EEEN20060 Communication Systems

Link Layer Protocols – part 2

Brian Mulkeen

UCD School of Electrical, Electronic and Communications Engineering

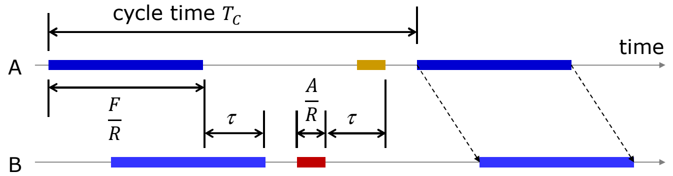
Scoil na hInnealtóireachta Leictrí, Leictreonaí agus Cumarsáide UCD

Analysis of Stop and Wait

- How well does it perform?
  - calculate *throughput* or *efficiency*
  - **throughput** U = rate of transfer of useful bits (payload) (unit: bit/s)
  - **efficiency** η = throughput as fraction of physical layer bit rate
- Define some symbols
  - D = maximum data bits in each frame
  - H = overhead bits in frame (header + trailer)
  - $F = D + H$ = max. total bits in frame
  - A = bits in acknowledgement (ACK or NAK)
  - R = physical layer bit rate
  - τ = time delay through physical layer

2

Time Used by One Frame – Cycle Time



- With everything going well…
  - time to transmit bits of frame $\frac{F}{R} = \frac{D+H}{R}$
  - delay before last bit is received τ
  - processing delay at receiver ?
  - time to transmit bits of ACK/NAK $\frac{A}{R}$
  - delay before last bit is received τ
  - processing delay at sender ?

3

Assumptions

Cycle Time $T_C = \frac{F}{R} + \tau + \frac{A}{R} + \tau = \frac{D+H+A+2\tau R}{R}$

- To simplify analysis, assume
  - no processing delays at either end
    - ACK sent as soon as frame received, etc.
  - plenty of data to waiting to be sent
    - so all frames carry full payload of D data bits
    - no delay waiting for more data to become available
  - timeout set just longer than time needed
    - so even if no reply, cycle time remains ~same
  - no enquiries – just re-send data block
- Cycle time is time used in sending one frame
  - but does it transfer useful data?

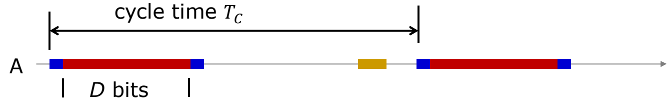
4

Probability of Success

- Success = frame received with no errors
  - and ACK received with no errors
  - so no need to re-send data
- More definitions
  - P_{SF} = probability of successful reception of frame
  - P_{SA} = prob. of successful reception of ACK
  - $P_S = P_{SF} P_{SA}$ = overall probability of success
- Multiplication of probabilities
  - independent events
  - need both to happen for overall success

5

Throughput



- With our assumptions
  - send one frame in each cycle time
  - some succeed – fraction P_S
  - if succeed, transfer D data bits
  - others fail, transfer nothing useful
- So throughput $U = \frac{DP_S}{T_C} = \frac{DP_S R}{D+H+A+2\tau R}$
  - efficiency $\eta = \frac{U}{R} = \frac{DP_S}{D+H+A+2\tau R}$

6

Efficiency Examples

- Frame: 2000 data bits, 50 overhead bits
  - physical layer 1 Mbit/s, so 2.05 ms to send
  - ACK or NAK: 50 bits, so 50 μs to send
  - probability of success 0.9
- Short link – 200 m, propagation delay 1 μs
  - cycle time 2.102 ms
  - throughput $U = \frac{2000 \text{ bits}}{2.102 \text{ ms}} 0.9 \approx$ 856 kbit/s
  - efficiency 0.856
- Long link – 5000 km, propagation delay 25 ms
  - cycle time 52.1 ms
  - throughput 34.5 kbit/s, efficiency 0.0345

7

Improving Efficiency

$$\eta = \frac{DP_S}{D + H + A + 2\tau R}$$

- Assume physical layer is fixed: τ, R, errors
  - we design link layer protocol
  - we control D, H, A
- What should we do?
- Make H, A small ?
  - but cannot reduce to zero…
- Make D very large ?
  - but P_S will depend on size of frame…
    - exact relationship depends on physical layer
  - longer frame more likely to have errors

