

CSBRIDGE – NET 5

PROF. KATZ

USER DATAGRAM PROTOCOL (UDP)

- A simple, connection-less layer 4 protocol for the transmission of non-critical data
- No reliability, No confirmation of delivery, No order of packets!
- Very Light protocol which sits on top of IP layer.
- Used by a number of ULPs including DNS, DHCP, and BootP.

PORT NUMBERS

- UDP introduces the concept of port numbers as a layer 4 address.
- Ports are used to indicate which layer 7 service should receive the packet and which produced it.
- When a server program starts it “binds” to the port it will use for communication (usually <1024), UDP then knows that data received for that port should be delivered to that program.

PORTS (CONT.)

- When a client service begins communication it can either request a specific port for communication or use a unique port provided by the OS.
- Server ports are static, client ports may be dynamic.

COMMON UDP PORT NUMBERS

- 67 – BootP Server
- 68 – BootP client
- 69 – Trivial FTP

UDP HEADER

0	1	2	3	4	5	6	7	8	9	0	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	3	3	
Source Port																Destination Port															
Length																checksum															
Data																															

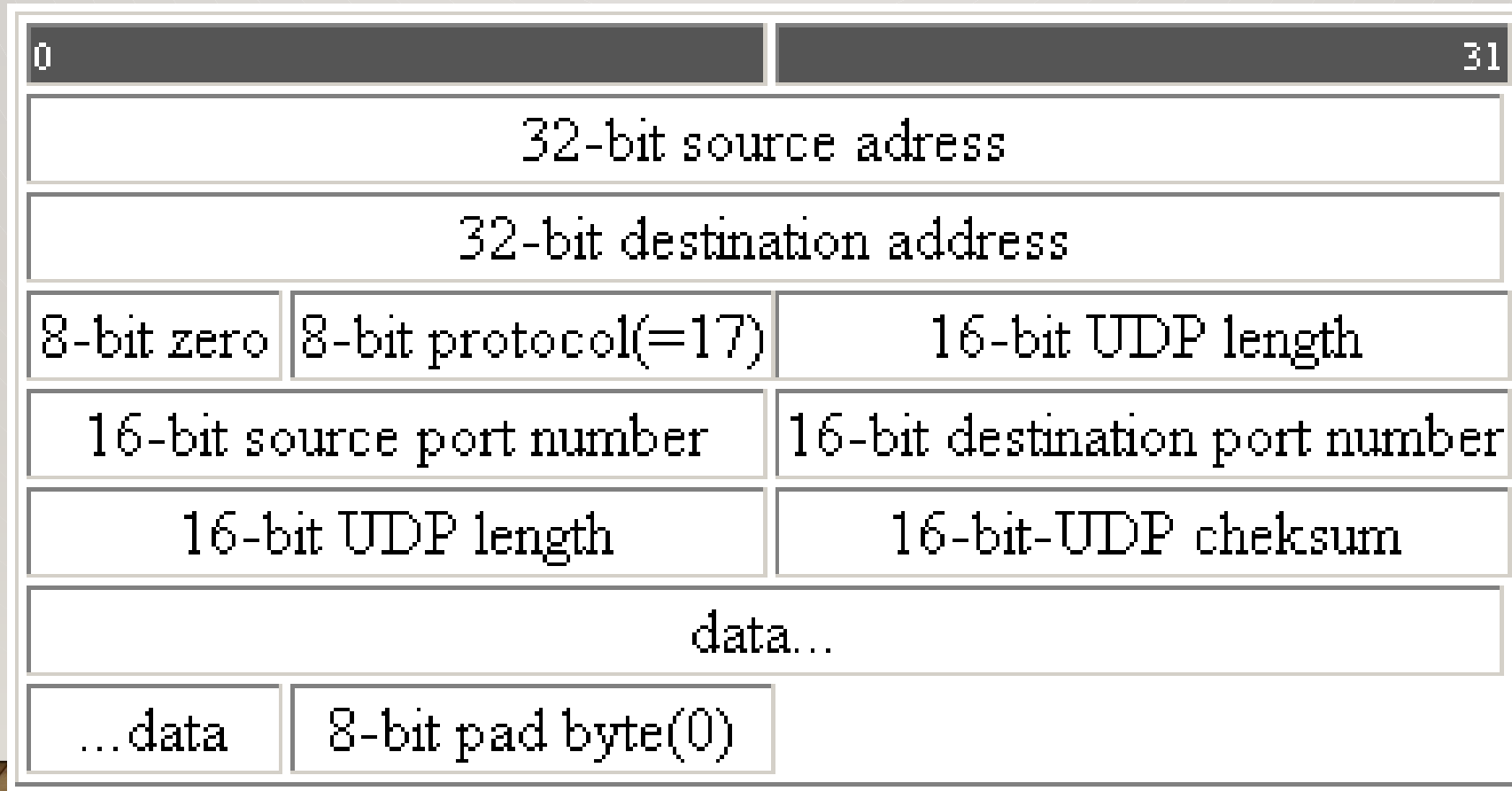
HEADER VALUES

- Source port – The port on the source host which generated this message
- Destination Port – The port on the destination host which should receive this message
- Length – The length of the header and data

UDP CHECKSUM

- The UDP checksum is calculated across the data and the header.
