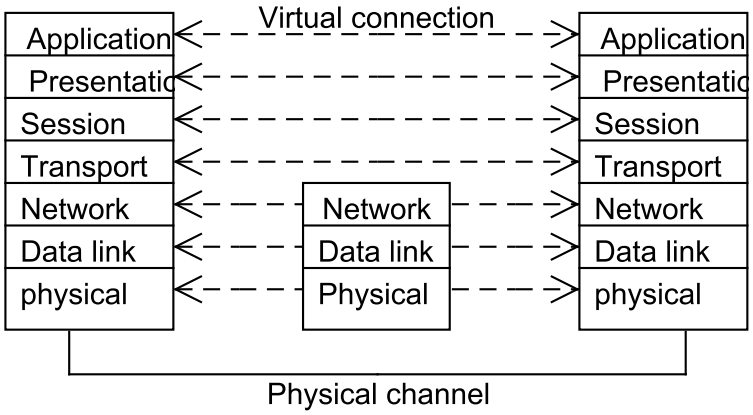

Figure 2.1: OSI model

There are other kind of models of networks. One popular model is TCP/IP model we mentioned before. This model is initialed from RFC standard.

There are criticisms for both models. Because our purpose is for understanding how the protocols work for computer networks, we will use OSI based model. This lecture is basically for computer science students. So we will focus on software protocols.

2.1 Physical layer

The physical layer handles the transmission of raw bits over a communication channel. Protocols in this layer specify the medium used for the transmission (electronic, optical or wireless), the signal format (serial or parallel, synchronous or asynchronous), and convert raw bit stream into common codes understandable by all the connected parties.

CCITT/ITU (International Telecommunication Union) has established X.21 - X.24 to specify the functions at the physical lever for leased circuits. Other standard such as EIA-232 and v.21 - v.24 are widely used for various purpose.

2.1.1 Data encoding

Encoding is the procedure that converts the original signals or raw data into a different form for different reasons (better utilization of existing facilities, increasing efficiency, reducing error rate, synchronization, security, etc.).

Here we give some examples of encoding methods.

NRZ-L(non-return-to-zero level): binary data 1s and 0s are simple represented by two different voltages.It is also possible to use different voltage levels to transmit more than one bit at a time.

NRZ-I (non-return-to-zero, invert-on-ones): similar to NRZ-L, but rather than measuring the absolute value of the signal element, two voltage are compared. If the two voltages are different, a 1 is transmitted, otherwise, a 0.

Manchester coding (add synchronization), CSMA/CD, etc are also used for encoding.

2.1.2 Multiplexing schemes

In a computer network, computers have to share physical links. Multiplexing (MUX) techniques, which allow more than one communication channels to be carried over a single physical connection.

- FDM (frequency division multiplexing): different carrier frequencies are used to separate one channel from another. This method can be used for wired or wireless connections.

- TDM (time division multiplexing): a large data rate connection is divided into many small time slots. The system cyclically scans the incoming data from the multiple input ports. This method needs synchronization for sender and receiver.

- CDMA (code division multiple access): uses some coding theory and complicated algorithms. It is used in cellular telephone system. CDMA uses some code which has orthogonal property:

$$c_i \cdot c_j = \frac{1}{n}\sum_{k=1}^{n} c_{ik}c_{jk} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

  Where $c_i = (c_{i_1}, c_{i_2}, \cdots, c_{i_n})$ and $c_i \cdot c_j$ is called inner product. One way of implementation can use c_i and $\overline{c_i}$ to represent a bit 1 or bit 0, where c_i is a vector of 1, -1 and $\overline{c_i}$ is obtained by exchange 1 and -1 of c_i. Then

$$c_i \cdot \overline{c_j} = \begin{cases} -1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

  For example, suppose we have 4 communication channels. Then we can use the following code.

	bit 1	Bit 0
channel 1	(1,1,1,1)	(-1,-1,-1,-1)
channel 2	(-1,-1,1,1)	(1,1,-1,-1)
channel 3	(-1,1,-1,1)	(1,-1,1,-1)
channel 4	(-1,1,1,-1)	(1,-1,-1,1)

  It is easy to check that c_1, c_2, c_3, c_4 are mutually orthogonal. Suppose in a moment, channel 1 is transmitting 0, channel 2 is transmitting 1, channel 3 is not transmitting and channel 4 is transmitting 0. Then

$$\begin{aligned} S &= (-1,-1,-1,-1) + (-1,-1,1,1) + (0,0,0,0) + (1,-1,-1,1) \\ &= (-1,-3,-1,1) \\ c_1 \cdot S &= \frac{(1,1,1,1)\cdot(-1,-3,-1,1)}{4} = -1 \\ c_2 \cdot S &= 1 \\ c_3 \cdot S &= 0 \\ c_4 \cdot S &= -1 \end{aligned}$$

- WDM (wave division multiplexing), used for conventional fiber for high-speed transmission. WDM is similar in principle to FDM, but use discrete colors of light. For a 128-channel dense WDM switch at full capacity, each fiber pair can deliver 1.28 terabits per second bandwidth.

2.2 Data link layer

2.2.1 ARP and link layer implementation

In a data link layer, the transmitting streams are divided into frames, and the frames are sent or received one by one. In data link layer, the addresses are the MAC addresses. There is a protocol called Address Resolution Protocol (ARP) (RFC 826) used to resolve the address problem. Basically, ARP will provide the MAC address from the IP address. In many ways ARP is analogous to DNS, which resolves host names to IP addresses. The main difference is that DNS resolves host names for hosts anywhere in the Internet, but ARP only resolves addresses for hosts and router interfaces on the same subnet.

To connect two subnets, there must be some router with more than one interface. Each of the interfaces has an IP address, while it has its unique MAC address. In this way, the hosts in different subnets can find out the MAC address from different ARP modules. An ARP request is sent as broadcast frame and the respond of ARP is sent by standard frame.

Most popular wired LAN (local area network) is Ethernet. Basically because of the simplicity and efficiency of the Ethernet (there are other LAN technologies such as FDDI, ATM). But the transmission of Ethernet is not reliable (the receiver will not send back acknowledgment). The reliability of the communication in Ethernet will be taken care at other layer of the network. But there are other LAN technologies which provide reliable delivery. Especially, in wireless communications, reliable delivery is used.

In a computer, usually part of the link layer is implemented in a network adapter, or sometimes called network interface card (NIC). Sometime we call this lower level of link-layer. The heart of the network adapter is the link-layer controller, usually a single, special-purpose chip that implement link-layer services (framing, link access, error detection etc.). That means most link layer controllers are implemented in hardware. Most NIC were physically separated cards, but now many computers are integrated it onto

its motherboard. Part of the link layer functionality (link layer addressing, activating the controller hardware, etc) is implemented in software and runs in CPU. These are referred as higher-level link layer functionality.

2.2.2 Asynchronous and synchronous communications

Inside a computer, many data transfers occur in parallel (between RAM and CPU, Disk and memory, etc) through host's bus. In digital communications and in network environments data are transmitted almost exclusively serially.

An asynchronous transmission system copes with timing and synchronization problems by splitting a character into a fixed number of bits and sending one character, or byte of data, at a time. To keep the communication synchronous, extra bits - a start bit and one or two stop bits - are added to the original data.

2.2.3 Error detection and correction

Electromagnetic waves traveling over a transmission medium may encounter noise and that will cause data errors. Single-bit errors are the most common type in data communication. However, multiple-bit errors or burst errors are possible too. To avoid passing the effects of bit errors to higher OSI layers, the data link layer must detect any errors in a received message.

- Parity: the parity bit is obtained based on the count of 1s in the data block. The system can use either odd or even parity. Single-bit parity can detect 1-bit error. (but cannot find out which bit is wrong). To detect two error bits, a two-dimensional parity mechanism may be used. A two-dimensional parity can correct one bit error or detect two error bits. Some error-correcting code can be used to detect and correct errors. Using error-correcting codes requires more redundancy.

d data bits	parity bit
0111010100010110	0
0110100010001000	1

Figure 2.2: One-bit even parity

- Arithmetic checksum: the sender divides the sending data unit into equal segments. Then ones-complement arithmetic is used to add the segments together to get the result sum. The sum is complemented and appended to the data as the checksum field. As an example, consider four 4-bit data units as 1000, 1101, 0101 and 1110. The sum of these units is 1010 ($1000 + 1101 = 0110, 0110 + 0101 = 1011, 1011 + 1110 = 1010$). The check sum will be 0101 which is attached after the four data units.

- Cyclic redundancy checksum (CRC): the CRC of data is generated at the transmitter end by means of a hardware that involves sequential circuits using shift registers and flip-flops. As as example, a bit sequence 11100110 is represented as $M(x) = X^7 + X^6 + X^5 + X^2 + X$. The CRC implementation using $G(x) = X^4 + X^3 + 1$ (11001), which is of degree 4. We have $X^4 M(x) \equiv X^2 + X \pmod{G(x)}$. So the message gives us 111001100110. Both sender and receiver know $G(x)$. So the receiver can use the last four bits to check if the string is correct. The CRC using $G(x)$ can detect up to 4 error bits. And also can detect more error bits in the probability of $1 - 0.5^4$. We omitted the detailed theory behind the CRC codes.

2.2.4 Framing and flow control

Asynchronous transmission consists of the movement of short sequences of fixed-length data. The frames help indicate the start and end of packets for the receiver.

- bit-oriented transmission uses a bit pattern 01111110 as a flag to indicate the frame's start and end.

Flag	Address	Control	Data (0 or more bytes)	CRC	Flag

- Character-oriented transmission uses an integral number of bytes. The frame start is indicated by a special synchronization character SYN followed by a DLE (data link escape) character and an STX (start-of-text) character. The end of the frame is indicated by DLE and ETX(end-of-text) characters.

SYN	DLE	STX	Header	Data	DLE	ETX	CRC	SYN

Some other technologies are used for framing and flow control.

- Bit stuffing and character stuffing: A flag in the bit-oriented protocol determines the start and end of a frame. However, the data and other fields may also have the same sequence and the receiver may take the wrong sequence as the end of frame flag. To avoid such an error, the sender end must be designed to ensure that no bit sequence identical to the flag will occur anywhere else in the data. A bit stuffing can be used for that purpose. For example, suppose the flag is 01111110. If any sequence of five 1 is found at the sender side after the start flag, a 0 is inserted. For example, if the data is as follows:

  $$01111011011111011111011110$$

  After bit stuffing, the data becomes:

  $$01111011011111o1011111o011110$$

  where o denote the 0 bit added.

  For character-oriented transmission, similar method can be used. When the pattern DLE is inside the data, another DLE is inserted. This prevents the pattern DEL ETX from appearing anywhere in the frame except at the end.

Flow control defines both the way in which many frames are sent and tracked and how the stations do error control. For example, if errors have been found in a frame, a request to resend the erroneous frame can be sent to the sender. This type of error control is called automatic repeat request (ARQ). Sometimes the receiver may need the sender to stop sending data (e.g., buffer space not enough) and resume after some time. XON/XOFF characters can be used to do that. This method is more applicable to character-oriented asynchronous transmission. For Bit-oriented or frame-oriented transmission, the following methods are used.

- Stop-and-wait protocol: sender sends out a frame and then wait until receiving a positive acknowledge message from the receiver. The receiver checks the frame when it is received. Then sends a positive acknowledge message to the sender if no errors found, or otherwise sends out a negative acknowledge message. The sender then can send the next frame of resend the previous frame. A timer mechanism must be set for the cases when a frame is lost. When time-out, the sender

retransmit the frame. However, the acknowledgment loss will cause duplicated frames. So a sequence bit will be put at each frame, that alternates between 0 and 1.

This protocol is not very efficient.

- Sliding window protocol: using this protocol, a sequence of packets may be sent and received simultaneously. A sequence number is used for each packet. For example, if the sequence number uses 3 bits, then the number can be 0 - 7. The sender can send out at most 8 frames at a time. To keep track of sent and received frames, windows are implemented that open and close as frames are being sent or received. In sender's window, the left of the window moves when a frame is being sent and the right side moves when an acknowledged frame is being received. The maximum number of pending frames may not exceed the half of window size (4 if the window size is 8). The receiver's window is designed in a similar fashion.

2.2.5 Broadcast link

There are two types of network links: point-to-point link and broadcast link.

In a broadcast link, multiple sending and receiving nodes all connected to the same single shared channel. Ethernet and wireless LAN are examples of broadcast link layer technologies. One important problem for broadcast link is how to solve collisions.

- Channel partition protocols: TDM, FDM and CDMA as we mentioned in physical layer. As an example, now we look at the TDM technology to partition a broadcast channel's bandwidth among all nodes sharing that channel. TDM divide time into time frames and each time frame is divided into N slots. Whenever a node has a packet to send, it transmits the packet's bits during its assigned time slot in the revolving TDM frame. One disadvantage of TDM is that some slots may be wasted if some nodes stop sending packets.

- Random access protocols: a transmitting node always transmits at the full rate of the channel. When there is a collision, each node involved in the collision repeatedly retransmits its frame (packet) until its frame gets through without a collision.

We use Slotted ALOHA as an example. In that protocol, all the frames are of the same size. The time is divided into slots such that each slot equals the time to transmit one frame. Nodes start to transmit frames only at the beginning of time slots. If two or more frames collide in a slot, then all the nodes detect the collision event before the slot ends. In the case of collision, the nodes retransmits its frame in each subsequent slot with probability p until the frame is transmitted without a collision. There is some method to calculate p to maximum efficiency of the slotted ALOHO. The drawback of this method is that when a collision happens, the slot is wasted.

There are many random access protocols. One of them is called carrier sense multiple access with collision detection (CSMA/CD) protocol.

- Taking-turns protocols: let nodes take turns so that no time slot is wasted.

  We introduce two important protocols. One is the polling protocol. In this protocol, one of the nodes is designated as a master node. The master node polls each of the nodes in a round-robin fashion. So the master node sends a message to node 1 saying that node 1 can transmit some maximum number of frames, then inform node 2 to transmit certain frames, etc. The procedure continues in this way, so that the nodes are transmitted in a cyclic manner. The drawback of this method is it introduces a poll delay-time needed for master node to notify the nodes. And if the master node fails, then the entire channel become inoperative.

  Another protocol is the token-passing protocol. In this protocol, there is no master node. A small frame know as a token is passing though the nodes in some fixed order. For example, node 1 passes the token to node 2, node 2 passes it to node 3 etc. When a node received a token, then it transmits data frames up to some maximum number. Then the node passes the token to the next one. If the node does not have frames to transmit, then it simply pass the token to the next node. One drawback of the protocol is that if some node is failure to pass the token.

  Many protocols based on above ideas are developed, that tried to treat possible failures.

2.2.6 Data center networking

Recently, Internet companies such as Google, Microsoft, Facebook, etc have build massive data centers, each housing a lot of hosts and concurrently supporting many distinct cloud applications. Each data center has its own data center network that interconnects its hosts with each other and interconnects the data center with the Internet.

The data center network supports two types of traffic: traffic flowing between external clients and internal hosts and traffic flowing between internal hosts. To handle flows between external clients and internal hosts, the data center network includes one or more border routers. Data center network design has been a research topic in recent years.

A data center usually provides many services. Inside the data center, the external requests are first directed to a load balancer whose job is to distribute requests to the hosts.

Figure 2.3 explains the topology of an example data center network. In this example, there is a border router which connects the data center network to the public Internet. This data center provide two applications. To support requests from external clients, each application is associated with a public visible IP address to an access router. In side the data center, the external requests are first directed to a load balancer that distributes to the hosts and balances the load across the hosts. The load balancer also translates the public external IP address to the internal IP address. The hosts in data center, called blades, are generally commodity hosts that include CPU, memory and disk storage. The hosts are stacked in racks, with each rack typically having 20 to 40 blades. At the top of each rack there is a switch, called Top of Rack (TOR) switch, that interconnects the hosts in the rack.

2.3 Network layer

The network layer of OSI architecture deals with the connection of two ends via a switching mechanism to allow the use of network links in a predetermined manner. The two services used are called connection-oriented services (CONS) and connectionless network service (CLNS).

In the CONS, there are 3 main phases of communication. In the first phase, a connection is established between the sender and the receiver, followed by the second phase consisting of data transfer. The connection may

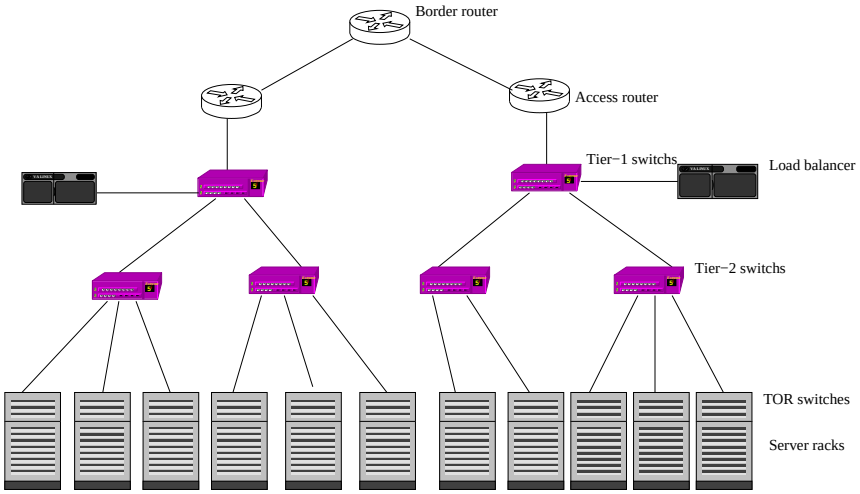

Figure 2.3: A data center network

be terminated by either side in the third phase when the data transfer is complete or for some other reasons. In CLNS, there are no connection establishment and termination phase. Rather, the stations transfer the data directly. The packet forming the data may take different routes to reach the destination.

2.3.1 Subnet concept

An internetwork consists of many smaller networks capable of performing switching, routing and forwarding on their own. These smaller networks are called subnets (or subnetworks). Each subnet is capable of operating on its own. Interconnection of subnets is possible by using special devices called intermediate systems (ISs) by ISO. Two types of IS are routers connecting the subnets at the network layer and switches (or bridges) connecting the subnets at the second layer.

A router may allow forwarding and routing of packets between different subnets operating under different protocols. However, a bridge does not change or modify the contents of a received packet, it simply acts as as address filter that picks up a packet on one subnet and forward it to another subnet.

The routers and switches may be configured to support either the CONS or CLNS.

2.3.2 Overview of switching techniques

For a multiplexed network, there are two ways of switching between communication channels: circuit switching and packet switching. A variety of switching mechanisms are between these two switching methods (multirate circuit switching, cell relay ATM, frame relay, virtual circuit packet switching, datagram, etc.).

- Circuit switching: like telephone systems, circuit switching requires a transmission path between source and destination (so it is CONS). Since the line is dedicated for the user, there is continuous transmission of data. If the network is not capable of handling fast traffic, the stations will know about it during connection establishment phase. Once the line has been established, that path will remain in effect for the entire conversation, and the network is not responsible for accommodating changes in demand by the user. However, if the network is experiencing heavy delays or if the destination station is busy, the path connection may be refused by means of a busy signal.

  A circuit switching system stores no data at the intermediate nodes.

- Packet switching: packet switching is specially designed to accommodate the bursty multiprocess communication commonly found in computer networks. Two networks connected by a circuit switch must operate at the same speed, packet switching can connect networks operating at different speeds. Because of the store-and-forward nature, packet switching often cause variation in delay. Packet switching can recover from failure in less time and with less effort than are required in circuit switching. Also packets may take different paths when a route because too crowed. This makes packet switching more robust. However, packets may not arrive in the order originally sent. Buffers are introduced for flow control in packet switching systems.

  The X.25 standard for packet switching is a lower three-layer equivalent of the OSI model. This protocol, based on a physical layer, a link layer, and packet layer, is standardized by the ITU-T and is defined as an interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE). It uses X.21 as physical layer standard of some other standard such as EIA 232. The link layer protocol is called LAP-B (Link Access Protocol - balanced). The packet layer

data are transmitted as packets over virtual circuits, which may be permanent or dynamically established. The DTE is the connecting device that allows up to 4095 simultaneous virtual circuits with other DTEs over a single physical link.

- Virtual circuit packet switching: statistical multiplexing means that paths (virtual circuits) are defined through the network. However, no bandwidth is allocated to the paths until actual data (real information) are ready for transmission. Then the bandwidth within the network is dynamically allocated on a packet-by-packet basis. If, for a short period of time, more data need to be transmitted than the transmission facilities can accommodate, the switched within the network buffers the data for later transmission. If the oversubscription persists, congestion control mechanisms must be invoked.

  A virtual circuit (VC) consists: (1) a path (series of links and routers) between the source and destination hosts, (2) VC numbers, one number for each link along the path, and (3) entries in the forwarding table in each router along the path. A packet belonging to a virtual circuit will carry a VC number in its header. Each intervening router will replace the VC number of each traversing packet with a new VC number. The VC number is from the forwarding table. As an explain example, suppose a router has three interfaces, the following is the forwarding table

in interface	incoming VC#	out interface	outgoing VC#
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87
2	36	3	33
...	...	...	...

  Whenever a new VC is established across a router, an entry is added to the forwarding table. When a VC terminates, the appropriate entries in each table along its path are removed.

  There are three phases in a virtual circuit:

  - VC setup: the sending transport layer contacts the network layer, specifies the receiver's address, and waits for the network to set

up the VC. The network layer may also reserve resources (for example, bandwidth) along the path of the VC

- Data transfer: packets begin to flow along the VC.
- VC teardown: The sender informs the network layer to terminate the VC. The network layer inform the end system of the call termination and update the forwarding tables in each of the packet routers on the path.

2.3.3 Routing strategies

Communications in a network depends on several factors such as delay, the number of intermediate nodes, and the distance between them. When the outgoing link costs are known, a router can figure out the best path for a packet. Several algorithms exist for performing these calculations.

Two strategies used are distance vector routing and link state routing.

- Distance vector (DV) routing: the routers exchange cost information about neighbors with one another. They also share the complete routing table, and the inputs received by other routers are used to update the current table. The link cost is considered to be one. The cost of sending data form one router to another in five hops would be 5.

- Link state (LS) routing: instead of sending the entire routing table, only the information about neighbors is sent. The routers send periodic updates to each neighboring router, which in turn sends the information to each of its neighbors, and so on. The cost in link state routing is expressed in terms of the weighted value based on traffic, link state and security levels.

In evaluating the shortest paths, most routers use one of the two algorithms: Dijkstra's algorithm (for LS routing) and the Bellman-Ford algorithm (for DV routing). Both of algorithms use graphs made up of nodes and arcs to calculate the shortest path between two nodes. We will go back to some details about routing algorithms later.

2.3.4 Congestion control

In network communication, we want to achieve the maximum throughput with controlled delay. When the throughput is increased, the delay is likely

to increase too. A region of mild congestion may reached. As the offered load is increased further, a period of severe congestion is reached, there upon the network throughput actually drops instead of increasing. The throughput decreases dramatically in this region. Congestion control mechanisms are used to avoid this region.

Congestion avoidance and recovery mechanisms are used to control the network to prevent complete collapse. When the network starts to drop packets as a result of congestion, these procedures are used.

In a router, there might be input port queues and output port queues. For output port queuing, a packet scheduler will be used to choose one packet among those queued for transmission. There are different schedulers. Some simple examples include first-com-first-served (FCFS) and weighted fair queuing (WFQ) which shares the outgoing link fairly among the different end-to-end connections that have packets queued for transmission. Packet scheduling plays a crucial role in providing quality-of-service guarantees.

For input port queues, when there is no memory of buffer, then some packet has to be dropped. One simple way is to drop the arriving packet, that is known as drop-tail. Another popular algorithm is the Random Early Detection (RED) algorithm. Under RED, a weighted average is maintained for the length of the output queue. If the average queue length is less than a minimum threshold min_{th}, when a packet arrives, the packet is admitted to the queue. If the queue is full or the average queue length is greater than a threshold max_{th}, when a packet arrives, the packet is marked or dropped. Finally, if the packet arrives when the average queue length is in the interval $[min_{th}, max_{th}]$, the packet is marked or dropped with a probability.

Another phenomenon is know as head-of-the-line (HOL) blocking in an input-queued switch (a queued packet in an input queue must wait for transfer through the fabric although its output port is free). An example of HOL in Figure 2.4 can be used to explain that phenomenon.

In Figure 2.4, the upper part is the situation before the switch fabric transfers the packets from the input queue to output queue. The numbers on the packets denote that destined output pot numbers of the packets. In first input queue, there are two packets destined to output port 1, in second input queue, there is a packet destined to output port 3, etc. The lower part of Figure explains the situation after the switch fabric transfer a packet from first and second queues to the output queues. In this case, the packet destined to output port 2 cannot transfer even though the output queue for port 2 is empty. There are researches about how to reduce the HOL.

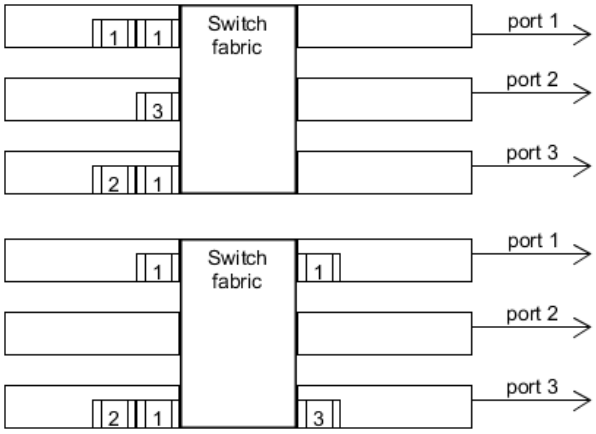

Figure 2.4: Example of HOL

Some topics about the network layer will be covered in the later Chapters.

2.4 Transport layer and session layer

The session layer of OSI is very small in practical network today. Most of the session layer tasks are usually built into applications. This layer is responsible for session management such as checking for user logon to a remote system.

Transport layer is responsible for providing reliable, cost-efficient data transport. The transport should be independent of the physical network in use. There are two types of transport services, connection oriented and connectionless. We will discuss the details of transport layer in other chapter.

2.5 Presentation layer and application layer

The presentation layer of OSI is concerned with the syntax and semantics of the transmitted information rather than with the reliable transmission of data. Data encoding, compression and security are some of the issues handled at this layer. For example, different computers may have different codes representing characters (ASCII or Unicode, etc), integers, floating-point values, and other data structures. To ensure a smooth exchange of

data between computers, the presentation layer is responsible for managing the abstract data structures and converting them form one form to another.

2.5.1 Data Compression

Data compression are used to save storage and transmission time. There are many compression algorithms and utilities. The performance of a compression scheme is largely characterized by its average compression ratio.

There are two types of data compression.

- Lossless compression scheme: the compressed data can be fully recovered by the uncompressing. (gzip, GIF, etc).
- Compression with not significant loss (lossy): the compression may be of slightly lower quality, but the compression ratio is good. Some image or movie compression use that kind of schemes.

There are many different data compression algorithms. For different kind of data, one algorithm's compression rate may be different. So usually special algorithms will be chosen for spacial type of data. Here we will not give detailed compression algorithms, but just introduce main ideas.

- Huffman encoding: note that characters or patterns usually appear in data with different probabilities. We want to encoding more frequent patterns with less number of bits. The main idea of Huffman encoding is as follows. First the character stream is analyzed to determine the character types and their relative frequencies. Then a binary tree is formed to determine the encoding. For example, assume the data contains only 4 characters a,b,c,d with probabilities $0.5, 0.3, 0.1, 0.1$. Then using a binary tree, the encoding method is as below.

character	encoding
a	1
b	01
c	001
d	000

  The average number of encoded bits per symbol would be: $1 \cdot 0.5 + 2 \cdot 0.3 + 3 \cdot 0.1 + 3 \cdot 0.1 = 1.7$. The actually Huffman encoding will be a little more complicated than the example.

The Huffman encoding are used in several compression algorithms such as JPEG, Facsimile compression.

- Run-length compression: try to shorten the expression of consecutively occurring data characters. For example, `AAABBCCCBBB` will be replaced by `3A2B3C3B`. For binary values the runs could be even longer, resulting in a better compression ratio. For example, an image may consist of runs having 25 zeros followed by a one or 55 zeros followed by a one. The run-length compression may be used for digital images by comparing pixel values that are adjacent and coding only the change in value.

- LZW (Lempel-Ziv-Welsh) compression: encode segments. The segments of the original text are stored in a dictionary which is built during the compression. When a segment appears later, it will be substituted with the index in the dictionary. The dictionary can be recovered from the compressed file so the dictionary needs not to be sent. `gzip` uses variations of that kind of method. It is understandable that this method is more efficient for big files. So `gzip` together with `tar` will be more efficient.

  In LZW compression, the dictionary is initialized with all strings of length 1, the characters of the possible input characters. When we have just read a segment W of the test and "a" is the next symbol (just following the given occurrence W), we do the follows. If Wa is not in the dictionary, we write the index of W in the output file, and add "Wa" in the dictionary. We then reset W to "a" and process the next symbol (following "a"). If Wa is in the dictionary, we process the next symbol, with segment Wa instead of W. Initially, the segment W is set to the first symbol of the source text. The algorithm works by scanning through the input string for successively longer substrings until it finds one that is not in the dictionary. When such a string is found, the index for the string without the last character (i.e., the longest substring that is in the dictionary) is retrieved from the dictionary and sent to output, and the new string (including the last character) is added to the dictionary with the next available code. The last input character is then used as the next starting point to scan for substrings.

  The decoding method is symmetric to the coding algorithm. The dictionary is recovered while the decompression process runs. Read a code

c in the compressed file, write into the output file the segment W having index c in the dictionary, and add the word Wa to the dictionary, where "a" is the first letter of the next segment.

- Image, audio and video compression: usually are not lossless compression. Some advanced techniques and some mathematics are used. Examples: Fractal compression, Facsimile compression, MPEG, MPEG-2.

  Wavelet compression can be either lossless or lossy.

2.5.2 Network Applications

The TCP/IP application layer is considered to be equivalent to the combined session, presentation and application layers of OSI model. Many applications based on TCP/IP have been developed over the years. Some popular ones are: telnet, FTP (file transfer protocol), SMTP (simple mail transfer protocol), SNMP (simple network management protocol), HTTP (hypertext transfer protocol).

A graphical user interface (GUI) is important in network applications.

2.6 Network performance

Network applications can be very chatty and bursty. Even if the network components improve in reliability, and the bandwidth is increased, the network may not function reliably. It is important to understand how the overall network performance is affected.

A network connection may be viewed as a pipe, where the bandwidth is the pipe width and the length of the pipe is the delay. The data present on the line may be found as the product of bandwidth times delay.

2.6.1 Delay

Delay refers to the amount of time needed for an offered frame to be delivered across the network to a remote user device. Because of the complexity of the network, it is difficult to calculate the exact delay. Approximate analytical and simulation models are used that closely replicate the actual situation.

There are many types of delay

- Processing delay: The time required to examine the packet's header and determine where to direct the packet. May also include other factors, such as the time needed to check for bit-level errors in the packet that occurred in transmitting the packet's bits from the upstream node to router.
- Queuing delay: by buffering of a packet at the queue as it waits to be transmitted onto the line. This delay depends on the traffic.
- Transmission delay: caused by limited media bandwidth. Suppose the length of the packet is L bits and the the transmission rate from one router to the other router is R bits/sec. Then the transmission delay is L/R.
- Propagation delay: the time required to propagate from the beginning of the link to the router. The propagation delay depends on the distance of two routers and the transmission rate.

In addition to the above delays, there can be other significant delays in the end systems.

2.6.2 Throughput and bandwidth

The throughput of a network is defined as the amount of data that can be transferred per unit time. It is not necessary equal to the offered media capacity in bits per second. Throughput and delay are often odds with each other. As the throughput is increased owing to higher offered load, the delay increases too. Over time, increasing the offered load causes an uncontrolled excessive delay, resulting in network congestion, whereupon throughput falls instead of increasing. Studies show that to keep the network stable, bandwidth utilization (the percentage of time it is busy) should be below 50%. However, this figure largely depends on the type of network being used. Bandwidth utilization should not serve as an independent metric. Higher utilization does not necessarily mean a better network design. On the other hand, it may sometimes reflect a need to increase the network bandwidth.

