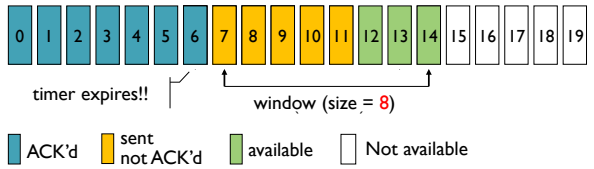
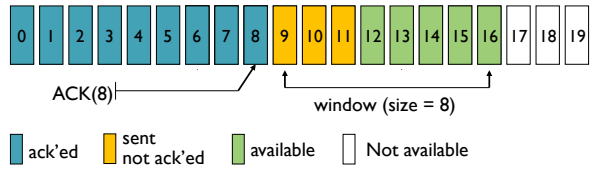
Go Back N



- Very simple receiver, only accepts packets that arrive in-order and discards others.
- Send each packet in window in turn, window moves on when ***first*** packet in window is acknowledged.
- Timeout on first packet, then all *UNACKNOWLEDGED* packets resent

16

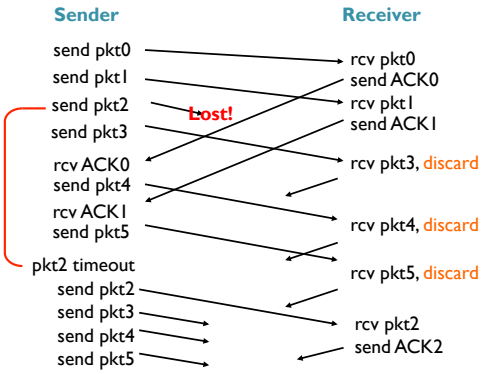
Go Back N (Cumulative ACK)



- Should a later packet be ACKed, consider ***all prior*** packets in the window to also be ACKed, so in other words:
  - ACK is cumulative, so an ACK for a later sequence number effectively ACKs all preceding packets.
- Lost ACK(6) and ACK(7)
- Get ACK(8)

17

Go Back N in action



Sender's sliding window size is 4

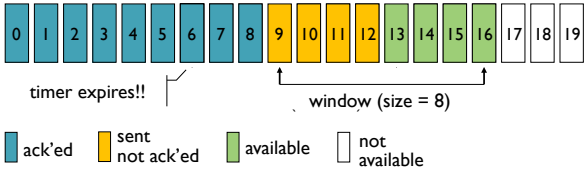
18

Quick exercise

- In a Go-Back-*N* protocol, if the window size is 63, what is the minimum range of sequence numbers?
  - A. 0-62
  - B. 0-63
  - C. 0-64
  - D. 1-63

19

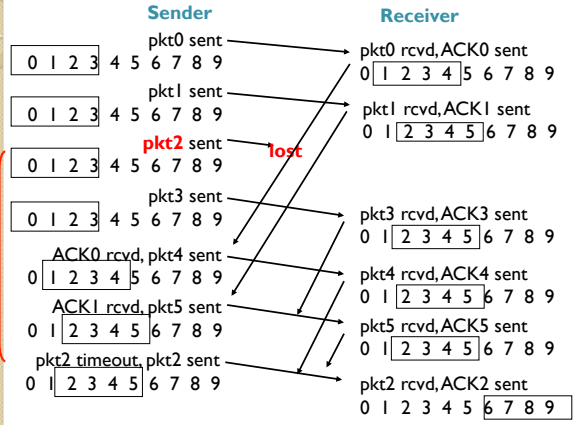
Selective Repeat



- Can send fewer packets at the cost of making the protocol more complicated.
- Packets are individually acknowledged.
- Only one packet resent on timer expiry.

21

Selective Repeat in Action



22

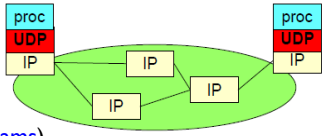
Quick exercise

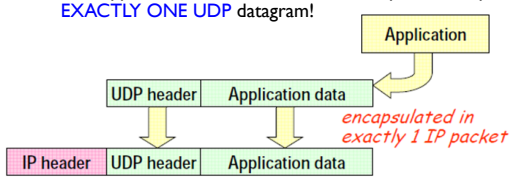
- In Selective Repeat, if 5 is the number of bits for the sequence number, then the maximum size of the sliding window must be _____
  - A. 15
  - B. 16
  - C. 31
  - D. 1

23

User datagram protocol

- Connection-Less
  - (no handshaking)
- UDP packets (Datagrams)
  - Each application interacts with UDP transport sw to produce EXACTLY ONE UDP datagram!