- The Checksum is the one's compliment of the 16-bit sum of the words in the header and data.
- Additionally the 32-bit source and destination, the 8 bit protocol field, and the UDP length are included in the sum.

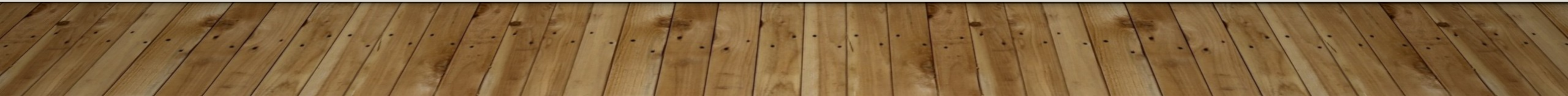
UDP GENERATED HEADER



UDP CHECKSUM

- On the receiving end all words are added to the checksum and the result should be all ones.
- If the result is not all ones then an error has occurred.

UDP IN ACTION (DNS DEMO)



TRANSPORT CONTROL PROTOCOL

- Provides a connection oriented layer 4 service
- Provides guarantees
- Provides sequence ability

TCP HEADER

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	2	2	2	2	2	2	2	2	2	3	3
Source Port																Destination Port																	
Sequence Number																																	
Acknowledgement Number																																	
Header len				reserved						U R G	A C K	P S H	R S T	S Y N	F I N	Window size																	
Checksum																Urgent Pointer																	
Options + padding																																	
Data																																	

TCP HEADER (CONT.)

- Source and destination ports – Which application produced the packet and which should receive it
- Sequence number – Identification of the relative location of the first byte in the packet to the first byte in the stream.
- Acknowledgement number – The sequence number of the next byte which the receiving station expects

TCP HEADER (CONT.)

- Urgent
- Acknowledgement
- Push
- Reset
- Synchronize sequence numbers
- Final packet

