INP Grenoble DÉPARTEMENT TÉLÉCOMMUNICATIONS

Computer Networking

Reliable Transport

Prof. Andrzej Duda
duda@imag.fr

http://duda.imag.fr

1

Reliable Transport

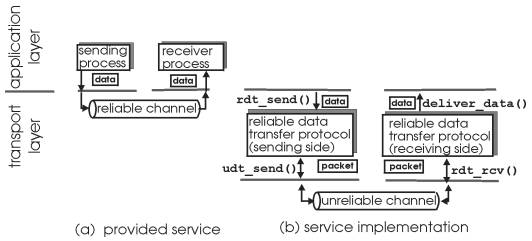
Reliable data transfer
- Data are received ordered and error-free
- Elements of procedure usually means the set of following functions
  - Error detection and correction (e.g. ARQ)
  - Flow Control

Automatic Repeat reQuest
- Sliding window
- Error and Loss detection
- Acknowledgements: short control packets
- Retransmission Strategies
  - Stop & Go
  - Go Back N
  - Selective Repeat

2

Principles of reliable data transfer

- important in app., transport, link layers
- data are received ordered and error-free

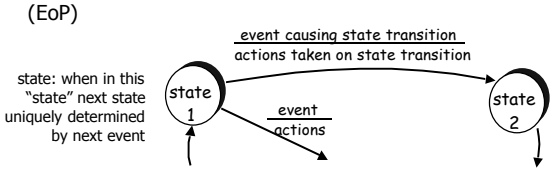


- characteristics of unreliable channel will determine complexity of reliable data transfer protocol (rdt)

3

Reliable data transfer

Our approach:
- analyze reliable data transfer protocol (rdt)
- use finite state machines (FSM) to specify the model
- introduce gradually the Elements of Procedure (EoP)



4

Elements of Procedure

- Elements of Procedure transform a raw, unreliable frame transport mechanism into a reliable data pipe
  - ordered delivery of packets
  - loss-free as long as the connection is alive
  - no buffer overflow
- Elements of procedure usually means the set of following functions
  - Error detection and correction (e.g. ARQ - Automatic Repeat reQuest)
  - Flow Control
  - Connection Management
- Elements of procedure exist in
  - reliable transport (TCP for the TCP/IP architecture) (layer 4)
  - also: in reliable data link (ex: over radio channels, over modems -layer 2) as HDLC
- Congestion Control

5

ACK/NAK handling

- ACKs & NAKs: short control packets
  - cumulative versus selective
  - positive (ACK) versus negative (NAK)
- Stop and wait
  - The sender waits for an ACK/NAK
  - Only one packet at time can be transmitted
- Go back N
  - packets are transmitted without waiting for an ACK
  - All following packets are resent on receipt of NAK
- Selective repeat procedure
  - Only packets negatively acknowledged are resent

6

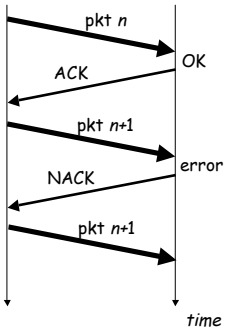
Retransmission Strategies

- underlying channel can garble and lose packets (data or ACKs)
  - checksum, seq. #, ACKs, retransmissions will be of help, but not enough
- to deal with loss & errors:
  - sender waits until certain time, then retransmits
  - duplicate
- Approach: sender waits "reasonable" amount of time for ACK
- retransmits if no ACK or NAK received in this time
- if pkt (or ACK) just delayed (not lost):
  - retransmission will be duplicate, but use of seq. #'s already handles this
  - receiver must specify seq # of pkt being ACKed
- requires countdown timer

7

Send and Wait (simple)

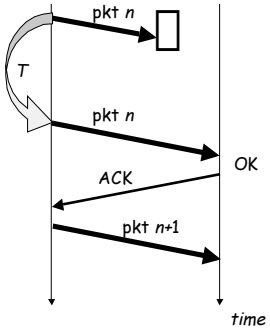
- Reliable transmission
- Flow control
  - sender can send the next packet after receiving ACK
- No losses



8

Losses

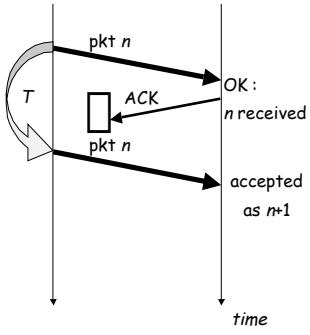
- Timer
  - if no response within a time interval, retransmission
  - the time interval must be longer than RTT



