Communications Networks

TCP/IP Protocol stack:

|  |
| --- |
| Applications |
| Transport  (TCP/UDP) |
| Internetwork  (IP) |
| Network Interface and Hardware |

Divided into layers to share work and ability to add in new parts into the system

UDP is fast but only use if you don’t mind losing certain data.

Internetwork Layer: Internet layer / network layer

* Provides virtual network view (hides underlying physical network)
* IP is the most important internetwork layer protocol
  + Internet protocol sends message across the internet, providing a routing function
  + Unreliable protocol, fragments might be lost/arrive out of place
  + Higher layer protocols add reliability to IP
  + IP DATAGRAM is the basic unit of information transmitted in a IP network.
    - Can be divided with a header

IP DATAGRAM HEADER

* + IP Datagram = IP Header + data (from higher layer protocols)
  + Header has a minimum length of 20 bytes (8bits)
* Version field
  + The IP protocol version being used 4 bits e.g. IPv4 is 0100 e.g. IPv6 is 0110
* Internet Header Length (IHL) field
  + The length of this IP header in 32-bit words 4 bits e.g. IHL value 6 (0110) → header is 6 ∗ 32 = 192 bits Minimum IHL value is 5 (0101)
* Service Type field Specifies
  + how the datagram should be handled in transmission (the quality of service requested) 8 bits
* Total length field.
  + Total number of bytes/octets (multiples of 8 bits) that the IP datagram (header plus data) takes up
    - 16 bits
    - maximum length: 65,535 octets, minimum length: 20 octets
* Identification field
  + A unique number assigned to a datagram fragment to help reconstructing message from fragments
    - 16 bits
    - Every fragment of the same message has the same identification number
* Flag fields: Informational bits
  + 3 bits for 3 flags: Reserved, Don’t Fragment (DF), and More Fragments (MF)
  + Reserved bit must be zero DF indicates if fragmentation of this data is allowed (0), or not allowed (1)
  + MF indicates if this is the last fragment of the datagram (0), or there are still more fragments to follow (1)
* Fragment offset field: Specifies where in the original message the fragment in this datagram starts
  + 13 bits
  + It is the position of this fragment from the start of the original message counted in 8 bytes units
* Time to live field
  + (8 bits).
  + How long this datagram is allowed to remain in the system before being deleted
* Protocol field
  + (8 bits).
  + The host-to-host transport layer protocol being used by the message sent over IP, e.g. TCP: 6
* Header checksum field
  + (16 bits).
  + Used to ensure integrity of the header in transmission
* Source field
  + (32 bits).
  + IP address of sending/source host, e.g. 137.73.9.232
* Destination field
  + (32 bits).
  + IP address of destination host (intended recipient)
* Options field
  + Optional arguments usable by IP processing software
* Padding field
  + 0s to make header up to multiple of 32 bits

FRAGMENTATION

* Each network has limit on size of message it can carry – Maximum Transmission Unit (MTU)
* Sender fragments message to fit local MTU
* May be fragmented further when moving from one network to another to fit that MTU
* Reassembled at destination

FRAGMENTATION ALGORITHM

* DF flag is checked to see if fragmentation is allowed:
  + If DF is set but message is too large for MTU an error is sent by ICMP
    - ICMP – protocol that provides errors to sender.
  + IMCP code 4: fragmentation needed and DF set
* Based on MTU data is split into parts
  + Every new data portion must be multiple of 8 bytes (32 bit) apart from last one
* Each data portion placed in IP datagram
* Each data portion treated as normal in IP datagram
  + Fragments can traverse different routers to destination
* The headers of fragmented datagrams are minor modifications of the original datagram’s header:
  + The MF flag bit is set in all fragments except the last
  + The fragment offset field in each is set to the location this data portion occupied in the original datagram, relative to the beginning of the original datagram
  + If options were included in the original datagram, some of the options may be copied to each of the fragments depending on the Copied flag in each option field
  + The header length field of the fragment is set
  + The total length field of the fragment is set
  + The header checksum field is re-calculated

TRANSPORT LAYER:

**Transmission Control Protocol (TCP)**

**Key Features**: - Connection oriented: implements mechanisms to setup and tear down a full duplex connection between end points

* + Reliable: implements mechanisms to guarantee error free and ordered delivery of information Flow and
  + Congestion controlled: implements mechanisms to control traffic

**Ports & Sockets**

* A process identifies itself to the TCP protocol by one or more ports
* A port is a 16-bit number used to identify to which application/process the message should be delivered
* Some ports are reserved for specific applications
  + e.g. FTP: 20/21, HTTP: 80, SMTP (e-mail): 25
* A socket is the combination of a host’s IP address and a port number
  + E.g. 137.73.9.232 : 8080
  + Every communication in TCP is between two sockets, i.e. two hosts using particular ports

TCP Connection Set-up:

* Snder sends a synchronise message (SYN) to sender
* Receiver sends message back acknowledging SYN (SYN ACK)
* Sender sends a message acknowledging the acknowledgement (ACK)

TCP Connection Tear Down:

* Sender sends a finalise message (FIN) to the receiver
* Receiver responds with an acknowledgement of the finalise (FIN + ACK)
* Finally, the sender responds with an acknowledgement of the acknowledgement (ACK)

**TCP Concepts for Reliability and Flow Control:**

**Basic Reliability**: Positive Acknowledgement with Retransmission (PAR)

* Message 2 cannot be sent without acknowledgement of message 1 returning

**Improved PAR**: similar structure to PAR but with message IDENTIFICATION and Send Limits

* Multiple message can be sent without waiting for acknowledgment
* Beter use of bandwidth (efficiency)
* Send Limit specifies maximum number of unacknowledged messages allowed at one time (BASIC FLOW CONTROL)

**TCP’s Sliding Window System**: Similar to Improved PAR

* Reliability. The receiver sends an acknowledgement back to the sender for every segment it receives
  + An acknowledgement states that the receiver has received all data in the message before a given sequence number
  + A segment is retransmitted on time out, or on acknowledgment repetition
* Flow Control. The amount of data that the receiver can receive is called the window size (WND)
  + It is the maximum number of unacknowledged bytes allowed from sender at one time
  + Determined initially by the receiver when the connection is established, but can vary during data transfer
  + Each ACK message will include the window size that the receiver is ready to deal with at that particular time
  + Usable Window: amount of the window that the sender is still allowed to send at any point in time

USABLE WINDOW SIZE = UNA + WND – NXT

**TCP Header:**

* Source Port: the port of the segment sender
* Destination Port: the port of the receiver
* Sequence Number: the sequence number of the start of the segment, or the Initial Sequence
* Number (if a synchronise message)
* Acknowledgement Number: the sequence number before which the receiver has all the message data
* Data Offset: The length of the TCP header in 32-bit words
* Reserved: Not currently used (all 0s)
* URG Flag: Marks that this message contains urgent data
* ACK Flag: Marks that this is an acknowledgement
* PSH Flag: Marks that this data was pushed
* RST Flag: Marks that this is a reset message
* SYN Flag: Marks that this is a synchronise message or acknowledgement of a synchronise
* FIN Flag: Marks that this is a finalise message or acknowledgement of a finalise
* Window: The current acceptable window size (sent in acknowledgement messages)
* Checksum: A checksum over the segment, used to check for corruption
* Urgent Pointer: Position of where the urgent data ends inside the segment
* Options: Various TCP options (varies in length)