Process to Process Delivery in Transport Layer

The data link layer is responsible for delivery of frames between two neighboring nodes over a link. This is called node-to-node delivery.

Process to Process Delivery:

The data link layer is responsible for delivery of frames between two neighboring nodes over a link. This is called node-to-node delivery. The network layer is responsible for delivery of datagrams between two hosts. This is called host-to-host delivery. Real communication takes place between two processes (application programs). We need process-to-process delivery. The transport layer is responsible for process-to-process delivery-the delivery of a packet, part of a message, from one process to another. Figure 4.1 shows these three types of deliveries and their domains

The transport layer is responsible for process-to-process delivery.

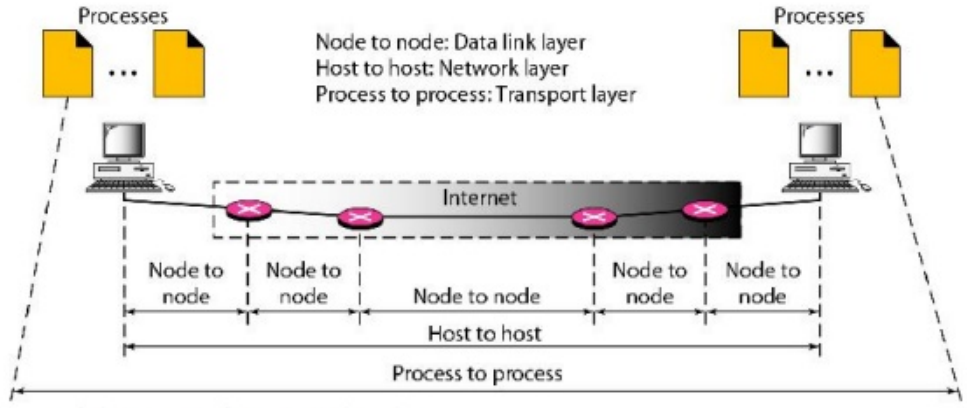

Figure 4.1 Types of data deliveries

1. Client/Server Paradigm

Although there are several ways to achieve process-to-process communication, the most common one is through the client/server paradigm. A process on the local host, called a client, needs services from a process usually on the remote host, called a server. Both processes (client and server) have the same name. For example, to get the day and time from a remote machine, we need a Daytime client process running on the local host and a Daytime server process running on a remote machine. For communication, we must define the following:

1. Local host
2. Local process
3. Remote host
4. Remote process

2. Addressing

Whenever we need to deliver something to one specific destination among many, we need an address. At the data link layer, we need a MAC address to choose one node among several nodes if the connection is not point-to-point. A frame in the data link layer needs a Destination MAC address for delivery and a source address for the next node's reply.

Figure 4.2 shows this concept.

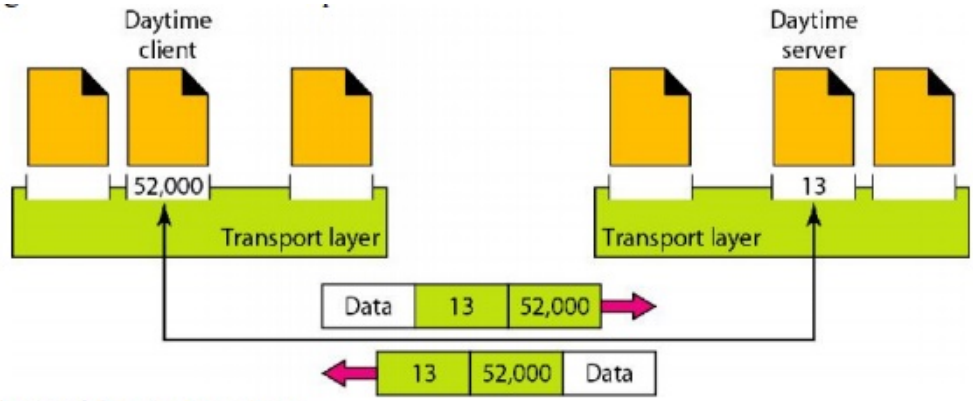

Figure 4.2 Port Numbers

The IP addresses and port numbers play different roles in selecting the final destination of data. The destination IP address defines the host among the different hosts in the world. After the host has been selected, the port number defines one of the processes on this particular host (see Figure 4.3).