9

Problem of duplicates

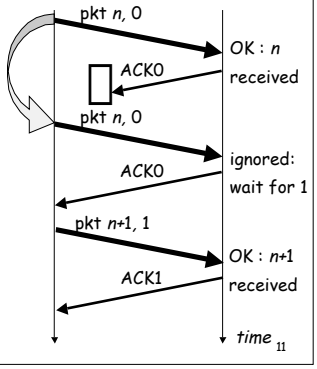
- Retransmission
  - next and retransmitted packets are confused
- Solution
  - sequence number



10

Sequence numbers

- Numbers of packets and ACKs
  - counter on k bits - mod 2^k
  - if block out of sequence, ignored
  - 1 bit is sufficient



11

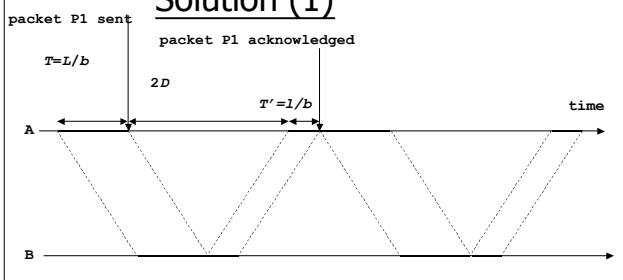
Performance

- **Question**: What is the maximum throughput assuming that there are no losses?

  notation:
  - packet length = L, constant (in bits)
  - acknowledgement length = l, constant
  - channel bit rate = b
  - propagation = D
  - processing time = 0

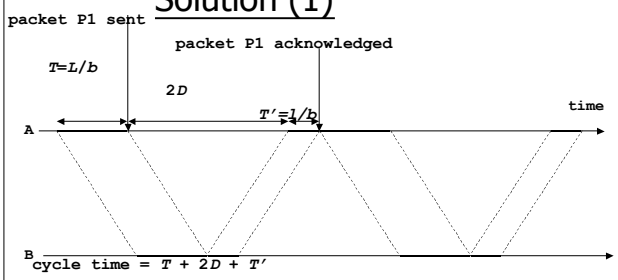
12

Solution (1)



13

Solution (1)



cycle time = $T + 2D + T'$

useful bits per cycle time = L

throughput = $Lb / (L + l + 2Db) = b / (\alpha + \beta/L)$

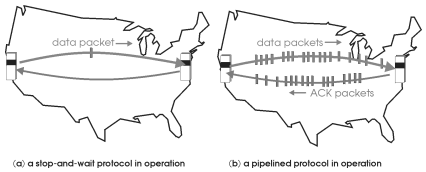
with $\alpha=(L+l)/L$=overhead and $\beta=2Db$=bandwidth-delay product

14

Pipelined protocols

Pipelining: sender allows multiple, "in-flight", yet-to-be-acknowledged pkts

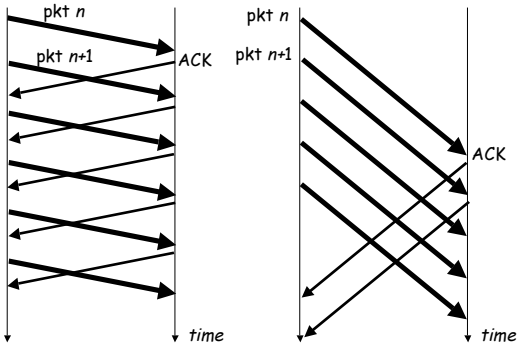
- range of sequence numbers must be increased
- buffering at sender and/or receiver



- Two generic forms of pipelined protocols: *go-Back-N, selective repeat*

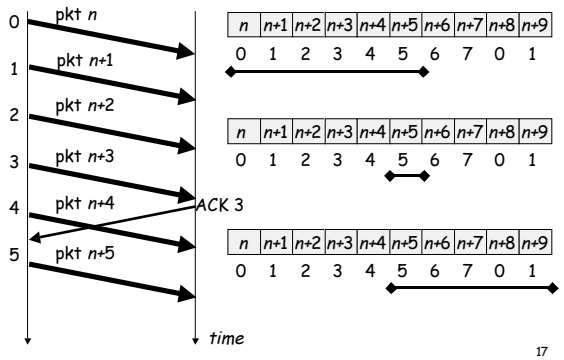
15

Window size



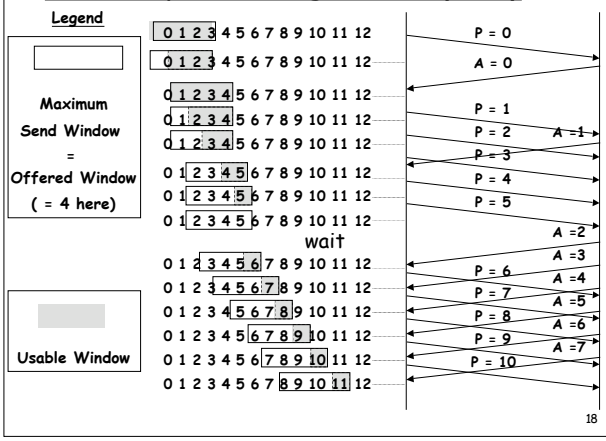
16

Sender window



17

The Principle of Sliding Window (W=4)



