Transmission Control Protocol (TCP)

Manar Rsheed Almunyif
4419603554
Department of Computer Science , College of Engineering and Computer Science , Prince Sattam bin Abdulaziz University
Riyadh , Kingdom Saudi Arabia
441960354@std.psau.edu.sa

***Abstract*— The Transmission Control Protocol (TCP) is intended for use as a highly reliable host-to-host protocol between hosts in packet-switched computer communication networks, and in interconnected systems of such networks.**

Keywords—TCP, protocol , transport , network .

I. INTRODUCTION

This document describes the TCP is a transport level protocol of the Internet that provides reliable, end-to-end communication between two processes. The requesting process, often known as the client, requests services from the server process. Both client and server processes are accessible on their respective machines by their TCP port numbers assigned to them. Many standard application layer services have *well-known* TCP port numbers assigned by a central authority. For example, a Simple Mail Transfer Protocol server operates at the well-known TCP port 25. TCP carries bytes of data from the higher-level process by packaging it into TCP segments. TCP segment data are then packaged by IP into the data portion of IP packets. This document represents a specification of the behavior required of any TCP implementation, both in its interactions with higher level protocols and in its interactions with other TCPs.

II. OPERATION OF THE TCP

the primary purpose of the TCP is to provide reliable, securable logical circuit or connection service between pairs of processes. To provide this service on top of a less reliable internet communication system requires facilities in the following areas: Basic Data Transfer , Reliability , Flow Control , Multiplexing , Connections , and Precedence and Security.

A. Basic Data Transfer

The TCP is able to transfer a continuous stream of octets in each direction between its users by packaging some number of octets into segments for transmission through the internet system. In general, the TCPs decide when to block and forward data at their own convenience. Sometimes users need to be sure that all the data they have submitted to the TCP has been transmitted. For this purpose a push function is defined. To assure that data submitted to a TCP is actually transmitted the sending user indicates that it should be pushed through to the receiving user. A push causes the TCPs to promptly forward and deliver data up to that point to the receiver. The exact push point might not be visible to the receiving user and the push function does not supply a record boundary marker.

B. Reliability

The TCP must recover from data that is damaged, lost, duplicated, or delivered out of order by the internet communication system. This is achieved by assigning a sequence number to each octet transmitted, and requiring a positive acknowledgment (ACK) from the receiving TCP. If the ACK is not received within a timeout interval, the data is retransmitted. At the receiver, the sequence numbers are used to correctly order segments that may be received out of order and to eliminate duplicates. Damage is handled by adding a checksum to each segment transmitted, checking it at the receiver, and discarding damaged segments. As long as the TCPs continue to function properly and the internet system does not become completely partitioned, no transmission errors will affect the correct delivery of data. TCP recovers from internet communication system errors.

C. Flow Control

TCP provides a means for the receiver to govern the amount of data sent by the sender. This is achieved by returning a "window" with every ACK indicating a range of acceptable sequence numbers beyond the last segment successfully received. The window indicates an allowed number of octets that the sender may transmit before receiving further permission.

D. Multiplexing

To allow for many processes within a single Host to use TCP communication facilities simultaneously, the TCP provides a set of addresses or ports within each host. Concatenated with the network and host addresses from the internet communication layer, this forms a socket. A pair of sockets uniquely identifies each connection. That is, a socket may be simultaneously used in multiple connections. The binding of ports to processes is handled independently by each Host. However, it proves useful to attach frequently used processes (e.g., a "logger" or timesharing service) to fixed sockets which are made known to the public. These services can then be accessed through the known addresses. Establishing and learning the port addresses of other processes may involve more dynamic mechanisms.

E. Connections

The reliability and flow control mechanisms described above require that TCPs initialize and maintain certain status information for each data stream. The combination of this information, including sockets, sequence numbers, and window sizes, is called a connection. Each connection is uniquely specified by a pair of sockets identifying its two sides. When two processes wish to communicate, their TCP's must first establish a connection (initialize the status

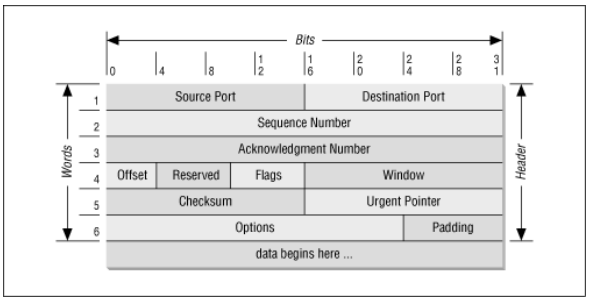
information on each side). When their communication is complete, the connection is terminated or closed to free the resources for other uses. Since connections must be established between unreliable hosts and over the unreliable internet communication system, a handshake mechanism with clock-based sequence numbers is used to avoid erroneous initialization of connections.

F. Precedence and Security

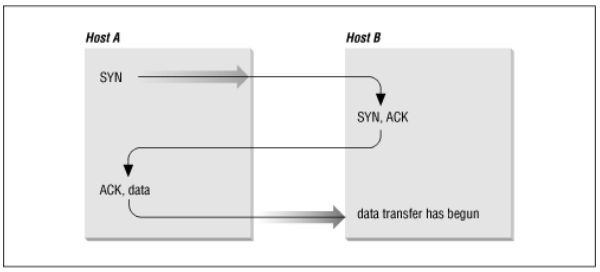
The users of TCP may indicate the security and precedence of their communication. Provision is made for default values to be used when these features are not needed. The TCP makes use of the internet protocol type of service field and security option to provide precedence and security on a per connection basis to TCP users. Not all TCP modules will necessarily function in a multilevel secure environment; some may be limited to unclassified use only, and others may operate at only one security level and compartment. Consequently, some TCP implementations and services to users may be limited to a subset of the multilevel secure case. TCP modules which operate in a multilevel secure environment must properly mark outgoing segments with the security, compartment, and precedence. Such TCP modules must also provide to their users or higher level protocols such as Telnet or THP an interface to allow them to specify the desired security level, compartment, and precedence of connections.