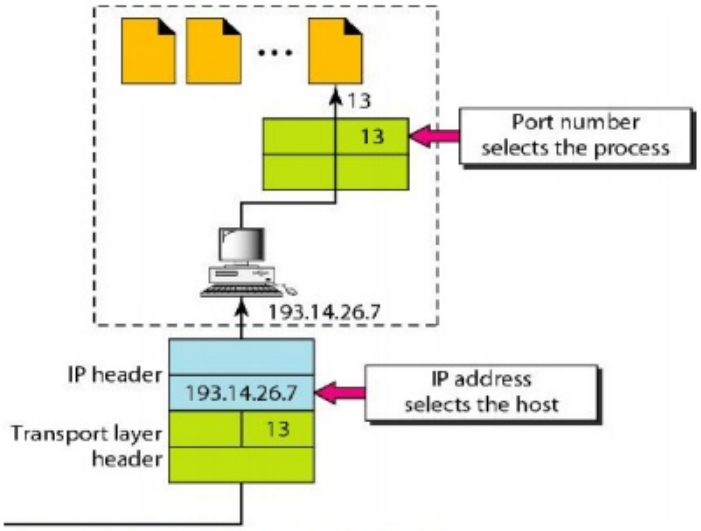

Figure 4.3 IP addresses versus port numbers

3. lANA Ranges

The lANA (Internet Assigned Number Authority) has divided the port numbers into three ranges: well known, registered, and dynamic (or private), as shown in Figure 4.4.

- **Well-known ports**. The ports ranging from 0 to 1023 are assigned and controlledby lANA. These are the well-known ports.

- **Registered ports.** The ports ranging from 1024 to 49,151 are not assigned orcontrolled by lANA. They can only be registered with lANA to prevent duplication.

- **Dynamic ports**. The ports ranging from 49,152 to 65,535 are neither controllednor registered. They can be used by any process. These are the ephemeral ports.

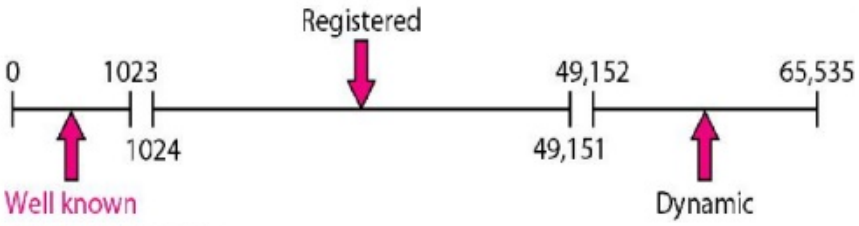

Figure 4.4 lANA Ranges

4. Socket Addresses

Process-to-process delivery needs two identifiers, IP address and the port number, at each end to make a connection. The combination of an IP address and a port number is called a socket address. The client socket address defines the client process uniquely just as the server socket address defines the server process uniquely (see Figure 4.5).

UDP or TCP header contains the port numbers.

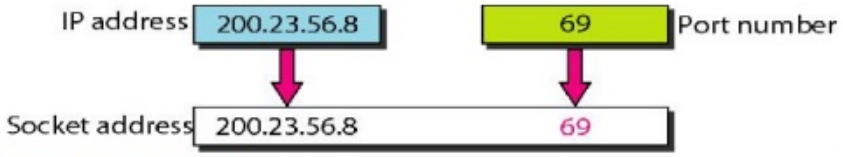

Figure 4.5 Socket Address

5. Multiplexing and Demultiplexing

The addressing mechanism allows multiplexing and demultiplexing by the transport layer, as shown in Figure 4.6.

Multiplexing

At the sender site, there may be several processes that need to send packets. However, there is only one transport layer protocol at any time. This is a many-to-one relationship and requires multiplexing.

Demultiplexing

At the receiver site, the relationship is one-to-many and requires demultiplexing. The transport layer receives datagrams from the network layer. After error checking and dropping of the header, the transport layer delivers each message to the appropriate process based on the port number.

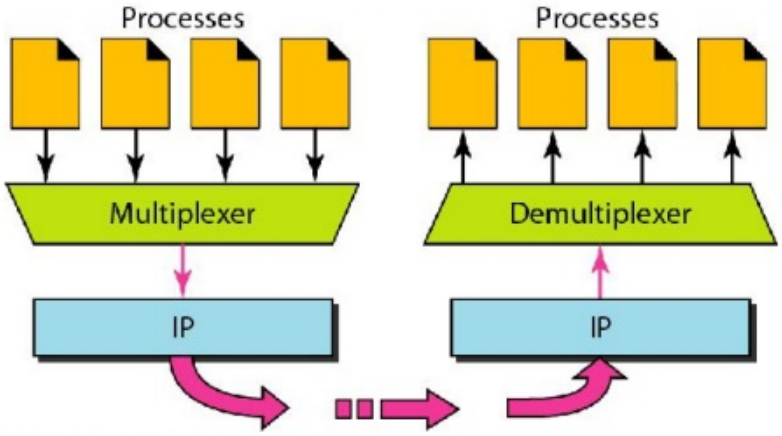

Figure 4.6 Multiplexing and Demultiplexing

6. Connectionless Versus Connection-Oriented Service

A transport layer protocol can either be connectionless or connection-oriented.