8

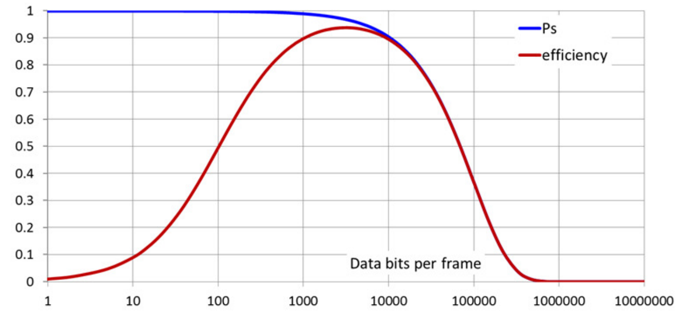
Probability of Success – Example 1

- One simple model of physical layer
  - independent decision on each bit (no bursts)
  - every bit has probability of error *p*
    - so probability of good bit is 1 – *p*
- For successful reception of frame
  - need all bits good, prob. $P_{SF} = (1-p)^{(D+H)}$
  - similarly for ACK, $P_{SA} = (1-p)^{A}$
  - so prob. overall success $P_S = (1-p)^{(D+H+A)}$
- Example as earlier (D = 2000, H = 50, A = 50)
  - $p = 10^{-5}$ gives P_{SF} = 0.9797, P_{SA} = 0.9995
  - overall prob. success P_S = 0.9792
  - $p = 10^{-4}$ gives P_S = 0.81
  - $p = 10^{-3}$ gives P_S = 0.122

9

Example

$$\eta = \frac{D(1-p)^{(D+H+A)}}{D+H+A+2\tau R}$$

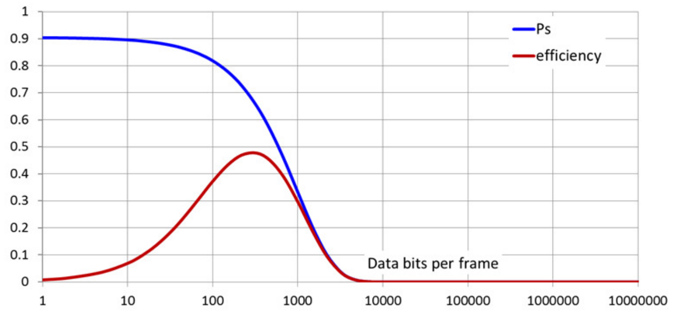


- Note broad peak – good for range of D values
  - graph is for $H = A = 50$ bit, R = 1 Mbit/s, τ = 1 µs, $p = 10^{-5}$

10

Example

$$\eta = \frac{D(1-p)^{(D+H+A)}}{D+H+A+2\tau R}$$



- Choose D to suit physical layer!
  - graph is for $H = A = 50$ bit, R = 1 Mbit/s, τ = 10 µs, $p = 10^{-3}$

11

Optimum Block Size

$$\eta = \frac{D(1-p)^{(D+H+A)}}{D+H+A+2\tau R}$$

- Differentiate with respect to D, set = 0 ?
  - D is integer, but η is continuous function of D
  - so OK to differentiate w.r.t. D
- To simplify, replace constants
  - use $B = H + A$, $C = H + A + 2\tau R$, $s = 1 - p$
  - then $\eta = \frac{Ds^{(D+B)}}{D+C}$
  - so $\frac{d\eta}{dD} = \frac{(D+C)\left[s^{(D+B)} + Ds^{(D+B)}\ln(s)\right] - Ds^{(D+B)}}{(D+C)^2} = 0$
  - $\Rightarrow (D+C)[1 + D\ln(s)] - D = 0$
  - $\Rightarrow D^2\ln(s) + DC\ln(s) + C = 0$

12

Solving…

$$D^2 \ln(s) + DC \ln(s) + C = 0$$

- quadratic equation: $D_{opt} = -\frac{C}{2} + \sqrt{\left(\frac{C}{2}\right)^2 - \frac{C}{\ln(s)}}$
- $\ln(s) = \ln(1-p) \approx -p$ for $p \ll 1$ (Taylor series)
- get $D_{opt} \approx \sqrt{\left(\frac{C}{2}\right)^2 + \frac{C}{p}} - \frac{C}{2} = \frac{C}{2}\left[\sqrt{1+\frac{4}{pC}} - 1\right]$
- note $C = H + A + 2\tau R$ = wasted bit times/cycle
  - = 102 in our 1 μs example
- with $p = 10^{-5}$, get $D_{opt} \approx 3143$ bit
- in our 10 μs example, $C = 120$ bit times
- with $p = 10^{-3}$, get $D_{opt} \approx 291$ bit

