

Design Notes

TCP Protocol

In this issue of Design Notes, we discuss the most widely used Transport layer protocol, TCP, and explain the fields in the TCP header. We also discuss how a TCP connection is opened and closed. Finally we explain some other popular protocols including ARP, UDP and ICMP.

TCP Protocol

The TCP protocol is used to carry upper layer data, for an application layer protocol such as SMTP, HTTP or FTP. Alternatively, it may be data in some proprietary format that doesn't conform to a particular protocol, and hence the TCP packet is used simply as a data carrier.

TCP is a point to point protocol, meaning communications occurs between two hosts, and only two hosts. It is not a multi broadcast system, hence one sender cannot send a message that will be interpreted or intended to be processed by more than one recipient.

The protocol uses a mechanism to ensure data sent by a host, is acknowledged by the recipient host. In other words, there is an acknowledgement mechanism present.

The protocol also provides flow control of data.

Data Sequencing And Acknowledgement

When two hosts are communicating and exchanging data, they encapsulate the raw data in the TCP packet data field. The header also contains various other fields as shown in Figure 1.

The most important fields are the sequence and acknowledgement numbers. Every byte of data sent by a host, has a 32 bit sequence number associated with it. Sequence numbers are sequential and don't have to start from zero. When a datagram is sent, the sequence number of the first byte in the datagram is included in the TCP header.

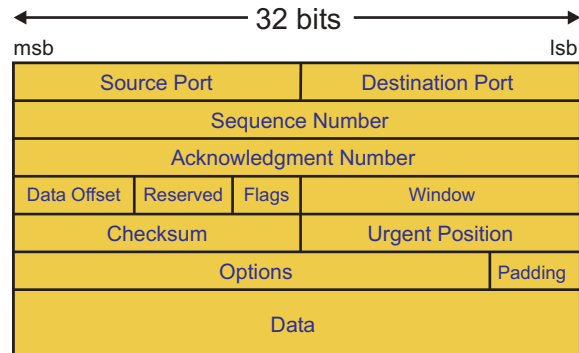


Figure 1: TCP Header

For instance, assume a datagram has 30 bytes of data, and the first byte's sequence number is 500. The final byte's sequence number will be 529. Say the following datagram has 55 bytes. The sequence number of the first byte will be 530, and the final byte 584. Finally, say the next datagram has 40 bytes. The sequence number of the first byte will be 585 and the last byte 624. The sequence number of the first byte of each datagram will be included in the TCP header of the respective datagram, thus the sequence number field of the 1st, 2nd and 3rd datagrams will be 500, 530, and 585 respectively.

The question must be asked: what is the purpose of sequence numbers? Their purpose is to allow a recipient to correctly order the received data, should the datagrams arrive out of order, as can happen over the internet or a LAN or WAN. By looking at the sequence numbers, the order of the packets can be deciphered. In addition, lost packets and duplicates can also be identified,



electronics by design

Electronics Design Services

- Microcontrollers
- Microprocessors, FPGA's,
- Telephony, Facsimile
- TCP/IP, Ethernet,
- Embedded C & Assembler
- Protel & EMC/EMI

Electronics By Design

Suite 25, 1-7 Jordan St Gladesville 2111

Phone : (02) 9816-3965

Fax: (02) 9816-3967

Web: www.electronicbydesign.com.au

through missing or doubled up sequence numbers. For instance, in the above example if the second datagram is lost, it will be obvious since the first datagram will be received and have data with sequence numbers 500-529. The next datagram received will have data with sequence numbers 585-624. Data with sequence numbers 530-584 will be missing, i.e. the 2nd datagram did not arrive at its destination.

The host receiving the data must now indicate to the sender that data is missing, but how is this done? The mechanism for doing this is a 32 bit number called the acknowledgement number. In TCP, the acknowledgement number indicates the position in the data stream up to which data has been received and acknowledged by the remote TCP host.

If the host sending the data does not receive an acknowledgement number within a timeout period, it retransmits the lost data.

Acknowledgement numbers are cumulative in the sense that they indicate how much of the data stream has been accumulated so far. It is possible for a single acknowledgement number to acknowledge bytes received in multiple datagrams. To minimise communications traffic, a remote host receiving data may send a single acknowledgement packet for every two or three packets it has received.