2.6.3 Error rate, congestion, and network reliability

High application performance requires both reliability and low delay. Many switches simply admit all traffic into the network, without regard to instanta-

neously available bandwidth in the network. If congestion occurs, the server module buffers will fill, and the delay will increase. Increased delay lowers throughput. Once the delay experienced by users has reached a critical level, the server module starts setting forward and backward explicit congestion notification bits (FECN/BECN) on all frames to notify end devices. If the congestion continues to grow, and the buffer is about to overflow, the DE bit is used to decide which frames are to be discarded first. When frames are discarded, throughput is lowered even more, as protocol windows shrink in size and frames are retransmitted.

Both low delay and low error rate are needed to achieve maximum throughput. Since the internetwork is more and more complicated and different switches and protocols are used, in general it is difficult to manage the network and to keep good performance.

Chapter 3

Application Layer

Network applications are the driving forces for the explosive development of the internet. Internet applications include the classic text-based applications such as text email, file transfers, newsgroups etc. Then applications as World Wide Web, instant messaging and P2P file sharing, voice-over-IP (VoIP), video conference (Skype), video distribution (YouTube), movies on demand (Netflix), on-line games (World of Warcraft), social networks (Facebook, Twitter), etc. New application are keeping invented.

In this chapter, we introduce a few network applications to get some basic idea about application layer of the networks.

3.1 Basics of network applications

Software of network applications usually need to run on the different end systems and communicate each other over the network. Since the each lower layer of the network protocols has implemented in different network devices, usually the application software, no matter written in C, Java or Python, needs not include the parts run on network core devices, such as routers or switches. Basically, we just need to know the interface of the lower layer, or the API of the lower layer.

3.1.1 Network application architectures

A network application is running on two or more computers. The network application architecture means the architecture how the application is struc-

tured over the various end systems. When develop an application, a special architecture must be chosen according to the specific application. Roughly, there are two architectural paradigms used in modern network applications.

- **Client-server architecture**. In this architecture, there are a server and many clients distributed over the network. The server is always on while a client can be randomly run. The server is listening on the network and a client initializes the communication. Upon the requests from a client, the server provide certain services to the client. Usually, there is no communication between two clients. The server has a fixed IP address and a client contact the server by sending a packet to the server's IP address. A server is able to communicate with many clients. The applications such as FTP, telnet, Web, e-mail etc use the client-server architecture.

  Original client-server architecture includes a single-server host. But now, in many cases, a single-server host is unable to keep up with all the requests from large number of clients. For this reason, a data center, housing a large number of hosts, is often used to create a powerful virtual server. For example, Google has 30 to 50 data centers distributed around the world, which collectively handle search, YouTube, Gmail, and other services. A date center can have hundreds of thousands of servers, that must be powered and maintained.

- **P2P architecture**. In this architecture, the application exploits direct communication between pairs of intermittently connected hosts, called peers. The peers are not owned by the service provider, but are instead desktops and laptops controlled by users. Many of today's most popular and traffic-intensive applications are based on P2P architecture. Examples include file sharing (BitTorrent), peer-assisted download acceleration (Xunlei), Internet telephone (Skype), IPTV (PPstream), etc.

  One of the most compelling feature of P2P architectures is their self-scalability. For example, in a P2P file sharing system, each peer generates workload by requesting files, but each peer also adds service capacity to the system by distributing files to other peers. There are also some challenges for the P2P applications.

  – ISP Friendly: most residential ISPs (DSL, cable ISPs etc) have been designed for "asymmetrical" bandwidth usage, i.e., for much

more downstream than upstream traffic. But many P2P applications shift upstream traffic from servers to residential ISPs, which significant stress on the ISPs.

- Security: Since the highly distribution and openness, P2P applications can be a challenge to security.
- Incentive: the success of P2P applications depend on convincing users to volunteer bandwidth, storage and computation resources to the applications.

There are also some applications using hybrid architectures, that combines both client-server and P2P elements. For example, servers are used to track the IP addresses of users, but user-to-user messages are sent directly between user hosts without passing through servers.

3.1.2 Processes communicating

When a program is running, it results processes. In the case of network application, processes will run on different hosts and include processes communication over the network.

A network application consists of pairs of processes that send messages to each other. Usually, we will label one of the two processes as the client and the other server. In the case of P2P, a process can both upload and download files. Usually, a process that initiates the communication is labeled as the client and the process that waits to be contacted to begin the session is labeled as the server.

Some important processes for the network applications:

- **The interface between the process and the network**

  In network application, any message sent from one process to another must go through the underlying network. Most of these processes send and receive messages through a software interface of the underlying network called socket. Socket is an Application Programming Interface (API) between the application layer and the transport layer within a host. The application developer has control of everything at the application layer side of the socket, but most of the transport layer side is controlled by operating system. A few things the application developer can control on the transport layer is choice of transport protocol

and fix a few transport-layer parameters such as maximum buffer and maximum segment sizes.

- **Addressing processes**

  In the Internet, the host is identified by its IP address. The message must include the IP address of the sending process and the IP address of the destination process. In addition to the addresses, the sending process also needs to identify the receiving process (the receiving socket), because a host may run several sockets at the same time. A destination port number serves this purpose.

3.1.3 Transport services for applications

Networks usually provide more than one transport-layer protocols for different applications. An application developer should choose certain protocol according to the type of applications. Different protocols may provide different services.

- **Reliability of data transfer**

  In some applications, data need to be guaranteed that it sent by one end of the application is delivered correctly and completely to the other end of the application. Some of the transport protocol provides to an application such a guaranteed data delivery service, that is said to provide reliable data transfer (such as TCP). Some protocol does not provide reliable transfer. Some of the data sent by the sending process may never arrive at the receiving process (e.g., UDP). This may be acceptable for loss-tolerant applications, such as multimedia applications, in which lost data might result in a small glitch in the audio/video.

- **Throughput**

  The throughput for the communications between two processes over the network means the rate at which the sending process can deliver bits to the receiving process. Since other hosts are using the network, the throughput can fluctuate with time. Some applications, called bandwidth-sensitive applications, need a guaranteed throughput. Some

transport protocol provides the service of guaranteed available throughput at some specified rate. With this service, the application could request a guaranteed throughput of r bits/sec, and the transport protocol would then ensure that the available throughput is always at least r bits/sec. Some applications may not need the least throughput limitation, such as email. These applications are called elastic applications.

- **Timing**

  A transport-layer protocol can also provide timing guarantees. For example, it guarantees that every bit that sender pumps into the socket arrives at the receiver's socket no more than 100 msec later. Many "real time" applications need such services (Internet telephony, multi-player games, etc).

- **Security**

  A transport layer protocol can provide an application with one or more security services.

Today's Internet can often provide satisfactory services to time-sensitive applications, but it cannot provide any timing or throughput guarantees.

3.2 HTTP and the web

The World Wide Web appears in the early 1990s. In 1989, CERN (the European Center for Nuclear Research) initialized the main idea. In 1994, CERN and MIT signed an agreement setting up the W3C (World Wide Web Consortium) , an organization devoted to further developing the Web. The appearance of Web dramatically changed the Internet. Web has many advantages for a lot of applications. It operates on demand so that the users receive what they want when they want it. It provides a easy way for everyone make information available over the world. Search engine and hyperlinks, graphics and multimedia, web based emails, social networks, web telephony, $\cdots$, applications keep emerging that elevated the Internet from just one of many data networks to essentially the one and only one data network.

Many tools are used for the web applications such as forms, JavaScript, Java applets and other devices, that let the developers use web as a platform many killer applications.

3.2.1 Overview of HTTP

The HyperText Transfer Protocol (HTTP) , the Web's application-layer protocol, is at the heart of the web. HTTP is defined in RFCs 1945 and 2616, which is implemented in two programs: a client program and a server program. The client program and the server program, executing separately, talk to each other by exchanging HTTP messages. HTTP defines the structure of these messages and how the client and server exchange the messages.

A web page consists of objects. An object is a file, such as an HTML (HyperText Markup Language) file, a JPEG image, a Java applet, a video chip etc, that is addressable by a single URL (Uniform Resource Locator). Most web pages consist of a base HTML file and several referenced objects. The base HTML file references the other objects in the page with the object's URLs. Each URL has two components: the hostname of the server that houses the object and the object's path name. The web browsers (such as Internet Explorer and Google Chrome) implement the client side of HTTP and the web servers (such as Apache and Microsoft Internet Information Server) implement the server side of HTTP.

When a user requests a web page, the browser sends HTTP request messages for the objects in the page to the server. The server receives the requests and responds with HTTP response messages that contain the objects.

HTTP uses TCP as its underlying transport protocol. The HTTP client first initiates a TCP connection with the server. Once the connection is established, the browser and the server processes access TCP through their socket interfaces.

HTTP is a stateless protocol, in which the server sends requested files to clients without storing any state information about the client. The web uses the client-server architecture. The web server is always on with a fixed IP address.

3.2.2 HTTP connections

In many internet applications, the client and server communicate for an extended period of time. When this client-server interaction takes place over TCP, a decision should be made: each request/response pair be sent over a separate TCP connection or the sequence of request/response pairs be sent over the same TCP connection. These different connections are called non-persistent connections or persistent connections. HTTP uses persistent con-

nections in its default mode, but HTTP client and server can be configured to use non-persistent connections as well.

- **HTTP with non-persistent connections**

  Suppose a web page (server) consists a base HTML file and 10 JPEG images. Under the non-persistent connection, the client first requests the web by creating a TCP connection to the server. The server then sends the base HTML file to the client over the TCP connection and then closes that TCP connection. When the client needs the JPEG image, a new TCP connection will be create. With non-persistent connection, each TCP connection only be used to send one object.

- **HTTP with persistent connections**

  With persistent connections, the server leaves the TCP connection open after sending a response. Subsequent requests and responses between the same client and server can be sent over the same connection. In particular, an entire web page (base HTML base file and other objects) can be sent over a single persistent TCP connection. Moreover, multiple web pages residing on the same server can be sent from the server to the same client over a single persistent TCP connection. Usually, the HTTP server closes a connection when it isn't used for a certain time. The default mode of HTTP uses persistent connections with pipelining.

Non-persistent connections have some shortcomings. To establish a new TCP connection, some simple packet exchange must perform. So create more TCP connections consumers more communication resources. Modern browsers can be configured to obtain parallel TCP connections. In their default modes, most browsers open 5 to 10 parallel TCP connections.

3.2.3 HTTP message format

HTTP specification include the definitions of the HTTP message formats. There are two formats.

- HTTP request message.
- HTTP response message

Let's first look at an example of HTTP request message:

```
GET /cs4453/sample.html HTTP/1.1
Host: ccc.cs.lakeheadu.ca
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
```

The message is written in ASCII text. The first line of the HTTP request message is called the request line. The subsequent lines are called the header lines. The request line has three fields: the method, the URL and the version. A request message can contain several header lines. A carriage return and a line feed are required to indicate the end of the message. The meaning of the header lines are as follows. The `Connection:close` means requesting a non-persistent connection. `User-agent:Mozilla/5.0` means the browser used is the Firefox. `Accept-language:fr` means French is the preferred language. If there is no French Web, then the default page will be sent. `Accept-language` is one of many content negotiation headers available in HTTP.

HTTP defined the request methods as follows:

Method	Description
GET	Read a Web page
HEAD	Read a Web page's header
POST	Append to a Web page
PUT	Store a Web page
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Connect through a proxy
OPTION	Query options for a page

The GET method means retrieve whatever information (in the form of an entity) is identified by the Request-URI. The HEAD method is identical to GET except that the server MUST NOT return a message-body in the response. The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line. The actual function performed by the POST method is determined by the server and is usually dependent on the Request-URI. The PUT method requests that the enclosed entity

be stored under the supplied Request-URI. If the Request-URI refers to an already existing resource, the enclosed entity SHOULD be considered as a modified version of the one residing on the origin server. If the Request-URI does not point to an existing resource, and that URI is capable of being defined as a new resource by the requesting user agent, the origin server can create the resource with that URI. The DELETE method requests that the origin server delete the resource identified by the Request-URI. This method MAY be overridden by human intervention (or other means) on the origin server. The client cannot be guaranteed that the operation has been carried out, even if the status code returned from the origin server indicates that the action has been completed successfully. The TRACE method is used to invoke a remote, application-layer loop- back of the request message. The specification reserves the method name CONNECT for use with a proxy that can dynamically switch to being a tunnel. The OPTIONS method returns the HTTP methods that the server supports for the specified URL. This can be used to check the functionality of a web server by requesting `'*'` instead of a specific resource.

Next let's see an example of HTTP response message.

```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 13 Jan 2015 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified:  Tue, 13 Jan 2015 15:11:04 GMT
Content-Length: 6821
Content-Type: text/html
... ... ...
... ... ...
```

The response message contains three sections: an initial status line, several header lines and then the entity body (the details of this section is missed). The status line has three fields: the protocol version, a status code and a corresponding status message. Many status codes are defined in HTTP including:

- `1xx` Informational
- `2xx` Success
- `3xx` Redirection

- `4xx` Client error
- `5xx` Server error

Some of these codes only supported by HTTP 1.1, but not HTTP 1.0. Microsoft, Nginx, and others also gave some extensions of these codes. Some frequently used codes are:

- `200 OK`: Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource.
- `301 Moved Permanently`: This and all future requests should be directed to the given URI.
- `400 Bad Request`: The server cannot or will not process the request due to something that is perceived to be a client error.
- `403 Forbidden`: The request was a valid request, but the server is refusing to respond to it.
- `404 Not Found`: The requested resource could not be found but may be available again in the future. Subsequent requests by the client are permissible.
- `500 Internal Server Error` A generic error message, given when an unexpected condition was encountered and no more specific message is suitable.
- `503 Service Unavailable`: The server is currently unavailable (because it is overloaded or down for maintenance). Generally, this is a temporary state.
- `505 HTTP Version Not Supported`: The server does not support the HTTP protocol version used in the request.

The header lines for the request messages are created by the web browser, that are determined by the version of browser and the user's configuration.

3.2.4 Caching and GET extension

A Web cache is a network entity that may used to response HTTP requests on the behalf of an origin Web server. The Web cache has its own disk storage and keeps copies of recently requested objects in this storage. The user's HTTP requests are first directed to the web cache. If the cache has the object requested, it returns the object within an HTTP response message to the client browser. If the cache does not have that object, then it connects to the original server and ask for the object. After it obtained the object, it then sends the object to the client.

Although caching may reduce the response time, it could send a old version of the object to the client. To avoid that happen, conditional GET method can be used. The semantics of the GET method change to a "conditional GET" if the request message includes an If-Modified-Since, If-Unmodified-Since, If-Match, If-None-Match, or If-Range header field. A conditional GET method requests that the entity be transferred only under the circumstances described by the conditional header field(s). The conditional GET method is intended to reduce unnecessary network usage by allowing cached entities to be refreshed without requiring multiple requests (the cache need not to check if the object is updated by the server for every request) or transferring data already held by the client.

The following is an example of using conditional GET:

```
GET /fruit/kiwi.fig HTTP1.1
Host: www.exoriguecuisine.com
If-modified-since: Wed, 7 Sep 2011 09:23:24
```

The response is:

```
HTTP/1.1 304 Not Modified
Date: Sat, 15 Oct 2011 15:39:29
```

Sometimes a Web cache is also called a proxy server. However, a proxy server has wider meaning: cache proxy is one type of the proxy servers. There are proxy servers for other purpose, such as load balancing, security, compression, spoon feeding etc.

Another extension of GET is "partial GET" method. The semantics of the GET method change to a "partial GET" if the request message includes a Range header field. A partial GET requests that only part of the entity be

transferred. The partial GET method is intended to reduce unnecessary network usage by allowing partially-retrieved entities to be completed without transferring data already held by the client.

Caching can also be used in client side. A browser can reduce download times significantly by saving a copy of each image in a cache on the user's disk and using the cached copy next time for visiting the same web page. The HEAD method can be used to see if the object is updated and a new download should be performed.

3.2.5 Cookies

HTTP servers are designed as stateless so that a Web server can handle thousands of simultaneous TCP connections. However, in many cases a Web site needs to identify users. For this purpose, HTTP uses cookies (RFC 6265), that allow sites to keep track of users.

When a user first time visit a site, the user may provide a user identification or some information. The web site can put a `Set-Cookie` header line in the response message. This line will provide a unique cookie value to the user. The user then store the cookie in the computer and add the `Cookie` line in the subsequent requests using the cookie provided by the server.

The following is an simple example to explain how to use the cookies.

First the browser sends a request:

```
GET /index.html HTTP/1.1
Host: www.example.ca
```

The server response:

```
HTTP/1.0 200 OK
Content-type: text/html
Set-Cookie: name=<value>
Set-Cookie: name2=<value2>; Expires=Wed, 09 Jun 2021 10:18:14 GMT

(content of page)
```

The browser continues the requests:

```
GET /spec.html HTTP/1.1
Host: www.example.ca
Cookie: name=<value>; name2=<value2>
Accept: */*
```

Cookie specifications suggest that browsers should be able to save and send back a minimal number of cookies. In particular, a web browser is expected to be able to store at least 3000 cookies of four kilobytes each, and at least 50 cookies per server or domain. The value of a cookie can be modified by the server by sending a new `Set-Cookie` line in response of a page request. The browser then replaces the old value with the new one. Cookies can also be set by JavaScript or similar scripts running within the browser.

The server will store the cookies associating with the users information (user identity, address, email, credit card number etc) to its database. The next time when the user visit the web site, the web server will get the information from the database so that the user needs not provide again.

Cookies may have different types. A session cookie, also referred as in memory cookie. When a expiry date of validity interval is not set at the `Set-Cookie` line, a session cookie is created. When a expiry is set, the cookie is called persistent cookie. Some secure cookie is used for security purpose for the https connection.

3.2.6 Dynamic Web pages

Now many Web pages are used for different applications and services. The applications run inside the browser, with user data stored on servers in Internet data centers. They use Web protocols to access information via the internet, and the browser to display a user interface. The advantage of this approach is that users do not need to install separate application programs, and user data can be accessed from different computers and back up by the service operator. For these Web applications, dynamic content of web is needed. The dynamic content can be generated by programs running on the server or in the browser. For example, consider a map service that lets the user enter a street address and presents a corresponding map of the location. Given a request for a location, the Web server must use a program to create a page that shows the map for location from a database of streets and other geographic information. There will be more to dynamic content. The page that is returned may contain other programs that run in the browser. In the map example, the program may let the user find routes and explore nearby areas etc. To update the page, the program need more data from the server and the server has to access different databases and other resources. Most of these requests and responds happen in the background and the user may not

know because the page URL and title typically do not change. By including client side programs, the page can present a more responsive interface than with server side program along.

First let us see how to generate server side dynamic web page. Several methods have been used for this purpose. Some of the most popular methods are as follows.

- CGI (Common Gateway Interface) (RFC 3875): CGI provides an interface to allow web servers to talk to back-end programs and scripts that can accept input and generate HTML pages in response. These programs can be implemented in different languages such as Python, Peal, Ruby and so on. Usually, the programs invoked via CGI live in a directory called `cgi-bin`.

- PHP (Hypertext preprocessor): In this approach, little scripts are embedded inside HTML pages and have them be executed by the server to generate the page. PHP is also a powerful programming language for interfacing the Web and a server database. PHP is open source code and widely used. Usually, servers identify Web pages containing PHP from the file extension `.php` rather than `html`.

- JSP (JavaServer Pages): It is similar to PHP, but the dynamic part is written in the Java instead of in PHP. Pages using this technique have the file extension `.jsp`. ASP.NET (Active Server Pages) is Microsoft's version of PHP and JavaServer Pages.

Next, we consider the generating of client side web pages. The above scripts let the server interact with databases and then form the web page and send the page to the client. However, they still can not handle some actions useful in client side, such as response to mouse movements or interact with users directly. For this purpose, we need some scripts embedded in HTML pages that are executed on the client machine rather than the server machine. Start with HTML 4.0, such scripts are permitted using the tag `<script>`.

The most popular scripting language for the client side is JavaScrpt. JavaScript is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is important to understand that JavaScript and PHP look similar as both of them

are embedded inside the HTML files, they are processed differently. In the case of PHP, the browser collect information into a long string and sends it off to the server as a request for a PHP page. The server loads the PHP file and executes the embedded PHP script to produce a new HTML page. That page is then sent back to the browser for display. But in the case of JavaScript, the work is done locally, inside the browser. There may be no contact with the server. Consequently, the result of a JavaScript is displayed virtually instantaneously, while the PHP may have a delay. There are some other scripts used at client side:

- VBScript: based on Visual Basic used on Windows platforms.
- Java applets: small Java programs that have been compiled into machine instructions for JVM (Java virtual machine).
- ActiveX controls: programs compiled to x86 machine language and executed on the bare hardware.

In general, JavaScript is easier to write, Java applets execute faster, and ActiveX controls run faster than others. But ActiveX is basically platform dependent while Java applets are more portable.

JavaScript is also used in server-side network programming with runtime environments such as Node.js, game development and the creation of desktop and mobile applications.

Despite some naming, syntactic, and standard library similarities, JavaScript and Java are otherwise unrelated and have very different semantics. The syntax of JavaScript is actually derived from C, while the semantics and design are influenced by Self and Scheme programming languages.

3.3 FTP

File Transfer Protocol (FTP) is first published in 1971 (RFC 114) that were implemented on hosts at MIT. Then many comments and improving proposals are published in RFC. In 1985, a standard FTP is published as RFC 959. Several extensions are published since then including anonymous FTP, secure FTP, FTP for IPv6 or NAT etc.

FTP is used for a local host transfers files from or to a remote host which runs an FTP server and the user has an account on it. The user initializes the communication and provides a user identification and a password. After

proving this authorization information, the user can transfer files between the local host and the remote host.

Both FTP and HTTP are file transfer protocols, but they have some important differences. FTP uses two parallel TCP connections to transfer file, a control connection and a data connection. The control connection is used for sending control information between the local and remote hosts (such as user identification, password, commands "put", "get" etc.). The data connection is used to transfer files. As we have seen in previous section, HTTP basically uses one TCP connection for both control and transfer.

When a user starts an FTP session, the client side of FTP initiates a control TCP connection with the server side on server port 21. The client side sends user identity and password and other commands over this control connection. When the server side receives a command for a file transfer over the control connection, the server side initiates TCP data connection to the client side. FTP sends exactly one file over the data connection and then closes the data connection. If the user wants to transfer another file in the session, FTP opens another data connection. Therefore, usually the control connection remains open throughout the duration of the user session, but a new data connection is created for each file transferred within a session.

During an FTP session, the FTP server must maintain the state about the user. For example, the server must keep track of the user's current direction as the user changes directions. Keeping track of this state information for each ongoing user session significantly constrains the number of sessions for a FTP server to maintain simultaneously. Recall that HTTP server uses stateless connections and thus can handle more clients.

3.3.1 FTP commands and reply codes

FTP defines many commands including:

- Access control commands, for example:
  - `USER` (user name): The argument field is a Telnet string identifying the user. The user identification is that which is required by the server for access to its file system.
  - `PASS` (password): The argument field is a Telnet string specifying the user's password.

  - `CWD` (change working directory): This command allows the user to work with a different directory or dataset for file storage or retrieval without altering his login or accounting information.
  - `QUIT` (logout): This command terminates a USER and if file transfer is not in progress, the server closes the control connection. If file transfer is in progress, the connection will remain open for result response and the server will then close it.

- Transfer parameter commands, for example:
  - `PORT` (data port): The argument is a HOST-PORT specification for the data port to be used in data connection. There are defaults for both the user and server data ports, and under normal circumstances this command and its reply are not needed.
  - `TYPE` (representation type): The argument specifies the representation type such as: ASCII (A), EBCDIC (E), Image (I), Local byte Byte size (L).

- FTP service commands, for example:
  - `RETR` (retrieve, i.e., "get"): This command causes the server to transfer a copy of the file, specified in the path name, to the server or user at the other end of the data connection.
  - `STOR` (store, i.e., "put"): This command causes the server to accept the data transferred via the data connection and to store the data as a file at the server site.
  - `LIST` (list): This command causes a list to be sent from the server to the other side. If the path name specifies a directory or other group of files, the server should transfer a list of files in the specified directory.
  - `HELP` (help): This command shall cause the server to send helpful information regarding its implementation status over the control connection to the user.

The FTP also defined the reply codes with . The followings are examples of some codes associate with messages:

- `200 Command okay`

- `503 Bad sequence of commands.`
- `221 Service closing control connection.`
- `125 Data connection already open; transfer starting.`
- `226 Closing data connection.`
- `230 User logged in, proceed.`
- `331 User name okay, need password.`
- `425 Can't open data connection.`
- `452 Requested action not taken. Insufficient storage space in system.`

3.4 Electronic mail

Electronic mail (now called email) is the most important internet application at the beginning of the Internet. It remains one of the most utilized applications now. E-mail is an asynchronous communication medium, i.e., people send and read messages when it is convenient for them without coordinating with other party's schedules.

In high level, an e-mail system has three major components:

- User agents: allow users to read, reply to, forward, save and compose messages.
- Mail servers: contain mailboxes for users for each user. A message is first sent to the sender's mail server and the senders mail server sends the message to the receiver's mail server. If something is wrong for the receiver's server, then the message is held in a message queue and will try to send later.
- Protocols: most important protocol is the SMTP (Simple Mail Transfer Protocol)

3.4.1 SMTP

SMTP (RFC 5321) is the most important (and old) protocol of the email system. There are some characteristics made this application different from others.

- Message body uses 7-bit ASCII code only.
- Normally, no intermediate mail servers used for sending mail.
- Mail transmissions across multiple networks through mail relaying.

Usually, mail servers are listening at port 25. The sending server initiates a TCP connection to the receiving mail server. If the receiver's server is down, the sending server will try later. If the connection is established, then the client and the server perform some application layer handshaking. The client indicates the e-mail address of the sender and the recipient. Then the client sends the message to the server over the same TCP connection.

In the following example, `S` is the recipient's mail server and `C` is the sender's mail server.

```
S: 220 smtp.example.com
C: HELO send.example.org
S: 250 Hello send.example.org, I am glad to meet you
C: MAIL FROM:<bob@example.org>
S: 250 bob@example.org ... Sender ok
C: RCPT TO:<alice@example.com>
S: 250 alice@example.com ... Recipient ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: Date: Tue, 15 January 2008 16:02:43 -0500
C: Subject: Test message
C:
C: Hello Alice.
C: This is a test message with 3 header fields and 4 lines.
C: Your friend,
C: Bob
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221   smtp.example.com closing connection
```

The client notifies the receiver of the originating email address of the message in a `MAIL FROM` command. The corresponding SMTP command is `RCPT TO` for the recipient's address. Each successful reception and execution of a command is acknowledged by the server with a result code and response message.

The transmission of the body of the mail message is initiated with a `DATA` command after which it is transmitted verbatim line by line and is terminated with an end-of-data sequence. This sequence consists of a new-line (`<CR><LF>`), a single full stop (period), followed by another new-line.

The server's positive reply to the end-of-data, as exemplified, implies that the server has taken the responsibility of delivering the message. A message can be doubled if there is a communication failure at this time, e.g. due to a power shortage: Until the sender has received that 250 reply, it must assume the message was not delivered.

The `QUIT` command ends the session. If the email has other recipients located elsewhere, the client would QUIT and connect to an appropriate SMTP server for subsequent recipients after the current destination(s) had been queued.

Some other commands are

- `RSET` (Reset): the current mail transaction will be aborted.
- `VRFY` (Verify): ask the receiver to confirm that the argument identifies a user or mailbox.
- `EXPN` (Expand): ask the receiver to confirm that the argument identifies a mailing list, and if so, to return the membership of that list.
- `HELP` (Help): let the server send helpful information to the client.

SMTP also defined reply codes. The following are some common used codes:

- `220 <domain> Service ready`
- `221 <domain> Service closing transmission channel`
- `250 Requested mail action okay, completed`
- `354 Start mail input; end with <CRLF>.<CRLF>`

- `421 <domain> Service not available, closing transmission channel`
- `451 Requested action aborted: local error in processing`
- `500 Syntax error, command unrecognized (This may include errors such as command line too long)`
- `501 Syntax error in parameters or arguments`

Mail message formats are defined in RFC 5322. The header lines and the body messages are separated by a blank line. Note that the message format is not the SMTP commands. It is the format of the message, which the sending server may used to create some SMTP commands. The following is an example.

```
From: "Joe Q. Public" <john.q.public@example.com>
To: Mary Smith <mary@x.test>, jdoe@example.org, Who? <one@y.test>
Cc: <boss@nil.test>,  <sysservices@example.net>
Date: Tue, 1 Jul 2003 10:52:37 +0200
Message-ID: <5678.21-Nov-1997@example.com>

Hi everyone.
```

After the message header, a blank line follows, then the message body (in ASCII) follows.

3.4.2 MIME

SMTP uses ASCII code only, which simplifies the structure but causes the limitations of using languages other than English. People also want to use email to send binary files. Multipurpose Internet Mail Extensions (MIME) was developed for solving these problems.

MIME defines five new message headers:

Header	Meaning
MIME-version:	Identifies the MIME version
Content-Description:	Human-readable string telling what is in the message
Content-ID:	Unique identifier
Content-Transfer-Encoding:	How the body is wrapped for transmission
Content-Type:	Type and format of the content

The `Content-Transfer-Encoding` tells how the body is wrapped for transmission through the network. MIME defines five transfer encoding schemes, plus an escape to new scheme – just in case.

- ASCII characters use 7 bits.
- 8-bit characters, that is, all values from 0 to 255 are allowed.
- Base64 encoding: change binary codes to a form that satisfies the rules of 7 bits (ASCII code).
- Quoted-printable encoding: used for contents mostly in ASCII code, but a small part is not ASCII code.
- Binary.

A user also can specify a user-defined encoding in the `Content-Transfor -Encoding` header.

MIME defines content type for the header `Content-Type`, which specifies the nature of the message body and has had an impact well beyond email. For example, content downloaded from the Web is labeled with MIME types so that the browser knows how to present it.

Initially, MIME types were defined in RFC 1521. Each type has one or more available subtypes. Hundreds of subtypes have been added since then. The following display the types.

Type	Example subtypes	Description
text	plain, html, xml, css	Test in various formats
image	gif, jpeg, tiff	Pictures
audio	basic, mpeg, mp4	Sounds
video	mpeg, mp4, quichtime	Movies
model	vrml	3D model
application	octet-stream, pdf, zip	Data produced by applications
message	http, rfc822	Encapsulated message
multipart	mixed, alternative, parallel	Combination of multiple types

The `multipart` type allows a message to contain more than one part, with the beginning and end of each part being clearly delimited. The `mixed` subtype allows each part to be a different type, with no additional structure

imposed. Many email program allow the user to provide one or more attachments to a text message. These attachments are sent using multipart type. The `alternative` subtype allows the same message to be included multiple times but expressed in different media. For example, a message could be sent in ASCII, in HTML and in PDF. A properly designed user agent would display it according to user preferences. The `alternative` subtype can also be used for multiple languages. The `parallel` subtype is used when all parts must be "viewed" simultaneously. For example, movies often have an audio channel and a video channel. Movies are more effective if these two channels are played back in parallel.

The type and subtype are separated by a slash, such as "`Content-Type: video/mpeg`".