18

Sliding window performance

- If there are no losses
  - if the window size satisfies:
    $W \geq b / L$
    where b is the bandwidth delay product, L the packet size.
  - sliding window protocol can have a throughput of 100% of link rate (if overhead is not accounted for)
  - counted in bytes, this means that **the minimum window size for 100% utilization is the bandwidth-delay product**.

19

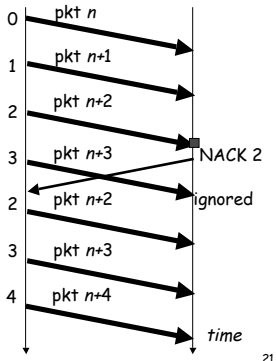
Elements of ARQ

- The elements of an ARQ protocol are:
  - Sliding window:
    - used by all protocols
  - Error detection
    - at receiver on error detection (code)
  - Loss detection
    - at sender on timeout versus at receiver on gap detection
  - Acknowledgements: short control packets
    - cumulative versus selective
    - positive (ACK) versus negative (NAK)
  - Retransmission Strategy
    - Selective Repeat
    - Go Back n
    - Others

20

Go back N (GBN)

- Retransmission of all packets starting from the bad one



21

GBN: sender

- The GBN sender must respond to three types of events:
- *packet to be sent.*
  - if the window is not full, send packet and variables are appropriately updated.
  - otherwise upper layer waits.
- *Cumulative ACK.*
  - Ack for packet n (sequence number) = all packets up to and including n have been correctly received.
- *A timeout.*
  - If a timeout occurs, resends *all* packets yet-to-be-acknowledged .
  - If ACK, timer is restarted for yet-to-be-acknowledged packets

22

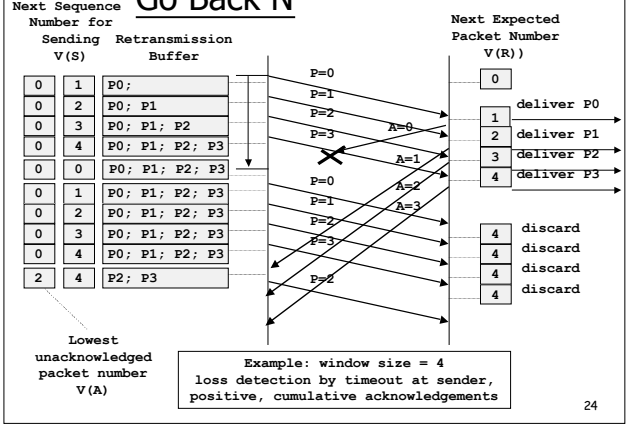
GBN: receiver

The receiver is simple:

- ACK-only: always send ACK for correctly-received pkt with highest *in-order* seq #
  - need only remember the expected seq #
  - cumulative ack
- out-of-order pkt:
  - discard (don't buffer) -> no receiver buffering! and no reordering
  - ACK pkt with highest in-order seq #

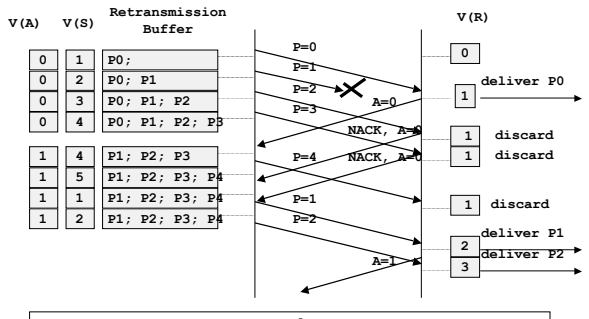
23

Go Back N



24

Go Back N with Negative Acks



Example:
window size = 4
loss detection by gap detection at receiver; negative acknowledgements

25

GBN problems

- large window size and bandwidth-delay product = many packets can wait for ACK
  - a single packet error -> retransmit a large number of packets
  - this is not necessary!
- As the probability of channel errors increases, the pipeline can become filled with these unnecessary retransmissions.