13

Probability of Success – Example 2

- Another simple model – burst errors
  - model error bursts as Poisson random process
  - average rate λ burst/s
- Poisson random process:
  - models occurrence of discrete events
  - no memory: events in non-overlapping time intervals are independent
  - in small time interval Δt, probability of exactly one event occurring $\rightarrow \lambda\Delta t$ as $\Delta t \rightarrow 0$
  - and probability of no event $\rightarrow 1-\lambda\Delta t$ as $\Delta t \rightarrow 0$
- Results:
  - in longer interval T, expect λT events
  - prob. no event in interval T is $e^{-\lambda T}$

14

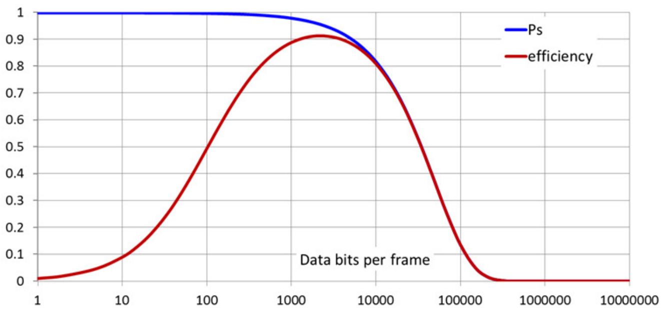
Success?

- Frame received successfully if no error burst
  - duration $\frac{D+H}{R}$, so $P_{SF} = e^{-\lambda\frac{D+H}{R}}$
- ACK received successfully if no error burst
  - duration $\frac{A}{R}$, so $P_{SA} = e^{-\lambda\frac{A}{R}}$
- Overall probability of success
  - $P_S = e^{-\lambda\frac{D+H}{R}} e^{-\lambda\frac{A}{R}} = e^{-\lambda\frac{D+H+A}{R}} = e^{-\frac{\lambda}{R}(D+H+A)}$
- Example as earlier (2000, 50, 50, 1 Mbit/s)
  - $\lambda = 2$ burst/s, $P_{SF} = 0.9959$, $P_{SA} = 0.9999$
  - overall prob. success $P_S = 0.9958$
  - $\lambda = 20$ burst/s gives $P_S = 0.959$
  - $\lambda = 200$ burst/s gives $P_S = 0.657$

15

Example

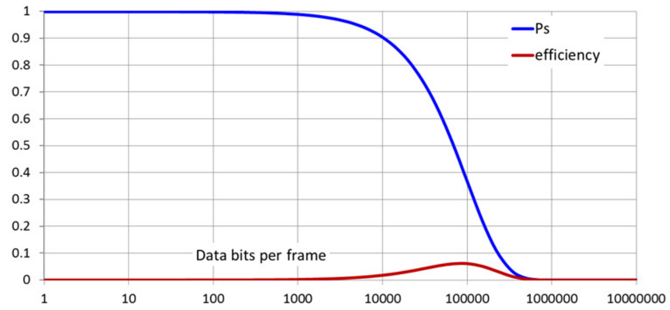
$$\eta = \frac{De^{-\frac{\lambda}{R}(D+H+A)}}{D + H + A + 2\tau R}$$



- Function has exactly same form as before
  - graph is for $H = A = 50$ bit, $R = 1$ Mbit/s, $\tau = 1$ μs, $\lambda = 20$ burst/s

16

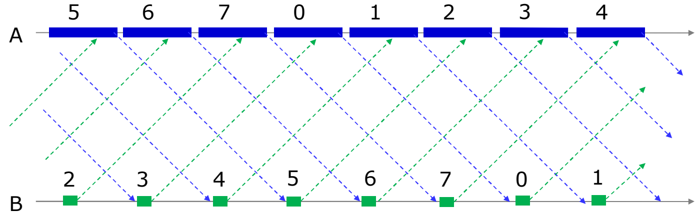
Problems with Stop and Wait



- Stop and Wait is OK for short links
  - with long delay (or at very high bit rate), waiting time wastes too many bit times…
  - example is satellite link, $\tau = 250$ ms

