Transmission Control Protocol — TCP

- Transmission Control Protocol (TCP) is the main user-visible part of the TCP/IP protocol suite.
- TCP “segments” are carried as payload within IP packets.
- Most (but not all) user services (e-mail, FTP, etc.) rely on TCP.

TCP is a transport-level protocol, with the primary responsibility of providing reliable, sequenced, data delivery from the unreliable IP datagram service. Thus TCP must —

- Provide *stream* data – bits in become bits out, *exactly*.
- Recover from lost packets, and duplicated packets
- Provide a full duplex Virtual Circuit service to the user.

The TCP segment

- The user submits data to TCP, in blocks of up to 65,535 bytes each.
- The TCP header (as below) is added to form a TCP *segment*
- TCP will *fragment* the segment into sizes to fit into IP *packets*
- The receiver reassembles the fragments to rebuild the TCP segment

<table>
<tr><td>0</td><td>4</td><td>8</td><td>12</td><td>16</td><td>20</td><td>24</td><td>28</td></tr>
<tr><td colspan="4">SOURCE PORT</td><td colspan="4">DESTINATION PORT</td></tr>
<tr><td colspan="8">SEQUENCE NUMBER</td></tr>
<tr><td colspan="8">ACKNOWLEDGEMENT NUMBER</td></tr>
<tr><td>HLEN</td><td colspan="2">– reserved –</td><td>FLAGS</td><td colspan="4">WINDOW</td></tr>
<tr><td colspan="4">CHECKSUM</td><td colspan="4">URGENT POINTER</td></tr>
<tr><td colspan="6">OPTIONS (IF ANY)</td><td colspan="2">PADDING</td></tr>
<tr><td colspan="8">DATA</td></tr>
</table>

IP and TCP headers, combined

(shading or underlined text shows area covered by TCP checksum)

0	4	8	12		20	24	28	
VERS	HLEN	SERVICE TYPE		**TOTAL LENGTH**				
IDENTIFICATION				FLAGS	FRAGMENT OFFSET			
TIME TO LIVE		**PROTOCOL**		HEADER CHECKSUM				IP Header
SOURCE IP ADDRESS								
DESTINATION IP ADDRESS								
IP OPTIONS (IF ANY)						PADDING		
SOURCE PORT				**DESTINATION PORT**				
SEQUENCE NUMBER								
ACKNOWLEDGEMENT NUMBER								TCP Header
HLEN	**– reserved –**		**FLAGS**	**WINDOW**				
CHECKSUM				**URGENT POINTER**				
OPTIONS (IF ANY)						**PADDING**		
USER DATA								TCP Data

Ports

These are "addresses" or identifying numbers for users of TCP, just like protocols or access points at lower layers —

- The LLC layer uses the DSAP to select the service, say IP
- IP uses its protocol field to select a protocol such as TCP (or ICMP etc.)
- TCP uses the Destination Port to select the end-user, e.g. FTP or e-mail

Within TCP itself

- Small port numbers (<1024) are reserved for defined or "well known" services, such as e-mail, FTP, Telnet, Name Server (see RFC 1340).
- The message sender must put the appropriate Source Port number in the TCP segment.

Sequence Numbers

TCP uses a sliding window protocol, with a variable window size. Sequencing is by a byte sequence count.

The sequence number is the sequential number of the first byte of this segment. (The very first byte is given a *random* sequence number.)

- The acknowledgement number is the number of the next byte which the receiver expects. It acknowledges all preceding bytes of the data stream. This field is used only if the ACK flag is set.
- Note that if the acknowledgement from one segment is lost, a later acknowledgement may well replace it, with no error seen.

The Header Length is the length of the TCP header, in 32-bit “words”, of all the options field(s), OR is the offset, in words, where the data starts.

TCP Flags

There are 6 flags —

- URG Urgent pointer field is valid
- ACK The Acknowledgement number is valid
- PSH The data so far is to be forwarded immediately, even if the segment is incomplete
- RST Reset the connection
- SYN Synchronise sequence numbers, especially for start-up
- FIN Finish of byte stream (ie close connection)

Window

Used for flow control, it tells how many bytes can be received beyond those acknowledged. The sender knows how many it has actually sent when it receives the acknowledgement, and therefore how many beyond that can still be received. The receiver may vary the window size.

Checksum

Take the 1-s complement sum of the segment (as 16 bit words) and store its complement as the checksum. (The checksum includes a 32-bit “pseudo-header” including the IP addresses, protocol and length – see later.)

Urgent Pointer

If URG = 1 this gives length of urgent data at the start of the segment. It will be acted on immediately at the receiver, bypassing normal data.