Connectionless Service

In a connectionless service, the packets are sent from one party to another with no need for connection establishment or connection release. The packets are not numbered; they may be delayed or lost or may arrive out of sequence. There is no acknowledgment either.

Connection~Oriented Service

In a connection-oriented service, a connection is first established between the sender and the receiver. Data are transferred. At the end, the connection is released.

7. Reliable Versus Unreliable

The transport layer service can be reliable or unreliable. If the application layer program needs reliability, we use a reliable transport layer protocol by implementing flow and error control at the transport layer. This means a slower and more complex service.

In the Internet, there are three common different transport layer protocols. UDP is connectionless and unreliable; TCP and SCTP are connection oriented and reliable. These three can respond to the demands of the application layer programs.

The network layer in the Internet is unreliable (best-effort delivery), we need to implement reliability at the transport layer. To understand that error control at the data link layer does not guarantee error control at the transport layer, let us look at Figure 4.7.

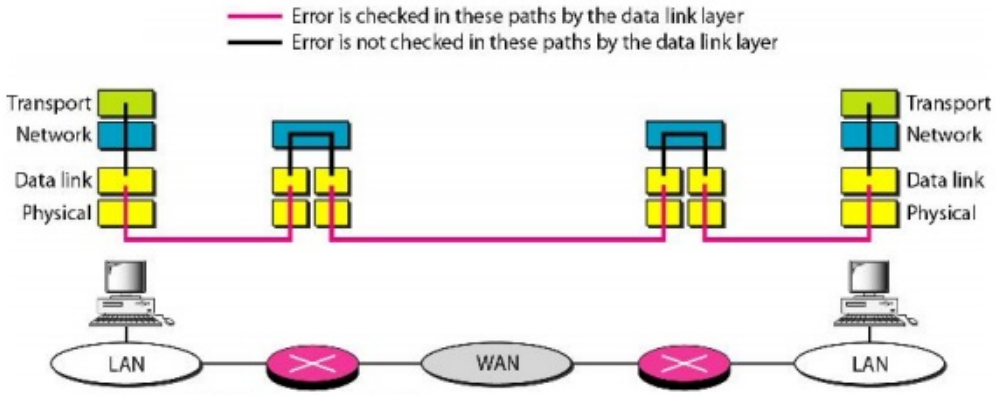

Figure 4.7 Error control

8. Three Protocols

The original TCP/IP protocol suite specifies two protocols for the transport layer: UDP and TCP. We first focus on UDP, the simpler of the two, before discussing TCP. A new transport layer protocol, SCTP, has been designed. Figure 4.8 shows the position of these protocols in the TCP/IP protocol suite.

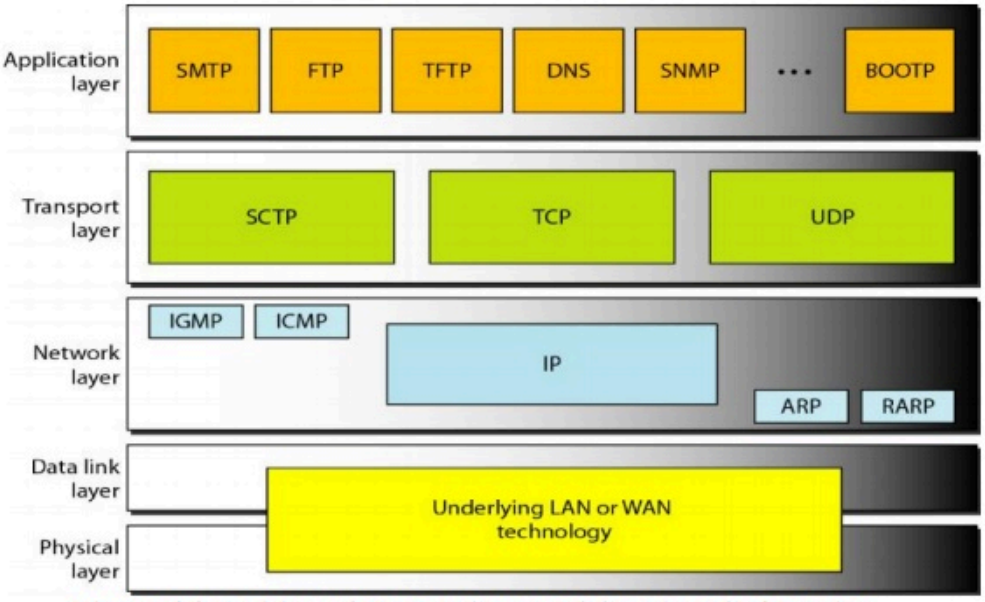

Figure 4.8 Position of UDP, TCP, and SCTP in TCP/IP suite