17

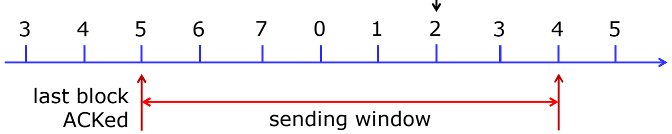
Better Protocols



- Need to keep transmitting while waiting
  - somehow keep track of data blocks and ACKs
    - need 2-way link, both directions at same time
  - called *sliding window* protocols…
  - *sending window* is range of sequence numbers that sender can send without getting ACK
  - *slides* upwards as ACKs arrive for old blocks…

18

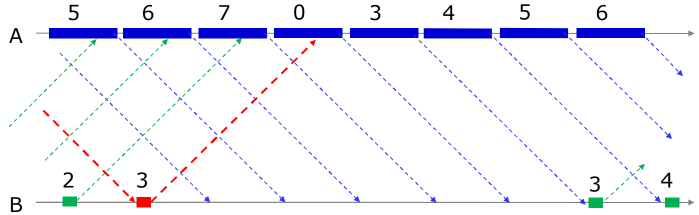
Sending Window



- Why limit number of un-ACKed data blocks?
- Limited storage to hold those data blocks
  - must be stored by link layer at sender
  - might be needed for re-sending if NAK arrives
- Limited range of sequence numbers
  - cannot have 2 un-ACKed blocks with same no.
- Design – choose limit large enough
  - so ACKs should arrive before limit is reached…

19

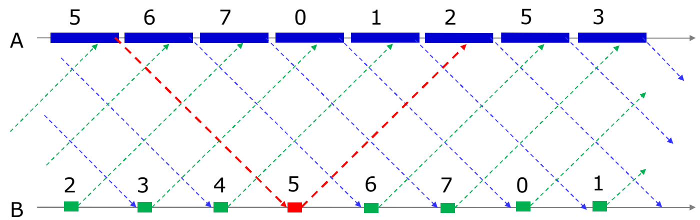
Receiver Behaviour ?



- Option 1 – no storage
  - receiver only accepts data blocks in sequence
    - discards anything that it cannot use immediately
    - so after NAK, no more ACKs until missing block arrives
  - when sender gets NAK
    - must go back and start again from bad block
  - called "go back N" operation

20

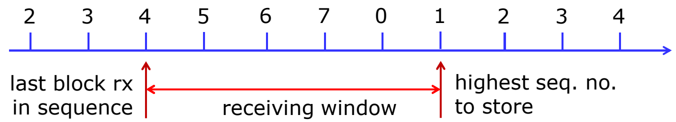
Receiver Behaviour ?



- Option 2 – with storage
  - receiver accepts blocks out of sequence
    - may acknowledge them if good (option)
    - stores them until missing blocks arrive
  - when sender gets NAK, resend rejected block
    - then resume where it was interrupted…
  - called "selective reject" operation

21

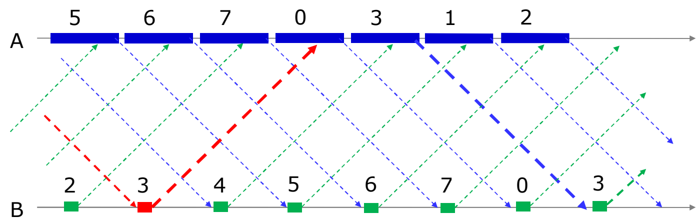
Selective Reject – Receiving Window



- Why limit number of blocks to receive out of sequence?
- Limited storage to hold those data blocks
  - must be stored by link layer at receiver
  - can only pass on to customer in sequence…
- Limited range of sequence numbers
  - cannot store 2 blocks with same number

22

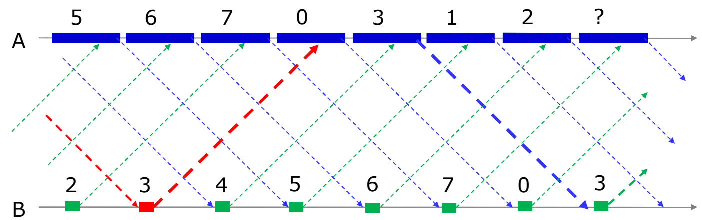
Design – Selective Reject



- How big should windows be?
- Depends on round-trip delay, frame time…
  - in example, delay rounds up to 6 frame times
    - e.g. from start of frame with block 5 to ACK 5 received
- If receiver sends NAK ?
  - cannot expect to get rejected data block again until 6 frame times have elapsed

