Latency：
- Propagation delay : distance / transmit speed(光速)
- Transmission delay : size / bandwidth
- Queueing delay

考古：x-bit message over a k-hop path in a circuit switched network and in a lightly packet switched network. The circuit setup time is s sec, the propagation delay is d sec per hop, the packet size is p bits, and the data rate of each link is b bps. Under what conditions does the packet switched network have a lower delay?

(Circuit switched) $s+kd+xb$

(Packet switched) $kd+xb+(k-1)pb$

Packet switched network has a lower delay if $s>(k-1)pb$

※Seven layers(1>7)：Physical > Data link > Network > Transport > Session > Presentation > application

UDP: User Datagram Protocol

→ "no frills," "bare bones" Internet transport protocol

→"best effort" service, UDP segments may be: lost or delivered out-of-order to app

→connectionless:

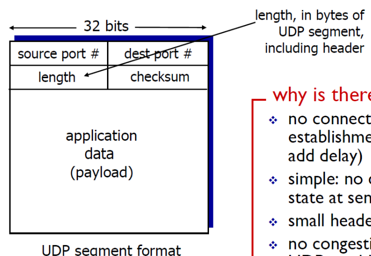
└no handshaking between UDP sender, receiver

└each UDP segment handled independently of others

TCP features：

Point-to-point, reliable, in-order byte stream, pipelined, connection-oriented, flow-controlled, full duplex data, send & receive buffers, packet-switching, fairness,

UDP: segment header



UDP segment format

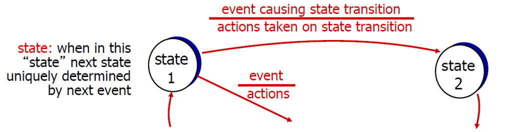
why is there a UDP?
- no connection establishment (which can add delay)
- simple: no connection state at sender, receiver
- small header size
- no congestion control: UDP can blast away as fast as desired

Sum：將資料以 n 個 bit 為單位切成多份後相加，如果溢位則 LSB+1

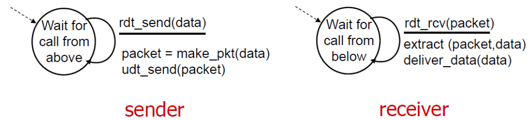
Check sum：sum 的 1's complement

Rdt : Reliable data transfer protocol

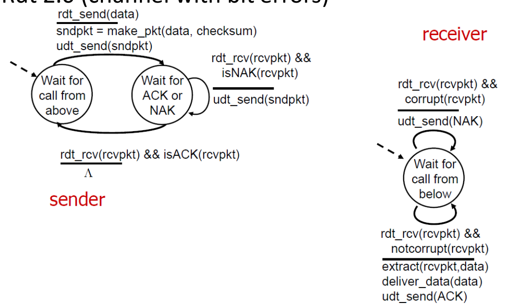
>use finite state machines (FSM) to specify sender, receiver



Rdt 1.0 (下圖)



Rdt 2.0 (channel with bit errors)



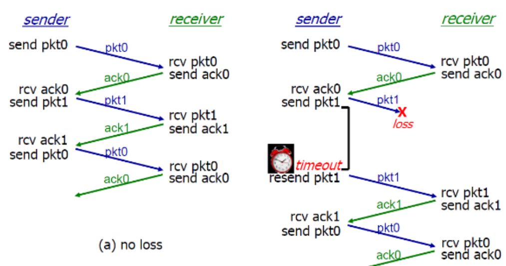
缺點：sender 不知道 receiver 狀況，可能重複傳造成 duplication

