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

- Connection-oriented services (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 be terminated by either side in the third phase when the data transfer is complete or for some other reasons.
- Connectionless network service (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.

Switching techniques The main switching techniques are:

- Circuit switching (circuit networks).
- Packet switching (datagram networks)
- Virtual circuit packet switching (virtual-circuit networks).

Circuit switching: 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.

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.

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.

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

- Distance vector 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 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 link state routing algorithm)
- The Bellman-Ford algorithm (for distance vector routing algorithm).

Both of algorithms use graphs made up of nodes and arcs to calculate the shortest path between two nodes.

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

Packet drop at router

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.

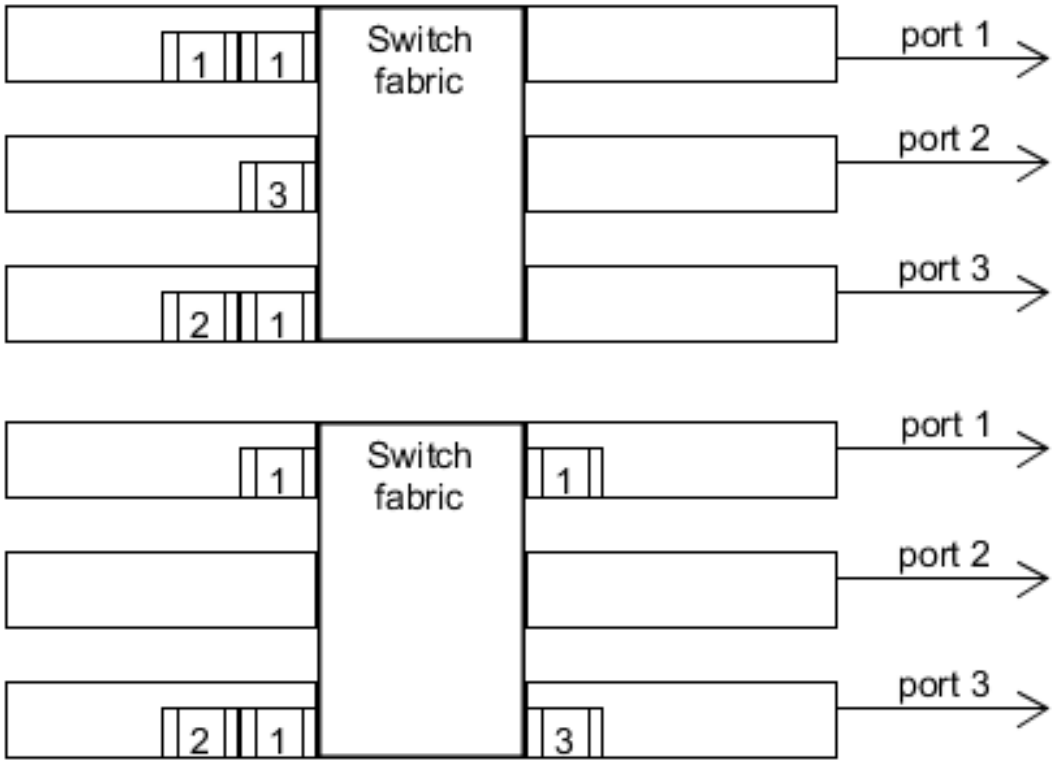
There are different schedulers. Some simple examples include first-com-first-served (FCFS) and weighted fair queuing (WFQ) which shares the outgoing link fairly among the different end-to-end connections that have packets queued for transmission.

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. Example algorithms:

- Drop-tail: drop the arriving packet.
- Random Early Detection (RED) algorithm: 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:



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.

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: the compression may be of slightly lower quality, but the compression ratio is good. Some image or movie compression use that kind of schemes.

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. 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.

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 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.

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 etc.

Wavelet compression can be either lossless or lossy.

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

For the performance of a network we need to consider

- Throughput and the utilization of the network
- Packet delay, loss, congestion and other errors
- Reliability
- Diagnostic assessment
- Trend assessment

Delay It is difficult to calculate the exact delay. Approximate analytical and simulation models are used.

- Processing delay: The time required to examine the packet's header and determine where to direct the packet, the time needed to check for bit-level errors, etc.
- Queuing delay: by buffering of a packet at the queue as it waits to be transmitted onto the line.
- Transmission delay: 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.

Throughput and bandwidth

The throughput of a network is defined as the amount of data that can be transferred per unit time.

Throughput and delay are often odds with each other. As the throughput is increased owing to higher offered load, the delay increases too.

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.

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 instantaneously available bandwidth in the network.

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.

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.