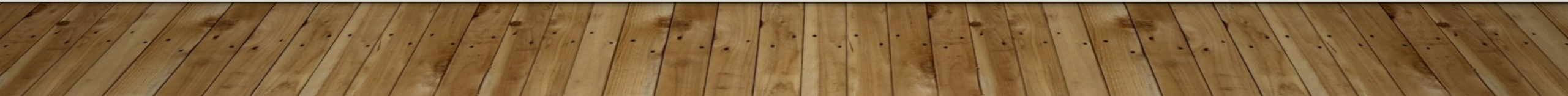
TCP CONNECTION ESTABLISHMENT

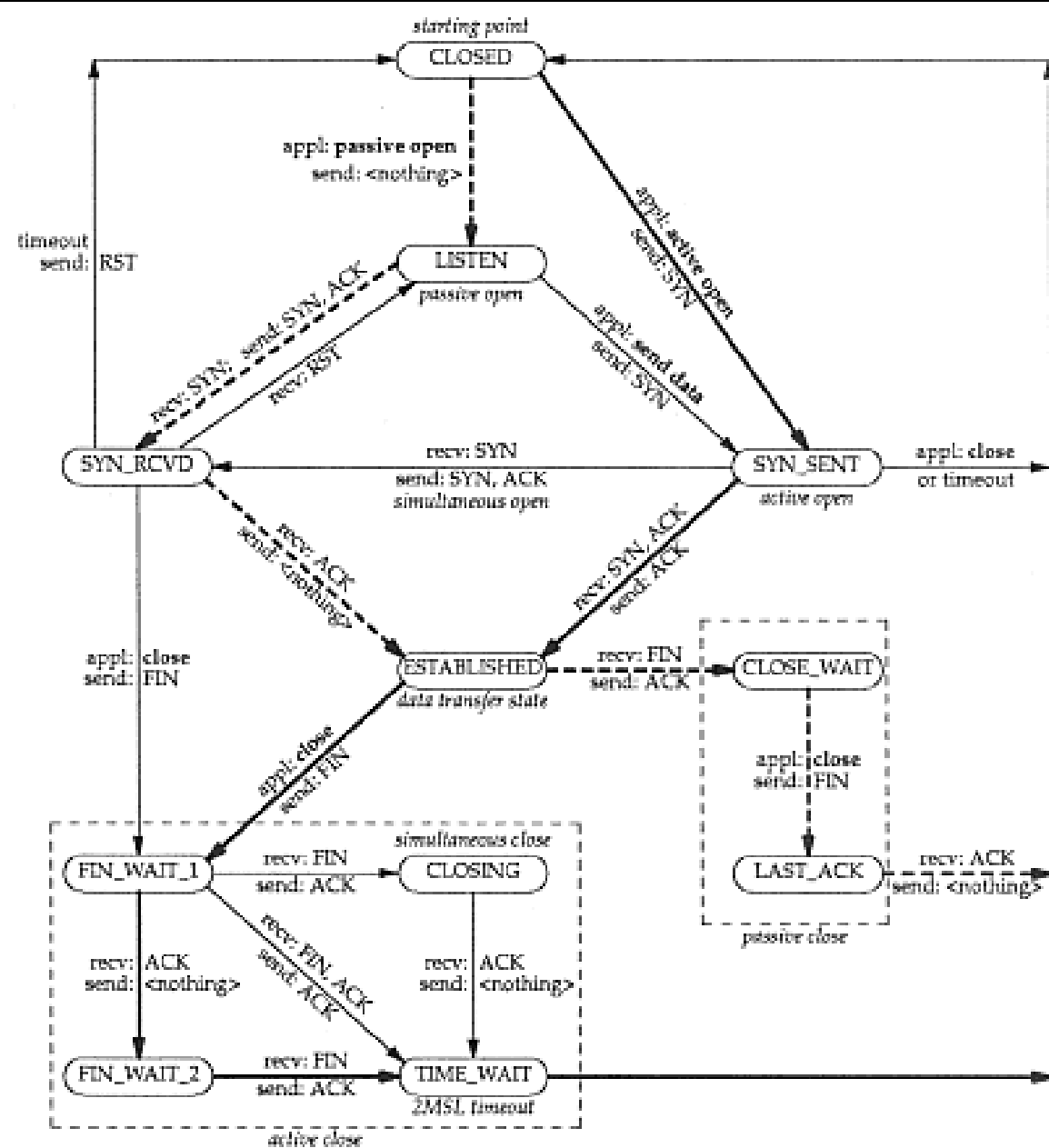
- Client Sends SYN
- Server Replies with SYN-ACK
- Client Replies with ACK

TCP CONNECTION MAINTENANCE

- The Sequence number is incremented for every packet sent
- An acknowledgement must be made before the window size is reached.
- An ack is sent for the next byte which the client expects to receive.

TCP STATE DIAGRAM





—→ normal transitions for client
 - - - → normal transitions for server
 appl: state transitions taken when application issues operation
 recv: state transitions taken when segment received
 send: what is sent for this transition