Transport Protocols

Connection-Oriented Transport Protocol Mechanisms

Two basic types of transport service are possible: connection oriented and connectionless or datagram service. A connection-oriented service provides for the establishment, maintenance, and termination of a logical connection between TS users.

This has, so far, been the most common type of protocol service available and has a wide variety of applications. The connection-oriented service generally implies that the service is reliable. This section looks at the transport protocol mechanisms needed to support the connection-oriented service.

A full-feature connection-oriented transport protocol, such as TCP, is very complex. For clarity, we present the transport protocol mechanisms in an evolutionary fashion. We begin with a network service that makes life easy for the transport protocol by guaranteeing the delivery of all transport data units in order and defining the required mechanisms. Then we will look at the transport protocol mechanisms required to cope with an unreliable network service.

1. Reliable Sequencing Network Service

Let us assume that the network service accepts messages of arbitrary length and, with virtually 100 percent reliability, delivers them in sequence to the destination.

Examples of such networks include the following:

• An IEEE 802.3 LAN using the connection-oriented LLC service

• A highly reliable connection-oriented packet-switching network, such as a frame relay, with the reliable connection option

In such cases, the transport protocol is used as an end-to-end protocol between two systems attached to the same network, rather than across an internet. The assumption of a reliable sequencing networking service allows the use of a quite simple transport protocol. Four issues need to be addressed:

- Addressing
- Multiplexing
- Flow control
- Connection establishment/termination

Addressing

The issue concerned with addressing is simply this: A user of a given transport entity wishes either to establish a connection with or make a data transfer to a user of some other transport entity using the same transport protocol. The target user needs to be specified by all of the following:

• User identification

• Transport entity identification

• Host address

• Network number

The transport protocol must be able to derive the information listed above from the TS user address. Typically, the user address is specified as (Host, Port).b The Port variable represents a particular TS user at the specified host. Generally, there will be a single transport entity at each host, so transport entity identification is not needed. If more than one transport entity is present, there is usually only one of each type. In this latter case, the address should include a designation of the type of transport protocol (e.g., TCP, UDP). In the case of a single network, Host identifies an attached network device. In the case of an internet, Host is a global internet address. In TCP, the combination of port and host is referred to as a socket.

Because routing is not a concern of the transport layer, it simply passes the Host portion of the address down to the network service. Port is included in a transport header, to be used at the destination by the destination transport protocol entity.

Multiplexing

With respect to the interface between the transport protocol and higher-level protocols, the transport protocol performs a multiplexing/demultiplexing function. That is, multiple users employ the same transport protocol and are distinguished by port numbers or service access points.

The transport entity may also perform a multiplexing function with respect to the network services that it uses. We define upward multiplexing as the multiplexing of multiple connections on a single lower-level connection, and downward multiplexing as the splitting of a single connection among multiple lower-level connections.

Flow Control

Whereas flow control is a relatively simple mechanism at the link layer, it is a rather complex mechanism at the transport layer, for two main reasons:

• The transmission delay between transport entities is generally long compared to the actual transmission time. This means that there is a considerable delay in the communication of flow control information.

• Because the transport layer operates over a network or internet, the amount of the transmission delay may be highly variable. This makes it difficult to effectively use a timeout mechanism for retransmission of lost data.

In general, there are two reasons why one transport entity would want to restrain the rate of segment transmission over a connection from another transport entity:

• The user of the receiving transport entity cannot keep up with the flow of data.

• The receiving transport entity itself cannot keep up with the flow of segments.

Connection Establishment and Termination

Even with a reliable network service, there is a need for connection establishment and termination procedures to support connection-oriented service. Connection establishment serves three main purposes:

• It allows each end to assure that the other exists.

• It allows exchange or negotiation of optional parameters (e.g., maximum segment size, maximum window size, quality of service).

• It triggers allocation of transport entity resources (e.g., buffer space, entry inconnection table).