The following is an example of using MIME types:

```
From: alice@cs.lakeheadu.ca
To: bob@ee.lakeheadu.ca
MIME-Version: 1.0
Message-Id:<09384756.AA785104@cs.lakeheadu.ca>
Content-Type: multipart/mixed; boundary=frontier
Subject: Try multi-part

This is a message with multiple parts in MIME format.
--frontier
Content-Type: text/plain

This is the body of the message.
--frontier
Content-Type: application/octet-stream
Content-Transfer-Encoding: base64

PGh0bWw+CiAgPGhlYWQ+CiAgPC9oZWFkPgogIDxib2R5PgogICAgPHA+VGhp
cyBpcyB0aGUgYm9keSBvZiB0aGUgbWVzc2FnZS48L3A+CiAgPC9ib2R5Pgo8
L2h0bWw+Cg==
--frontier--
```

3.4.3 Mail access protocols

Since mail servers should be always on and have fixed IP addresses, it is not realistic to run the mail severs on personal computers, laptops etc. Then the problem is how a person can access his email using his PC or laptop. Mail access protocols are used to solve this problem. There are several popular mail access protocols, including Post Office Protocol (POP3), Internet Mail Access Protocol (IMAP), and HTTP.

POP3 is a simple mail access protocol defined in RFC 1939. POP3 server will listen at port 110. The user agent at client's computer opens a TCP connection to the main server. POP 3 then progresses through three phases:

- Authentication: the user agent sends a user name and password to authenticate the user.
- Transaction: the user agent retrieves messages; the user agent can also make or remove a mark for a message deletion and get statistics of the user maildrop.
- Update: after the user issued a quit command in the transaction phase, the POP3 server removes all messages marked as deleted.

Let us see an example. At the first phase:

```
S:  +OK POP3 server ready
C:  USER Bob
S:  +OK
C:  PASS some string
S:  +OK user successfully logged on
```

In second phase, a user agent can be configured to "download and delete" or " download and keep" mode. In download-and-delete mode, commands `LIST` (list the mail number and size), `RETR` (download the message) and `DELE` (mark deletion) can be used by the user agent.

```
C:    STAT
S:    +OK 2 320
C:    LIST
S:    +OK 2 messages (320 octets)
S:    1 120
```

```
S:    2 200
S:    .
C:    RETR 1
S:    +OK 120 octets
S:    <the POP3 server sends message 1>
S:    .
C:    DELE 1
S:    +OK message 1 deleted
C:    RETR 2
S:    +OK 200 octets
S:    <the POP3 server sends message 2>
S:    .
C:    DELE 2
S:    +OK message 2 deleted
C:    QUIT
S:    +OK  POP3 server signing off (maildrop empty)
C:  <close connection>
S:  <wait for next connection>
```

Using the download-and-delete mode, the mails may partitioned and stored in different computers if the use uses different computers. That may cause inconvenience for a user. On the other hand, on the download-and-keep mode, the maildrop cannot be clear up. In the update phase, all the marked messages will be deleted. The POP3 server does not carry state information across POP3 sessions, that greatly simplifies the implementation of a POP3 server.

One disadvantage for the POP3 is that the user cannot manage his mails at the remote mail server. For example, the user cannot create folders hierarchy, move the mails to different folders, delete messages, etc. Usually, a user only can download all the mails to the local machine and manage the mails locally. But if a user prefers to use different computers, then there will be the management problem.

IMAP (RFC 3501) is another mail access protocol, which has more features than POP3. An IMAP server will associate each message with a folder. When a message first arrives at the server, it is associated with the recipient's INBOX folder. The recipient can then move the message into a new, user created folder, read the message, delete the message, and so on. IMAP also

provides commands that allow users to search remote folders for messages matching specific criteria. Different from POP3, an IMAP server maintains user state information across IMAP sessions (such as name of folders and which messages are associated with which folder, etc).

Another important feature of IMAP is that it has commands that permit a user agent to obtain components of messages. For example, a user agent can obtain just the message header of a message or just one part of a multipart MIME message.

IMAP has many good features so that user can manage the mails remotely and thus the user can use different devices to access the maildrop. But IMAP server is much complicated for implementation.

HTTP are now used for Web-based email accessing. Hotmail introduced Web-based access first. Now Web-based emails are provided by many corporations and universities including Google, Yahoo etc. In this approach, the user agent is an ordinary Web browser, and the user communicates with its remote server via HTTP.

3.5 P2P applications

Peer-to-peer architecture is different from client-server architecture. In P2P, each node (called peers) acts somehow as a client and server at the same time. The peers are not owned by a service provider and not supposed to be always listening on the Internet. The peers are dynamic, i.e., some peers will join some peers will leave from time to time.

3.5.1 P2P file distribution

One suitable application of P2P architecture is file distribution. One popular P2P file distribution protocol is BitTorrent which are used for several successful software products, such as Xunlei, Transmission, μTorrent, etc.

To simplify the discussion, we consider the following scenarios. Suppose a server has a large file, which N computers want to download. In the case of client-server architecture, each of the N computers will connect to the server and download a copy of the file to local. For the P2P architecture, a peer is not necessary download a copy from the server. It may download from other peer.

Let us look at the distribution time (the time it takes to get a copy of the file to all N peers). Suppose the length of the file is F. The upload rate for the server is u_s. The upload rate and the download rate for ith computer is u_i and d_i, respectively, as show in Figure 3.1

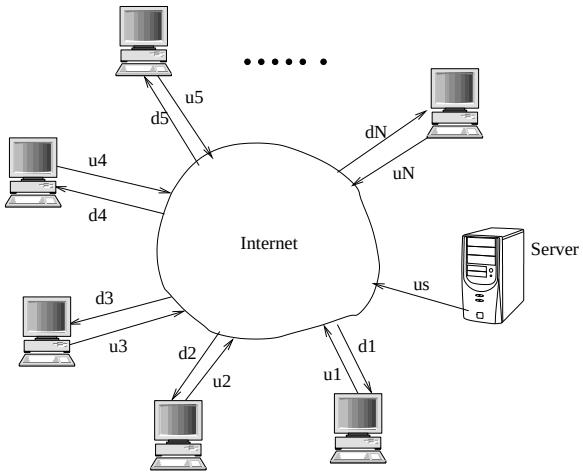

Figure 3.1: P2P architecture

First let us look at the distribution time for the client-server architecture, that is denoted as D_{cs}. Under this situation,

- The server must transmit one copy of the file to each of the N peers. So the server needs transmit NF bits. Since the server's upload rate is u_s, the distribution time is at least NF/u_s.
- Let d_{min} denote the download rate of the peer with the lowest download rate. The peer with the lowest download rate cannot obtain all F bits of the file in less than F/d_{min}.

Put the above two observations together, we have

$$D_{cs} \geq \max\left\{\frac{NF}{u_s}, \frac{F}{d_{min}}\right\}.$$

It is easy to see that when N is large, the bound of D_{cs} is $\frac{NF}{u_s}$ and controlled by N. So the bound of the distribution time increases linearly with the number of peers N.

Next we can look at the same situation, but using the P2P architecture. Let D_{P2P} denote the distribution time for the P2P mode.

- At the beginning of the distribution, only the server has the file. So the minimum distribution time is at least F/u_s.
- Similar to the situation of client-server architecture, the peer with the lowest download rate cannot obtain all the F bit in less than F/d_{min}. So the minimum distribution time is at least F/d_{min}.
- Observe that the total upload capacity of the system as a whole is $u_{total} = u_s + u_1 + u_2 \cdots + u_N$. Since the system must deliver F bits to each of the N peers, the minimum distribution time is at least $NF/(u_s + u_1 + u_2 \cdots + u_N)$.

So we have

$$D_{P2P} \geq \max\left\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_s + \sum_{i=1}^{N} u_i}\right\}.$$

Researchers have proved that the above bound can be achieved if each peer can redistribute a bit as soon as it received the bit. To compare the distribution time, we assume that all the upload rate for the peers is u. When N is large, D_{cs} is about $\frac{NF}{u_s}$, while D_{P2P} is about

$$\frac{NF}{u_s + \sum_{i=1}^{N} u} < \frac{F}{u}.$$

The above comparison shows that when N is large, the P2P architecture is not only consuming less distribution time, but also be self-scaling. The distribution time can be independent to the number of peers.

3.5.2 BitTorrent

Now, let us look some basic idea of the BitTorrent protocol. The details of BitTorrent is rather complicated. We only can give some outlines and some interesting techniques.

In BitTorrent lingo, the collection of all peers participating in the distribution of a particular file is called a torrent. Peers in a torrent download equal-size chunks of the file from one another, with a typical chunk size of 256 KBytes. When a peer first joins a torrent, it has no chunks. Over time,

it accumulates more and more chunks. While it downloads chunks it also uploads chunks to other peers. Once a peer has acquired the entire file, it may leave the torrent or remain in the torrent and continue to upload chunks to other peers. Also, any peer may leave the torrent at any time without download whole file and rejoin the torrent later.

Each torrent has an infrastructure node called tracker. When a peer joins a torrent, it registers itself with the tracker and periodically informs the tracker that it is still in the torrent. A given torrent may have fewer than ten or more than a thousand peers participating at any instant of time. When a new peer, Alice, joins the torrent, the tracker randomly selects a subset of peers from the set of participating peers, and sends the IP addresses of these peers to Alice. Alice then tries to establish concurrent TCP connections with all the peers on this list. We can call all the peers with which Alice succeeds in establishing a TCP connection neighboring peers. As time evolves, some of these peers may leave and other peers may attempt to establish TCP connections with Alice.

At any given time, each peer will have a subset of chunks from the file, with different peers having different subsets. Periodically, Alice will ask each of her neighboring peers for the list of chunks they have. If Alice has L different neighbors, she will obtain L lists of chunks. Alice will issue requests for chunks she currently does not have.

So at an instant time, Alice will have a subset of chunks and will know which chunks her neighbors have. Then Alice needs to decide which chunks she will ask first and which neighbors she will send the requested chunks. For these, Alice uses a technique called rarest first. The idea is to determine, from among the chunks she does not have, the chunks that are the rarest among her neighbors and then request those rarest chunks first. In this manner, the rarest chunks get more quickly redistributed, aiming to equalize the numbers of copies of each chunk in the torrent.

To determine which requests she responds to, Alice uses a trading algorithm. Alice gives priority to the neighbors that are currently supplying her data at the highest rate. Specifically, for each of her neighbors, Alice continually measures the rate at which she receives bits and determines the four peers that are feeding her bits at the highest rate. She then reciprocates by sending chunks to these four peers. Every 10 seconds, she recalculates the rates and possible modifies the set of four peers. In BitTorrent lingo, these four peers are said to be unchoked. Every 30 seconds, she also picks one additional neighbor at random and sends it chunks. In BitTrront lingo, it is

said to be optimistically unchoked. In other words, every 30 seconds, Alice will randomly choose a new trading partner and initiate trading with that partner. If the two peers are satisfied with the trading, they will put each other in their top four lists and continue trading with each other until one of the peers finds a better partner. All other neighboring peers besides these five peers are choked peer.

There are other interesting techniques used in BitTorrent.

3.5.3 Distributed hash table (DHT)

We can use P2P architecture to form a distributed database. We consider a simple database, which contains (key, value) pairs. In this architecture, each peer will only hold a small subset of the data. Any peer can query the distributed database with a particular key. The database will then locate the peers that have the corresponding (key, value) pairs and return the key-value to the querying peer. Any peer will also be allowed to insert new key-value pairs into the database. Such a distributed database is referred to as a distributed hash table (DHT). To construct the database, we need to use some hash functions. The input of a hash function can be a large number, but the output of the hash function is of fixed size bit string. The hash function has some collision-free property that it is very difficult to find two different input such that they have the same hash value (output). Examples of such hash functions include MD5, SHA, etc.

The outline of building a DHT is as follows.

- Assign an identifier to each peer, where the identifier is an n-bit string. So we can view the identifier as an integer at the range from 0 to $2^n - 1$.

- For a data pair (key, value), the hash value of the key is computed. We suppose that the hash value is of n-bit (with the same range as the identifiers). Then the data is stored in the peer whose identifier is closest to the key. For convenience, we define the closest peer as the closest successor of the key. For example, if the identifiers for the peers are $1, 3, 4, 5, 8, 10, 12$, and 15, then a data with hashed key value 11 will be stored in the peer with identifier 12. If the hash value equals to some peer's identifier, then the data is stored in that peer. If the hash value is greater than the largest identifier, then the data is stored in the peer with the smallest identifier.

- To insert or retrieve data, first we need to find the appropriate peer. It is not realistic to let the peer to store all of the other peer's identifiers (and the associated IP addresses) or arrange a data center to provide the service (that will damage the P2P architecture). So the DHT uses a circular arrangement. Figure 3.2 can be used to explain the idea.

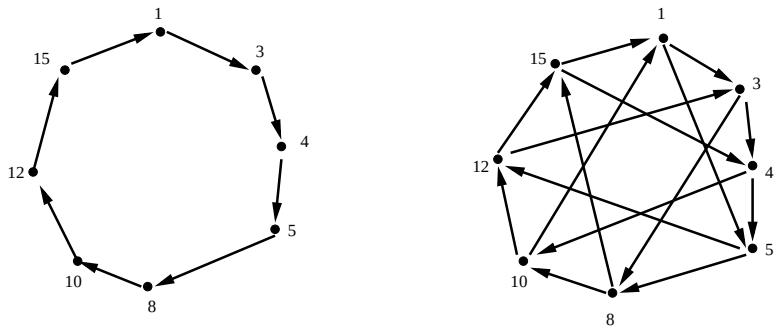

Figure 3.2: Circular DHT

In this example, the identifiers are range $[0, 15]$ (4-bit strings) and there are eight peers. In the circular arrangement, each peer is only aware of its immediate successor and predecessor (as show at the left side of Figure 3.2). So each peer just needs to track two neighbors. When a peer asks for a key, it sends the message clockwise around the circle. For example, the peer 4 sends a message saying "Who is responsible for key 11?". The message will forward through peers 5, 8, 10 and reach peer 12, who determines that it is the closest peer to key 11. At this point, peer 12 can send a message back to the querying peer 4, indicate that it is responsible for key 11. But when the circle is too large, a message may go through a large number of peers to get answer. There is trade off between the number of neighbors each peer has to track and the number of message that the DHT needs to send to resolve a single query. One method can be used to reduce the query time (but increase neighbors) is add "shortcuts" to the circular arrangement. The right side of Figure 3.2 explains the idea. In this example, peer 1 has a shortcut to peer 5, peer 3 has a shortcut to 8, etc. Now when peer 4 asks for the key 11, it will send to peer 10 instead of peer 5. Peer 10 then pass the message to peer 12 and peer 12 will reply the message to peer 4. In general, researchers provide some methods to determine

how to arrange shortcuts for peers. It is proved that the DHT can be designed so that both the number of neighbors per peer as well as the number of messages per query is $O(\log N)$, where N is the number of peers.

- To handle the situation that a peer may join or leave from time to time in a DHT, researchers also proposed various methods. One method for peers' leaving is that a peer keeps track s successors. So when a peer leaves, the other peers still can update their tackers. For example, let $s = 2$ in Figure 3.2. Suppose peer 5 left. Since each peer will periodically verify its two successors (by ping its successors, for example), peers 3 and 4 will know that peer 5 departed. Then peer 4 will replace the immediate successor as peer 8 and ask peer 8 for its successor's identifier (in this case, it is peer 10). Peer 3 will ask peer 4 for its updated immediate successor's identifier and update its second successor (Peer 8).

  Another case should be considered is if some peer will join the DHT. For example, if the peer 13 wants to connect to the DHT. When it joins, it only knows the peer 1. The peer 13 will send a request "what will be peer 13's predecessor and successor?". This message will get forwarded through the DHT until it reaches peer 12. Peer 12 knows that it is 13's predecessor and peer 15 is 13's immediate successor. So peer 12 will inform peer 13 and peer 13 will update its predecessor and successor. It also inform peer 15 and peer 12 to update their connection tracks.

DHT are used in BitTorrent to create a distributed tracker. Large commercial cloud services such as Amazon's Dyname also incorporate DHT techniques.

Chapter 4

TCP/IP and the Internet

We have discussed briefly about the OSI seven layer mode for networks. As we know the Internet is the network of networks. In this chapter, we consider Internet and focus on TCP/IP protocols. The materials will mostly cover protocols in transport layer and network layer, while sometimes the related protocols at other layers will be mentioned. Since TCP/IP is next to application layer, when we develop internet applications, these protocols need to be considered.

4.1 Internet architecture

4.1.1 Internet Addresses

IPv4 (Internet Protocol version 4) assigns to each host a 32-bit integer address called the Internet address, or IP addresses, which is different from a host's physical addresses. An IP address encodes the identification of the network to which a host attaches as well as the the identification of a unique host on that network.

In history, a classful network architecture is used. In this setting, each address is a pair (netid, hostid), where *netid* identifies a network, and *hostid* identifies a host on that network. There are three types of Internet address, class A, class B and class C as shown in Figure 4.1

Class A addresses used for large networks ($2^7 = 128$ networks) have about 2^{24} hosts per network: 7 bits for netid and 24 bits for nostid. Class B addresses used for intermediate-size networks have 14 bits for netid and 16

class	leading (bits)	first octet (decimal)	netid	hostid	# of networks	# of address
A	0	0-127	a.	b.c.d	2^7	2^{24}
B	10	128-191	a.b.	c.d	2^{14}	2^{16}
C	110	192-223	a.b.c.	d	2^{21}	2^8
D	1110	224-239	Multicast address			2^{28}
E	1111	240-255	Reserved			

Figure 4.1: IPv4 addresses

bits for hostid. Class C addresses have only 2^8 hosts for each network, which use 21 bits for netid.

Conventionally, the IP addresses are represented as 4 octets and written as 4 integers for 0 to 255. For example, 11000001 00100000 11011000 00001001 will be displayed as 193.32.216.9, which is a class B address, whose netid is 193.32. And 65.39.14.57 is a class A address. The netid is 65.

From Figure 4.1, we can see that use the classic method of IP address, there are at most $2^7 + 2^{14} + 2^{21}$ networks which is not sufficient for the fast development of Internet.

Later, some classless architecture, called CIDR (Classless Inter-Domain Routing) is defined. In this method, the length of netid is not fixed. For IPv4, the IP address will be a.b.c.d/n, where n is the prefix length (the length of netid in bit from most significant bit of the address), from 0 to 32. In that kind network, the number of addresses are 2^{32-n}. Using this method, we can define more networks (subnets).

4.1.2 Gateway addressing and subnets

The computers that are connected internally to a localized network as well as to an intermediate computer to pass the data to other networks are called Internet gateway or Internet routers. A gateway has at least two physical interfaces, and an IP address is required for each physical interface. An IP address specifies a connection to a network rather than to an individual machine. A machine that has n connection networks will have n IP address. Figure 4.2 displays an example of networks, where two routers are used. Each of the router has three interfaces. In this example, there are five sub networks. The netid for these networks are 222.23.i, where $i = 1, 2, 3, 4, 5$. Note that the two routers form one network. If a computer needs to communicate to a

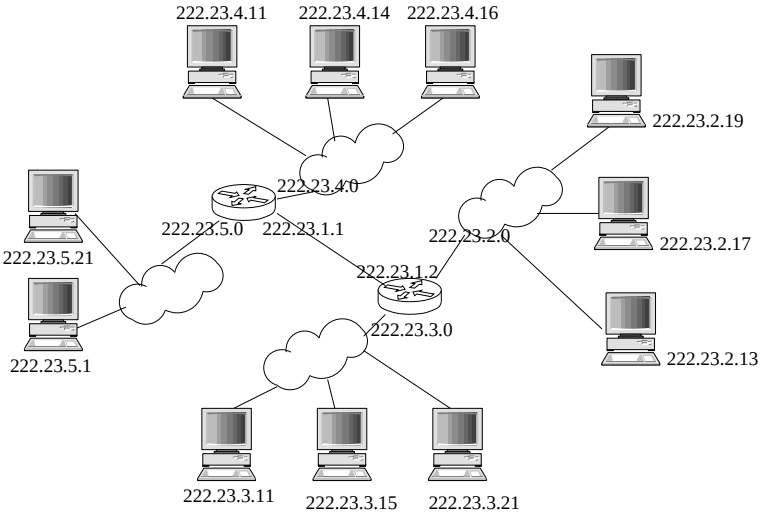

Figure 4.2: An example of networks

computer at other network, then the communication has to go through one or two gateways.

Usually, a network administrator will contact its ISP for a block of addresses from a larger block of addresses, which has been already allocated to the ISP. For example, the ISP has the address block: 200.23.16.0/20 (11001000 00010111 0001|0000 000000000). Then the ISP may assign blocks of addresses to different subnets as follows, where | is used to denote the separation of hostid and netid.

Subnet 1	200.23.16.0/23	11001000000101110001000|0000000000
Subnet 2	200.23.16.0/23	11001000000101110001001|0000000000
Subnet 3	200.23.18.0/23	11001000000101110001010|0000000000
...	...	...
Subnet 7	200.23.30.0/23	11001000000101110001111|0000000000

4.1.3 Subnet addresses

For any Internet class, subnetting is introduced to allocate a part of the host address space to network address and leave the remaining part to other physical networks. This adds flexibility and administrative benefits. If subnetworks are intended as part of the subnetting process, a network-wide netmask should be decided first. The netmask determines which bits in the

IP address space represent the subnetwork address and which bits represent host addresses. The netmask also determines how many subnetworks will be created and how many nodes are included in each subnetwork.

In the netmask file, a 1 at a particular position in the netmask indicates that the particular bit in an IP address ought to be the network address. A 0 indicates that the bit belongs to the host address.

For example, for a class B network with network address 131.31.0.0, the two leftmost octets are assigned to the network address, and the right-most octets are assigned to the host number. So that network can host up to 65,534 computers. However, subnetting allows an entire class B network to be partitioned into 254 subnetworks with up to 254 host computers on each, simply by specifying a netmask 255.255.255.0. This netmask indicates that not only the first two octets, but also the third octet serve as the network address and only the fourth octet is for the host addresses. Note that the binary version of 255.255.255.0 is 11111111.11111111.11111111.00000000. To create only two subnetworks with 32,766 hosts in each, a subnet netmask is 255.255.128.0 (i.e., 11111111.11111111.10000000.00000000). A subnet mask 255.255.192.0 creates 4 subnets.

4.1.4 Network and broadcast addressing

IP addresses with hostid consisting of all 1s are reserved for broadcasting. A broadcast address refers to all the hosts on the network. During start-up, a machine does not know its IP address. Thus until it learns its IP address, it uses the local network broadcast address which is a 32-bit string of 1s (i.e., 255.255.255.255).

Internet addresses can refer to networks as well as to hosts. An IP address with hostid 0 refers to "this" host, and netid 0 means "this" network. Using netid 0 is important when a host wants to communicate over a network but does not yet know the network IP address. The host uses network id 0 temporarily, and other hosts on the network interpret the address as meaning "this" network.

4.1.5 Loopback addressing

The address 127.0.0.0 (looks like a class A address) is reserved for loopback and is designed for testing and interprocess communication on the local machine. If a program uses the loopback address to send data, the protocol

software in the computer returns the data without sending any traffic across the network.

4.1.6 Mapping of physical and IP addresses

In a TCP/IP network, each machine is assigned an IP address and a physical address. The goal of the Address Resolution Protocol (ARP) is to provide low-level software that hides physical addresses and allows higher level program to work with Internet addresses only. Basically, ARP is used to map IP address to a lower layer address. ARP maintains a cache to store recently acquired IP-to-physical address bindings. ARP is not used for crossing networks.

APR is divided into two parts:

- When a host needs to send a packet to a destination host, it looks in the ARP cache to check if the binding between IP address and physical address is available. If the binding is available, the host extracts the physical address from the cache and uses it to send the data. Otherwise it broadcasts a request.

- Whenever an ARP packet arrives from the network, the receiving host extracts the sender's IP address and physical address. It then looks into the local cache to determine whether a binding for the sender IP address exists. If there is such a binding, the host updates the cache entry.

  If the incoming ARP packet is a request, the receiving machine must verify that it is the target of the request. If so, the ARP software forms a reply by supplying its physical hardware address and sends the reply directly back to the requester. The receiver also adds the sender's address pair to its cache if the pair is not present. If the IP address in the ARP request does not match the IP address of the receiver, the request is ignored.

  Another type of incoming ARP packet is the reply for a past request from this receiver. In this case, first the cache is updated for the address binding. Then the receiver tries to match the reply with a previously issued request. Between the time the machine broadcasts its ARP request and receives the reply, application program or higher level

protocol may generate additional requests for the same address. All requests for the same IP address are stored in a queue, and when a reply comes for the IP address, the ARP software removes items from the queue and supplies the address binding to each. If a machine does not issue a request for the IP address in reply, it is ignored.

For example, the computers Matterhorn and Washington are in an office, connected to each other on the office local area network by Ethernet cables and network switches, with no intervening gateways or routers. Matterhorn wants to send a packet to Washington. Through other means, it determines that Washington's IP address is 192.168.0.55. In order to send the message, it also needs to know Washington's MAC address. First, Matterhorn uses a cached ARP table to look up 192.168.0.55 for any existing records of Washington's MAC address (00:eb:24:b2:05:ac). If the MAC address is found, it sends the IP packet on the link layer to address 00:eb:24:b2:05:ac via the local network cabling. If the cache did not produce a result for 192.168.0.55, Matterhorn has to send a broadcast ARP message (destination FF:FF:FF:FF:FF:FF) requesting an answer for 192.168.0.55. Washington responds with its MAC address (00:eb:24:b2:05:ac). Washington may insert an entry for Matterhorn into its own ARP table for future use. The response information is cached in Matterhorn's ARP table and the message can now be sent.

4.1.7 Reverse address resolution protocol (RARP)

The IP address of a machine is usually read from its configuration file stored on a disk file. However, a diskless machine is usually booted from ROM, which has minimum booting information. The ROM is installed by the manufacturer. It cannot include the IP address because the IP addresses on a network are assigned by the network administrator. The machine can get its physical address (by reading its NIC, network interface card, for example), which is unique locally. It can then use the physical address to get the logical address by using the RARP protocol. A RARP request is created and broadcast on the local network. Another machine on the local network that knows all the IP addresses will respond with a RARP reply. The requesting machine must be running a RARP client program; the responding machine must be running a RARP server program. RARP has been rendered obsolete by the Bootstrap Protocol (BOOTP) and the modern Dynamic Host

Configuration Protocol (DHCP).

4.1.8 Dynamic Host Configuration Protocol (DHCP)

DHCP is a network protocol (RFC standard) that is used to configure network devices so that they can communicate on an IP network. A DHCP client uses the DHCP protocol to acquire configuration information, such as an IP address, a default route and one or more DNS (Domain Name System) server addresses from a DHCP server. The DHCP client then uses this information to configure its host. Once the configuration process is complete, the host is able to communicate on that network.

The DHCP server maintains a database of available IP addresses and configuration information. When it receives a broadcast query from a client, the DHCP server determines the network to which the DHCP client is connected, and then allocates an IP address or prefix that is appropriate for the client, and sends to client. DHCP servers typically grant IP addresses to clients only for a limited interval. DHCP clients are responsible for renewing their IP address before that interval has expired, and must stop using the address once the interval has expired, if they have not been able to renew it. The query is typically initiated immediately after booting, and must complete before the client can initiate IP-based communication with other hosts. Upon disconnecting, the IP address is returned to the pool for use by another computer. This way, many other computers can use the same IP address within minutes of each other.

The DHCP server may have three methods of allocating IP-addresses:

- dynamic allocation: a network administrator assigns a range of IP addresses to DHCP, and each client computer on the LAN is configured to request an IP address from the DHCP server during network initialization. A lease concept with a controllable time period, allows the DHCP server to reclaim (and then reallocate) IP addresses that are not renewed.

- automatic allocation: The DHCP server permanently assigns a free IP address to a requesting client from the range defined by the administrator and keeps a table of past IP address assignments, so that it can preferentially assign to a client the same IP address that the client previously had.

- static allocation: The DHCP server allocates an IP address based on a table with MAC address/IP address pairs, which are manually filled in. Only clients with a MAC address listed in this table will be allocated an IP address. This feature is not supported by all DHCP servers.

The DHCP server and DHCP client must be connected to the same network link. In larger networks, this is not practical. In this case, one or more DHCP relay agents (usually routers) are used. These DHCP relay agents receive messages from DHCP clients and forward them to DHCP servers. DHCP servers send responses back to the relay agent, and the relay agent then sends these responses to the DHCP client on the local network link.

DHCP was first defined as a standards track protocol in RFC 1531 in October 1993, as an extension to the Bootstrap Protocol (BOOTP). DHCP is very useful for a network, which laptops and other mobile devices may randomly access.

4.1.9 Network address translation (NAT)

NAT (Network address translation) was proposed to slow down the speed of running out of IP address in 2001.

The basic idea behind NAT is to assign each company a single IP address or a small number of IP addresses for Internet traffic. Within the company, every computer gets a unique IP address, which is used for routing intramural traffic. However, when a packet exits the company, an address translation takes place. Three ranges of IP addresses have been declared as private. No packets containing these address may appear on the Internet. The three ranges are:

10.0.0.0	-	10.255.255.255/8	(16,777,216 hosts)
172.16.0.0	-	172.31.255.255/12	(1,048,576 hosts)
192.168.0.0	-	192.168.255.255/16	(65,536 hosts)

When a packet leaves the company premises, it passes through a NAT box that converts the internal IP source address in to the company's true IP address. The NAT box is often combined in a single device with a firewall or integrated into the company's router.

To allow the reply packet reach the internal host, the NAT box need to remember the host's internal IP address with a TCP source port number.

That means when the packet go outside, not only the IP address is replaced, the TCP port number is replace as well.

Home network is typically using a NAT-enabled router. Addressing within the home network is 10.0.0.0/24. Suppose the IP address assigned by the ISP is 138.76.29.17. The NAT translation table could be as in Figure 4.3.

NAT translation table	
WAN side	LAN side
138.76.28.17,5001	10.0.0.1,2334
138.76.28.17,4009	10.0.0.2,1344
138.76.28.17,2109	10.0.0.1,1344
...	...

Figure 4.3: NAT table

For a outgoing packet, the source IP address 10.0.0.1 with port number 2334 will be changed to 138.76.28.17 with port number 5001. For a coming packet, the destination IP address 138.76.28.17 with port number 5001 will be changed to 10.0.0.1 with port number 2334. In this way, several devices at home can share one IP address.

There are criticisms about NAT. Because it violates the architectural model of IP (mixed up two layers). If the NAT box crashes and its mapping table is lost, all its TCP connection are destroyed. Another problem is that a server behind an NAT should have a fixed IP address so that a client can contact it. But in P2P usually a peer behind an NAT will not have a fixed IP address. So another peer outside the NAT cannot initiate a TCP connection with it. To solve that problem for P2P application, some method called connection reversal is used. Suppose peer A is outside an NAT and B is inside of the NAT. If A wants to initiates a TCP connection with B, it first find some peer C which is also outside that NAT and C has a TCP connection with B. By the help of C, A then find the way to establish a TCP connection with B.

NAT traversal is increasingly provided by Universal Plug and Play (UPnP), which is a protocol that allows a host to discover and configure a near by NAT. UPnP requires that both the host and the NAT be UPnP compatible. With UPnP, an application running in a host can request a NAT mapping between its (private IP , port) and (public IP, port) for some requested public

port number. If the NAT accepts the request and creates the mapping, then nodes from the outside can initiate TCP connections to the public IP and the port. Furthermore, UPnP lets the application know the value of (public IP, port), so that the application can advertise it to the outside world. For example, a BitTorrent peer inside a NAT can ask the NAT to create a "hole" that maps (10.0.0.1, 3345) to (138.76.28.17, 5001) and advertise to its tracker that it is available at (138.76.28.17, 5001). The external peer then may send a TCP SYN packet to (138.76.28.17, 5001). When NAT receives the SYN packet, it will change the destination IP address and the port number in the packet to (10.0.0.1, 3345) and forward the packet through the NAT.