Pseudo-Header

- The TCP checksum (and the UDP checksum later) covers
  - the *user payload and TCP header (as expected)*,
  - the *source IP address*,
  - the *destination IP address*,
  - the *segment length* (TCP) or *packet length* (UDP)
  - the *protocol type*

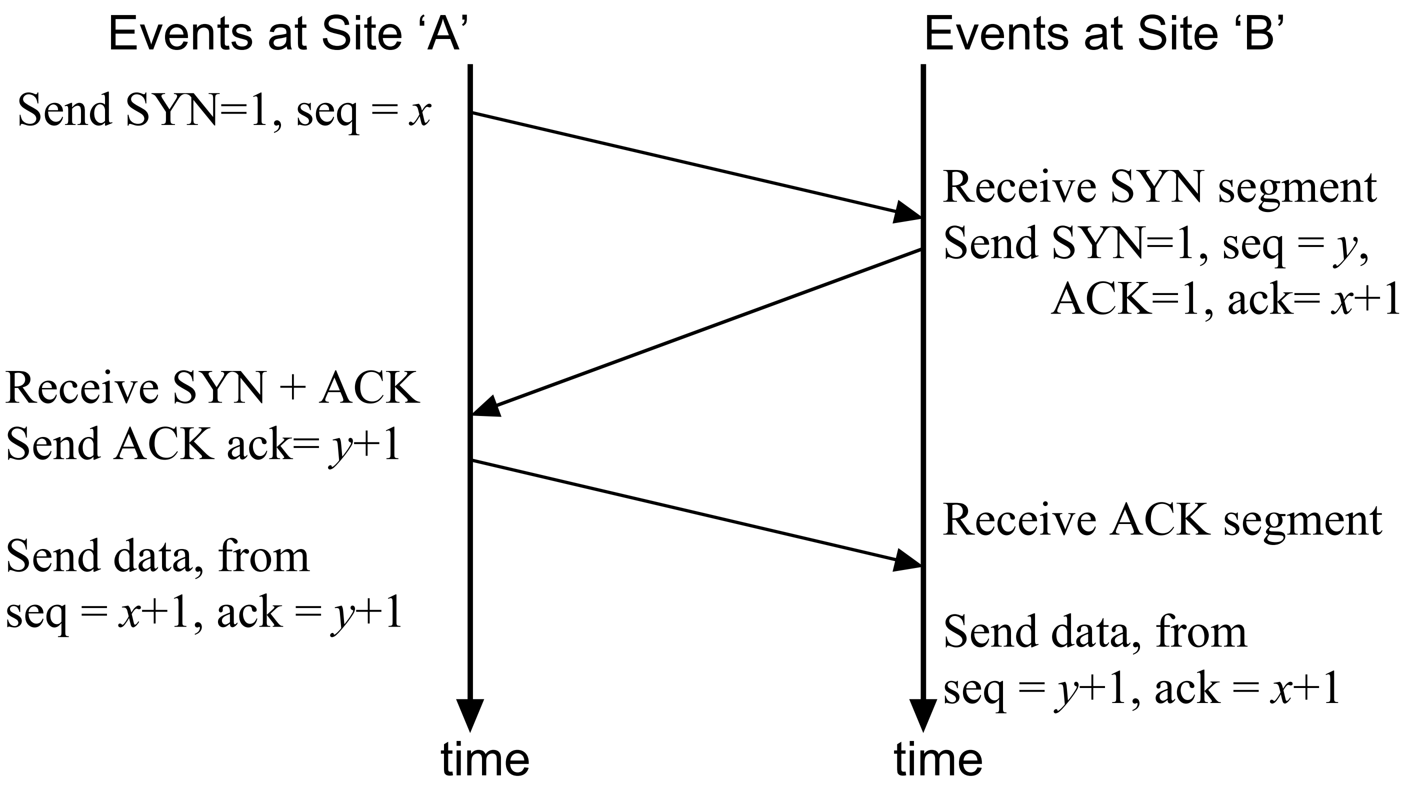
  The last four items are known as the "pseudo header"

0	8	16	31
SOURCE IP ADDRESS			
DESTINATION IP ADDRESS			
ZERO	PROTOCOL	TCP LENGTH	

- Including the addresses protects against misrouted datagrams
- Including protocol and length protects against other message corruption. (The 1's complement checksum does not detect missing 0-words.)