2. Unreliable Network Service

A more difficult case for a transport protocol is that of an unreliable network service. Examples of such networks include the following:

• An internetwork using IP

• A frame relay network using only the LAPF core protocol

• An IEEE 802.3 LAN using the unacknowledged connectionless LLC service

The problem is not just that segments are occasionally lost, but that segments may arrive out of sequence due to variable transit delays. As we shall see, elaborate machinery is required to cope with these two interrelated network deficiencies. We shall also see that a discouraging pattern emerges. The combination of unreliability and nonsequencing creates problems with every mechanism we have discussed so far.

Generally, the solution to each problem raises new problems. Although there are problems to be overcome for protocols at all levels, it seems that there are more difficulties with a reliable connection-oriented transport protocol than any other sort of protocol.

In the remainder of this section, unless otherwise noted, the mechanisms discussed are those used by TCP. Seven issues need to be addressed:

- Ordered delivery

- Retransmission strategy

- Duplicate detection

- Flow control

- Connection establishment

- Connection termination

- Failure recovery

Ordered Delivery

With an unreliable network service, it is possible that segments, even if they are all delivered, may arrive out of order. The required solution to this problem is to number segments sequentially. We have seen that for data link control protocols, such as HDLC, each data unit (frame, packet) is numbered sequentially with each successive sequence number being one more than the previous sequence number. This scheme is used in some transport protocols, such as the ISO transport protocols. However, TCP uses a somewhat different scheme in which each data octet that is transmitted is implicitly numbered. Thus, the first segment may have a sequence number of 1. If that segment has 200 octets of data, then the second segment would have the sequence number 201, and so on.

Retransmission Strategy

Two events necessitate the retransmission of a segment. First, a segment may be damaged in transit but nevertheless arrive at its destination. If a checksum is included with the segment, the receiving transport entity can detect the error and discard the segment. The second contingency is that a segment fails to arrive. In either case, the sending transport entity does not know that the segment transmission was unsuccessful.

Duplicate Detection

If a segment is lost and then retransmitted, no confusion will result. If, however, one or more segments in sequence are successfully delivered, but the corresponding ACK is lost, then the sending transport entity will time out and one or more segments will be retransmitted. If these retransmitted segments arrive successfully, they will be duplicates of previously received segments. Thus, the receiver must be able to recognize duplicates. The fact that each segment carries a sequence number helps, but, nevertheless, duplicate detection and handling is not simple. There are two cases:

- A duplicate is received prior to the close of the connection.

- A duplicate is received after the close of the connection.

Flow Control

The credit allocation flow control mechanism described earlier is quite robust in the face of an unreliable network service and requires little enhancement.

Connection Establishment

As with other protocol mechanisms, connection establishment must take into account the unreliability of a network service. Recall that a connection establishment calls for the exchange of SYNs, a procedure sometimes referred to as a two-way handshake. Suppose that A issues a SYN to B. It expects to get a SYN back, confirming the connection. Two things can go wrong: A’s SYN can be lost or B’s answering SYN can be lost.

Connection Termination

The state diagram of Figure 15.3 defines the use of a simple two-way handshake for connection establishment, which was found to be unsatisfactory in the face of an unreliable network service. Similarly, the two-way handshake defined in that diagram for connection termination is inadequate for an unreliable network service. Disordering of segments could cause the following scenario. A transport entity in the CLOSE WAIT state sends its last data segment, followed by a FIN segment, but the FIN segment arrives at the other side before the last data segment. The receiving transport entity will accept that FIN, close the connection, and lose the last segment of data. To avoid this problem, a sequence number can be associated with the FIN, which can be assigned the next sequence number after the last octet of transmitted data. With this refinement, the receiving transport entity, upon receiving a FIN, will wait if necessary for the late-arriving data before closing the connection.

Failure Recovery

When the system upon which a transport entity is running fails and subsequently restarts, the state information of all active connections is lost. The affected connections become half open because the side that did not fail does not yet realize the problem. The still active side of a half-open connection can close the connection using a keep alive timer. This timer measures the time the transport machine will continue to await an acknowledgment (or other appropriate reply) of a transmitted segment after the segment has been retransmitted the maximum number of times. When the timer expires, the transport entity assumes that the other transport entity or the intervening network has failed, closes the connection, and signals an abnormal close to the TS user.