4.2 Internet protocol and datagrams

4.2.1 IP datagram format

The IP header of IPv4 is shown in Figure 4.4

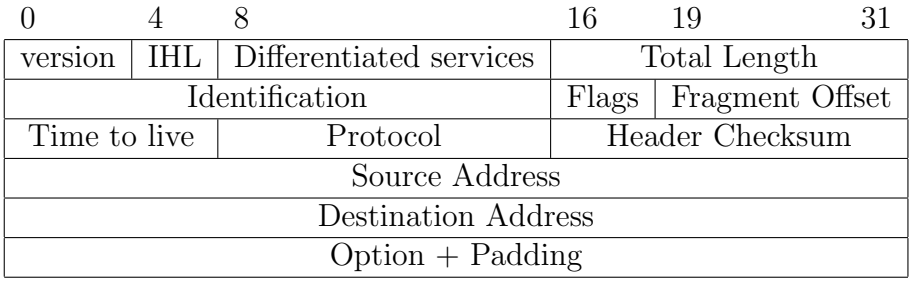

Figure 4.4: IPv4 header

The size of the IPv4 header is a minimum of 20 octets, or 160 bits. The items in IPv4 header are as follows.

- Version (4 bits): The version of IP that was used to create the datagram. Machines reject the datagrams with protocol versions that differ from theirs.

- Internet Header Length (IHL) (4 bits): Length of header in 32-bit words. The minimum value is 5.

- Differentiated services (8 bits): Previous called Type of Services (TOS). It provides guidance to end IP modules and to routers along the packet's path about the packet's relative priority. The Type of service field is broken into five subfields as follows:

Precedence	D	T	R	Unused

  The first three bits (0-7) indicate datagram precedence, allowing the sender to indicate the importance of each datagram,. Bit sets D, T and R specify that the type of transport: D for low delay, T for high throughput and R for high reliability. Since the name changed to Differentiated service (RFC 2474), now the top 6 bits are used to mark the packet with its service class (DSCP). The last 2 bits are used to carry explicit congestion notification information (ECN), such as whether the packet has experienced congestion. The main reason for changing this field is that many different types of services, including voice, video, streaming music, etc, have some different requirements for forwarding. We will see a little details later when we look at multimedia networks.

- Total length (16 bits): Total IP packet length, in octets. This includes both header and data. Since the field is only 16 bits long, the maximum size of an IP datagram is 64KB.

- Identification (16 bits): A unique integer identifies the datagram, created from the source address, destination address and user protocol. Retransmissions of IP datagram contains the same identification number. All the fragments of a packet contains the same identification number.

- Flags (3 bits): Indicates whether it is the last fragment of the original datagram. First bit is reserved (unused). Second bit signals DF (Don't fragment) and third bit signals MF (More fragments).

- Fragment Offset (13 bits): Indicate where in the original datagram this fragment belongs, measured in 64-bit units (sequence number of fragments).

- Time to live (8 bits): Specifies how long (in seconds) a packet is allowed to remain in the Internet. Gateways and hosts that process datagram must decrement the TTL field as time passes and remove from the network when time has expired. In practice, the field has become a hop

count, when the datagram arrives at a router, the router decrements the TTL field by one. When the TTL field hits zero, the router discards the packet and typically sends an ICMP Time Exceeded message to the sender.

- Protocol (8 bits): Indicates the next higher level protocol.
- Header Checksum (16 bits): An error-detecting code (for the header only. The data will be checked at transport layer). Since some header fields may change during transit, this is reverified and recomputed at each router. If the checksum is wrong, the router will discard the fragment.
- Source Address (32 bits): Coded to allow a variable allocation of bits to specify the network and the end system attached to the specified network.
- Destination Address (32 bits): Same characteristics as source address.
- Options (variable): Encodes the options requested by the sending user, such as security label, source routing, record routing, and timestamp. The list of options may be terminated with an EOL (End of Options List, 0x00) option.
- Padding (variable): Used to ensure that the packet header is a multiple of 32 bits in length.

The underlying physical network transports datagram. Each datagram travels in a distinct physical frame, for efficient Internet transportation. Ideally the entire IP datagram should fit into one physical frame. A field in the frame header identifies the data being carried. However, different physical networks allow a different size date. For example, proNET-10 allows 2044 bytes per frame, and 2044 bytes is called this network's maximum transfer unit (MTU). Some MTU size can be quite small ($\leq$ 128 bytes). Since a datagram may travel across many types of physical network, IP should select a maximum datagram size to ensure that each datagram will always fit into one frame. Limiting datagram to fit the smallest possible MTU in the Internet makes transfers inefficient. On the other hand, allowing a datagram to be larger than a network MTU means that some datagram will not fit into a single network frame.

TCP/IP software chooses a convenient initial datagram size and arranges a way to divide large datagrams into smaller piece when they need to travel over a network with smaller MTU. The smaller pieces are called fragments, the process of dividing datagrams into smaller pieces is called fragmentation. Fragmentation occurs at a gateway somewhere along the path between the datagram source and its ultimate destination.

Each fragment contains a datagram header that duplicates most of the original datagram header, except for a bit in the flag field, followed by as much data as can be carried in the fragment of a limited MTU. Fragments must be reassembled to produce a complete copy of the datagram before it can be processed at the destination.

4.2.2 Internet control message protocol (ICMP)

To allow gateways in the Internet to report errors or provide information about unexpected circumstances, ICMP (RFC 792) protocols are added to the TCP/IP family. ICMP is often considered part of IP but architecturally it lies above IP, as ICMP messages are carried inside IP datagram as the IP payload.

About a dozen types of ICMP messages are defined. The most important ones are listed in Table (4.1).

Type	Code	Description
0	0	Echo reply (used to ping)
3	0	Destination network unreachable
3	1	Destination host unreachable
3	2	Destination protocol unreachable
3	3	Destination port unreachable
3	6	Destination network unknown
5	0	Redirect Datagram for the Network
8	0	Echo request (used to ping)
10	0	Router discovery/selection/solicitation
11	0	TTL expired in transit
13	0	Timestamp
14	0	Timestamp reply

Table 4.1: ICMP types

The format of ICMP message is as follows.

0	8	15	31	
type	code	checksum	optional	IP header and first 64 bits of datagram

ICMP is not restricted to gateways. Any machine can send an ICMP message to another machine. Thus a way exists to report errors to the original source. Most errors are from the original source, but some do not. However, the datagram only contains field that specify the original source and the ultimate destination and gateways can establish and change their own routing table. So it cannot know the set of intermediate machines that processed the datagram. So basically, ICMP reports to the original source.

One example of applications of ICMP is the Traceroute program. To determine the names and addresses of the routers between source and destination, Traceroute in the source sends a series of ordinary IP datagrams to the destination. Each of these datagrams carries a UDP segment with an unlikely UDP port number. The first of these datagrams has a TTL 1, the second of 2, the third of 3, and so on. The source also starts timers for each of the datagrams. When the nth datagram arrives at the nth router, the router observes that the TTL of the datagram has just expired. So the router will discard the datagram and sends an ICMP warning message (type 11, code 0) to the sender. This warning message includes the router's IP address. When the final daragram arrived the traced destination, the destination host will send back a unreachable ICMP message (type 3, code 3) because the port number is unlikely UDP port.

4.2.3 IPv6

IPv6 was developed by the Internet Engineering Task Force (IETF) to deal with the long-anticipated problem of IPv4 address exhaustion. On February 3, 2011, the Internet Assigned Numbers Authority (IANA) assigned the last batch of five/8 address blocks to the Regional Internet Registries.

IPv6 uses a 128-bit address, allowing for 2^{128}, or approximately 3.4×10^{38} addresses, or more than 7.9×10^{28} times as many as IPv4, which uses 32-bit addresses.

IPv6 addresses consist of eight groups of four hexadecimal digits separated by colons, for example 2001:0db8:85a3:0042:1000:8a2e:0370:7334. The hexadecimal digits are not case-sensitive; e.g., the groups 0DB8 and 0db8 are equivalent.

An IPv6 address may be abbreviated by using one or more of the following rules:

1. Remove one or more leading zeros from one or more groups of hexadecimal digits(For example, convert the group 0042 to 42.)

2. Omit one or more consecutive sections of zeros, using a double colon (::) to denote the omitted sections. The double colon may only be used once in any given address, as the address would be indeterminate if the double colon was used multiple times. (For example, 2001:db8::1:2 is valid, but 2001:db8::1::2 is not permitted.)

Hybrid dual-stack IPv6/IPv4 implementations recognize a special class of addresses, the IPv4-mapped IPv6 addresses. In these addresses, the first 80 bits are zero, the next 16 bits are one, and the remaining 32 bits are the IPv4 address. The first 96 bits written in the standard IPv6 format, and the remaining 32 bits written in the customary dot-decimal notation of IPv4. For example, ::ffff:192.0.2.128 represents the IPv4 address 192.0.2.128. A deprecated format for IPv4-compatible IPv6 addresses is ::192.0.2.128.

An IPv6 packet has two parts: a header and payload. The header consists of a fixed portion with minimal functionality required for all packets and may be followed by optional extensions to implement special features. The fixed header occupies the first 40 octets (320 bits) of the IPv6 packet. It contains the source and destination addresses, traffic classification options, a hop counter, and the type of the optional extension or payload which follows the header. This Next Header field tells the receiver how to interpret the data which follows the header. If the packet contains options, this field contains the option type of the next option. The "Next Header" field of the last option, points to the upper-layer protocol that is carried in the packet's payload. Extension headers carry options that are used for special treatment of a packet in the network, e.g., for routing, fragmentation, and for security using the IPsec framework. The format of the header is as in Figure 4.5.

The fields of IPv6 are as follows:

- Version (4 bits) The constant 6 (bit sequence 0110).

- Traffic Class (8 bits) The bits of this field hold two values. The 6 most-significant bits are used for DSCP, which is used to classify packets. The remaining two bits are used for ECN; priority values subdivide

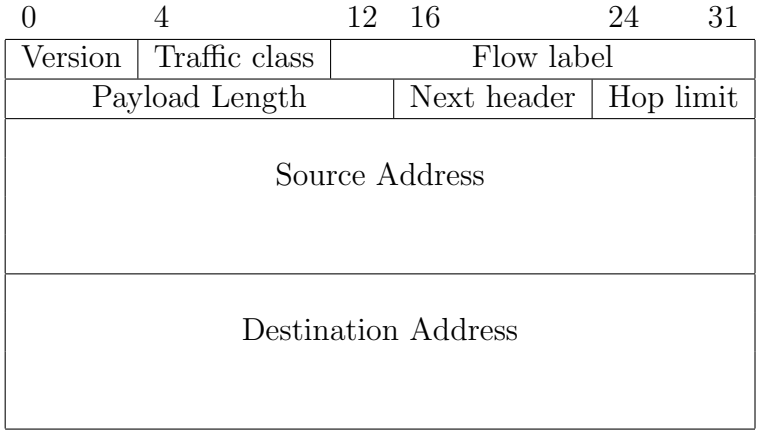

Figure 4.5: IPv6 header

into ranges: traffic where the source provides congestion control and non-congestion control traffic.

- Flow Label (20 bits) Originally created for giving real-time applications special service. The flow label when set to a non-zero value now serves as a hint to routers and switches with multiple outbound paths that these packets should stay on the same path so that they will not be reordered. It has further been suggested that the flow label be used to help detect spoofed packets.

- Payload Length (16 bits) The size of the payload in octets, including any extension headers. The length is set to zero when a Hop-by-Hop extension header carries a Jumbo Payload option.

- Next Header (8 bits) Specifies the type of the next header. This field usually specifies the transport layer protocol used by a packet's payload. When extension headers are present in the packet this field indicates which extension header follows. The values are shared with those used for the IPv4 protocol field, as both fields have the same function.

- Hop Limit (8 bits) Replaces the time to live field of IPv4. This value is decremented by one at each intermediate node visited by the packet. When the counter reaches 0 the packet is discarded.

- Source Address (128 bits) The IPv6 address of the sending node.

- Destination Address (128 bits) The IPv6 address of the destination node(s).

In order to increase performance, and since current link layer technology is assumed to provide sufficient error detection, the header has no checksum to protect it.

In IPv6, the packet header and the process of packet forwarding have been simplified. Although IPv6 packet headers are at least twice the size of IPv4 packet headers, packet processing by routers is generally more efficient.

- The packet header in IPv6 is simpler than that used in IPv4, with many rarely used fields moved to separate optional header extensions.
- IPv6 routers do not perform fragmentation. IPv6 hosts are required to either perform path MTU discovery, perform end-to-end fragmentation, or to send packets no larger than the IPv6 default minimum MTU size of 1280 octets. If a IPv6 datagram received by a router is too large to be forwarded, the router simply drops the datagram and send a "Packet Too Big" ICMP error message to the sender.
- The IPv6 header is not protected by a checksum; integrity protection is assumed to be assured by both link-layer and higher-layer (TCP, UDP, etc.) error detection. UDP/IPv4 may actually have a checksum of 0, indicating no checksum; IPv6 requires UDP to have its own checksum. Therefore, IPv6 routers do not need to recompute a checksum when header fields (such as the time to live (TTL) or hop count) change.
- The TTL field of IPv4 has been renamed to Hop Limit, reflecting the fact that routers are no longer expected to compute the time a packet has spent in a queue.
- The IPv6 defined subnet *anycast addresses* (RFC 2526) , that is assigned to one or more network interfaces (typically belonging to different nodes), with the property that a packet sent to an anycast address is routed to the "nearest" interface having that address, according to the routing protocols' measure of distance.

The IPv6 specification currently defines 6 Extension Headers:

- Routing Header - Similar to the source routing options in IPv4. Used to mandate a specific routing.

- Authentication Header (AH) - A security header which provides authentication and integrity.

- Encapsulating Security Payload (ESP) Header - A security header which provides authentication and encryption.

- Fragmentation Header - The Fragmentation Header is similar to the fragmentation options in IPv4.

- Destination Options Header - This header contains a set of options to be processed only by the final destination node. Mobile IPv6 is an example of a Destination Options Header.

- Hop-by-Hop Options Header - A set of options needed by routers to perform certain management or debugging functions.

A new version of ICMP has been defined for IPv6 (RFC 4443). ICMPv6 added new types and codes required by the new IPv6 functionality. These include the "Packet TooBig" type and an "unrecognized IPv6 options" error codes. It also includes some group management protocol.

After the IPv6 has been used, the IPv4 are still using. Some methods are used to handle that situation. Dual-stack (or native dual-stack) refers to side-by-side implementation of IPv4 and IPv6. That is, both protocols run on the same network infrastructure, and there's no need to encapsulate IPv6 inside IPv4 or vice-versa (using tunneling). Dual-stack is defined in RFC 4213. Although this is the most desirable IPv6 implementation, it is not always possible, since outdated network equipment may not support IPv6. Some network equipment (such as a CMTS) or customer equipment (like cable modems) may require software updates or hardware upgrades to support IPv6. This means cable network operators must resort to "tunneling" until the backbone equipment supports native dual-stack. The basic idea of tunneling is that suppose an IPv6 datagram has to go through a router that only support IPv4, then the whole datagram is put into the data field of an IPv4 datagram. The source address of the IPv4 datagram will be the last IPv6 supported router and the destination address is the next IPv6 supported router. Between these two routers, a IPv4 tunnel is formed.

4.3 Routing protocols

4.3.1 Routing Tables and routing mechanisms

A routing table has columns for the destination network (or nods) with the corresponding cost and the next router address to reach a destination. Additional information may vary depending on the routing protocol being used.

Distance vector (DV) routing

In the distance vector-based routing algorithm, (Bellman-Ford algorithm), routers pass periodic copies of a routing table to other routers to indicate the changes in the topology. Each router receives a routing table from its direct neighbor. The router adds a distance vector number (such as number of hops), increasing the distance vector then passes the table to next neighbor. The same process occurs in all directions between direct-neighbors. In this way, the algorithm accumulates network distances and is able to maintain a database of network topology information.

One basic idea for DV algorithm is as follows. For two hosts v and y, let $d(v, y)$ denote the least cost path from v to y. For a host x and its neighbor v the cost from x to v is $c(x, v)$. Then

$$d(x, y) = \min\{c(x, v) + d(v, y) : v \text{ is a neighbor of } x\}. \tag{4.1}$$

Suppose N is a network. The DV Algorithm for N is as follows. At each node x:
Initialization:

```
for all destination y in N:
 D(x, y) = c(x, y)
  // if y is not a neighbor, c(x,y) = infinite.
for each neighbor w
 D(w, y) = ?
for each neighbor w
 send vector D(x) = [D(x, y) : y in N] to w
```

loop

```
wait (until a link cost change to some neighbor w,
       or received a distance vector from neighbor w)
```

```
for each y in N:
D(x, y) = min{c(x, v) + D(v, y): v is a neighbor of x}
if D(x,y) changed for any destination y
 send distance vector D(x) = [D(x, y): y in N] to all neighbors
```

To understand the algorithm, let us look at a simple example of network in Figure 4.6. In this network, there are 4 nodes and the weights (cost) for the links are different.

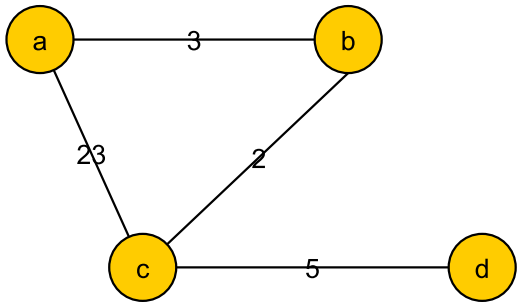

Figure 4.6: An example for DV algorithm

After initialization, the vectors in each node are as in Table 4.2.

D(a)	(a,b):3	(a,c):23	(a,d):∞	(b,c):?	(b,d):?	(c,d):?
D(b)	(a,b):3	(b,c):2	(b,d):∞	(a,d):?	(c,d):?	(a,c):?
D(c)	(a,c):23	(b,c):2	(c,d):5	(a,b):?	(b,d):?	(a,d):?
D(d)	(c,d):5	(b,d):∞	(a,d):∞	(c,d):?	(a,b):?	(c,b):?

Table 4.2: Initialized vectors

After node a received the vectors from its neighbors b and c, node a will update its vector as follows. The vector D(b) tells that D(b, c) = 2 and the vector D(c) tells that D(c, d) = 5. So node a can compute an update as:

D(a)	(a,b):3	(a,c):5	(a,d):10	(b,c):2	(b,d):?	(c,d):5

Each node will update their vector at the same time. If their vectors changed, then they will send the vector to the neighbors again. Therefore the node a will update the vector again until no changes are received.

The DV routing algorithm is decentralized algorithm. In this algorithm, each node does not need to know the topology of the network. All the updates are performed by communicating to the neighbors. It can be proved that the distance vectors will be convergent to the least costs.

After the router has a complete vector, then it will be easy to find which way to forward a packet by using the equation (4.1). In fact, the equation tells which neighbor v should the node forward to.

The above discussion assumes that the cost for each link is stable. If the cost changes, then the above algorithm will have problems. If some cost is reduced, then the above algorithm can easily get the updated vectors. However, if some link cost increases, then the above algorithm will take a lot of loops to update. Some technique called poisoned reverse has been used to solve this problem. We omitted the details here.

Link-state (LS) routing

This routing algorithm is also know as shortest-path-first (SPF) algorithm. A link state routing algorithm maintains full knowledge of distant routers and how they interconnect.

Link sate routing uses link state advertisements(LSAs), a topological data base, the SPF algorithm, the resulting SFP tree, and finally, a routing table of paths and ports to each network.

Each router does the following

- Keeps track of its neighbors: the neighbor's name, it is up or down, the the cost of the link to the neighbor.
- Construct an LSA packet that lists its neighbor router names and link costs. This includes new neighbors, changes in link costs, and links to neighbors that have gone down.
- Send out this LSA packet so that all other routers receive it.
- Records each LSA packet it receives in its database immediately, to ensure that it has the most recently generated LSA packet from every other router.
- Using accumulated LSA packet data to construct a complete map of the internetwork topology, proceeds from this common starting point

to return the SPF algorithm and compute routes to every network destination.

The routing table updates in the link state routing method use the following process.

- Routers exchange LSAs with each other, Each router begins with directly connected networks for which it has direct information.
- Each router, in parallel with the others, constructs a topological database consisting of all the LSAs from the internetwork.
- The SPF algorithm computes network reachability, determining the shortest path first to each other network in the link state protocol internetwork. Each router constructs this logical topology of shortest paths as SPF tree. With itself as root, this tree expresses paths from the router to all destinations.
- Each router lists its best paths and the ports to these destination networks in the routing table. It also maintains other database of topology elements and status details.

With all the links and costs in the table, the router views the network as a graph and uses Dijkstra's algorithm as SPF algorithm to find out the shortest path. LS is basically a centralized algorithm, in which the node has the topology picture of the whole network.

Routing policies

There are mainly two policies to form the routing tables, manual and automatic. In the manual policy, the system administrator sets a router's table at start-up. This is called static routing. When the router receives a packet with a particular destination address, it looks at the table for the next hop. If the address is not found in the table, the router forwards the packet to a default router connected to it.

In the other case, the routers accept information from other router to periodically update their entries. This type of routing policy is called dynamic routing.

Static router are explicitly configured and entered into the routing table. When the router uses a combination of static and dynamic policies, static routes take precedence.

A static route in the network reflects some special knowledge of the networking situation known to the network administrator. Routing updates are not sent on a link if defined only by a static route, thereby conserving bandwidth.

4.3.2 LAN-to-LAN and LAN-to-WAN routing

Routers seamlessly handling packets encapsulated into different lower level frames without changing the packet's layer 3 addressing. For example, a packet traffic from source host 4 on Ethernet needs a path to destination host 5 on network 2. The LAN hosts depend on the router and its consistent network addressing to find the best path. When the router checks its router table entries, it discovers that the best path to destination uses outgoing port To 0, the interface to a token ring LAN.

Although the lower layer framing must change as the router switches packet traffic from the Ethernet on network 1 to the token ring on network 2, the layer 3 addressing for source and destination remains the same.

The network layer must interface with various lower layers for LAN-to-WAN traffic. The path taken by a packet might encounter several relay points and a variety of data link types beyond the LANs. For example, a packet may be encapsulated in a token ring frame by a router, and then another router needs to remove the packet from the token ring frame and encapsulates it in a frame of frame relay, or uses Ethernet frame, etc. The routers enable LAN-to-WAN packet flow by keeping the end-to-end source and destination addresses constant while encapsulating the packet at the port to a data link that is appropriate for the next hop along the path.

One of the main problem for internet routing is the size of the internet. The number of hosts in the internet is huge. Therefore it is not possible for a router to have a table for all nodes on the internet. For example, we cannot use LS routing to find the path for every node in the internet because we cannot form a graph which includes all nodes in the internet.

In internet routers are organized into *autonomous systems* (ASs). Each AS consists of a group of routers that are typically under the same administrative control and running the same routing algorithm. The routing algorithm running within an autonomous system is called an intra-autonomous system routing protocol. There should be routers connect ASs to each other. So some routers in an AS should be responsible for forwarding packets to destinations outside the AS. These routers are usually called gateway routers.

If a AS has only one gateway router, then the routing is simple because any packet not for the hosts inside the AS must go to that gateway router. How ever, if an AS has two or more gateway routers, then we need some methods to decide which gateway router should be used for specific outside hosts. In this case, the gateway routers need to obtain reachability information from neighboring ASs and propagating the reachability information to all routers internal to the AS. That kind protocol is called inter-AS routing protocol. In Internet, all ASs run the same inter-AS routing protocol, called BGP4.

4.3.3 Intra-AS routing

Now let us look at two Intra-AS routing protocols used in the Internet.

RIP

Routing Information Protocol (RIP) is specified in RFC 1058. RIP is very close to the DV protocol we examined in previous sections, but it uses hop count as a cost metric. In RIP, costs are actually from source router to a destination subnet. RIP uses the term hop, which is the number of subnets traversed along the shortest path from the source router to destination subnet, including the destination subnet. The maximum cost is limited to 15, thus limiting the use of RIP to AS that are fewer than 15 hops in diameter. In RIP, routing updates are exchanged between neighbors approximately every 30 seconds using a RIP response message (or RIP advertisement). The response message contains a list of up to 25 destination subnets within the AS, as well as the sender's distance to each of those subnets. If a router does not hear from its neighbor at least every 180 seconds, that neighbor is considered to be no longer reachable. When this happens, RIP modifies the local routing table and then propagates this information by sending advertisements to its neighboring routers. A router can also request information about its neighbor's cost to a given destination using RIP's request message. Routers send RIP request and response messages to each other over UDP using port number 520.

OSPF

Open Shortest Path First (OSPF): is defined in RFC 2328. It is based on link-state information and a Dijkstra's least-cost path algorithm. With OSPF,

a router constructs a complete topological map of the entire AS. Then the router uses the Dijkstra algorithm to determine the shortest paths tree with itself as the root node. Individual link costs are configured by the network administrator. With OSPF, a router broadcasts routing information to all other routers in the AS. A router broadcasts link-state information whenever there is a change in a link's state. It also broadcasts a link's state periodically, even if the link's state has not changed. OSPF advertisements are contained in OSPF messages that are carried directly by IP. The OSPF protocol also checks that links are operational (via HELLO message that is sent to an attached neighbor) and allows an OSPF router to obtain a neighboring router's database of network-wide link state. An OSPF autonomous system can be configured hierarchically into areas. Each area runs its won OSPF routing algorithm, with each router in an area broadcasts its link stake to all other routers in the area. Within the area, one or more area border routers are responsible for routing packets outside the area. One OSPF area in the AS is configured to be the backbone. The main function of the backbone area is to route traffic between the other areas in the AS. The backbone area consists of all the border routers in the AS and may contain some other routers. OSPF also defines some advanced features. It considered the security of the routing, so passwords among the routers are used.

4.3.4 Inter-AS routing

Internet uses the Border Gateway Protocol (BGP) (RFC4271, 4274) for inter-AS routing. BGP is one of the most important protocol for Internet. Without that protocol, the ASs cannot be glue together and the Internet cannot be formed. On the other hand, BGP is a very complicated protocol and there are entire books for BGP.

In BGP, pairs of routers exchange routing information over semi-permanent TCP connections using port 179. There is typically one BGP TCP connection for each link that directly connects two routers in two different ASs. There are also semi-permanent BGP TCP connections between routers within an AS. A common configuration is one TCP connection for each pair of routers internal to an AS. For each TCP connection, the two routers at the end of the connection are called BGP peers, and the TCP connection along with all the BGP messages sent over the connection is called a BGP session. A BGP session that spans two ASs is called an external BGP (eBGP) session and a BGP session between routers in the same AS is called an internal BGP

(iBGP) session.

BGP allows each AS to learn which destinations are reachable via its neighboring ASs. In BGP, destinations are not hosts but instead are netids (prefix). We will use Figure 4.7 as an example to explain the BGP session.

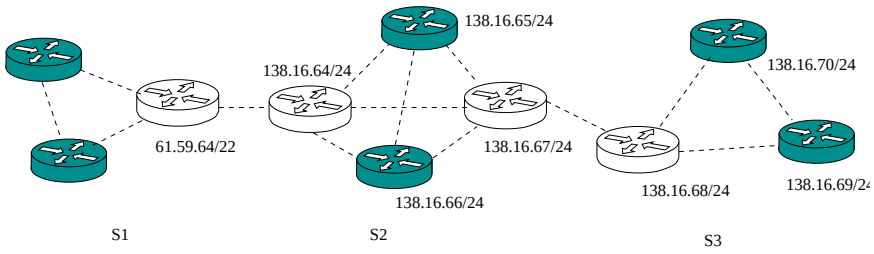

Figure 4.7: BGP sessions

In this example, there are 3 ASs, each of AS contains several routers. The dashed lines in the picture are not physical links, but BGP TCP semi-permanent connections. The white routers are BGP peers. There are four subsets attached to AS2: 138.16.64/24, 138.16.65/24, 138.16.66/24, and 138.16.67/24. AS2 will aggregate the netid for these four subnets and use BGP to advertise the single netid to 138.16.64/22. Similarly, the AS3 can advertise the netid 138.16.68/22. In this case, if the subnet 138.16.71/24 is at some AS other than AS3, it can still advertise 138.16.71/24, because routers use longest-prefix matching for forwarding datagrams. Now eBGP sessions between connected BGP peers will send reachable netid information each other. On the other hand, when a BGP peer receives eBGP-learned prefixes, the peer router uses its iBGP sessions to distribute the prefixes to the other routers in the AS. When a router learns about a new prefix, it creates an entry for the prefix in its forwarding table.

In BGP, an AS is identified by its globally unique autonomous system number (ASN) (RFC 1930). ASNs, like IP addresses, are assigned by ICANN (Intenet Corporation for Assigned Names and Numbers) regional registries. When a router advertises a prefix across a BGP session, it includes with the prefix a number of BGP attributes. These information are called a route. Two of the more important attributes are AS-PATH and NEXT-HOP.

- The attribute AS-PATH contains the ASs through which the advertisement for the prefix has passed. When a prefix is passed into an AS, the AS adds its ASN to the AS-PATH. For example, if AS3 advertises

its prefix to AS2 and then AS2 advertises the prefix to AS1, then the AS-PATH will be AS3 AS2. Routers use the AS-PATH attribute to detect and prevent looping advertisements. If a router sees that its AS is contained in the path list, then it will reject the advertisement. Routers also use the AS-PATH attribute in choosing among multiple paths to the same prefix.

- The NEXT-HOP is the IP address of the router interface that begins the AS-PATH. For example, when the peer in AS1 advertises some AS-PATH to AS2 using eBGP, it contains the IP address of its interface as the NEXT-HOP attribute. After routers inside AS2 learned the advertisement (route) by iBGP, it can use that IP address as forwarding address if it wants to use that path. So the router can put it into its forwarding table.

- It is possible that two ASs are connected by two peering links. The two routers could have the same AS-PATH but different NEXT-HOP values. In this situation, the router can use the NEXT-HOP values and the intra-AS routing algorithm to determine the cost of the path to each peering link and determine the appropriate interface to use.

By eBGP and iBGP, a router may learn about more than one route to a prefix. In this case, the route must select one of the possible route. The input of the route selection processor is the set of all routes that have been learned and accepted by the router (the router usually has its import policy to decide whether accept or filter the route). If there are two or more routes to the same prefix, the BGP will use elimination rules outline as follows:

- Routes are assigned a local preference value as one of their attributes. The local preference of a route could be set by the router or by another router in the same AS. This is a policy decision that is left up to the AS's network administrator. The routes with the highest local preference values are selected.

- From the remaining routes (all with the same local preference value), the route with the shortest AS-PATH is selected. If this rule were the only rule for route selection, then BGP would be using a DV algorithm for path determination, where the distance metric uses the number of AS hops rather than the number of router hops.

- From the remaining routers (all with the same local preference values and the same AS-PATH length), the route with the closest NEXT-HOP router is selected.

- If more than one route still remains, the router uses BGP identifiers to select the router.

We omitted most details about BGP and just gave some conceptual description. These information just outline the main idea of BGP. To fully understand or really use BGP, some detailed book should be read.

4.3.5 Broadcast and multicast routing

Broadcasting information for a host in the network means sending the information to every host on the network. It is not realistic to use the unicast (point-to-point) routing methods discussed previously for the broadcast routing. This is not just because using unicast routing for broadcast is not efficient, but also because a host usually does not know addresses of all the hosts on the network before broadcasting.

There are several methods used for broadcast routing. In what follows, we will view the network as a graph $G = (N, E)$, where the vertices N contains all the hosts on the network and the edges contains all the links of the network.

Controlled flooding

To flood a packet means a host sends the packet to all of its neighboring hosts and the neighboring host forwards the packet to its neighbors except the neighbor from which it received the packet. If the network is acyclic (i.e., the network is a tree), then the flooding method is an efficient way. Unfortunately, usually the network contains a lot of cycles, that makes the flooding method problematical. In fact a packet may be forwarded along the cycle forever. To avoid dead loops for flooding, several methods are used to control the flooding.

- Sequence-number-controlled flooding: A source node puts its address (or other unique identifier) as well as a broadcast sequence number into a broadcast packet. Each node maintains a list of the source address and sequence number of each broadcast packet it has already

received, duplicated and forwarded. When a node receives a broadcast packet, it checks whether the packet is in this list. If so, the packet is dropped, otherwise the packet is duplicated and forwarded to all the node's neighbors (except the node from which the packet has just been received).