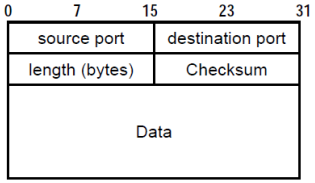


This is why, improperly, we use the term UDP packets

25

UDP datagram format

8 bytes header + variable payload



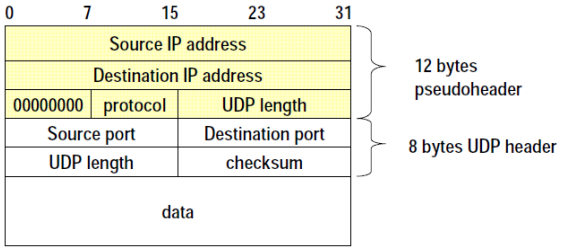
- UDP length field
  - all UDP datagram
  - (header + payload)
- payload sizes allowed:
  - Empty (0)
  - 65527 bytes (65535-8)

- UDP functions limited to:
  - Error checking
    - which may even be **disabled for performance**
  - addressing
    - which is the only strictly necessary role of a transport protocol

26

IPv4 pseudo header

- The pseudo header is only used for the checksum computation



27

Quick exercise

- Which of the following is not true about disabling checksum?
  - A. Checksum in UDP is not always necessary since IP packet checksum includes packet payload.
  - B. UDP checksum is often disabled to speed up implementation.
  - C. Lack of UDP checksum is tolerable for communication in LANs.
  - D. Lack of UDP checksum is definitely dangerous in the Internet.

28

UDP: a lightweight protocol

- No connection establishment
  - no initial overhead due to handshaking
- No connection state
  - greater number of supported connections by a server!
- Small packet header overhead
  - 8 bytes only vs 20 in TCP
- originally intended for simple applications, oriented to short information exchange
  - DNS
  - management (e.g. SNMP)
  - Distributed file system support (e.g. NFS)
  - etc

29

Unregulated send rate in UDP

- No rate limitations
- No re-transmission
- Extremely important features for today multimedia applications
- *Be careful: UDP ok for multimedia because it does not provide anything at all (no features = no limits!).*

30

Quick exercise

- Which of the following about UDP is correct?
  - A. UDP supports a self-regulating "throttle" feature that prevents network saturation
  - B. UDP consumes fewer computer resources by not maintaining connection state
  - C. UDP guarantees that individual packets of a transmission will arrive "in order"
  - D. None of the above

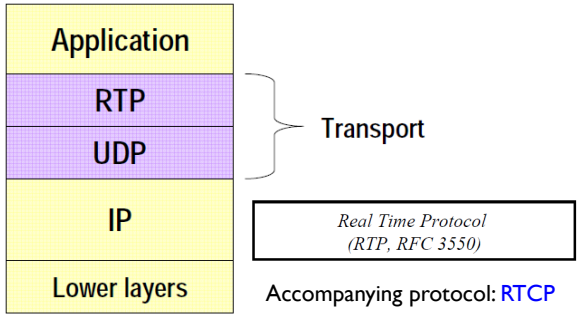
31

Question to ponder

- Why would a TCP and UDP "phone call" likely have equivalent performance characteristics in practice?

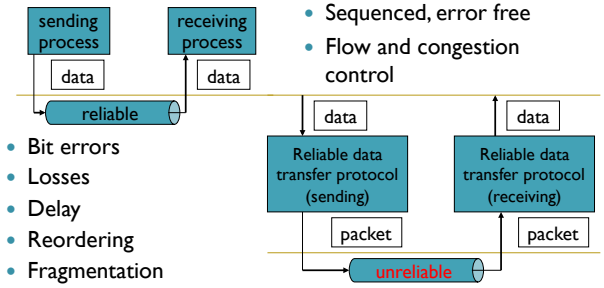
32

RTP: sublayer of transport



33

Connection Oriented Transport

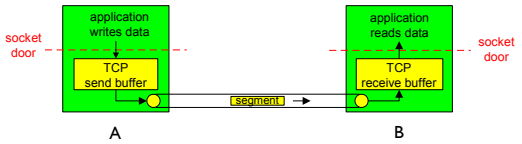


- Process to process
- Sequenced, error free
- Flow and congestion control

- Bit errors
- Losses
- Delay
- Reordering
- Fragmentation

TCP

- point-to-point:
  - one sender, one receiver
  - no multicast
- reliable, in-order *byte steam:*
  - no "**message boundaries**"
- full duplex data:
  - bi-directional data flow in same connection
- connection-oriented:
  - handshaking (exchange of control msgs) init's sender, receiver state before data exchange



Quick exercise

- TCP does not support virtual circuit, why?
  - A. TCP relies on IP and IP does not support virtual circuit
  - B. TCP maintains connection information at two communicating end systems only
  - C. Virtual circuit can only be established at the network layer
  - D. None of the above

39

The TCP Header

- Byte stream sent as sequence of segments
  - Segment may be 0 to 64k bytes

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0			NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size															
16	128	Checksum																Urgent pointer (if URG set)															
20 ...	160 ...	Options (if Data Offset > 5, padded at the end with "0" bytes if necessary) ...																															

41