23

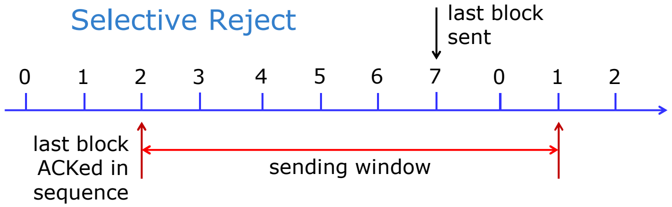
Design – Selective Reject continued



- So receiving window > round trip delay
  - delay measured in frame times, rounded up
  - to be efficient, must accept **all** blocks after NAK
- Sender sends block 3
  - cannot expect ACK for 6 frame times
  - but if get NAK instead, how long to get ACK?

24

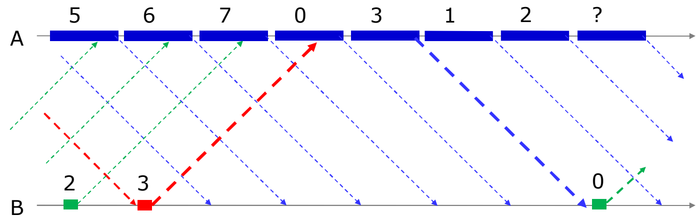
Selective Reject



- Sending window > round trip delay
  - minimum, as in go back N
  - but will have to stop after NAK
- To continue non-stop after NAK
  - need > twice round trip delay
  - this is reasonable design criterion
    - allows continuous sending most of the time
    - could still fail if NAK, re-send, NAK again...

25

Selective Reject – Alternatives



- Some systems only ACK blocks in sequence
  - no ACKs for blocks received after NAK sent
  - ACK means "all up to this block received OK"
- HDLC (High-level Data Link Control)
  - bit-oriented protocol, basis for many others
  - ACK/NAK can travel in data frame...
  - ACK uses seq. no. of **Next** block expected...

26

Example – HDLC Frame

Header			Data	Trailer	
Flag 01111110	Address 8 bit	Control 8 bit	Data (if any)	Frame Check Sequence (CRC) 16 bit	Flag 01111110

0	Sequence Number		P/F	Next

0 = Information frame, with data and seq. no., also carries ACK...

1	0	Type	P/F	Next

10 = Supervisory frame, no data. Type: Receiver Ready, Receiver not Ready, Reject (NAK), Selective Reject.

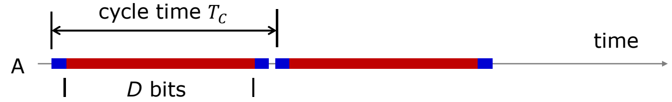
1	1	Type	P/F	Modifier

11 = Un-numbered frame, no data. Control messages: reset, disconnect, change mode, etc.

Option for 7-bit seq. no.s, adds another byte to header...

27

Analysis – Selective Reject

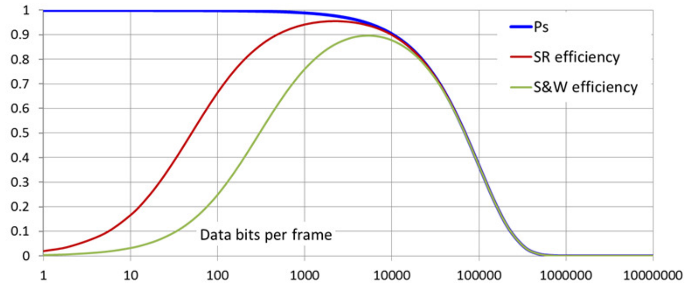


- Assume well designed...
  - sender never has to stop?
    - stops with very low probability, which we ignore
  - only reason to re-send block is error
- Cycle time is now frame transmit time
  - send one frame in each cycle time $T_C = \frac{D+H}{R}$
  - some succeed – fraction P_S
  - if succeed, transfer D data bits
  - others fail (NAK), transfer nothing useful