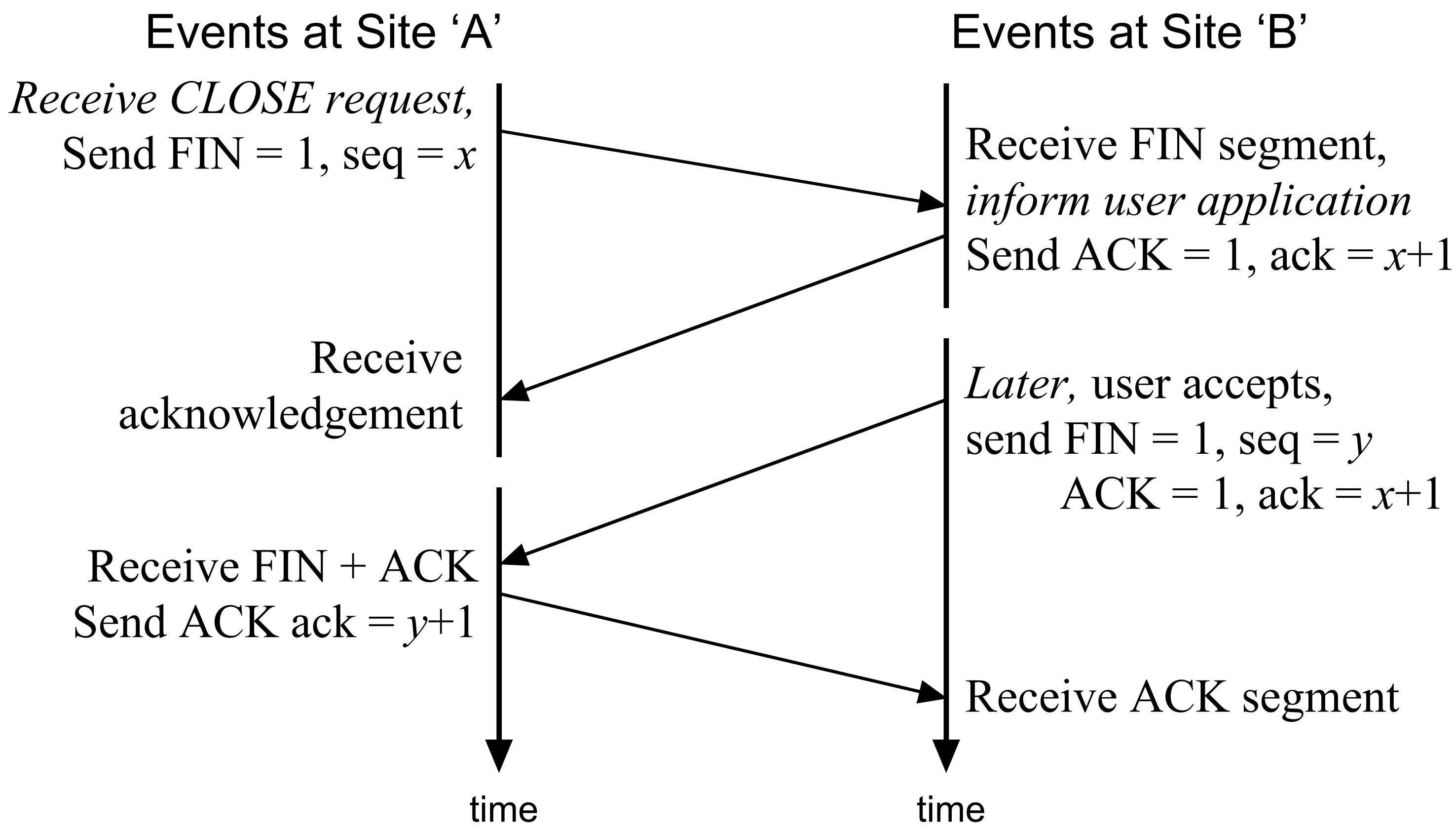
TCP Connection Protocol

- TCP uses a modified 3-way handshake, to guard against lost or duplicated packets or segments. (See Shay 2ed pp 522,523, pp 36,37 or 3ed p 578)
- Sequence numbers are chosen at random, *not starting from 1*



TCP Disconnection Protocol

Again, use a modified 3-way handshake, to guard against lost or duplicated packets or segments



TCP Flow Control

TCP uses a form of credit flow control, or window advertisement.

- At connection establishment, the end-nodes (A $\rightarrow$ B) agree on window sizes, a *transmission window* (T) in A and a *reception window* (R) in B.
- A also maintains a *congestion window* (C), $0 \le C \le T$, initially $C = 1$.
- As transmission proceeds and acknowledgements are received, C is increased by 1 *for each acknowledged segment*, as long as $C \le T$. (The effect is to double C for each window of segments.)
- If however a packet is lost (time-out, ICMP notification), A will immediately halve the congestion window (C=C/2), (but watch for multiple congestion notifications).
- A then "ramps" C up again, until either we get congestion, or $C = T$.
- At any time, B can advise a different transmission window.