- Reverse path forwarding (RPF): When a router receives a broadcast packet with a given address, it transmits the packet on all of its outgoing links (neighbors except the one on which it was received) only if the packet arrived on the link that is on its own shortest unicast path back to the source. Otherwise, the router simply discards the incoming packet. Such a packet can be dropped because the router knows it either will receive or has already received a copy of this packet on the link that is on its own shortest path back to the sender. The RPF does not use unicast routing to deliver a packet to a destination, nor does it require that a router know the complete shortest path from itself to the source. RPF need only know the next neighbor on its unicast shortest path from itself to the source and determine whether or not to flood a received broadcast packet.

Spanning-tree broadcast

Using controlled flooding still cannot completely avoid transmission of redundant broadcast packets. Some nodes will receive duplicated packets from different paths and discards all except one. Ideally, every node should receive only one copy of the broadcast packet. As we already mentioned, if the network is a tree, then the simple flooding method can achieve the broadcast. So the concept of spanning tree of a connected graph can be used here. A spanning tree of a connected graph $G = (N, E)$ is a graph $G' = (N, E')$ such that E' is a subset of E, G' is connected and contains no cycles. If each link has an associated cost (the graph is then called weighted graph) and the cost of a tree is the sum of the link costs, then a spanning tree whose cost is the minimum of all the graph's spanning tree is called a minimum spanning tree. Therefore one approach to the broadcast routing is first constructing a spanning tree of the network (minimum will be better).

There are many algorithms for finding a spanning tree for a graph. One simple method is called center-based approach (but it will not create the minimum spanning tree in general). In this method, a center node (also

know as a rendezvous point or a core) is defined. Nodes then unicast tree-join messages addressed to the center node. A tree-join message is forwarded using unicast routing toward the center until it either arrives at a node that already belongs to the spanning tree or arrives at the center. In either case, the path that the tree-join message has followed defines the branch of the spanning tree between the edge node that initiated the tree-join message and the center.

Multicast

Multicast service means that some message is delivered to a subset of network nodes. In this case, both unicast and broadcast are not efficient. A number of emerging network applications require the delivery of packets from one or more senders to a group of receivers. Examples include bulk data transfer (software upgrade), streaming continuous media (a live lecture to lecture participants), shared data applications (teleconference), data feeds (stock quotes), interactive gaming (multiplayer games), etc.

In multicast communication, we first need to decide two things: how to identify the receivers of a multicast packet and how to address a packet sent to these receivers. It is not practical to include all the IP addresses of the receivers in the packet, if the size of the receivers is large. In the Internet architecture, a multicast packet is addressed using address indirection. That is, a single identifier is used for the group of receivers, and a copy of the packet that is addressed to the group using this single identifier is delivered to all of the multicast receivers associated with that group. In the Internet, the single identifier that represents a group of receivers is a class D (first octect 224-239) multicast IP address. To manage the multicast addresses, an Internet Group Management Protocol (IGMP) is developed.

IGMP and multicast routing

Internet Group Management Protocol (IGMP) defined in RFC 3376 operates between a host and its directly attached router (one-hop router). IGMP provides the means for a host to inform its attached router that an application running on the host wants to join a specific multicast group. Given that the scope of IGMP interaction is linited to a host and its attached router, another protocol is clearly required to coordinate the multicast routers throughout the Internet. This functionality is accomplished by network-layer multicast

routing algorithm. Therefore, the multicast consists of two components: IGMP and mlticast routing protocols.

Like ICMP, IGMP messages are carried within an IP datagram, with IP protocol number of 2. IGMP has three message types.

- `membership_query` message is sent by a router to all hosts on an attached interface to determine the set of all multicast groups that have been joined by the hosts on that interface.

- `membership_report` message is used to response the `membership_query` message. This message can also used by a host when an application first joins a multicast group without the query message from the router.

- `leave_group` message is used to indicate the host's leaving of the group. This message is optional. If a host does not use `membership_report` message to reply the `membership_query` message, then the router will change the state of the host as leaving.

Several multicast routing protocols have been proposed. Basic idea is to find a tree of links that connects all of the routers that have attached hosts belonging to the multicast group. Multicast packets will then be routed along this tree from the sender to all of the hosts belonging to the multicast group. Some methods are used for forming the tree.

- Multicast routing using a group-shared tree: based on building a spanning tree that includes all edge routers with attached hosts belonging to multicast group. Usually, a center-based approach is used to construct the multicast routing tree, with edge routers with attached hosts belonging to the multicast group sending join messages addressed to the center node. A join message is forwarded using unicast routing toward the center until it either arrived at a router that already belongs to the tree or arrived at the center.

- Multicast routing using a source-based tree: each source in the muticast group constructs a routing tree within the multicst group. In practice, an RPF (reverse path forwarding) algorithm is used to construct a multicast forwarding tree for multicast datagrams originating at that source.

4.4 User datagram protocol (UDP)

UDP, a connectionless datagram transmission, is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols. UDP does not perform handshake and mainly just indicate the port numbers for the process.

UDP uses a port number corresponding to each program. Each UDP message contains the port information for source and destination machine. Port numbers are used to keep track of different conversations crossing the network at the same time. Application software developers agree to use well-known port numbers to do specific tasks that are defined in RFC 1700. Conversations that do not involve an application with a well-known port number are assigned numbers randomly chosen from within a specific range:

- Numbers below 255 are for public applications.
- Numbers from 255 to 1023 are assigned to companies for salable applications.
- Numbers above 1023 are unregulated.

Protocols that use UDP include Trivial File Transfer Protocol (TFTP), Simple Network Management Protocol (SNMP), Network File System (NFS), and the domain name system (DNS), Routing Information Protocol (RIP), etc.

The UDP frame consists of 16-bit source and destination ports numbers. The other fields include the checksum (16 bits) and the data length (16 bits), allowing a maximum of 64 KB of data (including the header). The minimum value of length field is 8. The frame format is shown in Figure 4.8.

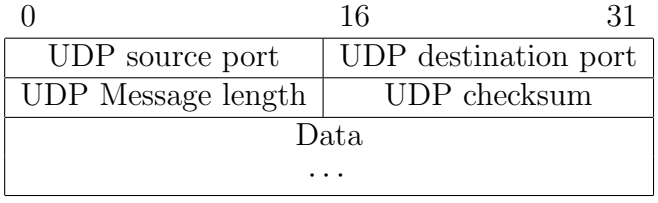

Figure 4.8: UDP frame format

In IPv6 Jumbograms, it is possible to have UDP packets of size greater than 64KB. This allows for a maximum length value of 4,294,967,295 bytes ($2^{32} - 1$) with 8 bytes representing the header and 4,294,967,287 bytes for data.

The checksum field is used for error-checking of the header and data. If no checksum is generated by the transmitter, the field uses the value all-zeros. However, this field is not optional for IPv6. UDP checksum uses the arithmetic checksum on 16-bit words (we have introduced arithmetic checksum in Chapter 2). As an example, suppose that we have three 16-bit words:

```
0110011001100000
0101010101010101
1000111100001100
```

Adding these three words we have the sum

```
0100101011000010
```

Note that the addition had overflow, which was wrapped around. The 1s complement of the sum is `1011010100111101` which is the checksum. When the receiver checks the correctness of the message, he add all the 16-bits words together with the checksum. If the result is `111111111111111`, then the message is correct.

UDP is an efficient transport protocol, but it is not reliable transport protocol and not including congestion control. That may cause some problem.

4.5 Transmission control protocol (TCP)

TCP is one of the most widely used transport protocols on the Internet. It provides a reliable and in-order bytes stream between two ends as it frees applications from worrying about missing or reordered data. It also provides a flow control mechanism for the byte stream. The flow control mechanism allows the receiver to limit the number of bytes it accepts from the sender. TCP is a connection-oriented protocol meaning that the data transfer is preceded by a connection establishment phase, and a connection termination phase follows the data transfer.

TCP is a protocol at the transport layer. So it is implemented at the end systems, but not in the intermediate device such as switches and routers.

The intermediate devices do not maintain TCP connection state. It is not an end-to-end TDM or FDM circuit, or a virtual circuit. A TCP connection provides a full-duplex service. With a TCP connection, either end system can send datagram to the other end system. So each end system maintains TCP send buffer and receive buffer.

4.5.1 Headers and services

TCP is a reliable transmission protocol based on IP. It allows multiple application programs on one machine to communicate to one another through a demultiplexing operation. The demultiplexing operation makes it possible for two or more application programs running on same or different hosts to simultaneously carry out a data transfer. For this purpose, it uses port numbers like UDP. The connection is considered to be an abstraction consisting of a virtual circuit between two applications running on usually two different machines. The host machine address and port number for each machine serve as the end points of such a virtual circuit. The frame format of a TCP message is shown in Figure 4.9.

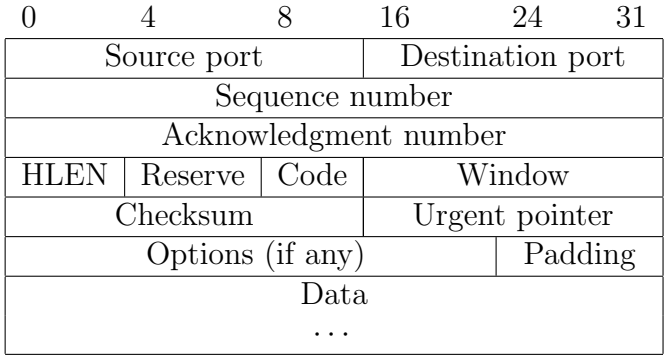

Figure 4.9: TCP frame format

The fields are as follows.

- The source port and destination port fields contains the TCP port numbers that identify the application programs at the two ends of the connection.
- The sequence number identifies the position in the sender's byte stream of the data in the segment. If the SYN flag (see code below) is set (1),

then this is the initial sequence number. The sequence number of the actual first data byte and the acknowledged number in the corresponding ACK are then this sequence number plus 1. TCP views data as an unstructured, but ordered, stream of bytes. So the sequence number for a segment is the byte-stream number of the first byte in the segment. If the SYN flag is clear (0), then this is the accumulated sequence number of the first data byte of this segment for the current session.

The maximum amount of data that can be placed in a segment is limited by the maximum segment size (MSS). The MSS is typically determined by the length of the largest link-layer frame that can be sent by the local sending host (maximum transmission unit, MTU). The MSS is usually set to ensure that a tCP segment will fit into a single link-layer frame. Both Ethernet and PPP link layer protocols have an MTU of 1500 bytes. Thus a typical value of MSS is 1460 bytes.

For example, suppose Host A wants to send some data which consisting of $500,000$bytes, and the MSS is $1,000$ bytes. Suppose the first byte of the data stream is numbered 0. then the second segment gets assigned sequence number 1,000, the third segment gets assigned sequence number 2,000 and so on.

- The acknowledgment number identifies the number the source expects to receive next. If the ACK flag is set then the value of this field is the next sequence number that the receiver is expecting. This acknowledges receipt of all prior bytes (if any). The first ACK sent by each end acknowledges the other end's initial sequence number itself, but no data.

  Suppose Host A received one segment from Host B containing bytes 0 through 535 and another segment containing bytes 900 through 1000. In this case, A is still waiting for byte 536 and beyond. So A's next segment to B will contain 536 in the acknowledgment number field.

- HLEN specifies the size of the TCP header in 32-bit words. The minimum size header is 5 words and the maximum is 15 words thus giving the minimum size of 20 bytes and maximum of 60 bytes, allowing for up to 40 bytes of options in the header. This field gets its name from the fact that it is also the offset from the start of the TCP segment to the actual data.

- The reserve field is for future use and should be set to zero.
- The code field consists 8 bits in original TCP header. From the left-most to right, flags are:
  - CWR: Congestion Window Reduced field used from the TCP sender to inform the receiver that the sender has slowed down so that the receiver can stop sending the ECE-Echo.
  - ECE: Explicit Congestion Notification is used to signal congestion.
  - URG: Urgent pointer field is valid. It is seldom used.
  - ACK: Acknowledgment field is valid. All packets after the initial SYN packet sent by the client should have this flag set.
  - PSH: This segment requests a push. The receiver is requested to deliver the data to the application upon arrival and not buffer it until a full buffer has been received.
  - RST: Reset the connection. It is also used to reject an invalid segment or refuse an attempt to open a connection.
  - SYN: Synchronize sequence numbers. Only the first packet sent from each end should have this flag set.
  - FIN: Sender has reached the end of its byte stream.

  Original TCP header contains a code field of 6 bits. More bits from the reserved field are used for additional flags by later RFC documents.
- The Window indicates the size of the receive window, which specifies the number of window size units (by default, bytes) (beyond the sequence number in the acknowledgment field) that the sender of this segment is currently willing to receive (see Flow control and Window size below).
- The checksum field is used for error-checking of the header and data.
- Urgent pointer: if the URG flag is set, then this field is an offset from the sequence number indicating the last urgent data byte.
- Options: The length of this field is determined by the HLEN field. Options have up to three fields: Option-Kind (1 byte), Option-Length (1 byte), Option-Data (variable). The Option-Kind field indicates the

type of option, and is the only field that is not optional. Depending on what kind of option we are dealing with, the next two fields may be set: the Option-Length field indicates the total length of the option, and the Option-Data field contains the value of the option, if applicable. Some options may only be sent when SYN is set.

- Padding: The TCP header padding is used to ensure that the TCP header ends and data begins on a 32 bit boundary. The padding is composed of zeros

4.5.2 Connection establishment and termination

Connection establishment in TCP is done via a three-way handshake between the sending and receiving hosts.

To initial the handshake, the sending host A sends a SYN signal to the receiver host B indicating that it is ready to transmit and would like to synchronize the byte sequence number x. After B has received the SYN signal, B sends a SYN-ACK signal back to A with a sequence number, say y, and an ACK field equal to $x + 1$. The sequence number y in the SYN signal indicates the starting sequence number of backward traffic from B to A. When A receives this SYN signal from B, it sends an ACK signal to B indicating its expectation of sequence $y+1$ onward. After that, a full-duplex communication is established.

When an endpoint wishes to stop its half of the connection, it transmits a FIN packet, which the other end acknowledges with an ACK. Therefore, a typical tear-down requires a pair of FIN and ACK segments from each TCP endpoint. After both FIN/ACK exchanges are concluded, the side which sent the first FIN before receiving one waits for a timeout before finally closing the connection, during which time the local port is unavailable for new connections; this prevents confusion due to delayed packets being delivered during subsequent connections. Usually the timeout is set for twice the time a packet might live on the Internet (120 seconds).

A connection can be "half-open", in which case one side has terminated its end, but the other has not. The side that has terminated can no longer send any data into the connection, but the other side can. The terminating side should continue reading the data until the other side terminates as well.

It is also possible to terminate the connection by a 3-way signals, when host A sends a FIN and host B replies with a FIN-ACK (merely combines

2 steps into one) and host A replies with an ACK. This is perhaps the most common method. It is possible for both hosts to send FINs simultaneously then both just have to ACK. This could possibly be considered a 2-way signals since the FIN/ACK sequence is done in parallel for both directions.

4.5.3 Flow control and window size advertising

TCP uses a sliding window flow control protocol. In each TCP segment, the receiver specifies in the receive window field the amount of additionally received data (in bytes) that it is willing to buffer for the connection. The sending host can send only up to that amount of data before it must wait for an acknowledgment and window update from the receiving host.

When a receiver advertises a window size of 0, the sender stops sending data and starts the persist timer. The persist timer is used to protect TCP from a deadlock situation that could arise if a subsequent window size update from the receiver is lost, and the sender cannot send more data until receiving a new window size update from the receiver. When the persist timer expires, the TCP sender attempts recovery by sending a small packet so that the receiver responds by sending another acknowledgment containing the new window size. If a receiver is processing incoming data in small increments, it may repeatedly advertise a small receive window. This is referred to as the silly window syndrome, since it is inefficient to send only a few bytes of data in a TCP segment, given the relatively large overhead of the TCP header. TCP senders and receivers typically employ flow control logic to specifically avoid repeatedly sending small segments.

4.5.4 TCP time-out and retransmission

TCP guarantees a reliable delivery of data even in the case of transmission losses and delays. This is possible because each segment of data is retransmitted if an ACK is not received within a certain period of time. The time-out period is set as a function of the round-trip time (RTT). However, because of the variation of the RTTs between hosts on the Internet, we need some method to decide the RTT. One simple technique uses a recursive estimation of the RTT based on the current estimate and the sample RTT as below.

$$\text{estimated RTT} = a\times \text{ estimated RTT } + b\times \text{ sample RTT},$$

where $a + b = 1$, and a and b are shosen to smooth the estimated RTT. If b

is large, the changes in RTT are tracked. A large a is more stable but not quick enough to adapt to real changes in the RTT. When an estimate for the RTT is obtained, the time-out period is simply taken to be twice as long.

There are some algorithms that modify the foregoing simple algorithm for the computation of the RTT. Enhancing TCP to reliably handle loss, minimize errors, manage congestion and go fast in very high-speed environments are ongoing areas of research and standards development. As a result, there are a number of TCP congestion avoidance algorithm variations.

4.6 Domain name system (DNS)

The Domain Name System (RFC 1034, 1035) is a hierarchical distributed naming system for computers, services, or any resource connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities. Most prominently, it translates domain names meaningful for users to the numerical IP addresses needed for the purpose of locating computer services and devices worldwide. By providing a worldwide, distributed keyword-based redirection service, the Domain Name System is an essential component of the functionality of the Internet.

The Domain Name System distributes the responsibility of assigning domain names and mapping those names to IP addresses by designating authoritative name servers for each domain. Authoritative name servers are assigned to be responsible for their particular domains, and in turn can assign other authoritative name servers for their sub-domains.

DNS provides a few other services in addition to translating host names to IP addresses.

- Host aliasing: a host with a complicated host name can have one or more alias names. For example, a host name

  `wei-office-ccc.cs.lakeheadu.ca` may have alias such as

  `ccc.cs.lakeheadu.ca` and `www.ccc.cs.lakeheadu.ca`.

  The host name is called the canonical hostname. DNS can be invoked by an application to obtain the canonical hostname for a supplied alias host name as well as the IP address of the host.

- Mail server aliasing: DNS can be invoked by a mail application to obtain the canonical hostname for a supplied alias hostname as well as IP addresses of the host. In fact, the MX record (we will discuss the record type later) permits an institute's server and web server to have identical aliased hostname.

- Load distribution: DNS is also used to perform load distribution among replicated servers, such as replicated web servers. Some site, such as `cnn.com`, are replicated over multiple servers, with each server running on a different end system and each having a different IP address. For replicated web servers, a set of IP addresses is associated with one canonical hostname. The DNS database contains this set of IP addresses. When client make a DNS query for a name mapped to a set of addresses, the server responds with the entire set of IP addresses, but rotates the ordering of the addresses within each reply. Because a client usually uses the first IP address in the set, DNS rotation distributes the traffic among the replicated servers. Similar method is used for multiple mail servers, content distribution companies.

4.6.1 Mapping domain names to IP addresses

The DNS implementation is just like a tree in which each node represents one possible label. The right-most label corresponds to the node closes to the root, whereas the left-most label corresponds to the host name and it is farthest node from the root.

Roughly to say, there are three classes of DNS servers:

- Root DNS servers: In the Internet, there are 13 root DNS servers (labeled A through M), most of which are located in North America. Each of the 13 root servers actually is a network of replicated server, for both security and reliability purpose. There are 247 root servers as of fall 2011.

- Top-level domain (TLD) servers: There servers are responsible for top-level domains such as com, org, edu, ca, jp, cn etc.

- Authoritative DNS server: Every organization with publicly accessible hosts on the Internet must provide publicly accessible DNS records that map the name of those hosts to IP addresses. An organization's authoritative DNS server houses theses DNS records.

There is another important type of DNS server called the local DNS server. Each ISP has a local DNS server (or several servers). When a host connects to an ISP, the ISP provides the host with the IP address of one or more of its local DNS servers.

The organization domain consists of labels describing the types of client organization. Some examples of labels are:

.com	Commercial organization
.edu	Educational institution
.gov	Government organization
.int	International organization
.mil	Military
.net	Network-related support center
.org	Other organization

It is noticed that the number of levels in the hierarchy is not limited to three. The country name is used as suffix after the organization type. For examples, .us for United States, .fr for France, .cn for China, etc.

The DNS servers store resource records (RRs), that provide hostname-to-IP address mappings. Each DNS reply message carries one or more RRs.

A resource record is a four-tuple that contains the fields: `(Name, Value, Type, TTL)`. The meaning of `Name` and `Value` depends on `Type`:

- If `Type=A`, then `Name` is a hostname and `Value` is the IP address for the hostname.
- If `Type=NS`, the `Name` is a domain and `Value` is the hostname of an authoritative DNS server that knows how to obtain the IP addresses for hosts in the domain. This record is used to route DNS queries further along in the query chain.
- If `Type=CNAME`, then `Value` is a canonical hostname for the alias hostname `Name`. This record can provide querying hosts the canonical name for an alias hostname.
- If `Type=MX`, then `Value` is the canonical name of a mail server that has an alias hostname `Name`. MX records allow the hostnames of mail servers to have simple aliase. By using the MX record, a company can have the same aliased name for its mail server and for one of its other servers. To obtain the canonical name for the mail server, a DNS client would query for an MX record; to obtain the canonical name for the other server, the DNS client would query for the CNAME record.

4.6.2 Name servers

The Domain Name System is maintained by a distributed database system, which uses the client-server model. The nodes of this database are the name servers. Each domain has at least one authoritative DNS server that publishes information about that domain and the name servers of any domains subordinate to it.

When domain names are registered with a domain name registrar, their installation at the domain registry of a top level domain requires the assignment of a primary name server and at least one secondary name server. The requirement of multiple name servers aims to make the domain still functional even if one name server becomes inaccessible or inoperable. The designation of a primary name server is solely determined by the priority given to the domain name registrar. For this purpose, generally only the fully qualified domain name of the name server is required, unless the servers are contained in the registered domain, in which case the corresponding IP address is needed as well. Primary name servers are often master name servers, while secondary name server may be implemented as slave servers.

To insert records into the DNS database, the domain names need to register at a registrar. A registrar is a commercial entity that verifies the uniqueness of the domain name, enters the domain mane into the DNS database, and collects a small fee for its services. A list and some other information about the registrars can be found at `www.internic.net`

4.6.3 DNS resolvers

The client-side of the DNS is called a DNS resolver. It is responsible for initiating and sequencing the queries that ultimately lead to a full resolution (translation) of the resource sought, e.g., translation of a domain name into an IP address.

A DNS query may be either a non-recursive query or a recursive query:

- A non-recursive query is one in which the DNS server provides a record for a domain for which it is authoritative itself, or it provides a partial result without querying other servers.

- A recursive query is one for which the DNS server will fully answer the query (or give an error) by querying other name servers as needed. DNS servers are not required to support recursive queries.

The resolver, or another DNS server acting recursively on behalf of the resolver, negotiates use of recursive service using bits in the query headers.

Resolving usually entails iterating through several name servers to find the needed information. However, some resolvers function more simply by communicating only with a single name server. These simple resolvers (called "stub resolvers") rely on a recursive name server to perform the work of finding information for them.

A reverse lookup is a query of the DNS for domain names when the IP address is known. Multiple domain names may be associated with an IP address.

Users generally do not communicate directly with a DNS resolver. Instead DNS resolution takes place transparently in applications such as web browsers, e-mail clients, and other Internet applications. When an application makes a request that requires a domain name lookup, such programs send a resolution request to the DNS resolver in the local operating system, which in turn handles the communications required.

DNS primarily uses User Datagram Protocol (UDP) on port number 53 to serve requests. DNS queries consist of a single UDP request from the client followed by a single UDP reply from the server. The Transmission Control Protocol (TCP) is used when the response data size exceeds 512 bytes, or for tasks such as zone transfers. Some resolver implementations use TCP for all queries.

Chapter 5

Wireless and Mobile Networks

A wireless network communication takes place over a wireless channel (which is usually a radio channel, or sometimes an infrared channel). The challenges for wireless networks are quite different from that of wired networks, especially at the data link layer and the network layer. Mobile networks require more interesting techniques.

5.1 Cellular networks

Originally, cellular networks provided only voice communications services and they could also be used to send and receive short text messages. Today, the range of application is much wider, including data communications, Internet access, multimedia applications (video telephony), and mobile payment services, etc.

Cellular networks are infrastructure-based networks. The infrastructure consists of base stations and a wired backbone network that connects the base stations together, as well as to the wired telephone system and to the Internet. Each base station serves only a limited physical area, called a cell. All the base stations of a given network operator together can cover a large area. Different network operators can jointly provide ubiquitous coverage and enable continent wide and ever worldwide mobility for users. In cellular networks, the only wireless part in the system is the link between the mobile phone and the base station. The rest is wired network. Base stations are connected to the mobile switching center (MSC) which is connected to the public switched telephone network (PSTN). The frequency spectrum allocated to

wireless communications is very limited. Each cell is assigned a certain number of channels. To avoid radio interference, the channels assigned to one cell must be different from the channels assigned to its neighboring cells. However, the same channels can be reused by two cells that are far apart.

5.1.1 GSM

Global System for Mobile Communications (GMS) is a European initiated standard which is a prominent example of cellular network.

The cellular technologies are often classified to one of several "generations". The earliest generations were designed primarily for voice traffic. First generation (1G) systems were analog FDMA systems designed exclusively for voice-only communication. These 1G systems are almost extinct now, having been replaces by digital 2G systems. The original 2G systems were also designed for voice, but later extended to support data as well as voice service. The 3G systems that currently are deployed also support voice and data, but with an ever increasing emphasis on data capabilities and higher-speed radio access links.

2G Cellular network

In GSM, each cell contains a base transceiver station (BTS) that transmits signals to and receivers signals from the mobile station in its cell. The coverage area of a cell depends on many factors, such as the transmitting power of BTS, the transmitting power of the user devices, obstructing buildings in the cell, and the height of base station antennas. Originally, each cell contains one base station residing in the center of the cell, many systems today place the BTS at corners where three cells intersect, so that a single BTS with directional antennas can service three cells.

2G cellular systems uses combined FDM/TDM for the air interface. With pure FDM, the channel is partitioned into a number of frequency bands with each band devoted to a call. With pure TDM, time is partitioned into frames with each frame further partitioned into slots and each call being assigned the use of a particular slot in the revolving frame. In combined FDM/TDM system, the channel is partitioned into a number of frequency sub-band, and within each sub-band, time is partitioned into frames and slots. Therefore in a combined FDM/TDM system, if the channel is partitioned into F sub-bands and time is partitioned into T slots, then the channel will be able to

support $F \cdot T$ simultaneous calls. GSM systems consist of 200-kHz frequency bands with each band supporting eight TDM calls. GSM encodes speech at 13 kbps and 12.2kbps.

A GSM network's base station controller (BSC) will typically service several tens of base transceiver stations. The role of the BSC is to allocate BTS radio channels to mobile subscribers, perform paging (finding the cell in which a mobile user in resident), and perform handoff (change base station during a call) of mobile users. The base station controller and its controlled base transceiver stations collectively constitute a GSM base station system (BSS). The mobile switching center (MSC) is for user authorization and accounting, call establishment and teardown, and handoff. A single MSC will typically contain up to five BSCs resulting in approximately 200K subscribers per MSC. A cellular provider's network will have a number of MSCs, with special MSCs know as gateway MSCs connecting the provider's cellular network to the larger public telephone network.

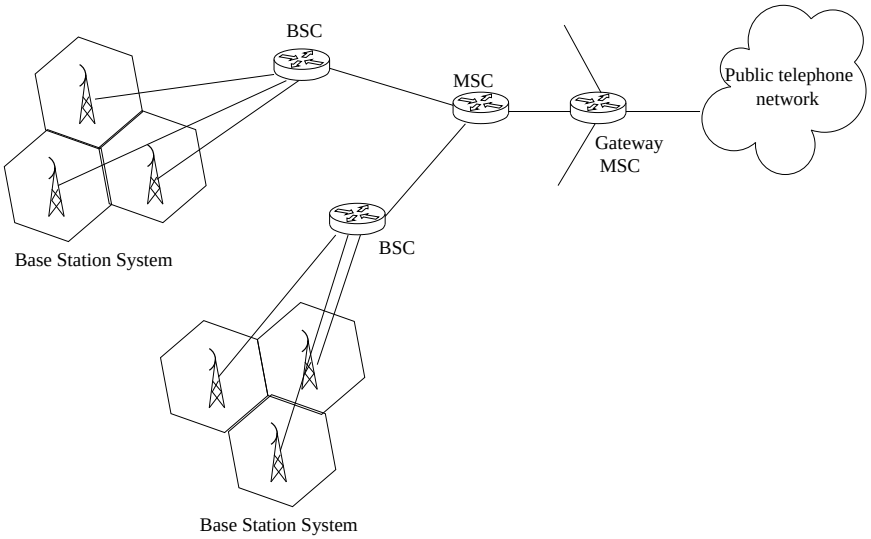

Figure 5.1: GSM 2G network

Figure 5.1 gives a simple example of GSM 2G cellular network architecture.

3G Cellular network

2G cellular network focuses on the voice communication. When we want to use smart phones to communicate with data and access the internet, we need to run a full TCP/IP protocol stack and connect into internet via the cellular data network. Unfortunately, we do not have a single official body that sets requirements for 2.5G, 3G, 3.5G or 4G technologies, making it harder to sort out the differences among competing standards.

In what follows, we will focus on the UMTS (Universal Mobile Telecommunication Service) 3G standards developed by the 3rd Generation Partnership project (3GPP), a widely deployed 3G technology.

The 3G core cellular data network connects radio access network to the public Internet. The basic idea of the designers of 3G data service is: leave the existing core GSM cellular voice network untouched, adding additional cellular data functionality in parallel to the existing cellular voice network.

There are two types of nodes in the 3G core network: Serving GPRS Support Nodes (SGSNs) and Gateway GPRS Support Nodes (GGSNs). Here GPRS stands for Generalized Packet Radio Service, an early cellular data service in 2G networks. An SGSN is responsible for delivering datagrams to/from the mobile nodes in the radio access network to which the SGSN is attached. The SGSN interacts with the cellular voice network's MSC for that area, providing user authorization and handoff, maintaining location information about active mobile nodes, and performing datagram forwarding between mobile nodes in the radio access network and a GGSN. The GGSN acts as a gateway, connecting multiple SGSNs into the larger Internet. A GGSN is thus the last piece of 3G infrastructure that a datagram originating at a mobile node encounters before entering the larger Internet. To the outside world, the GGSN looks like any other gateway router. The mobility of the 3G nodes within the GGSN's network is hidden from the outside world behind the GGSN. The Figure 5.2 explains the 3G system architecture.

Figure 5.2 gives a simple example of GSM 3G cellular network architecture. The 3G radio access network is the wireless first-hop network that we see as a 3G user. The Radio Network Controller (RNC) typically controls several cell base transceiver stations similar to the base stations in 2G system. Each cell's wireless link operates between the mobile nodes and a base transceiver station. The RNC connects to both the circuit-switched cellular voice network via an MSC, and to the packet-switched Internet via an SGSN.

