Transport Layer

TCP/IP Protocol Stack

The OSI model is more of an academic thing. It was initially intended to be a real network protocol stack, but it is never fully implemented anywhere. It is just a guide for implementing a protocol stack. The TCP/IP stack, however, is implemented.

Layers	Protocols	Addresses	Communication Type	Data Name
Application				
Transport	TCP	Ports (16 bit)	Process to process	Segment
Network	IP	IPv4 (32 bit)	End to end	Datagram
Link	(MAC) Ethernet	MAC Addresses (48 bits)	Node to node	Frame
Physical				

The connections between layers are called **access ports**.

Port Numbers

Used at transport layer.

Well-Known

0 – 1023

Examples:
21 – FTP
23 – SSH
80 – HTTP

Registered

1024 – 49151

Dynamic (Ephemeral)

49152 – 65535

Other Protocols at the Transport Layer

UDP – User Datagram Protocol
UDP is connectionless and unreliable and is much easier to implement.

TCP is connection oriented and is considered reliable. It keeps on trying to send data until it gets to the receiver correctly. It does do flow control and error control. Flow control is done using a sliding window type of protocol that is a little different than HDLC. Every byte will have a sequence number.

UDP Data Format

8-byte header, followed by the data

Composed of four fields:

Source Port Number – 16 bits
Destination Port Number – 16 bits
Total length – 16 bits
    Header + Data
Checksum – 16 bits
    Used for error detection

If it receives a segment with an error, it gets discarded, but no one gets notified of it.

The checksum is performed on everything, not just the header, but the header and the data.

TCP Header

More complicated – see handout

Source Port Number (16 bits)

Identifies the application on the source end

Destination Port Number (16 bits)

Identifies the application at the remote end

Sequence Number (32 bits)

Used to identify the first byte in the data that is being sent.

Socket – combination of the IP address and the port number (application that expects the data to be returned to it)

In TCP, you don't have to start off with 0 as the first sequence number. This is something that is negotiated when transmission begins (in link establishment).

Acknowledgement Number

Tells which sequence number it expects to get next.

This only has meaning if the ACK flag is set to a 1.

Used for flow control

Data Offset

The number of 32-bit words in the header. (Header length)
Typically, this is 5.

Reserved

Always 0

Control Field

Flags:

- **ACK –** If it is equal to 1, it means that the acknowledgement field is a meaningful number and contains valid data
- **SYN –** Synchronization. Used during link setup
- **FIN –** When the sending device has no more data to send, it will set this flag to 1. It is used to indicate that the source has no more data.
- **PSH** - Informs the destination application that the segment has to be responded to immediately. (The application has to deliver the segment immediately.) It will bypass all the stuff in the queue and go right to the top.
- **URG –** Kind of like PSH. Indicates that the urgent pointer field is valid. The urgent pointer points to the end of the urgent data.

Window

Iindicates how many bytes the receiver can accept.

There is an initial window size that is agreed upon when the link is established.