Congestion control in Internet

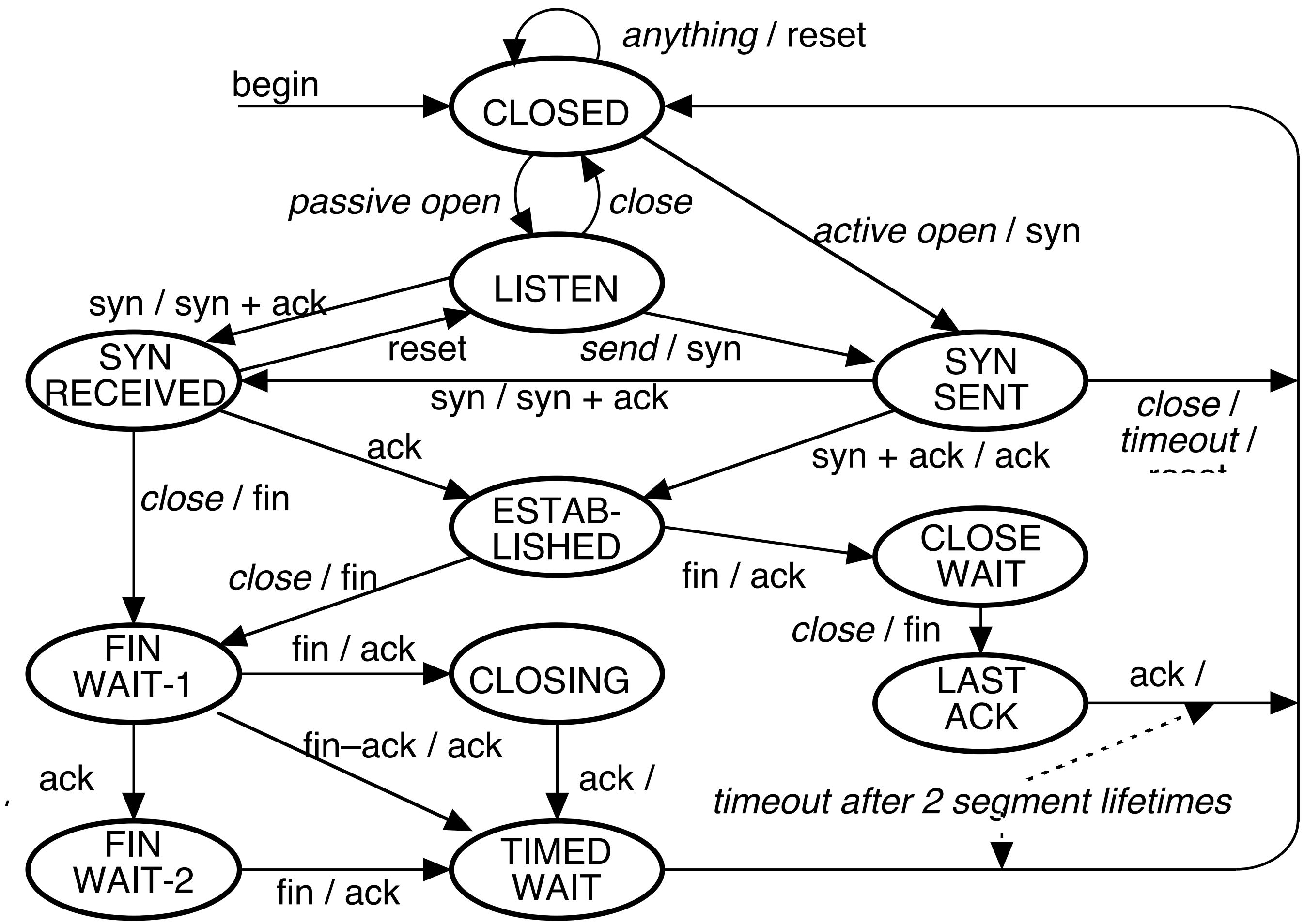
1. A node becomes congested.
2. An incoming IP packet (not necessarily related to congestion) arrives at the congested node. It *may be* selected at random and discarded
3. An ICMP "SOURCE QUENCH" packet is returned to the original source with the important details from the selected IP packet.
4. The source receives the ICMP packet and passes it to TCP as a congestion notification.
5. TCP immediately halves its Congestion Window to reduce its traffic into the network to try to avoid congestion.
6. Later acknowledgements allow TCP to "wind up" its Congestion Window", back towards its previous, larger, value.