A significant change in 3G UMTS over 2G networks is that UMTS uses a

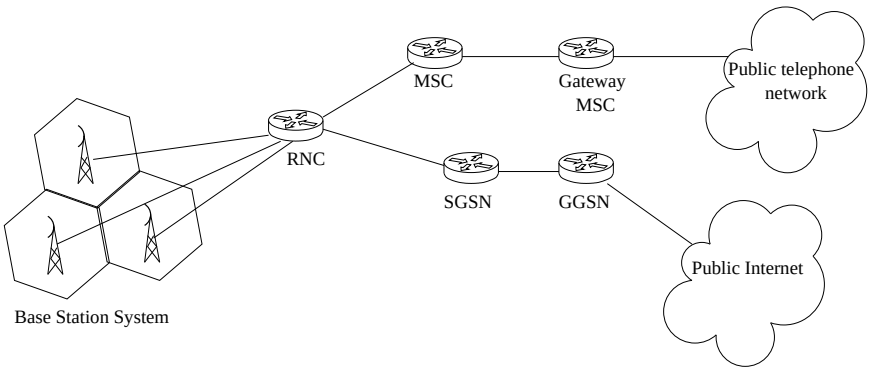

Figure 5.2: GSM 3G network

CDMA (Direct Sequence Wideband CDMA, DS-WCDMA) technique within TDMA slots rather than the FDMA/TDMA scheme. On the other hand, the TDMA slots are available on multiple frequencies. So this is an interesting applications combining of 3 multiplexing techniques. This change requires a new 3G cellular wireless-access network operating in parallel with the 2G BSS radio network. The data service associated with the WCDMA specification is known as HSP (High Speed Packet Access) and promises downlink data rates of up to 14 Mbps.

4G: LTE and WiMAX

Two 4G candidate systems are commercially deployed: the Mobile WiMAX (Worldwide Interoperability for Microwave Access) standard (first used in South Korea in 2007), and the first-release Long Term Evolution (LTE) standard (in Oslo, Norway and Stockholm, Sweden since 2009). It has however been debated if these first-release versions should be considered to be 4G or not.

In the United States, Sprint (previously Clearwire) has deployed Mobile WiMAX networks since 2008, while MetroPCS became the first operator to offer LTE service in 2010. USB wireless modems were among the first devices able to access these networks, with WiMAX smartphones becoming available during 2010, and LTE smartphones arriving in 2011. 3G and 4G equipment made for other continents are not always compatible, because of different frequency bands. Mobile WiMAX is currently (April 2012) not available for the European market.

In March 2008, the International Telecommunications Union-Radio communications sector (ITU-R) specified a set of requirements for 4G standards, named the International Mobile Telecommunications Advanced (IMT-Advanced) specification, setting peak speed requirements for 4G service at 100 megabits per second (Mbps) for high mobility communication (such as from trains and cars) and 1 gigabit per second (Gbps) for low mobility communication (such as pedestrians and stationary users).

Since the first-release versions of Mobile WiMAX and LTE support much less than 1 Gbit/s peak bit rate, they are not fully IMT-Advanced compliant, but are often branded 4G by service providers. According to operators, a generation of network refers to the deployment of a new non-backward-compatible technology. On December 6, 2010, ITU-R recognized that these two technologies, as well as other beyond-3G technologies that do not fulfill the IMT-Advanced requirements, could nevertheless be considered 4G, provided they represent forerunners to IMT-Advanced compliant versions and a substantial level of improvement in performance and capabilities with respect to the initial third generation systems now deployed.

Mobile WiMAX Release 2 (also known as WirelessMAN-Advanced or IEEE 802.16m') and LTE Advanced (LTE-A) are IMT-Advanced compliant backwards compatible versions of the above two systems, standardized during the spring 2011 and promising speeds in the order of 1 Gbps.

As opposed to earlier generations, a 4G system does not support traditional circuit-switched telephony service, but all-Internet Protocol (IP) based communication such as IP telephony. As seen below, the spread spectrum radio technology used in 3G systems, is abandoned in all 4G candidate systems and replaced by OFDMA multi-carrier transmission and other frequency-domain equalization (FDE) schemes, making it possible to transfer very high bit rates despite extensive multi-path radio propagation (echoes). The peak bit rate is further improved by smart antenna arrays for multiple-input multiple-output (MIMO) communications.

LTE has two important innovations over 3G systems.

- Evolved Packet Core (EPC): This is a simplified all-IP core network that unifies the separate circuit-switched cellular voice network and the packet-switched cellular data network. The EPC allows multiple types of radio access networks, including legacy 2G and 3G radio access networks, to attach to the core network.

- LTE Radio Access Network: LTE uses a combination of frequency di-

vision multiplexing and time division multiplexing on the downstream channel, know as orthogonal frequency division multiplexing (OFDM). In LTE, each active mobile node is allocated one or more 0.5 ms time slots in one or more of the channel frequencies. By being allocated increasingly more time slots, a mobile node is able to achieve increasingly higher transmission rates. Slot reallocation among nodes can be performed as often as once every millisecond. Another inovation in the LTE radio network is the use of multiple-input, multiple-output (MIMO) antennas.

The particular allocation of time slots to mobile nodes is not mandated by the LTE standard. Instead, the decision of which mobile nodes will be allowed to transmit in a given time slot on a given frequency is determined by the scheduling algorithm provided by the LTE equipment vendor and the network operator.

WiMAX refers to interoperable implementations of the IEEE 802.16 family of wireless-networks standards ratified by the WiMAX Forum. WiMAX Forum certification allows vendors to sell fixed or mobile products as WiMAX certified, thus ensuring a level of interoperability with other certified products, as long as they fit the same profile.

The original IEEE 802.16 standard (now called "Fixed WiMAX") was published in 2001. WiMAX adopted some of its technology from WiBro, a service marketed in Korea. The term fixed arises because the technology does not provide for handoff among access points. Thus it is designed to provide connections between a service provider and a fixed location.

Mobile WiMAX (originally based on 802.16e-2005) is the revision that was deployed in many countries, and is the basis for future revisions such as 802.16m-2011. The technology of the Mobile WiMAX offers handoff among access points, which means the system can be used with portable devices such as laptop computers and cell phones.

WiMAX can be used for a number of applications including broadband connections, cellular backhaul, hotspots, etc. It is similar to Wi-Fi, but it can enable usage at much greater distances.

Some key features of WiMAX can briefly summarized as follows:

- Uses licensed spectrum (i.e., offed by carriers).
- Each cell can cover a radius of 3 to 10 km.

- Uses scalable orthogonal FDM.
- Guarantees quality of services (for voice or vidio).
- Can transport 70 Mbps in each direction at short distances.
- Provides 10 Mbps over a long distance (10 km)

5.1.2 Security

One important security requirement of GSM is subscriber authentication. In addition to subscriber authentication, GSM also provides some countermeasures for the inherent weaknesses of the wireless channel. GSM provides confidentiality for voice communications and signaling over the wireless interface, and it protects the privacy of subscribers by hiding their identity from eavesdroppers. Being a wide area system, GSM security services operate in a multi-party environment.

In GSM, a subscriber and a network operator (called the home network operator) have a contractual relationship which is represented as a long-term secret key. The secret key and other identity related information of the subscriber are not stored in the mobile phone, but in a separate security unit, called the SIM (Subscriber Identity Module).

Subscriber authentication in GSM is based on a challenge-response principle. The subscriber receives an unpredictable random number as a challenger, and she must compute a correct response in order to be authenticated. The correct response is computed from the challenge and the secret key of the subscriber, which is shared with the home network.

The GSM subscriber authentication protocol can be described as follows. Assume that the subscriber roams into a foreign network, usually refereed to as the visited network. First the mobile reads the IMSI (International Mobile subscriber Identity) from the SIM, and sends it to the visited network. Based on the IMSI, the visited network determines the identity of the home network of the subscriber. Then the visited network forwards the IMSI to the home network via the backbone. The home network looks up the secret key K that corresponds to the subscriber by the IMSI. It then creates a triplet $(RAND, SRES, CK)$, where $RAND$ is an unpredictable random number used as the challenge, $SRES$ is the correct response to the challenge, and CK is a key to be used for encrypting communications over the wireless

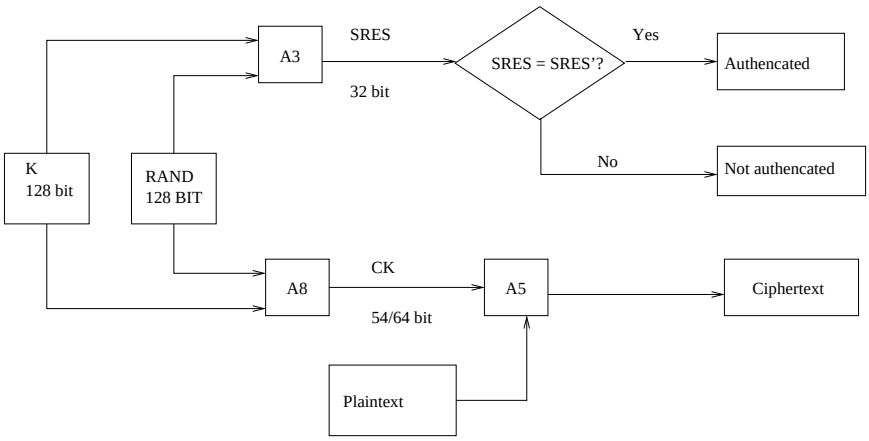

Figure 5.3: GMS security model

interface between the mobile phone and the base station of the visited network. $RAND$ is generated by a pseudo random number generator. $SRES$ and CK are computed from $RAND$ and K using the algorithms denoted by A3 and A8, respectively, in the GSM specifications. The triplet is sent to the visited network, which challenges the mobile phone with $RAND$. After the successful authentication of the subscriber, the communication between the mobile phone and the base station of the visited network are encrypted and decrypted with CK by using the stream cipher denoted by A5 in the GSM specification. It requires some trust in the home network operator by the visited network operator, which is established by signing roaming agreements between the two operators. In practice, the home network can transfer several triplets to the visited network when the subscriber first authenticates herself. In this way, the visited network does not need to contact the home network every time the subscriber needs to be authenticated.

The identity of the subscriber is hidden from eavesdroppers on the wireless interface as follows. After each successful authentication, the subscriber receives a temporary identifier called TMSI (Temporary Mobile Subscriber Identifier) from the visited network. The TMSI is encrypted with the freshly established key CK. In the next authentication request, the mobile phone uses the TMSI, instead of IMSI, to identify the subscriber. The TMSI is mapped to the IMSI by the visited network.

When the subscriber moves into another visited network, the new network

contacts the previous one and sends it the TMSI received from the mobile phone. The previous network looks up the data associated with the TMSI and transfers the IMSI of the subscriber and the remaining triplets (if any) to the new network, so that the new network can continue serving the subscriber. In case the data associated with the TMSI are no longer available in the previous network, the new network requests the mobile phone to send the IMSI in order to bootstrap the TMSI mechanism again.

Some flaws and attacks for GSM:

- No authentication of the network is provided to the user. It is possible for an attacker to set up a false base station with the same mobile network code as the subscriber's network.
- Common implementation of A3/A8 is flawed. This is because of the algorithms use the procedure COMP128.
- Vulnerabilities in the subscriber identity confidentiality mechanism. When a network loses track of a TMSI, it must ask the subscriber to submit its IMSI over the radio link, using a special mechanism for identity request. Thus the IMSI is sent in plain text. An attacker may use this to map a TMSI to its IMSI.
- Over the air cracking of K. An attacker first obtains the identity of a mobile phone (from TMSI and IMSI). Then the attacker requests a lot $SRES$ by sending $RAND$s. In this way, the attacker collects the $(RAND, SRES)$ pairs until he gains enough information to derive the key K (known plaintext attack).

UMTS security

Universal Mobile Telecommunications System (UMTS) is an extension and improvement of GSM, which is the third generation cellular network in Europe. It provides necessary mechanisms for authenticating the network to the subscriber and considering integrity protection over the wireless interface.

In UMTS, the GSM triplet are replaced by authentication vectors that have five elements: $(RAND, XRES, CK, IK, AUTN)$. $RAND$ is an unpredictable random number used as a challenge in the subscriber authentication protocol, $XRES$ is the expected response to $RAND$ and CK is an encryption key to be used between the mobile phone and the base station of the

visited network. Both $XRES$ and CK are computed from $RAND$ and the long-term secret key K of the subscriber. In addition, IK is an integrity protection key and $AUTN$ is a token that authenticates the home network to the subscriber and proves the freshness of $RAND$. $AUTN$ consists of three fields: $AUTN = (SQN \oplus AK, AMF, MAC)$, where

- SQN is a sequence number maintained synchronously by both the subscriber and the home network;
- AK is called the anonymity key, and it is used to hide the value of SQN from eavesdroppers. AK is generated from $RAND$ and K;
- AFM is an authentication and key management field used to pass parameters form the home network to the subscriber, but is is not fully specified in the UMTS standard;
- MAC is a message authentication code computed over $RAND, SQN$, and AMF using the long-term key K.

The subscriber authentication protocol is modified in such a way that, upon request, the visited network receives an authentication vector from the home network and it passes not only the challenge $RAND$ to the subscriber, but also the authentication token $AUTN$. The subscriber first generates the anonymity key AK and decodes the sequence number SQN received in $AUTN$. SQN is encoded with AK to protect the privacy of the subscriber. Otherwise, an eavesdropper could associate different executions of the authentication protocol with consecutive sequence numbers to the same subscriber. Once SQN is obtained, the subscriber verifies the MAC. If this verification is successful, then she knows that $RAND$ originates from her home network. Then the subscriber verifies if SQN is greater than the last sequence number stored by the subscriber. If it does not hold, then the protocol fails. This prevents the subscriber from accepting an old challenge. Finally, the subscriber computes a response RES to $RAND$ and sends it back to the visite network. The subscriber also computes CK and IK. Naturally, these computations are not performed by the subscriber herself, but the security unit of her mobile phone, which is called USIM.

The visited network compares RES to $XRES$, and if they are equal, then the authentication of the subscriber succeeds. After that, the mobile phone and the base station of the visited network protect the integrity and

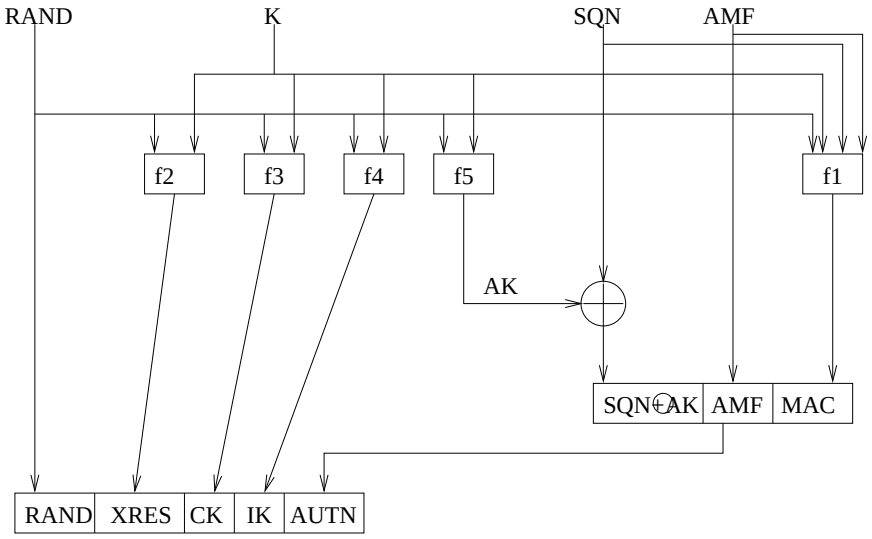

Figure 5.4: UMTS authentication vector

the confidentiality of their communications with IK and CK, respectively. Figure 5.4 describes the construction of UMTS authentication vector.

One weakness of the UMTS subscriber authentication protocol is that the home network does not pass any confirmation regarding the identity of the visited network to the subscriber in the authentication token. Therefore the visited network is not authenticated to the subscriber. This allows a malicious network operator X to masquerade as network Y to the subscriber. It would still authenticate itself as X to the home network, but the subscriber would not know this, and she would believe that she is served by Y. This can be a problem, as X and Y could use different tariffs and the subscriber would learn that she actually used a more expensive network when she receives her bill.

5.2 Wireless LANs: WiFi

Although many technologies and standards for wireless LAN were developed, the most common and important standard is the IEEE 802.11 wireless LAN, also known as WiFi. There are several 802.11 standards for wireless LAN technologies, including 802.11b, 802.11a and 802.11g. The main characteris-

tics of these standards are as follows:

Standard	Frequency Range (United States)	Data Rates
802.11a	5.1-5.8 GHz	up to 54 Mbps
802.11b	2.4-2.485 GHz	up to 11Mbps
802.11g	2.4-2.485 GHz	up to 54 Mbps

A number of dual-mode (802.11a/b) and tri-mode (802.11a/b/g) devices are also available. The three standards share many characteristics. Some of these characteristics are:

- They all use the same medium access protocol CSMA/CA.
- They all use the same frame structure for their link-layer frames.
- They have the ability to reduce their transmission rate in order to reach out over greater distances.
- All of them allow for both "infrastructure mode" and "ad hoc" mode.

The 802.11b and 802.11g wireless LANs operate in the unlicensed frequency band of 2.4 - 2.485 GHz, competing for frequency spectrum with 2.4 GHz phones and microwave ovens. 802.11a wireless LANs can run at significantly higher rates, but do so at higher frequencies. Due to the higher frequency, 802.11a LANs have a shorter transmission distance for a given power level and suffer more from multipath propagation.

A relatively new WiFi standard 802.11n uses multiple input multiple output (MIMO) antennas that are transmitting/receiving different signals. Depending on the modulation scheme used, transmission rates of several hundred megabits per second are possible with 802.11n.

5.2.1 The 802.11 architecture

The basic building block of the 802.11 architecture is the basic service set (BSS). A BSS contains one or more wireless stations and a central base station, know as an access point (AP). The AP in each BSS connects to an interconnection device such as a router, which in turn leads to the Internet. In a typical home network, there is one AP and one router (typically integrated together as on unit) that connects the BSS to the Internet.

Each 802.11 wireless station has a 6-bite MAC address that is stored in the firmware of the station's adapter (network interface card). Each AP also has a MAC address for its wireless interface. As with Ethernet, there MAC addresses are administered by IEEE and are globally unique.

Wireless LANs that deploy APs are often refereed to as infrastructure wireless LANs, with the infrastructure being the APs along with the wired Ethernet infrastructure that interconnects the APs and a router.

IEEE 802.11 stations can also group themselves together to form an ad hoc network, a network with no central control and with no connections to the "out side" world.

Channels and association

In 802.11, each wireless station needs to associate with an AP before it can send or receive network layer data. When a network administrator installs an AP, the administrator assigns a one or two-word Service Set Identifier (SSID) to the access point. The administrator must also assign a channel number to the AP. To understand channel numbers, recall that 802.11 operates in the frequency range of 2.4-2.485 GHz. Within this 85 MHz band, 802.11 defines 11 partially overlapping channels. Any two channels are non-overlapping if an only if they are separated by four or more channels. In particular, the set of channels 1, 6 and 11 is the only set of three non-overlapping channels. This means that an administrator could create a wireless LAN with an aggregate maximum transmission rate of 33 Mbps by installing three 802.11b APs at the same physical location, assigning channels 1, 6, and 11 to APs, and interconnecting each of the APs with a switch.

The 802.11 standard requires that an AP periodically send beacon frames, , each of which includes the AP's SSID and MAC address. The wireless station (laptop computer, for example), knowing that AP's are sending out beacon frames, scans the 11 channels, seeking beacon frames from any AP's that may be out there. Having learned about available AP's from the beacon frames, the wireless station selects one of the AP's for association.

The process of scanning channels and listening for beacon frames is known as passive scanning. A wireless host can also perform active scanning, by broadcasting a probe frame that will be received by all APs within the wireless host's range. APs respond to the probe request frame with a probe response frame. The wireless host can then choose the AP with which to associate from among the responding APs.

After selecting the AP with which to associate, the wireless host sends an association request frame to the AP, and the AP responds with an association response frame. Once associated with an AP, the host will want to join the subnet to which the AP belongs. Thus the host will typically send a DHCP discovery message into the subnet via the AP in order to obtain an IP address on the subnet. Once the address is obtained, the rest of the Internet views that host simply as another host with an IP address in that subnet.

In order to create an association with a particular AP, the wireless station may be required to authenticate itself to the AP. 802.11 wireless LANs provide several alternatives for authentication and access. One approach, used by many companies, is to permit access to a wireless network based on a station's MAC address. A second approach, employs usernames and passwords. In both cases, the AP communicates with an authentication server, relaying information between the wireless end-point station and the authentication server using a protocol such as RADIUS (RFC 2865) or DIAMETER (RFC 3588). Separating the authentication server from the AP allows one authentication server to serve many APs, centralizing the decisions of authentication and access within the single server, and keeping AP costs and complexity low.

5.2.2 The 802.11 MAC protocol

Once a wireless station is associated with an AP, it can start sending and receiving data frames to and from the access point. Since multiple stations may want to transmit data frames at the same time over the same channel, some multiple access protocol is needed to coordinate the transmissions. The designers of 802.11 chose a random access protocol for 802.11 wireless LANs. This access protocol is referred to as CSMA with collision avoidance, or CSMA/CA. Here CSMA stands for "carrier sense multiple access", meaning that each station senses the channel before transmitting, and refrains from transmitting when the channel is sensed busy. Instead of using collision detection (CSMA/CD) in Ethernet, 802.11 uses collision-avoidance technique. Because of the relatively high bit error rates of wireless channels, 802.11 uses a link-layer acknowledgment/retransissin (ARQ) scheme.

First let us examine the 802.11's link-layer acknowledgment scheme. When a station in a wireless LAN sends a frame, the frame may not reach the destination station intact a variety of reasons. To deal with this problem, the 802.11 MAC protocol applies link-layer acknowledgments. When the desti-

nation station receives a frame that passes the CRC, it waits a short period of time known as the Short Inter-fram Spacing (SIFS) and then sends back an acknowledgment frame. If the transmitting station does not receive an acknowledgment within a given amount of time, it assumes that an error has occurred and retransmits the frame, using the CSMA/CA protocol to access the channel. If an acknowledgment is not received after some fixed number of retransmissions, the transmitting station gives up and discards the frame.

The outline for CSMA/CA protocol is as follows. Suppose that a station (wireless station or an AP) has a frame to transmit.

1. If initially the station senses the channel idle, it transmits its frame after a short period of time know as the Distributed Inter-frame Space (DIFS).

2. Otherwise, the station chooses a random backoff value using binary exponential backoff (some method to decide the value) and counts down this value when the channel is sensed idle. While the channel is sensed busy, the counter value remains frozen.

3. When the counter reaches zero (note that this can only occur while the channel is sensed idle), the station transmits the entire frame and then waits for an acknowledgment.

4. If an acknowledgment is received, the transmitting station knows that its frame has been correctly received at the destination station. If the station has another frame to send, it begins the CSMA/CA at step 2. If the acknowledgment is not received, the transmitting station reenters the backoff phase in step 2, with the random value chosen from a larger interval.

Note that different from Ethernet protocol, CSMA/CA uses SIFS and random backoff value to hold off the transmission for a short time. The reason is as follows. In the case of wireless LAN, when a station transmits a frame, it does not detect collision. There are some reasons for that. To detect collision, the station requires the ability to send and receive at the same time, which will cost the adapter hardware a lot in the case of wireless. Moreover, in some situation, the station will not be able to detect a collision because of the hidden terminal problem (there are some physical obstacle between two transmitting stations, or it is caused by the fading of the signal

strength). Therefore, if two stations find that the channel is in idle and start to transmit, then a collision occur. But if two stations choose different random value of backoff, then the collision can be avoided.

RTS and CTS

The 802.11 MAC protocol also includes (but optional) a reservation scheme that helps avoid collision even in the presence of hidden terminals. Considering the example of hidden terminal in Figure 5.5.

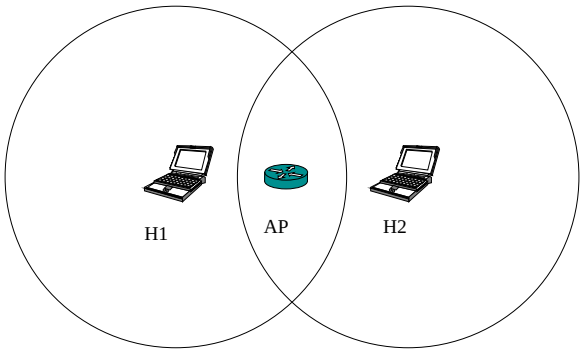

Figure 5.5: Hidden terminal example

In this example, both stations H1 and H2 are within range of the AP and are associated with the AP. However, due to the fading, the signal from H1 cannot reach H2 and the signal from H2 cannot reach H1. So each of the wireless stations is hidden from the other, but neither is hidden from the AP. Now, suppose station H1 is transmitting a frame and halfway through H1's transmission, station H2 wants to send a frame to the AP. H2 first wait a DIFS interval and then transmit the frame, resulting a collision (H1 has not finished the transmission). The channel will therefore be wasted during the entire period of H1's transmission as well as during the H2's transmission.

To avoid this problem, IEEE 802.11 protocol allows a station to use a short Request to Send (RTS) control frame and a short Clear to Send (CTS) control frame to reserve access to the channel. When a sender wants to send a data frame, it can first sent an RTS to the AP, indicating the total time required to send the data frame and the acknowledgment (ACK) frame. When the AP receives the RTS frame, it responds by broadcasting a CTS frame. This CTS frame servers two purpose: it gives the sender explicit

permission to send and also instructs the other stations not send for the reserved duration. The use of RTS and CTS can improve the performance because:

- The hidden station problem is mitigated, since a long data frame is transmitted only after the channel has been reserved.
- Because the RTS and CTS frames are short, a collision involving an RTS or CTS frame will last only for the duration of the short RTS or CTS frame. Once the RTS and CTS frames are correctly transmitted, the following data and ACK frames should be transmitted without collisions.

Although the RTS/CTS exchange can help reduce collisions, it also introduces delay and consumes channel resources. For this reason, the RTS/CTS exchange is only used when the frame is long. In practice, each wireless station can set an RTS threshold such that the RTS/CTS sequence is used only when the frame is longer than the threshold.

5.2.3 The IEEE 802.11 frame

The format of IEEE 802.11 frame is shown as follows:

2	2	6	6	6	2	6	0-2312	4
frame control	duration	addr 1	addr 2	addr 3	seq control	addr 4	payload	CRC

The numbers above each of the fields in the frame represented the lengths of the field in bytes. The fields are as follows:

- Payload: typically consists of an IP datagram of an ARP packet. Usually it is fewer than 1,500 bytes, although this field can be as long as 2,312 bytes.
- CRC: a 32-bit cyclic redundancy checksum.
- Address field: it has four address fields, each of which can hold a 6-byte MAC address. Three address fields are needed for internetworking purpose-specifically, for moving the network-layer datagram from a

wireless station through an AP to a router interface. The fourth address field is used when APs forward frames to each other in ad hoc mode. The first three address fields are defined as follows:

  - Address 2 is the MAC address of the station that transmits the frame. If a wireless station transmits the frame, that station's MAC address is inserted in the address 2 field. Similarly, if an AP transmits the frame, the AP's MAC address is inserted to the address 2 field.
  - Address 1 is the MAC address of the wireless station that is to receive the frame. For example, if a mobile wireless station transmits the frame, address 1 contains the MAC address of the destination AP.
  - To understand address3, recall that the BSS is part of a subnet, and that this subnet connects to other subnets via some router interface. Address 3 contains the MAC address of this router interface.

- Sequence control: whenever a station correctly receives a frame from another station, it sends back an acknowledgment. If the ACK frame get lost, the sending station may send multiple copies of a given frame. The use of sequence numbers allows the receiver to distinguish between a newly transmitted frame and the retreansmission of a previous frame.

- Duration: 802.11 allows a transmitting station to reserve the channel for a period of time that includes the time to transmit its data frame and the time to transmit an acknowledgment. This duration value is included in this field.

- Frame control: This field includes many subfields including: Protocol version, Type, Subtype, To AP, From AP, More fragment, Retry, Power mgt, More data, WEP, Reserved. The type and subtype fields are used to distinguish the association, RTS, CTS, ACK, and data frame. The to and from fields are used to define the meanings of different address fields. WEP field indicates whether encryption is used.

5.2.4 Gigabit WiFi

Since the speed of Ethernet standard has extended in the gigabit per second range, the speed of WiFi is also requested to extend. IEEE 802.11 has introduced two new standards for that purpose.

IEEE 802.11ac

IEEE 802.11ac operates in the 5 GHz band, same as 802.11a and 802.11n. The new standard achieves much higher data rates than 802.11n by means of enhancements in three areas:

- Bandwidth: The maximum bandwidth of 802.11n is 40MHz, but the maximum bandwidth of 802.11ac is 160 MHz.
- Signal encoding: 802.11n uses 64 QAM with OFDM, and 802.11ac uses 256 QAM with OFDM. (Quadrature Amplitude Modulation is a popular analog signaling technique). Thus more bits are encoded per symbol. Both schemes use forward error correction with a code rate of 5/6 (ratio of data bits to total bits).
- MIMO: With 802.11n, there can be a maximum of 4 channel input and 4 channel output antennas. 802.11 ac increases this to 8×8.

802.11ac includes the option of multiuser MIMO (MU-MIMO). This means that on the downlink, the transmitter is able to use its antenna resources to transmit multiple frames to different stations, all at the same time and over the same simultaneously spectrum. This enables the AP to deliver significantly more data in many environments.

IEEE 802.11ad

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

However, at 60 GHz, 802.11ad is operating in the millimeter range, which has some undesirable propagation characteristics:

- Since free space loss increases with the square of the frequency, losses are much higher in this range.
- Multipath losses can be quite high.
- Millimeter wave signals generally don't penetrate solid objects.

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

Whereas 802.11ac supports a MIMO antenna configuration, 802.11ad is designed for single antenna operation. And 802.11ad has a huge channel bandwidth of 2160 MHz.

5.2.5 WEP

The IEEE 802.11 wireless LAN standard featured a security architecture called WEP (Wired Equivalent Privacy). WEP is intended to increase the level of difficulty of attacking wireless LANs such that it becomes comparable to the difficulty of attacking wired LANs. However, researchers discovered the weakness of WEP and tools that automate the cracking of WEP keys appeared on the web.

IEEE came up with a new security architecture for wireless LANs, described in an extension to the 802.11 standard, which is called IEEE 802.11i.

There are two basic security problems in wireless LANs:

- Due to the broadcast nature of radio communications, wireless transmissions ca be easily eavesdropped.
- Connecting to the network does not require physical access to the network Access Point (AP), thus any device can try to illegitimately use the services provided by the network.

WEP attempts to solve the first problem by encrypting messages. The second problem is addressed by requiring the authentication of the mobile stations (STAs) before allowing their connect to the network.

The authentication of the STA is based on a simple challenge-response protocol, similar to that used in GSM systems. Once authenticated, the

STA communicates with the AP by encrypted messages. The key used for encryption is the same as the one used for authentication. The encryption algorithm specified by WEP is based on the RC4 stream cipher.

One problem of the encryption is addressed in WEP: if the same key are used to encryption two messages M_1 and M_2, then it is easy to get $M_1 \oplus M_2$: which is equal to $(M_1 \oplus K) \oplus (M_2 \oplus K)$. Definitely this is a weak encryption, which is likely to be broken by the statistical method. Therefore, WEP uses an IV (Initialization Vector) to the secret key before initializing the RC4 algorithm, where the IV changes for every message. The receiver should also know the IV. Thus the IV is sent in clear together with the encrypted message.

The sender also attaches an integrity check value(ICV) to the clear message. The purpose of this value is to enable the receiver to detect any malicious modifications of the message by an attacker. In WEP, the ICV is a CRC (cyclic redundancy check) value computed for the clear message. The CRC value is also encrypted by the secret key.

The key distribution in WEP is as follows. The standard states that each STA has its own key, know only to that STA and the AP. This makes the key management on the AP's side complicated, because the AP must store a key for every STA. For this reason, most implementations do not actually support this option. The standard also specifies a default key, known to every STA and the AP. Originally, this key was intended to be used for the encryption of broadcast messages originated by the AP. But most WEP implementations support only this default key. Therefore, in many WLANs, there is only one single common key, which only can be used to protect the communications from an outside attacker.

Some security problems of WEP.