A host (call it the local host) can send a TCP packet with no data, possibly to act as a heartbeat to the remote host. In this case, the sequence number field will contain the sequence number of the next byte of data, which will be sent in a future packet.

It must be pointed out, communications between the hosts is full duplex. Each host can send and receive data, hence every packet has sequence and acknowledgement number fields.

From the above it can be seen why TCP guarantees delivery of a packet. If the packet is not received, the sender is informed through the acknowledgement number.

Flow Control

Flow control is sometimes required because a host may have CPU, memory or bandwidth

restrictions. The window parameter in the TCP header indicates to a remote host, how many bytes of data it can send at any time. The receiver can change the window size at any time by sending a TCP packet to the sender, with a different window size. The window parameter reflects the available buffer size of the receiver. If the receiver cannot accept any further data, it sets the window parameter to zero.

Port Numbers

A local host may have multiple TCP connections open simultaneously, possibly with the same remote host. When data arrives at the local host, how does it know which application the data should be directed towards? This is solved using a 16-bit port number.

The port number in a TCP header may be application specific, and should be registered with an organisation called IANA, if it's proprietary. If it's not proprietary, it is known as a "well known" port number. Such numbers include 80 for the World Wide Web (HTTP) and 25 for mail protocol (SMTP).

A TCP header contains source and destination port numbers. Usually they are the same as data is sent between hosts using the same application, however in some instances, different numbers are required, to define associations between processes.

TCP Packet Flags

A TCP header has a number of flags as follows:

Urgent (URG)

The urgent flag indicates to a receiving host to process the data immediately, jump the queue and process this packet before any other pending packets.

Acknowledgement (ACK)

Setting this flag indicates the packet's acknowledgment number valid.

Push (PSH)

The push flag tells the TCP engine to send the packet through to the correct application.

Reset (RST)

The reset flag is used to abruptly terminate a TCP connection, usually to recover from an error state.

A reset causes both hosts to terminate the connection without any further handshaking. TCP connections should use this, as a last resort and only if a graceful close has been attempted and has failed.

Synchronisation (SYN)

The SYN flag is used to indicate the opening of a virtual circuit connection.

Finish (FIN)

The FIN flag is used to terminate a connection. It is set in the TCP header of packets used to terminate a virtual circuit connection.

Data Offset Field

The data offset field indicates the number of double words (32 bit) in the TCP header. This field will only be used if options are added to the header. At the time of writing, the only options are the Maximum Segment Size (MSS). The MSS is the maximum allowable size of the packet.

Checksum

Every packet of data has a checksum to verify the integrity of the data.

Opening And Closing TCP Connections

Description

A TCP connection is known as a virtual connection, because it is not a physical connection like a telephone line. Instead, the connection can best be described as an "understanding" between two hosts that they will exchange data with each other on an on-going basis. The physical link however, must also be in place to allow data to be transferred.

Opening A Connection

A TCP/IP connection opening is shown in Figure 2 and occurs as follows:

The initiator called the client, sends a TCP/IP packet to the target host called the server. The packet has the SYN flag set along with a sequence number which is called the Initial Sender Sequence Number (ISN). The acknowledgement number is set to zero because the client does not know what it should be as it has not communicated with the server yet. Port numbers, window size and MSS are also set as required by the client. The IP address and IP header must also be set appropriately, including the source and target IP addresses.