26

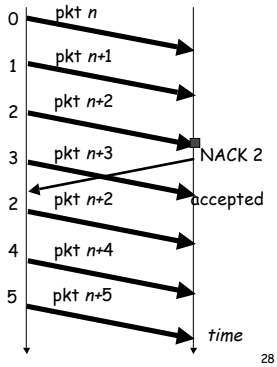
Selective Repeat Protocol (SRP)

- receiver *individually* acknowledges all correctly received pkts
  - buffers pkts, as needed, for eventual in-order delivery to upper layer
- sender only resends pkts for which ACK not received
  - sender timer for each unACKed pkt
- sender window
  - N consecutive seq #'s
  - again limits seq #s of sent, unACKed pkts

27

Selective Repeat

- Retransmission of the bad packet



28

Selective Repeat: sender

data to send from upper layer
- if next available seq # in window, send packet

timeout(n):
- resend packet n, restart timer

ACK(n):
- mark packet n as received
- if n smallest unACKed packet, advance window base to next unACKed seq #

29

Selective Repeat: receiver

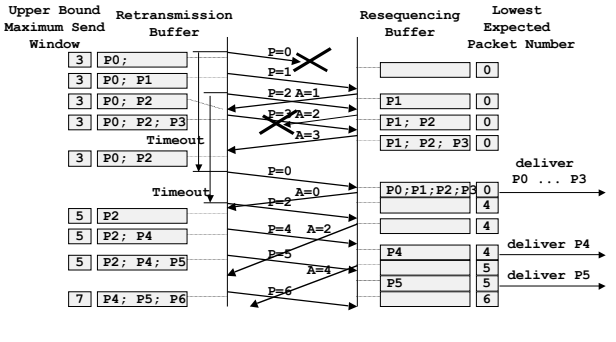
packet n expected or higher:
- send ACK(n)
- out-of-order: buffer
- in-order: deliver up (also deliver buffered, in-order packets), advance window to next not-yet-received packet

packet n smaller less than N than expected:
- ACK(n)

otherwise:
- ignore

30

Selective Repeat



31

Flow Control

- Purpose: prevent buffer overflow **at receiver**
  - receiver not ready (software not ready)
  - many senders to same receiver (overload focused on receiver)
  - receiver slower than sender
- Solutions: Backpressure, Sliding Window, Credit
- ***Flow Control*** is not the same as ***Congestion control*** (inside the network)

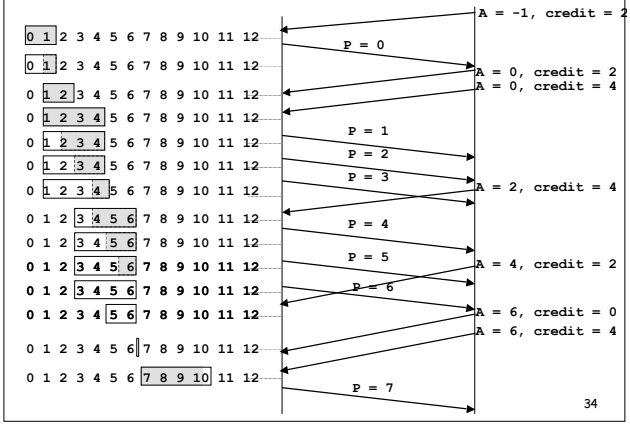
32

Sliding Window Flow Control

- Number of packets sent but unacknowledged <= *W*
- Included in SRP and Go Back N protocols
  - assuming acknowledgements sent when receive buffer freed for packets received in order
- Receiver requires storage for at most *W* packets per sender

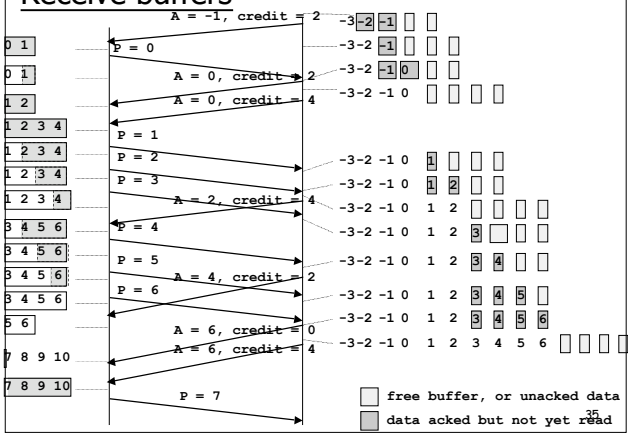
33

Credit Based Flow Control



34

Receive buffers



35

Reliable Transport - summary

- Principles behind transport layer services - reliable data transfer:
  - Sliding window
  - Error and Loss detection: ARQ procedures
  - Retransmission Strategies
    - Stop & Go
    - Selective Repeat
    - Go Back n
  - Flow control

36