- Authentication problems.
  - Authentication is not mutual, meaning that the AP does not authenticate itself to the STA.
  - Authentication and encryption use the same secret key.
  - STA is authenticated only at the time when it tries to connect to the network.
  - WEP uses RC4 in the authentication protocol for encrypting the random challenge. The attacker can easily obtain the challenge C and the encrypted challenge $R = C \oplus K$ by overhearing the

exchange. Thus the attacker can easily obtain K which can be used to impersonate the STA later on. the IV mechanism of WEP does not mitigate this problem, because the value of IV is selected by the sender and the attacker can always use the same IV as R. It will make things worse, if any STA uses a same secret key.

- Integrity problems.

  The encrypted message in WEP can be written as $(M||CRC(M)) \oplus K$ where M is the message, K is the pseudo-random sequence produced by RC4, and $(CRC(M)$ is the ICV. CRC is linear with respect to the XOR operation, i.e., $CRC(X \oplus Y) = CRC(X) \oplus CRC(Y)$. Based on this, an attacker can manipulate a WEP message. Suppose the attacker knows $(M||CRC(M)) \oplus K$ and she wants to change the message to $M \oplus \Delta M$. Then she can just XOR the $\Delta M||CRC(\Delta M)$ to the original message. In fact,

$$\begin{aligned} & ((M||CRC(M)) \oplus K) \oplus (\Delta M||CRC(\Delta M)) \\ = \; & ((M \oplus \Delta M)||CRC(M) \oplus CRC(\Delta M))) \oplus K \\ = \; & ((M \oplus \Delta M)||CRC(M \oplus \Delta M)) \oplus K \end{aligned}$$

  Another related problem is WEP does not considered a replay attack.

- Confidentiality problems.

  In WEP, the IV is only 24 bits long. A WiFi device can transmit approximately 17 million possible 500 full-length frames in a second, thus the whole IV space is used up in a few hours. Once all the IV values have been used, they start to repeat. That means a same key sequence will be repeated.

  Another practical problem is that in many WEP implementation, the IV is initialized with 0 at startup, and then incremented by one after each message sent. So several devices may use the same IV value. If they use a same key, then the things are even worse.

  The weakness of RC4 cipher causes the serious problem of WEP. It is known that there exist weak RC4 keys which let RC4 produces an output that does not look random. Security experts suggest always throwing away the first 256 bytes of the RC4 output, but WEP did not adopt it. Also due to the ever changing IV value, a weak key can

be encountered sooner or later, and the attacker can easily know that a weak key is used, because the IV is transmitted in clear. Based on this, some cryptographers constructed a method that break the full 104-bit secret key by eavesdropping on only a few hundred thousands messages.

5.2.6 IEEE 802.11i

IEEE began to develop a new security architecture for WiFi networks described in the 802.11i specification. The new concept is called RSN (Robust Security Network). It includes a new method for authentication and access control, which is based on the model defined in the 802.1X standard. The AES is used instead of RC4. However, since the old devices only support RC4 and not AES, the 802.11i included an optional protocol which still uses the RC4 but fixes the flaws in WEP. This protocol is called TKIP (Temporal Key Integrity Protocol).

Manufacturers immediately adopted TKIP. They did not wait until the 802.11i architecture was finalized by the lengthy standardization procedure, but they issued their own specification, called WPA (WiFi Protected Access), based on TKIP. WPA is a specification supported by WiFi manufacturers, which contains a subset of RSN. WPA can run on old devices that support only the RC4 cipher. Authentication and access control, as well as key management, are the same in WPA and in RSN. The difference between the two concepts lies in the mechanisms used for integrity protection and confidentiality. RSN is also called WPA2 by many manufacturers.

Authentication and access control

The 802.1X model distinguishes three entities in the authentication procedure: the supplicant, the authenticator and the authentication server. The supplicant wants to access the network and wants to authenticate itself. The authenticator controls access to the network, that is represented by controlling the state of a port. The default state of the port is “closed”, which means that data traffic is disabled. The authenticator can “open” the port if this is authorized by the authentication server. So the supplicant tries to authenticate itself to the authentication server, and if this authentication is successful, then the authentication server grants access to the network by instructing the authenticator to open the port.

In the WiFi networks, the supplicant is the mobile device and the authenticator is the AP. The authentication server is a process that can run on the AP in the case of smaller network, or on a dedicated server machine in the case of larger networks. A port in WiFi is not a physical connector, but a logical control implemented in software running on the AP.

In 802.1X standard, a port can only connect to one supplicant. Since in the wireless case, once the STA authenticates itself and associates with the AP, someone else may try to steal its session by spoofing its MAC address. So 802.11i extends 802.1X with the requirement of setting up a session key between the STA and the AP.

The authentication procedure in 802.11i uses EAP (Extensible authentication Protocol, RFC 3748) which has four message types: request, response, success and failure. EAP is only a carrier protocol at data link layer. EAP request and response messages are used to carry the messages of embedded authentication protocol from the STA to the server, and from the server to the STA, respectively. The EAP success and failure messages are used to signal the result of the authentication to the supplicant. The embedded authentication protocol can be higher layer protocols such as the EAP-TLS Handshake and GSM authentication protocols.

The EAP protocol and the embedded authentication protocol are executed by the mobile device and the authentication server. The AP relays messages without interpreting them. The AP only understand the success and failure messages.

EAP messages between the mobile device and the AP are carried by the EAPOL (EAP over LAN) protocol (802.1X). EAP messages between the AP and the authentication server can be carried by various protocols. WPA mandates the use of RADIUS (Remote Authentication Dial In User Service, RFC 3579), but RSN just specifies RADIUS as an option.

The authentication process in WiFi also establishes a session key to protect further communication between the mobile device and the AP. However, as authentication takes place between the mobile device and the authentication server, the session key is also established between them. The RADIUS uses MS-MPPE-Recv-Key RADIUS attribute, which has been specified for key transfer purpose, to securely transfer the session key to the AP. The session key is transferred in encrypted form, where the encryption uses a long-term key shared by the AP and the authentication server.

Key management

The session key established between the mobile and the AP is called the pairwise master key (PMK) which is not used directly for encryption or integrity protection. Both the mobile device and the AP derived four keys from the PMK: a data-encryption key, a data-integrity key, a key-encryption key and a key-integrity key. These four keys together are called the pairwise transient key (PTK). When AES-CCMP is used, the data encryption and integrity use the same key. In this case, PTK contains only three keys. The derivation of the PTK uses PMK together with the MAC addresses of the parties and two random numbers generated by the parties.

The mobile device and PA use four-way handshake protocol to exchange the random numbers, which are carried by the EAPOL protocol in EAPOL message of type Key.

1. The AP sends its random number to the mobile device. With this number, the mobile device can computes the PTK.

2. The mobile device sends its random number to the AP. This message also carries a Message Integrity Code (MIC), computed by the mobile device using the key-integrity. Upon reception of this message, the AP can compute the PTK and then verify th MIC.

3. The AP sends a message that contains a MIC to the mobile device. The MIC is computed using the key-integrity key of the PTK. This message contains the starting value of a sequence number that will be used to number further data packets, and to detect replay attacks. This message signals the mobile device that the AP has installed the keys and will encrypt subsequent data packets.

4. The mobile device acknowledges the reception of the third message.

For the purpose of broadcast messages, AP generates additional key material called the group transient key (GTK). The GTK contains a group encryption key and a group integrity key and it is sent to each mobile device separately encrypted with the key-encryption key of the given mobile device.

TKIP

AES-CCMP (AES CTR mode and CBC MAC Protocol) need new hardware that supports the AES algorithm. TKIP still uses RC4 but tries to fix flaws

in WEP.

- TKIP uses a new integrity protection mechanism, called Michael. TKIP also uses the IV as a sequence number.
- In TKIP, the IV size is increased from 24 bits to 48 bits and each message is encrypted with a different key. The message keys are generated from the data-encryption key of the PTK. The 48-bit IV is divided into a 32-bit upper part and a 16-bit lower part. The upper part of the IV is combined with the 128-bit data-encryption key and the MAC address of the device. The result of this computation is then combined with the lower part of the IV in order to obtain the 104-bit message key. The RC4 seed value for TKIP is obtained by concatenating the message-key to the lower part of the IV and a dummy byte.

5.3 Bluetooth

Bluetooth is a wireless technology that uses short-range digital radio communications and offers fast and reliable transmission of both voice and data. Bluetooth is defined in the IEEE 802.15.1 standard.

Now Bluetooth is managed by the Bluetooth Special Interest Group (SIG), which has more than 25,000 member companies in the areas of telecommunication, computing, networking, and consumer electronics. The IEEE no longer maintains the standard. The Bluetooth SIG oversees development of the specification, manages the qualification program, and protects the trademarks.

It is essentially a low-power, short-range, low-rate "cable replacement" technology for interconnection devices. Bluetooth incorporates a radio frequency transceiver and a full set of networking protocols on a single chip that is small enough to be included in cellular and cordless phone, portable PCs, headsets, etc. Sometimes, 802.15.1 networks are referred as wireless personal area networks (WPANs).

802.15.1 networks operate in the 2.4GHz unlicensed radio band in a TDM manner, with time slots of 625 microseconds. During each time slot, a sender transmits on one of 79 channels, with the channel changing in a known but pseudo random manner from slot to slot. This form of channel hopping, known as frequency-hopping spread spectrum (FHSS), spreads transmissions in time over the frequency spectrum. It can provide data rate up to 4 Mbps.

Bluetooth protocol stack includes:

- Bluetooth Radio layer: This is the lowest defined layer of the bluetooth specification. It is not a protocol, but defines the requirements and operations of the bluetooth transceiver device, transmitting and receiving radio frequency signals in the 2.4GHz ISM band.

- Baseband: This is the physical layer protocol of the bluetooth specification and it lies on top of the bluetooth radio layer.

- Link Manager Protocol(LMP): LMP performs the link setup, configuration and authentication process within the bluetooth stack. After the Link Manager(LM) of one device discovers the LM of another device, the LMP then communicates with the remote LM to establish a link.

- Host Controller Interface (HCI): HCI provides a commend interface between the baseband control and the LM.

- L2CAP (Logical link control and adaptation protocol): The L2CAP resides on data-link layer (OSI model) and provides the link functions for the baseband protocol.

- RFCOMM (Radio frequency communication): is the cable replacement protocol in the bluetooth stack. It creates a virtual seral port through which RF communications can be passed using standard EIA/TIA (Electronics Industries Association/Telecommunications Industry Association) 232 standard, which is the serial port communication standard.

- Object Exchange Protocol (OBEX): Originally specified by the Infrared Data Association (IrDA), OBEX is used to transfer data, graphics and voice objects between devices. OBEX is used in serval devices, such as PDAs (personal digital assistants), mobile phones and computer systems.

- vCard/vcal: A protocol used to store and transfer virtual business cards and personal calenders on mobile devices.

- TCP (Telephony control protocol): Used to set up and control speech and data calls between Bluetooth devices. The protocol is based on

the ITU-T standard Q.931, with the provisions of Annex D applied, making only the minimum changes necessary for Bluetooth.

TCP is used by the intercom (ICP) and cordless telephony (CTP) profiles.

- AT command set: It includes the control commands used to control dial-up modem functions and actions.
- Telephone Control Specification–Binary(TCS BIN): It is a bit-oriented protocol used to establish voice and data links between bluetooth devices.
- Service Discovery Protocol (SDP): Bluetooth requires an SDP to identify and manage the services available on portable devices moving into and out of range with each other. The Bluetooth SDP is specifically designed for bluetooth devices.
- Synchronous connection-oriented (SCO)link: The type of radio link used for voice data. An SCO link is a set of reserved timeslots on an existing ACL (Asynchronous Connection-Less) link. Each device transmits encoded voice data in the reserved timeslot. There are no retransmissions, but forward error correction can be optionally applied. SCO packets may be sent every 1, 2 or 3 timeslots.

  Enhanced SCO (eSCO) links allow greater flexibility in setting up links: they may use retransmissions to achieve reliability, allow a wider variety of packet types, and greater intervals between packets than SCO, thus increasing radio availability for other link.

Unlike WLAN, in the case of bluetooth, communications are between wireless stations (no APs). The operation of bluetooth networks (called piconets) is based on the master-slave principle, where one station is the master and other stations (up to 7) become the slaves. Bluetooth technology can also provide a link to a wired network in a similar way that 802.11 access point do, by installing a bluetooth access point that is connected to a wired network.

An ad-hoc bluetooth network piconet can include up to 8 bluetooth devices. When more than 8 device are attempting to associate with a piconet, the piconet is divided into two or more piconets, and then interconnected into what is called a scatternet.

A bluetooth device cannot act as a master for two piconet. If a piconet master device is also linked to another piconet in a scatternet, it must participate as a master device in one piconet and a slave device in a second piconet. The slaves that share the same master device belong to the same pibonet and to remain as a member of a piconet, each slave must interact with the master on a time interval negotiated between the master and the salve. If the master leaves the piconet, the other devices must elect a new master and the piconet is reformed. When a bluetooth device is a member of two piconets, it can be used as a link between the piconets.

Any bluetooth device is capable of serving as a master or a slave, depending on the networking situation it encounters on-the-fly. A piconet can be formed using one of the following methods:

- A master device actively scans for slave devices and, when it detects one in its range, it can invite the device to join a piconet as a slave.
- A master device can passively wait for a slave to contact it, and then invite the slave to join the piconet as a slave.

Bluetooth devices each have a unique clock signal and device address, which are combined to provide a unique identity. The difference or offset between one device's clock signal and the clock signal of another device is the basis for the FHSS (frequency-hopping spread-spectrum) sequence used between the devices to transmit data. Bluetooth device use frequency hopping in order to avoid interference with other devices that operate in the same unlicensed ISM (Industrial Scientific and Medical) band. The frequency-hopping scheme uses 79 different channels and changes frequency 1600 times per second in a pseudo-random matter. This makes eavesdropping slightly more difficult. Bluetooth is a short range radio technology enabling communications over a few meters only (mostly in class 2 that is for 10 meters). That means that an attacker must be physically close to the victims in order to eavesdrop the communications, which also reduces the likelihood of attacks.

5.3.1 Blutooth security

The bluetooth security architecture is concerned with the establishment of a secured wireless link between two bluetooth devices. This involves the authentication of the devices each other and setting up a confidential channel

between them. Cryptographic functions used here are E_1, E_{21}, E_{22} and E_3, which are based on the SAFER+ block cipher.

There are two ways to establish a link key. The first method is used when one of the devices has memory limitations and can store only one key, otherwise the second method is used. Both methods start by setting up a temporary initialization key K_{init}. First, one device selects a random number IN_RAND and sends it to the other device. Then both devices compute K_{init} as a function of IN_RAND, a share PIN and the length L of the PIN. The length of PIN can be vary between 1 and 16 bytes. The PIN can be shared between the devices in several ways. If both devices have some input facility, then the user can choose a random PIN and enter it into both devices. If only one device has input facility, then the user can enter the pre-configured PIN of the other device into the first device.

Now suppose one device, A, has memory limitation. A sends its long-term unit key K_A to the other device B encrypted with the key K_{init}. B decrypts the cipher text and K_A become the link key.

When none of the devices has memory limitation, both A and B choose a random number $RAND_A$ and $RAND_B$, respectively. A computes LK_K_A as a function of $RAND_A$ and its unique device address BD_ADDR_A. B does the similar computation. Then they exchange $RAND_A$ and $RAND_B$ encrypted with K_{init}. So A can get LK_K_B and uses $LK\_K_A \oplus LK\_K_B$ as the link key. B can also compute the link key.

After two devices share a link key, they authenticate each other using a simple challenge-response protocol as follows. One of the device, referred as the "verifier", generates a random number AU_RAND and sends it to the other device called " claimant". Both of them compute an authentication response $SRES$ from AU_RAND, BD_ADDR of the claimant, and the link key Key_{link} by security function E_1. The procedure also generate ACO (authentication cipher offset). The claimant sends the value $SRES$ to the verifier who then check its correctness. If the protocol above fails, then the verifier device will wait some time before a new attempt can be made. This makes it impractical for an attacker to defeat authentication by trying different keys in rapid succession.

The encryption key K_{enc} is computed by both devices as a function of three elements: link key K_{link}, the authentication cipher offset ACO generated during the authentication protocol, and a random number EN_RAND generated by the master device. A stream cipher E_0 is used for encryption. Besides the encryption key, E_0 also inputs the address BD_ADDR_{amster} of

the master device, and the clock value $CLOCK_{master}$ of the master.

Some weakness of bluetooth:

- The strength of the system is based on PIN which is typically a 4-digit number. An attacker can eavesdrop the communication, then try 10000 possible values off-line.
- For a memory-constrained device, the link key is the long-term unit key. An attacker can obtain the unit key by establishing a link key with it.
- There is also a privacy problem that stems from the use of fixed and unique device addresses. An attacker can track the whereabout of the person by tracking the use of the given device address.
- The encryption algorithm E_0 has some weakness.

5.4 Mobility management

Now we consider the situation about a mobile user how to maintain ongoing connections while moving between notworks. In the case of mobile, the mobile node (such as a smartphone or a laptop) needs a "permanent home address" known as home network, and an entity within the home network that performs the mobility management functions on behalf of the mobile node known as the home agent. The network in which the mobile node is currently residing is known as the foreign (or visited) network, and the entity within in the foreign network that helps the mobile node with the mobility management functions is known as a foreign agent. A correspondent is the entity wishing to communicate with the mobile node.

5.4.1 Mobile IP

One role of the foreign agent is to create a so called care-of address (COA) for the mobile node, with the network portion of the COA matching that of the foreign network. So there are two addresses associated with a mobile node, its permanent address and its COA, sometimes known as a foreign address. Although we have separated the functionality of the mobile node and the foreign agent, it is worth to note that the mobile node can also assume the responsibility of the foreign agent.

The Internet architecture and protocols for supporting mobility, collectively known as mobile IP, are defined primarily in RFC 5944 for IPv4. Mobile IP is a flexible standard, supporting many different modes of operation, multiple ways for agents and mobile nodes to discover each other, use of single or multiple COAs, and multiple forms of encapsulation.

The mobile IP standard consists of three main pieces:

- Agent discovery.
- Registration with the home agent.
- Indirect routing of datagrams.

Security considerations are prominent through the mobile IP standard. Authentication of a mobile node is one of the most important problems for mobile IP. Here we will address issues other than these security issues.

Agent discovery

Agent discovery can be accomplished in one of two ways: via agent advertisement or via agent solicitation.

With agent advertisement, a foreign or home agent advertises its services using an extension to the existing router discovery protocol. The agent periodically broadcast an ICMP message with a type of 9 (router discovery) on all links to which it is connected. The router discovery message contains the IP address of the router (the agent), thus allowing a mobile node to learn the agent's IP address. The router discovery message also contains a mobility agent advertisement extension that contains additional information needed by the mobile node. The format of the ICMP router discovery message with mobility agent advertisement extension is as in Figure 5.6.

In Figure 5.6, the upper part contains the standard ICMP fields and the lower part is the mobility agent advertisement extension. Some of the fields are explained below.

- Registration required bit (R): Registration with this foreign agent (or another foreign agent on this link) is required even when using a co-located care-of address.
- Busy bit (B): The foreign agent will not accept registrations from additional mobile nodes.

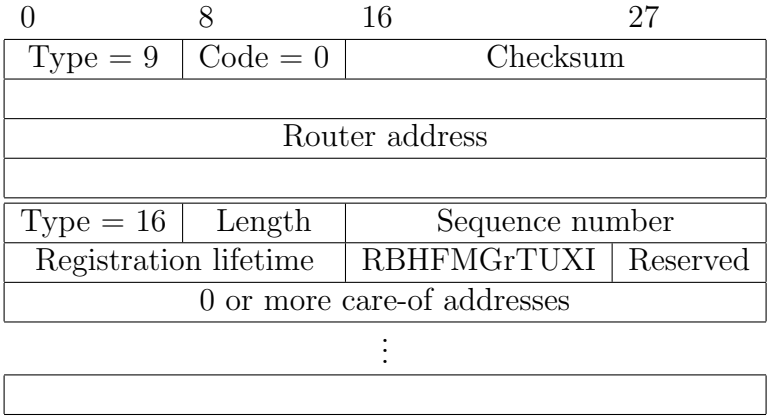

Figure 5.6: Discovery message

- Home agent bit (H): Indicates that the agent is a home agent for the network in which it resides.

- Foreign agent bit (F): Indicates that the agent is a foreign agent for the network in which it resides.

- Registration required bit (R): Indicates that a mobile user in this network must register with a foreign agent. In particular, a mobile user cannot obtain a care-of address in the foreign network (for example, using DHCP) and assume the functionality of the foreign agent for itself, without registering with the foreign agent.

- M, G encapsulation bits: Indicate whether a form of encapsulation other than IP-in-IP encapsulation will be used.

- Care-of address fields: A list of one or more care-of addresses provided by the foreign agent.

For agent solicitation, a mobile node wanting to learn about agents without waiting to receive an agent advertisement can broadcast an agent solicitation message, which is simply an ICMP message with type value 10. An agent receiving the solicitation will unicast an agent advertisement directly to the mobile node, which can then proceed as if it had received an unsolicited advertisement.

Registration with the home agent

Once a mobile IP node received a COA, that address must be registered with the home agent. This can be done either via the foreign agent (who then registers the COA with the home agent) or directly by the mobile IP node itself.

Registering by the foreign agent is as follows.

1. By receipting a foreign agent advertisement, a mobile node sends a mobile IP registration message to the foreign agent. The registration message is carried within a UDP datagram and sent to port 434. The registration message carries a COA advertised by the foreign agent, the address of the home agent (HA), the permanent address of the mobile node (MA), the requested lifetime of the registration, and a 64-bit registration identification. The registration identifier acts like a sequence number and serves to match a received registration reply with a registration request.

2. The foreign agent receives the registration message and records the mobile node's permanent IP address. The foreign agent now knows that it should be looking for datagram containing an encapsulated datagram whose destination address matches the permanent address of the mobile node. The foreign agent then sends a mobile IP registration message to port 434 of the home agent. The message contains the COA, HA, MA, encapsulation format requested requested registration lifetime, and registration identification.

3. The home agent receives the registration request and checks for authenticity and correctness. The home agent binds the mobile node's permanent IP address with the COA. The home agent sends a mobile IP registration reply containing the HA, MA, actual registration lifetime, and the registration identification of the request that is being satisfied with this reply.

4. The foreign agent receives the registration reply and then forwards it to the mobile node.

Indirect routing of datagrams

Let us see how mobile nodes, home agents, and (possibly) foreign agents cooperate to route datagrams to or from mobile nodes that are connected to a foreign network.

The mobile node informs its home agent of its current location using the registration procedure described above. Home agents and foreign agents will support tunneling datagrams using IP-in-IP encapsulation. Any mobile node that uses a care-of address will support receiving datagrams tunneled using IP-in-IP encapsulation.

When connected to its home network, a mobile node operates without the support of mobility services. That is, it operates in the same way as any other (fixed) host or router. ICMP Router Advertisement is one such method as we discussed before.

When registered on a foreign network, the mobile node chooses a default router.

Upon receipt of an encapsulated datagram sent to its advertised care-of address, a foreign agent compares the inner Destination Address to those entries in its visitor list. When the Destination does not match the address of any mobile node currently in the visitor list, the foreign agent will not forward the datagram. Otherwise, the foreign agent forwards the decapsulated datagram to the mobile node.

The home agent is able to intercept any datagrams on the home network addressed to the mobile node while the mobile node is registered away from home.

The home agent must examine the IP Destination Address of all arriving datagrams to see if it is equal to the home address of any of its mobile nodes registered away from home. If so, the home agent tunnels the datagram to the mobile node's currently registered care- of address or addresses.

The mobile IP standard allows many additinal scenarios and capabilities in addition to the above description. RFC 5944 contains more than 100 pages. Mobile IPv6 is defined in RFC 6275, which we omit here.

5.5 Mobile Sensor Networks

In this section, we consider wireless sensor networks (WSN) which usually integrate a large number of low-power, low-cost sensor nodes. They are

largely deployed to monitor a specific environment.

There are many different applications of WSNs:

- Military applications: Wireless sensor networks be likely an integral part of military command, control, communications, computing, intelligence, battlefield surveillance, reconnaissance and targeting systems.
- Area monitoring: In area monitoring, the sensor nodes are deployed over a region where some phenomenon is to be monitored. When the sensors detect the event being monitored (heat, pressure etc), the event is reported to one of the base stations, which then takes appropriate action.
- Transportation: Real-time traffic information is being collected by WSNs to later feed transportation models and alert drivers of congestion and traffic problems.
- Health applications: Some of the health applications for sensor networks are supporting interfaces for the disabled, integrated patient monitoring, diagnostics, and drug administration in hospitals, telemonitoring of human physiological data, and tracking and monitoring doctors or patients inside a hospital.
- Environmental sensing: The term Environmental Sensor Networks has developed to cover many applications of WSNs to earth science research. This includes sensing volcanoes, oceans, glaciers, forests etc.
- Structural monitoring: Wireless sensors can be utilized to monitor the movement within buildings and infrastructure such as bridges, flyovers, embankments, tunnels etc enabling Engineering practices to monitor assets remotely with out the need for costly site visits.
- Industrial monitoring: Wireless sensor networks have been developed for machinery condition-based maintenance (CBM) as they offer significant cost savings and enable new functionality. In wired systems, the installation of enough sensors is often limited by the cost of wiring.
- Agricultural sector: using a wireless network frees the farmer from the maintenance of wiring in a difficult environment. Irrigation automation enables more efficient water use and reduces waste.

Sensor networks often have one or more points of centralized control called base stations. sensor nodes are small in size and able to sense, process data and communicate with each other, typically over an RF (radio frequency) channel. In general, there are three categories of traffic:

- Many-to-one traffic.
- One-to-many traffic.
- Local traffic: The nodes in a limited area send localized messages to discover the neighbouring nodes and coordinate with each other. May be broadcast or send messages intended for a single neighbour.

Some factors of WSNs which make more difficulties for the security of the communication.

- Sensor nodes have limited storage, computation and power resources.
- The WSN does not have a fixed infrastructure and does not have a static topology.
- The sensing and communication tasks are often performed in a hostile environment where the gathered events are subjected to numerous threats.
- The detected events are forwarded through the sensor nodes themselves.

5.5.1 WSN Features

The basic features of WSNs:

- Self-organizing capabilities. The WSNs are able to cope with topology variability and infrastructure variations.
- Short-range broadcast communication and multirouting. The sensor nodes have reduced radio ranges and should cooperate to achieve complete routing of information.
- Dense deployment and cooperative effort of sensor nodes. The shortage of the radio range and the need to have efficient sensing call for a dense deployment of sensors.

- Limitations of energy, transmit power, memory and computing power. WSNs cope with limitation of resources and frequent changes of topology due to fading and node failures.

Some of the challenges for WSNs:

- Extension of lifetime. A typical alkaline battery, for example, provides about 50 watt-hours of energy. Given the expense and the potential infeasibility of monitoring and replacement of batteries for a large WSN, significantly longer lifetimes would be desired.

- Responsiveness. A simple solution to extending network lifetime is to operate the nodes in a duty-cycled manner with periodic switching between sleep and wake-up modes. This causes the time synchronizing requirement, and the responsiveness of and the effectiveness of the sensors.

- Robustness. The use of large number of inexpensive devices characterizes the WSNs. It is important to ensure that the global performance of the system is not sensitive to individual device failure. It is also often desirable that the performance of the system degrade as gracefully as possible with respect to component failure.

- Synergy. Design synergistic protocol, which ensures that the system as a whole is more capable than the sum of the capabilities of its individual component. The protocols must provide an efficient collaborative use of storage, computation and communication resources.

- Self-configuration. WSNs are inherently unattended distributed systems. Autonomous operation of the network is therefore a key design. Nodes in a wireless sensor network have to be able to configure their own network topology, synchronize, and calibrate themselves; coordinate inter-node communication; and determine other important operating parameters.

- Privacy and security. The large scale, prevalence, and sensitivity of the information collected by WSN (as well as their potential deployment in hostile locations) give rise to the final key challenge of ensuring both privacy and security.

5.5.2 Structure of WSN

Structure of a Wireless Sensor Network includes different topologies for radio communications networks. A short discussion of the network topologies that apply to wireless sensor networks are outlined below:

- Star network (single point-to-multipoint): A star network is a communications topology where a single base station can send and/or receive a message to a number of remote nodes. The remote nodes are not permitted to send messages to each other. The advantage of this type of network for wireless sensor networks includes simplicity, ability to keep the remote nodes power consumption to a minimum. It also allows low latency communications between the remote node and the base station. The disadvantage of such a network is that the base station must be within radio transmission range of all the individual nodes and is not as robust as other networks due to its dependency on a single node to manage the network.

- Mesh network: A mesh network allows transmitting data to one node to other node in the network that is within its radio transmission range. This allows for what is known as multi-hop communications, that is, if a node wants to send a message to another node that is out of radio communications range, it can use an intermediate node to forward the message to the desired node. This network topology has the advantage of redundancy and scalability. If an individual node fails, a remote node still can communicate to any other node in its range, which in turn, can forward the message to the desired location. In addition, the range of the network is not necessarily limited by the range in between single nodes; it can simply be extended by adding more nodes to the system. The disadvantage of this type of network is in power consumption for the nodes that implement the multi-hop communications are generally higher than for the nodes that dont have this capability, often limiting the battery life. Additionally, as the number of communication hops to a destination increases, the time to deliver the message also increases, especially if low power operation of the nodes is a requirement.

- Hybrid star Mesh network: A hybrid between the star and mesh network provides a robust and versatile communications network, while maintaining the ability to keep the wireless sensor nodes power consumption to a minimum. In this network topology, the sensor nodes

with lowest power are not enabled with the ability to forward messages. This allows for minimal power consumption to be maintained. However, other nodes on the network are enabled with multi-hop capability, allowing them to forward messages from the low power nodes to other nodes on the network. Generally, the nodes with the multi-hop capability are higher power, and if possible, are often plugged into the electrical mains line. This is the topology implemented by the up and coming mesh networking standard known as ZigBee.

In WSN, the main task of a sensor node is to sense data and sends it to the base station in multi hop environment for which routing path is essential. For computing the routing path from the source node to the base station there is huge numbers of proposed routing protocols exist. The design of routing protocols for WSNs must consider the power and resource limitations of the network nodes, the time-varying quality of the wireless channel, and the possibility for packet loss and delay. To address these design requirements, several routing strategies for WSNs have been proposed.

The first class of routing protocols adopts a flat network architecture in which all nodes are considered peers. Flat network architecture has several advantages, including minimal overhead to maintain the infrastructure and the potential for the discovery of multiple routes between communicating nodes for fault tolerance.

A second class of routing protocols imposes a structure on the network to achieve energy efficiency, stability, and scalability. In this class of protocols, network nodes are organized in clusters in which a node with higher residual energy, for example, assumes the role of a cluster head. The cluster head is responsible for coordinating activities within the cluster and forwarding information between clusters. Clustering has potential to reduce energy consumption and extend the lifetime of the network.

A third class of routing protocols uses a data-centric approach to disseminate interest within the network. The approach uses attribute-based naming, whereby a source node queries an attribute for the phenomenon rather than an individual sensor node. The interest dissemination is achieved by assigning tasks to sensor nodes and expressing queries to relative to specific attributes. Different strategies can be used to communicate interests to the sensor nodes, including broadcasting, attribute-based multicasting, geo-casting, and any casting.

A fourth class of routing protocols uses location to address a sensor node. Location-based routing is useful in applications where the position of the node within the geographical coverage of the network is relevant to the query issued by the source node. Such a query may specify a specific area where a phenomenon of interest may occur or the vicinity to a specific point in the network environment.

Routing protocols

Some of the major routing protocols and algorithms to deal with the energy conservation issue in the literatures are as follows.

- Flooding: Flooding is a common technique frequently used for path discovery and information dissemination in wired and wireless ad hoc networks. The routing strategy of flooding is simple and does not rely on costly network topology maintenance and complex route discovery algorithms. Flooding uses a reactive approach whereby each node receiving a data or control packet sends the packet to all its neighbors. After transmission, a packet follows all possible paths. Unless the network is disconnected, the packet will eventually reach its destination. Furthermore, as the network topology changes, the packet transmitted follows the new routes.