Solution：add sequence number to each pkt

Rdt 2.1, 2.2 都略過><

rdt3.0: channels with errors and loss

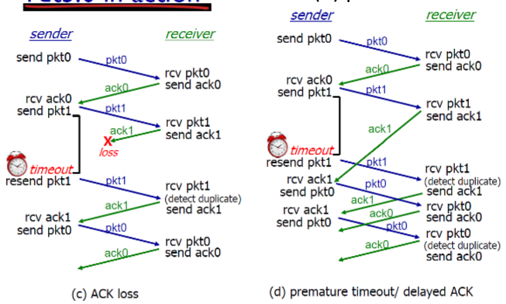
rdt3.0 in action



(a) no loss

↑(b) packet loss

rdt3.0 in action



(c) ACK loss

(d) premature timeout/ delayed ACK

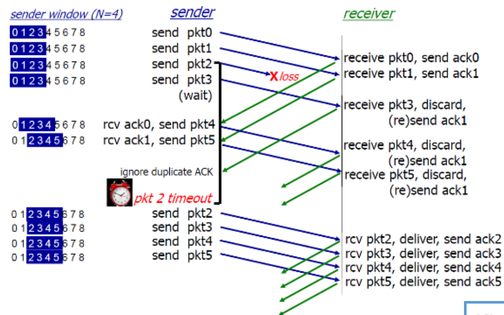
rdt 3.0之後的下一步：pipeline

Pipelined protocols: overview

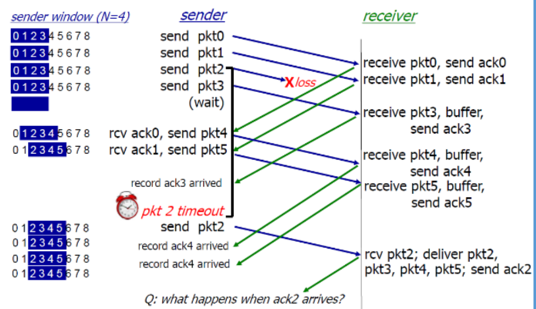
Go-back-N:
- sender can have up to N unacked packets in pipeline
- receiver only sends *cumulative ack*
  - doesn't ack packet if there's a gap
- sender has timer for oldest unacked packet
  - when timer expires, retransmit *all* unacked packets

Selective Repeat:
- sender can have up to N unack'ed packets in pipeline
- rcvr sends *individual ack* for each packet
- sender maintains timer for each unacked packet
  - when timer expires, retransmit only that unacked packet

GBN in action



Selective repeat in action



Rwnd : receiver window

Cwnd : congestion window

TCP congestion control:

additive increase

multiplicative decrease

>approach: sender increases transmission rate (window size), probing for usable bandwidth, until loss occurs

>additive increase:

increase cwnd by 1 MSS every RTT until loss detected

>multiplicative decrease:

cut cwnd in half after loss

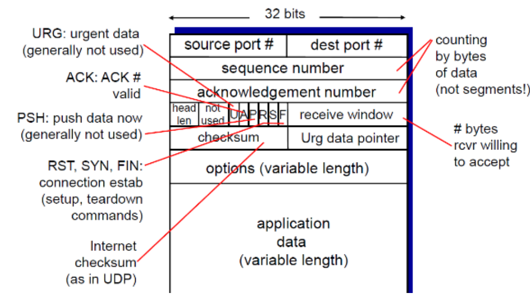
Internet transport protocols services

TCP service:
- *reliable transport* between sending and receiving process
- *flow control:* sender won't overwhelm receiver
- *congestion control:* throttle sender when network overloaded
- *does not provide:* timing, minimum throughput guarantee, security
- *connection-oriented:* setup required between client and server processes

UDP service:
- *unreliable data transfer* between sending and receiving process
- *does not provide:* reliability, flow control, congestion control, timing, throughput guarantee, security, orconnection setup,

Q: why bother? Why is there a UDP?

TCP segment structure



以下為 congestion control

- sender limits transmission:

  LastByteSent- LastByteAcked ≤ cwnd

- **cwnd** is dynamic, function of perceived network congestion

TCP sending rate:
- *roughly:* send cwnd bytes, wait RTT for ACKS, then send more bytes

  rate ≈ $\frac{\text{cwnd}}{\text{RTT}}$ bytes/sec