TCP User Actions

See the text for a complete list of requests and indications, but this is a representative sample.

Action Name	Reqst	Indic'n	Confirm	Description
Active-Open	•			This user initiates a connection
Active-Open w/Data	•			Initiate connection, with data
Open Success			•	The previous Active-Open succeeded
Full Passive Open	•			Can receive conn. requests, specified site
Unspec Passive Open	•			Can receive conn. requests, any site
Close	•			Request to close connection
Closing		•		Other site requested close
Deliver		•		Data has been received
Send	•			Request to send data
Terminate			•	The connection has ended

The TCP state diagram is rather nasty (and need not be learned!)



Labels on the transitions show the "*input or actions* / output" (output may be empty).

User Datagram Protocol (UDP)

- UDP is really just an extension of IP to handle user data.
- It adds to the data an 8-byte header with the two ports, a length and a checksum (also using a pseudo-header, as for TCP).
- The additions are just enough to allow multiple users of the IP service and to provide checking of the data.

0 4 8 12	16 20 24 28
SOURCE PORT	DESTINATION PORT
UDP MESSAGE LENGTH	UDP CHECKSUM
USER DATA	

- UDP guarantees that those packets which are delivered are reliable, but that is about all – there is no check for non-delivery.

TCP versus UDP

- TCP is used mostly for user data, which may be of variable length and for which reliable delivery is essential. even with some inefficiency

- UDP is used more for network management, where everything is under the control of well-defined software and short simple messages with minimal overhead are desirable.
  For example nodes regularly exchange routing messages; it doesn't matter much if one gets lost because the previous one will probably do nearly as well, and another one will be sent anyway in the near future.

Top TCP Ports by Octets

	TCP Port	Octets	Octets/Flow	Description
1	80	427,152,954	5,352	WWW HTTP
2	22	99,879,084	36,639	SSH Remote Login Protocol
3	1086	99,733,773	600,805	CPL Scrambler Logging
4	1022	85,940,887	1,562,561	Reserved
5	1090	77,108,633	510,653	FF Fieldbus Message Spec.
6	25	70,624,928	8,872	Simple Mail Transfer
7	1118	69,908,375	743,706	Unassigned
8	1096	53,691,660	590,018	Name Resolution Protocol
9	2027	50,566,262	632,078	Shadowserver
10	1273	46,822,652	514,534	EMC-Gateway

Top TCP Ports by Flows

	TCP Port	Flows	Octets / Flow	Description
1	21	87,397	79	File Transfer [Control]
2	80	79,805	5,352	WWW Http
3	25	7,960	8,872	Simple Mail Transfer
4	1080	6,913	5,209	Socks
5	8000	4,575	9,903	iRDMI
6	443	4,310	3,070	http protocol over TLS/SSL
7	113	3,885	84	Authentication Service
8	8888	3,251	983	NewsEDGE server TCP (TCP 1)
9	22	2,726	36,639	SSH Remote Login Protocol
10	1984	1,305	609	BB
11	8875	1,218	210	
12	53	1,060	262	Domain Name Server

Top UDP Ports by Octets

	UDP Port	Octets	Octets/Flow	Description
1	53	202,474,386	21,344	Domain Name Server
2	49406	168,869,855	2,345,414	Dynamic and/or Private Port
3	520	114,009,748	791,734	Local routing process (on site)
4	56322	107,745,704	1,496,468	Dynamic and/or Private Port
5	137	44,655,066	64,251	NETBIOS Name Service
6	123	35,685,040	109,800	Network Time Protocol
7	2910	31,389,908	429,998	TDAccess
8	4597	31,151,244	420,962	Unassigned
9	4035	30,644,080	419,781	WAP Push OTA-HTTP port
10	2641	26,328,349	392,960	HDL Server
11	4239	26,302,920	398,529	VRML Multi User Systems
12	6901	25,289,808	1,580,613	Unassigned

Top UDP Ports by Flow

	UDP Port	Flows	Octets/Flow	Description
1	53	9,486	21,344	Domain Name Server
2	27015	2,290	489	Half Life game
3	33450	2,103	40	Traceroute
4	33449	2,087	40	Traceroute
5	33447	2,077	40	Traceroute
6	33437	2,076	39	Traceroute
7	33444	2,071	39	Traceroute
8	33441	2,063	39	Traceroute
9	33440	2,062	40	Traceroute
10	33442	2,061	40	Traceroute