- Gossiping: To address the shortcomings of flooding, a derivative approach, referred to as gossiping, has been proposed. Similar to flooding, gossiping uses a simple forwarding rule and does not require costly topology maintenance or complex route discovery algorithms. Contrary to flooding, where a data packet is broadcast to all neighbors, gossiping requires that each node sends the incoming packet to a randomly selected neighbor. Upon receiving the packet, the neighbor selected randomly chooses one of its own neighbors and forwards the packet to the neighbor chosen. This process continues iteratively until the packet reaches its intended destination or the maximum hop count is exceeded.

- Protocols for Information via Negotiation (SPIN): Sensor protocols for information via negotiation (SPIN), is a data-centric negotiation-based family of information dissemination protocols for WSNs. The main objective of these protocols is to efficiently disseminate observations

gathered by individual sensor nodes to all the sensor nodes in the network. Simple protocols such as flooding and gossiping are commonly proposed to achieve information dissemination in WSNs. Flooding requires that each node sends a copy of the data packet to all its neighbors until the information reaches all nodes in the network. Gossiping, on the other hand, uses randomization to reduce the number of duplicate packets and requires only that a node receiving a data packet forward it to a randomly selected neighbor.

- Low-Energy Adaptive Clustering Hierarchy (LEACH) Low-energy adaptive clustering hierarchy (LEACH) is a routing algorithm designed to collect and deliver data to the data sink, typically a base station. LEACH adopts a hierarchical approach to organize the network into a set of clusters. Each cluster is managed by a selected cluster head. The cluster head assumes the responsibility to carry out multiple tasks. The first task consists of periodic collection of data from the members of the cluster. Upon gathering the data, the cluster head aggregates it in an effort to remove redundancy among correlated values. The second main task of a cluster head is to transmit the aggregated data directly to the base station over single hop. The third main task of the cluster head is to create a TDMA-based schedule whereby each node of the cluster is assigned a time slot that it can use for transmission. The cluster head announces the schedule to its cluster members through broadcasting. To reduce the likelihood of collisions among sensors within and outside the cluster, LEACH nodes use a code-division multiple accessbased scheme for communication. The basic operations of LEACH are organized in two distinct phases. The first phase, the setup phase, consists of two steps, cluster-head selection and cluster formation. The second phase, the steady-state phase, focuses on data collection, aggregation, and delivery to the base station. The duration of the setup is assumed to be relatively shorter than the steadystate phase to minimize the protocol overhead.

- Two hierarchical routing protocols called TEEN (Threshold-sensitive Energy Efficient sensor Network protocol), and APTEEN (Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network protocol): These protocols were proposed for time-critical applications. In TEEN, sensor nodes sense the medium continuously, but the data transmission

is done less frequently. A cluster head sensor sends its members a hard threshold, which is the threshold value of the sensed attribute and a soft threshold, which is a small change in the value of the sensed attribute that triggers the node to switch on its transmitter and transmit. Thus the hard threshold tries to reduce the number of transmissions by allowing the nodes to transmit only when the sensed attribute is in the range of interest. The soft threshold further reduces the number of transmissions that might have otherwise occurred when there is little or no change in the sensed attribute. A smaller value of the soft threshold gives a more accurate picture of the network, at the expense of increased energy consumption. Thus, the user can control the trade-off between energy efficiency and data accuracy. When cluster-heads are to change, new values for the above parameters are broadcast. The main drawback of this scheme is that, if the thresholds are not received, the nodes will never communicate, and the user will not get any data from the network at all.

- Power-Efficient Gathering in Sensor Information Systems (PEGASIS): Power-efficient gathering in sensor information systems (PEGASIS) and its extension, hierarchical PEGASIS, are a family of routing and information-gathering protocols for WSNs. The main objectives of PEGASIS are twofold. First, the protocol aims at extending the lifetime of a network by achieving a high level of energy efficiency and uniform energy consumption across all network nodes. Second, the protocol strives to reduce the delay that data incur on their way to the sink. The network model considered by PEGASIS assumes a homogeneous set of nodes deployed across a geographical area. Nodes are assumed to have global knowledge about other sensors positions. Furthermore, they have the ability to control their power to cover arbitrary ranges. The nodes may also be equipped with CDMA-capable radio transceivers. The nodes responsibility is to gather and deliver data to a sink, typically a wireless base station. The goal is to develop a routing structure and an aggregation scheme to reduce energy consumption and deliver the aggregated data to the base station with minimal delay while balancing energy consumption among the sensor nodes. Contrary to other protocols, which rely on a tree structure or a cluster-based hierarchical organization of the network for data gathering and dissemination, PEGASIS uses a chain structure.

- Directed Diffusion: Directed diffusion is a data-centric routing protocol for information gathering and dissemination in WSNs. The main objective of the protocol is to achieve substantial energy savings in order to extend the lifetime of the network. To achieve this objective, directed diffusion keeps interactions between nodes, in terms of message exchanges, localized within limited network vicinity. Using localized interaction, direct diffusion can still realize robust multi-path delivery and adapt to a minimal subset of network paths. This unique feature of the protocol, combined with the ability of the nodes to aggregate response to queries, results into significant energy savings. The main elements of direct diffusion include interests, data messages, gradients, and reinforcements. Directed diffusion uses a publish-and-subscribe information model in which an inquirer expresses an interest using attributevalue pairs. An interest can be viewed as a query or an interrogation that specifies what the inquirer wants.

- Geographic Adaptive Fidelity (GAF): GAF is an energy-aware location-based routing algorithm designed mainly for mobile ad hoc networks, but may be applicable to sensor networks as well. The network area is first divided into fixed zones and forms a virtual grid. Inside each zone, nodes collaborate with each other to play different roles. For example, nodes will elect one sensor node to stay awake for a certain period of time and then they go to sleep. This node is responsible for monitoring and reporting data to the BS on behalf of the nodes in the zone. Hence, GAF conserves energy by turning off unnecessary nodes in the network without affecting the level of routing fidelity.

5.5.3 WSN security requirements

Data confidentiality

Data confidentiality is the most important issue in network security. Every network with any security focus will typically address this problem first. In sensor networks, the confidentiality relates to the following:

- A sensor network should not leak sensor readings to its neighbours. Especially in a military application, the data stored in the sensor node may be highly sensitive.

- In many applications nodes communicate highly sensitive data, e.g., key distribution, therefore it is extremely important to build a secure channel in a wireless sensor network.
- Public sensor information, such as sensor identities and public keys, should also be encrypted to some extent to protect against traffic analysis attacks.

The standard approach for keeping sensitive data secret is to encrypt the data with a secret key that only intended receivers possess, thus achieving confidentiality.

Data integrity

With the implementation of confidentiality, an adversary may be unable to steal information. However, this does not mean the data is safe. The adversary can change the data, so as to send the sensor network into disarray. For example, a malicious node may add some fragments or manipulate the data within a packet. This new packet can then be sent to the original receiver. Data loss or damage can even occur without the presence of a malicious node due to the harsh communication environment. Thus, data integrity ensures that any received data has not been altered in transit.

Data freshness

Even if confidentiality and data integrity are assured, we also need to ensure the freshness of each message. Informally, data freshness suggests that the data is recent, and it ensures that no old messages have been replayed. This requirement is especially important when there are shared-key strategies employed in the design. Typically shared keys need to be changed over time. However, it takes time for new shared keys to be propagated to the entire network. In this case, it is easy for the adversary to use a replay attack. Also, it is easy to disrupt the normal work of the sensor, if the sensor is unaware of the new key change time. To solve this problem a nonce, or another time-related counter, can be added into the packet to ensure data freshness.

Availability

Adjusting the traditional encryption algorithms to fit within the wireless sensor network is not free, and will introduce some extra costs. Some ap-

proaches choose to modify the code to reuse as much code as possible. Some approaches try to make use of additional communication to achieve the same goal. What is more, some approaches force strict limitations on the data access, or propose an unsuitable scheme (such as a central point scheme) in order to simplify the algorithm. But all these approaches weaken the availability of a sensor and sensor network for the following reasons:

- Additional computation consumes additional energy. If no more energy exists, the data will no longer be available.

- Additional communication also consumes more energy. What is more, as communication increases so too does the chance of incurring a communication conflict.

- A single point failure will be introduced if using the central point scheme. This greatly threatens the availability of the network.

The requirement of security not only affects the operation of the network, but also is highly important in maintaining the availability of the whole network.

Self-organization

A wireless sensor network is a typically an ad hoc network, which requires every sensor node be independent and flexible enough to be self-organizing and self-healing according to different situations. There is no fixed infrastructure available for the purpose of network management in a sensor network. This inherent feature brings a great challenge to wireless sensor network security as well. For example, the dynamics of the whole network inhibits the idea of pre-installation of a shared key between the base station and all sensors. Several random key predistribution schemes have been proposed in the context of symmetric encryption techniques. In the context of applying public-key cryptography techniques in sensor networks, an efficient mechanism for public-key distribution is necessary as well. In the same way that distributed sensor networks must self-organize to support multi-hop routing, they must also self-organize to conduct key management and building trust relation among sensors. If self-organization is lacking in a sensor network, the damage resulting from an attack or even the hazardous environment may be devastating.

Time synchronization

Most sensor network applications rely on some form of time synchronization. In order to conserve power, an individual sensor's radio may be turned off for periods of time. Furthermore, sensors may wish to compute the end-to-end delay of a packet as it travels between two pairwise sensors. A more collaborative sensor network may require group synchronization for tracking applications, etc. researchers propose a set of secure synchronization protocols for sender-receiver (pairwise), multihop sender-receiver (for use when the pair of nodes are not within single-hop range), and group synchronization.

Secure localization

Often, the utility of a sensor network will rely on its ability to accurately and automatically locate each sensor in the network. A sensor network designed to locate faults will need accurate location information in order to pinpoint the location of a fault. Unfortunately, an attacker can easily manipulate nonsecured location information by reporting false signal strengths, replaying signals, etc. A technique called verifiable multilateration (VM) is proposed. In multilateration, a device's position is accurately computed from a series of known reference points. Authenticated ranging and distance bounding are used to ensure accurate location of a node. Because of distance bounding, an attacking node can only increase its claimed distance from a reference point. However, to ensure location consistency, an attacking node would also have to prove that its distance from another reference point is shorter. Since it cannot do this, a node manipulating the localization protocol can be found. For large sensor networks, the SPINE (Secure Positioning for sensor NEtworks) algorithm is used. It is a three phase algorithm based upon verifiable multilateration.

A SeRLoc (Secure Range-Independent Localization) can be described as follows. Its novelty is its decentralized, range-independent nature. SeRLoc uses locators that transmit beacon information. It is assumed that the locators are trusted and cannot be compromised. Furthermore, each locator is assumed to know its own location. A sensor computes its location by listening for the beacon information sent by each locator. The beacons include the locator's location. Using all of the beacons that a sensor node detects, a node computes an approximate location based on the coordinates of the locators. Using a majority vote scheme, the sensor then computes an overlap-

ping antenna region. The final computed location is the "center of gravity" of the overlapping antenna region. All beacons transmitted by the locators are encrypted with a shared global symmetric key that is pre-loaded to the sensor prior to deployment. Each sensor also shares a unique symmetric key with each locator. This key is also pre-loaded on each sensor.

Authentication

An adversary is not just limited to modifying the data packet. It can change the whole packet stream by injecting additional packets. So the receiver needs to ensure that the data used in any decision-making process originates from the correct source. On the other hand, when constructing the sensor network, authentication is necessary for many administrative tasks (e.g. network reprogramming or controlling sensor node duty cycle). From the above, we can see that message authentication is important for many applications in sensor networks. Informally, data authentication allows a receiver to verify that the data really is sent by the claimed sender. In the case of two-party communication, data authentication can be achieved through a purely symmetric mechanism: the sender and the receiver share a secret key to compute the message authentication code (MAC) of all communicated data.

Adrian Perrig et al. propose a key-chain distribution system for their μTESLA secure broadcast protocol. The basic idea of the μTESLA system is to achieve asymmetric cryptography by delaying the disclosure of the symmetric keys. In this case a sender will broadcast a message generated with a secret key. After a certain period of time, the sender will disclose the secret key. The receiver is responsible for buffering the packet until the secret key has been disclosed. After disclosure the receiver can authenticate the packet, provided that the packet was received before the key was disclosed. One limitation of μTESLA is that some initial information must be unicast to each sensor node before authentication of broadcast messages can begin.

Liu and Ning propose an enhancement to the μTESLA system that uses broadcasting of the key chain commitments rather than μTESLA's unicasting technique. They present a series of schemes starting with a simple pre-determination of key chains and finally settling on a multi-level key chain technique. The multi-level key chain scheme uses pre-determination and broadcasting to achieve a scalable key distribution technique that is designed to be resistant to denial of service attacks, including jamming.

Chapter 6

Multimedia Networking

Basically, multimedia networking application means applications that employs audio or video.

6.1 Video and audio on-line

The most salient characteristic of video is its high bit rate. Video distributed over the internet typically ranges from 100 kbps for low-quality video conferencing to over 3 Mbps for streaming high-definition movies.

Another important characteristic of video is that it can be compressed, there by trading off video quality with bit rate. A video is a sequence of images, typically being displayed at a constant rate, for example, at 24 or 30 images per second. There are two types of redundancy in video which can be used for compression.

- Spatial redundancy. Ah image that consists of mostly white space has a high degree of redundancy and can be compressed without significantly sacrificing image quality.

- Temporal redundancy. An image and the subsequent image may be the same.

By using compression, we can create multiple versions of the save video, each at a different quality level.

Digital audio has lower bandwidth requirement comparing to the video. But it has its own characteristics. The analog audio signal is sampled at

some fixed rate, for example, 8,000 samples per second. Each of the samples is then rounded to one of a finite number of values, for example, 256 quantization values. Each of the quantization values is represented by one byte. The bit representation of the samples are then concatenated together to form the digital representation of the signal. For playback, the digital signal can be converted back, but only in an approximation of the original signals. By increasing the sampling rate and the number of quantization values, the decoded signal can better approximate the original analog signal. Compression are also used for the audio.

Although audio bit rates are generally much less than those of video, users are generally much more sensitive to audio glitches than video glitches.

6.1.1 Types of multimedia network applications

The multimedia applications can be classified into three broad categories:

- Streaming stored audio/video.
- Conversational voice-/video-over-IP.
- Streaming live audio/video.

Streaming stored audio and video

Streaming stored video has some key distinguishing features.

- Streaming. The client typically begins video play out within a few seconds after it begins receiving the video from the server. This means that the client will be playing out from one location in the video while at the same time receiving later parts of the video from the server.
- Interactivity. The user may pause, reposition forward, reposition backward, fast-forward, and so on through the video content.
- Continuous play out. Once play out of the video begins, it should proceed according to the original timing of the recording. So data must be received from the server in time for its play out.

The most important performance measure for streaming video is average throughput. By using buffering and prefetching, it is possible to provide

continuous playout even when the throughput fluctuates, as long as the average throughput remains above the video rate.

For many streaming video applications, prerecorded video is stored on, and streamed from, a CDN rather than from a single data center. P2P video streaming applications are also used for many applications.

Conversational voice- and video-over-IP

Real-time conversational voice over the internet is often referred as Voice-over-IP (VoIP) or Internet telephony. Timing considerations are important because audio and video conversational applications are highly delay-sensitive. The delay from when a user speaks or moves until the action is manifested at the other side should be less than a few hundred milliseconds. For voice, delays smaller than 150 milliseconds are not perceived by a human listener, delay between 150 and 400 milliseconds can be acceptable and delays exceeding 400 milliseconds can result in frustrating. On the other hand, conversational multimedia application are loss-tolerant–occasional loss only causes occasional glitches in audio/video playback, and these losses can often be partially of fully concealed.

Streaming live audio and video

These applications allow a user to receive a live radio or television transmission. Live, broadcast-like applications often have many users who receive the same audio/video program at the same time. Although the distribution of live audio/video to many receivers can be efficiently accomplished using the IP multicasting techniques, multicast distribution is more often accomplished via application-layer multicast (using P2P network or CDNs).

6.1.2 Streaming stored video

A common characteristic of the video streaming is the extensive use of client-side application buffering to mitigate the effects of varying end-to-end delays and varying amounts of available bandwidth between server and client.

Streaming video systems can be classified into three categories.

- UDP streaming: UDP streaming typically uses a small client-side buffer, big enough to hold less than a second of video. Before passing the video chunks to UDP, the server will encapsulate the video chunks within

transport packets specially designed for transporting audio and video, using the RTP (which we will discuss later) or a similar scheme. In addition to the server-to-client video stream, the client and the server also maintain, in parallel, a separate control connection over which the client sends commands regarding session state changes. The Real-Time Streaming Protocol (RTSP) defined in RFC 2326 is a popular open protocol for that. UDP streaming has some drawbacks. The unpredictable and varying amount of available bandwidth between server and client may cause problems since the server uses a fixed transmitting rate. The additional control connection increases the cost and complexity of the transmitting. Some firewall are configured to block UDP traffic which also causes problems.

- HTTP streaming: The video is stored in an HTTP server as an ordinary file with a specific URL. Upon requesting, the server sends the video as quickly as possible (as the TCP congestion control and flow control allow). One client side, the bytes are collected in a client application buffer and start to playback when the buffer exceeds a predetermined threshold. The client also can pre-fetch video frames that are to be consumed in the future. Note that the application buffer is different from the TCP receiving buffer. The application buffer works together with the TCP buffer to smooth the playback. When the available rate in the network is less than the video rate, playout will alternate between periods of continuous play out and period of freezing. Otherwise, after the initial buffering delay, the user will enjoy continuous playout until the video ends. HTTP streaming does not require a control connection. HTTP byte-range header is used to specify the range of bytes the client currently wants to retrieve from the desired video. Using this header, the user can reposition the video palyout.

- Adaptive HTTP streaming: One shortcoming of the HTTP streaming is that the client cannot choose the version of the video even the video can be encoded into different versions from high-definitions to low-definitions. Dynamic Adaptive Streaming over HTTP (DASH) is developed for improving that problem. With DASH, each video version is stored in the HTTP server with different URL. The HTTP server also has a manifest file, which provides a URL for each version along with its bit rate. The client first requests the manifest file and learns about

the various versions. Then the client selects one chunk at a time by specifying the URL. While downloading chunks, the client also measures the received bandwidth and runs a rate determination algorithm to select the chunk to request next. DASH therefore allows the client to freely switch among different quality levels. By dynamically monitoring the available bandwidth and client buffer level, and adjusting the transmission rate with version switching, DASH can often achieve continuous playout at the best possible quality level without frame freezing or skipping. In many implementations, the server not only stores many versions of the video but also separately stores many versions of the audio, so that the client can dynamically select both video and audio chunks, and locally synchronizes audio and video playout.

6.1.3 Content distribution networks

Many companies need to distribute on-demand streaming videos to clients all over the world. A single massive data center does not quite fit that purpose. Most major video-streaming companies now make use of Content Distribution Networks (CDNs). A CDN manages servers in multiple geographically distributed locations, stores copies of the videos (and other types of Web content, including documents, images, and audio) in its servers, and attempts to direct each user request to a CDN location that will provide the best user experience. The CDN may be a private CDN, i.e., owned by the content provider itself (for examples, Google's CDN distributes YouTube videos). The CDN may alternatively be a third-party CDN that distributes content on behalf of multiple content providers (for example, Akamai's CDN distributes Netflix and Hulu).

CDNs typically adopt one of two different server placement philosophies.

- Enter Deep. Pioneered by Akamai, it deploys server clusters in access ISPs (ISPs direct accessing end users) all over the world. The goal is to get close to end users, thereby improving user-perceived delay and throughput by decreasing the number of links and routers between the end user and then CDN cluster from which it receives content. Because of this highly distributed design, the task of maintaining and managing the clusters become challenging.

- Bring home. Taken by Limelight and other CDN companies, it brings the ISPs home by building large clusters at a smaller number of key

locations and connecting these clusters using a private high-speed network. Instead of getting inside the access ISPs, these CDNs typically place each cluster at a location that is simultaneously near the point of presence of many tier-1 ISPs. Compared with the enter-deep design, the bring-home design typically results in lower maintenance and management overhead, possibly at the expense of higher delay and lower throughout to end users.

Once its clusters are in place, the CDN replicates content across its clusters. Usually, the CDN does not place a copy of every video in each cluster. Many CDNs do not push videos to their clusters but instead use a simple pull strategy: if a client requests a video from a cluster that is not storing the video, then the cluster retrieves the video and stores a copy locally while streaming the video to the client at the same time. When a cluster's storage becomes full, it removes videos that are not frequently requested.

CDN operation

Most CDNs take advantage of DNS to intercept and redirect requests. For example, suppose a content provider *LUvideo*, employs the third-party CDN company, *CCC*, to distribute its video to its customers. On the *LUvideo* web pages, each of its videos is assigned a URL that includes the string "video" and a unique identifier for the video itself; for example, Transformers 7 might be assigned `http://video.LUvideo.ca/6Y7B23V`. Then the following steps occur:

1. The user visits the web page at *LUvideo*.

2. When the user clicks on the link `http://video.LUvideo.ca/6Y7B23V`, the user's host sends a DNS query for `video.LUvideo.ca`.

3. The user's Local DNS server (LDNS) relays the DNS query to an authoritative DNS server for LUvideo, which observes the string "video" in the hostname `video.LUvideo.ca`. To hand over the DNS query to *CCC*, instead of returning an IP address, the LUvideo authoritative DNS server returns to the LDNS a hostmane in the *CCC*'s domain, for example, `a1105.ccc.com`.

4. From this point on, the DNS query enters into *CCC*'s private DNS infrastructure. The user's LDNS then sends a second query, now for

`a1105.ccc.com`, and *CCC*'s DNS system eventually returns the IP addresses of a *CCC* content server to the LDNS. It is thus here, within the *CCC*'s DNS system, that the CDN server from which the client will receive its content is specified.

5. The LDNS forwards the IP address of the content-serving CDN node to the user's host.

6. Once the client received the IP address for a *CCC* content server, it establishes a direct TCP connection with the server at the IP address and issues a HTTP GET request for the video. If DASH is used, the server will first send to the client a manifest file with s list of URLs, one for each version of the video, and the client will dynamically select chunks from the different versions.

Cluster selection strategies

Cluster selection strategy is a mechanism for dynamically directing clients to a server cluster or a data center within the CDN. As we just saw, the CDN learns the IP address of the client's LDNS server via the client's DNS lookup. After learning this IP address, the CDN needs to select an appropriate cluster based on this IP address. CDNs generally employ proprietary cluster selection strategies.

One simple strategy is to assign the client to the cluster that is geographically closest. (Using commercial geo-location databases, each LDNS IP addresses is mapped to a geographic location). Such a solution can work reasonably well for a large fraction of the clients. However, for some clients, the solution may perform poorly, since the geographically closest cluster may not be the closest cluster along the network path. And a problem inherent with all DNS-based approaches is that some end-users are configured to use remotely located LDNSs. Moreover, this simple strategy ignores the variation in delay and available bandwidth over time of Internet paths, always assigning the same cluster to a particular client.

One method CDNs can be used is performing periodic real-time measurements of delay and loss performance between their clusters and clients. For instance, a CDN can have each of its clusters periodically send probes to all of the LDNSs around the world. One drawback of this approach is that many LDNSs are configured to not respond to such probes.

An alternative to sending extraneous traffic for measuring path properties is to use the characteristics of recent and ongoing traffic between the clients and CDN servers. Another alternative method is to use DNS query traffic to measure the delay between clients and clusters.

A very different approach to matching clients with CDN servers is to use IP anycast (RFC 1546). The idea behind IP anycast is to have the routers in the internet route the client's packets to the "closest" cluster, as determined by BGP. During the IP-anycast configuration stage, the CDN company assigns the same IP address to each of its clusters, and uses standard BGP to advertise this IP address from each of the different cluster locations. When a BGP router receives multiple route advertisements for this same IP address, it treats these advertisements as providing different paths to the same physical location (in fact for different physical locations). Following standard operating procedures, the BGP router will then pick the "best" router to the IP address according to its local route selection mechanism. For example, if one BGP route is only one AS hop away from the router, and all other BGP routers are two or more AS hops away, then the BGP router would typically choose to route packets to the location that needs to traverse only one AS. This approach has the advantage of finding the cluster that is closest to the client rather than the cluster that is closest to the client's LDNS. However, the IP anycast strategy again does not take into account the dynamic nature of the internet over short time scales.

6.2 VoIP

6.3 Protocols for real time applications

On-line real-time conversational applications, including VoIP and video conferencing, are very popular now. Therefore IETF and ITU have been busy for trying standard protocols.

6.3.1 RTP

Real-time Transport Protocol (RTP) is defined in RFC 3550, which can be used for transporting common formats such as PCM, ACC and MP3 for sound and MPEG and H.263 for video.

RTP basics

RTP typically runs on top of UDP. The sending side encapsulates a media chunk within an RTP packet, then encapsulates the packet in a UDP segment, and then hands the segment to IP.

If an application incorporates RTP–instead of a proprietary scheme to provide payload type, sequence numbers, or timestamps–then the application will more easily interoperate with other network multimedia applications. For example, if two different companies develop VoIP software and they both incorporate RTP into their product, then there may be some hope that a user using one of the VoIP products will be able to communicate with a user using the other VoIP product.

RTP allows each source (for example, a camera or a microphone) to be assigned its own independent RTP stream of packets. For example, for a video conference between two participants, four RTP streams could be opened–two streams for transmitting the audio and two streams for transmitting the video. However, many popular encoding techniques bundle the audio and video into a single stream during the encoding process. When the audio and video are bundled by the encoder, then only one RTP stream is generated in each direction.

RTP packets are not limited to unicast applications. They can also be sent over one-to-many and many-to-many multicast trees. For a many-to-many multicast session, all of the session's senders and sources typically use the same multicast group for sending their RTP streams. RTP multicast streams belonging together, such as audio and video streams emanating from multiple senders in a video conference application, belongs to an RTP session.

It should be noted that RTP does not provide any mechanism to ensure timely delivery of data or provide other quality-of-service guarantees. In fact, RTP encapsulation is seen only at the end systems.

RTP header

The format of RTP header is as follows.

<table>
<tr><td></td><td>Payload type</td><td>Sequence number</td><td>Timestamp</td><td>Sychronization source identifier</td><td>Misellaneous fields</td></tr>
</table>

Some explanations for the main fields are as follows.

- Payload type: This field is of 7 bits long. For an audio stream, the payload type is used to indicate the type of audio encoding (e.g., PCM, adaptive delta modulation, linear predictive encoding) that is being used. If a sender decides to change the encoding in the middle of a session, the sender can inform the receiver of the change through this payload type field. The sender may want to change the encoding in order to increase the audio quality or to decrease the RTP stream bit rate. Some of the audio payload types supported by RTP are listed at Figure 6.1. Similarly for a video stream, the payload type is used to indicate the type of video encoding.

- Sequence number: This field is 16 bits long. The sequence number increments by one for each RTP packet sent, and may be used by the receiver to detect packet loss and to restore packet sequence. For example, if the receiver side of the application receives a stream of RTP packets with a gap between sequence number s 86 and 89, then the receiver knows that packets 87 and 88 are missing. The receiver can then attempt to conceal the lost data.

- Timestamp: This field is 32 bits long. It reflects the sampling instant of the first byte in the RTP data packet. The timestamp is derived from a sampling clock at the sender. As an example, for audio the timestamp clock increment by one for each sampling period (for example, each 125 μsec for an 8 kHz sampling clock); if the audio application generates chunks consisting of 160 encoded samples, then the timestamp increases by 160 for each RTP packet when the source is active.

Payload-Type	Audio format	Sampling rate	rate
0	PCM μ-law	8 kHz	64 kbps
1	1016	8 kHz	4.8 kbps
3	GSM	8 kHz	13 kbps
7	LPC	8 kHz	2.4 kbps
9	G.722	16 kHz	48-64kbps
14	MPEG Audio	90 kHz	
15	G.728	8 kHz	16kbps

Figure 6.1: Audio payload types

6.3.2 SIP

The Session Initiation Protocol (SIP), defined in RFC 3261, 5411, is an open and lightweight protocol which typically works by exchanging short lines of ASCII text. SIP is a protocol at the application layer that interworks well with other internet protocols but less well with existing telephone system signaling protocols. SIP mainly does the following:

- It provides mechanisms for establishing calls between a caller and a callee over an IP network. It allows the caller to notify the callee that it wants to start a call or end calls. It allows the participants to agree on media encoding.
- It provides mechanisms for the caller to determine the current IP address of the callee. Users do not have a fixed IP address. They may also have multiple IP devices, each with a different IP address.
- It provides mechanisms for call management, such as adding new media streams during the call, changing the encoding during the call, inviting new participants during the call, call transfer, an call holding.

SIP addresses

First we look at a simple example, in which Alice and Bob's computers are both equipped with SIP-based software for making and receiving phone calls. Alice knows Bob's IP address and initiates the call.

- Alice sends an `INVITATION` message witch is similar to an HTTP request message. The message is sent over UDP to the port 5060 for SIP (SIP message can also be sent over TCP). The message includes an identifier for Bob (bob@193.64.210.89), an indication that Alice desires to receive audio, which is to be encoded in forat AVP 0 and encapsulated in RTP, and an indication that she wants to receive the RTP packet on port 38060.
- After receiving Alice's message, Bob sends an SIP response message, which is similar to HTTP response message. This response message is also sent to the SIP port 5060. Bob's message includes a 200 OK as well as an indication of his IP address, his desired encoding and packetization for reception, and his port number to which the audio packets should be sent.

- After receiving Bob's response, Alice sends Bob an SIP acknowledgment message.

After this SIP transaction, Bob and Alice can talk each other. Bob will encode and packetize the audio as requested and send the audio packets to port number 38060 of Alice's IP address. Alice will also encode (may different from Bob's encoding) and packetize the audio as requested by Bob and send the audio packets to port number and IP address as Bob indicates. This simple example shows that the SIP messages are sent and received in sockets that are different from those used for sending and receiving the media data. The SIP messages are ASCII-readable and resemble HTTP message. SIP requires all messages to be acknowledged, as it can rum over UDP or TCP.

In general, the SIP addresses are not IP addresses. One kind of addresses resembles email address, such as `sip:bob@domain.com`. Other possible SIP addresses could be legacy phone number or even name (assuming it is unique). So now suppose that Alice knows only Bob's SIP address `bob@domain.com`. In this case, Alice needs to obtain Bob's current IP address. To do this, Alice create an INVITE message begins with `INVITE bob@domain.com SIP/2.0` and sends this message to an SIP proxy. The proxy will respond with an SIP reply that might include the IP address of Bob's current IP address. However, the proxy may respond with the IP address of Bob's voicemail box or a URL of a web page (that says "Bob is sleeping."). Also the result returned by the proxy might depend on caller: If the call is from Bob's wife, he might accept the call and supply his IP address; if the call is from Bob's mother-in-law, he might respond with the URL.

SIP has another device called SIP registrar, that will give the information to the SIP proxy about the users' address. Every SIP user has an associated registrar. Whenever a user launches an SIP application on a device, the application sends an SIP register message to the registrar, informing the registrar of its current IP address. Bob's registrar keeps track of Bob's current IP address. Often SIP registrars and SIP proxies are run on the same host.

The above discussion has focused on call initiation for voice calls. SIP, being a signaling protocol for initiating and ending calls in general, can be used for video conference calls as well as for text-based sessions. In fact, SIP has become a fundamental component in many instant messaging applications.

Bibliography

[1] A. Tanenbaum and D.J. Wetherall, Computer Networks, 5th Edt., Prentice Hall.

[2] J.F. Kurose and K.W. Ross, Computer Networking: A Top-Down Approach, 6th Edt., Pearson.

[3] D.E. Comer, Computer Networks and Internets, 6th Edt., Pearson.

[4] Y. Zheng and S. Akhtar Networks for Computer Scientists and Engineers, Oxfod.

[5] W. Stallings, Data and Computer Communications, 10th Edt, Pearson.

Index