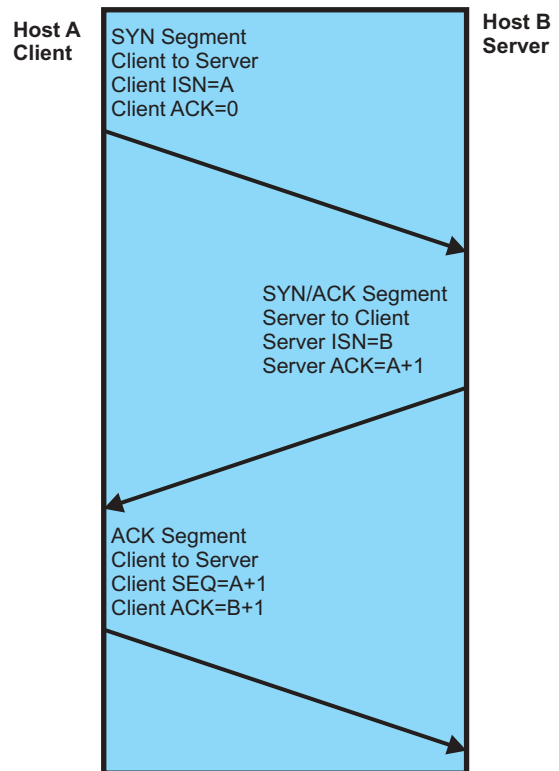


Figure 2: Opening a TCP Connection

The server responds with a TCP/IP packet. The TCP packet has the server's sequence number called the Initial receiver Sequence Number, and an acknowledgement number (which is the client's previous sequence number + 1). The SYN flag is set since the hosts are setting up a connection, and the ACK flag is also set because the server is sending an acknowledgement number in response to the client's previous packet. Port numbers, window size and MSS are also set as required by the server. The IP address and IP header must also be set appropriately, including the source and target IP addresses.

The final packet in the handshaking process is sent by the client. It acknowledges the connection has been set up. The sequence number in the packet is the client's previous sequence number + 1, that was sent by the client when it initiated the handshake process. The acknowledgement number is the server's previous sequence number. The SYN flag is not set. The ACK flag is set, as is the PSH flag because the application has to be informed the connection is now open. Port numbers are set as required by the server. The IP address and IP header must also be set appropriately, including the source and target IP

addresses. The connection is now open.

Closing A Connection

A connection is closed using a handshaking process, called a "Graceful Close". Either the client or the server can initiate the process of closing the connection and it's done in the following way:

- The initiator (client or server) sends a data packet with the FIN flag set. The data packet does not contain any data, however the acknowledgment and sequence numbers are valid. Port number remains unchanged and the window size parameter is set to zero.
- The recipient responds with a data packet. The FIN flag is set and the acknowledgment and sequence numbers are valid. Port number remains unchanged and the window size parameter is set to zero.
- The initiator sends the final packet with the FIN flag set. Once again the packet does not contain data, however the acknowledgment and sequence numbers are valid.
- The connection is now closed and no further data can be sent.

Closing A Connection The Brute Force Way

The TCP packet contains a flag called RST. If a connection has been established and the initiator (client or server) wishes to terminate it abruptly, say due to a non recoverable error, it can send a TCP/IP packet with the RST flag set to one. The recipient will see the RST flag has been asserted and terminate the connection. No confirmation will be sent from the recipient to the initiator. This method is sometimes called a "Hard Reset" and should only be used if a graceful close cannot be done.

Other Protocols

Address Resolution Protocol (ARP)

An ARP is used to translate a node's IP address to a corresponding MAC Address (Ethernet Address) of the underlying hardware. A request message is placed in the Ethernet packet, and broadcast to all hosts on the network. Only the receiving host whose IP address matches the

request, sends a response which contains it's MAC address. The requestor extracts the MAC address and uses it in subsequent packets addressed to the target node. ARP packets should be sent often enough to detect changes in hardware configurations which a network administrator may make. Typically an ARP is sent when data is to be exchanged between hosts and there has been no communications between them for over 5 minutes.

User Datagram Protocol (UDP)

UDP is a connectionless protocol. It differs from TCP because it doesn't have a handshaking mechanism (i.e. acknowledgment and sequence numbers) so packets are not guaranteed to arrive at their destination and are not guaranteed to arrive in sequence. UDP doesn't have a mechanism for opening a connection so there is minimal overhead in transferring data. A UDP packet encompasses a CRC field, thus providing error checking.

UDP is useful in applications that are command/response oriented. It is also ideal for sending multicast or broadcast packets because a connection does not have to be opened and closed with each recipient so if there are 1000 recipients (say) it represents a greatly reduced level of overhead.

A UDP packet replaces a TCP packet in the ISO 7 layer model, thus it is encapsulated inside the data field of the IP packet.

UDP packets are used for simple protocols such as TFTP, DNS or NFS.

Internet Control Message Protocol (ICMP)

ICMP packets are used to convey errors, status messages and implement debugging tools. The most common ICMP command is PING, which is used to test if a node's IP address is accessible and hence if the node is on-line.

References

- [1] Inside TCP/IP by Siyan Karanjit
- [2] An Introduction to TCP/IP by Z-World Inc