III. SEGMENT FORMAT

TCP provides reliability with a mechanism called Positive Acknowledgment with Re-transmission (PAR). Simply stated, a system using PAR sends the data again, unless it hears from the remote system that the data arrived successfully. The unit of data exchanged between cooperating TCP modules is called a segment . Each segment contains a checksum that the recipient uses to verify that the data is undamaged. If the data segment is received undamaged, the receiver sends a positive acknowledgment back to the sender. If the data segment is damaged, the receiver discards it. After an appropriate time-out period, the sending TCP module re-transmits any segment for which no positive acknowledgment has been received. In below overview about segment format .



TCP is connection-oriented. It establishes a logical end-to-end connection between the two communicating hosts. Control information, called a handshake, is exchanged between the two endpoints to establish a dialogue before data is transmitted. TCP indicates the control function of a segment by setting the appropriate bit in the Flags field in word 4 of the segment header. The type of handshake used by TCP is called a three-way handshake because three segments are exchanged. The figure shows the simplest form of the three-way handshake. Host A begins the connection by sending host B a segment with the Synchronize sequence numbers (SYN) bit set. This segment tells host B that A wishes to set up a connection, and it tells B what sequence number host A will use as a starting number for its segments. (Sequence numbers are used to keep data in the proper order.) Host B responds to A with a segment that has the Acknowledgment (ACK) and SYN bits set. B's segment acknowledges the receipt of A's segment and informs A which Sequence Number host B will start with. Finally, host A sends a segment that acknowledges receipt of B's segment and transfers the first actual data.



After this exchange, host A's TCP has positive evidence that the remote TCP is alive and ready to receive data. As soon as the connection is established, data can be transferred. When the cooperating modules have concluded the data transfers, they will exchange a three-way handshake with segments containing the no more data from sender bit (called the FIN bit) to close the connection. It is the end-to-end exchange of data that provides the logical connection between the two systems.

A. Sequence Number

The sequence number of the first data octet in this segment (except when SYN is present). If SYN is present the sequence number is the initial sequence number (ISN) and the first data octet is ISN+1.

B. Acknowledgment Number

If the ACK control bit is set this field contains the value of the next sequence number the sender of the segment is expecting to receive. Once a connection is established this is always sent.

C. Window

The number of data octets beginning with the one indicated in the acknowledgment field which the sender of this segment is willing to accept.

D. Checksum

The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header and text. If a segment contains an odd number of header and text octets to be checksummed , the last octet is padded on the right with zeros to form a 16 bit word for checksum purposes. The pad is not transmitted as part of the

segment. While computing the checksum, the checksum field itself is replaced with zeros.

IV. Advantages And Disadvantages

A. *Here are Advantages of TCP*

- It helps you to establish/set up a connection between different types of computers.
- It operates independently of the operating system.
- It supports many routing-protocols.
- It enables the internetworking between the organizations.
- TCP/IP model has a highly scalable client-server architecture.
- It can be operated independently.
- Supports several routing protocols.
- It can be used to establish a connection between two computers.

B. *Here are Disadvantages of TCP*

- TCP never conclude a transmission without all data in motion being explicitly asked.
- You can't use for broadcast or multicast transmission.
- TCP has no block boundaries, so you need to create your own.
- TCP offers many features that you don't want. It may waste bandwidth, time, or effort.
- In this, model the transport layer does not guarantee delivery of packets.
- Replacing protocol in TCP/IP is not easy.
- It doesn't offer clear separation from its services, interfaces, and protocols.

C. *Differenc Between TCP And UDP*

1. TCP stands for "Transmission Control Protocol" while UDP stands for "User datagram Protocol".
2. TCP is the connection-oriented protocol while UDP is connectionless protocol.
3. TCP is more reliable than UDP.
4. TCP uses both error detection and error recovery. UDP works on a "best-effort" basis
5. UDP is faster for data sending than TCP.
6. UDP makes error checking but no reporting but TCP makes checks for errors and reporting.
7. TCP gives a guarantee that the order of data at receiving end is the same as on sending end while UDP has no such guarantee.
8. Header size of TCP is 20 bytes while that of UDP is 8 bytes.
9. TCP is heavyweight as it needs three packets to set up a connection while UDP is lightweight.
10. TCP has acknowledgement segments but UDP has no acknowledgement.
11. TCP is used for an application that requires high reliability but less time-critical whereas UDP is used for an application that is time-sensitive but requires less reliability.

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an MSW document, this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your frame, use the MSWord "Format" pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.

V. Using the Template

References

[1] Robert J. Shimonski, Yuri Gordienko, in Sniffer Pro Network Optimization and Troubleshooting Handbook, 2002.

[2] Cerf, V., and R. Kahn, "A Protocol for Packet Network Intercommunication", IEEE Transactions on Communications.

[3] Postel, J. (ed.), "Internet Protocol - DARPA Internet Program Protocol Specification", RFC 791, USC/Information Sciences Institute.

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.