28

Efficiency

Example for $H = A = 50$ bit, $p = 10^{-5}$, $R = 1$ Mbit/s, $\tau = 100$ μs



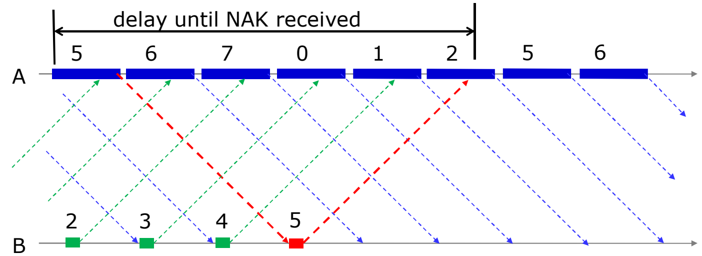
– throughput $U = \frac{DP_S}{T_C} = \frac{DP_S R}{D+H}$

– efficiency $\eta = \frac{U}{R} = \frac{DP_S}{D+H}$

– bit rate and time delay irrelevant (ideally)

29

Analysis – Go Back N



- Assume well-designed, as for selective reject
  - sender never has to stop sending
  - but now NAK means N blocks re-sent
  - N = delay until NAK rx ÷ frame time, rounded up
  - $N = \left\lceil \frac{D+H+A+2\tau R}{D+H} \right\rceil = 1 + \left\lceil \frac{A+2\tau R}{D+H} \right\rceil$

30

Analysis – Go Back N

- Still send one frame in time $T_C = \frac{D+H}{R}$
- How many frames to send one data block?
  - if succeed on first attempt, use 1 frame
    - probability P_S
  - if succeed on second try, use $N + 1$ frames
    - fail once, then succeed, probability $(1 - P_S)P_S = P_F P_S$
  - if succeed on third try, use $2N + 1$ frames
    - fail twice, then succeed, probability ${P_F}^2 P_S$

$$\text{average} = \sum_{k=1}^{\infty} \{(k-1)N + 1\} {P_F}^{k-1} P_S$$

$$= P_S \sum_{k=0}^{\infty} (kN + 1){P_F}^k \quad = P_S \left[N \sum_{k=0}^{\infty} k{P_F}^k + \sum_{k=0}^{\infty} {P_F}^k \right]$$

31

Analysis…

- Geometric series:

$$\sum_{n=0}^{\infty} ar^n = \frac{a}{1-r} \quad \text{for } |r| < 1, \quad \text{so } \sum_{n=0}^{\infty} r^n = \frac{1}{1-r}$$

$$\text{differentiating this: } \sum_{n=0}^{\infty} nr^{n-1} = \frac{1}{(1-r)^2}$$

- So average no. frames to send one block

$$= P_S \left[NP_F \frac{1}{(1-P_F)^2} + \frac{1}{1-P_F} \right] = \frac{N(1-P_S) + P_S}{P_S}$$

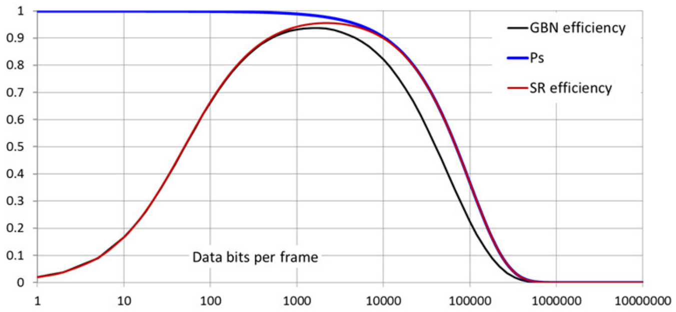
- So fraction of frames carrying useful data

$$= \frac{P_S}{P_S + N(1-P_S)}$$

32

Efficiency

Example for $H = A = 50$ bit, $p = 10^{-5}$, $R = 1$ Mbit/s, $\tau = 100$ μs



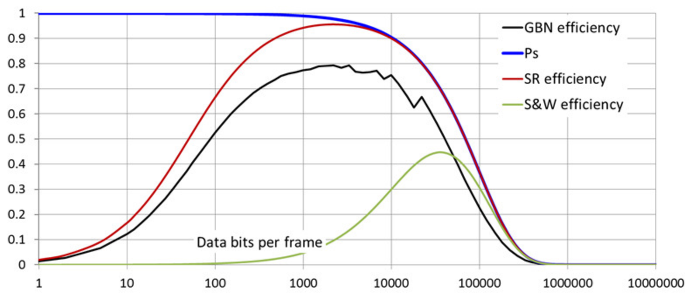
– throughput $U = \frac{D}{T_C}\frac{P_S}{P_S+N(1-P_S)} = \frac{DR}{D+H}\frac{P_S}{P_S+N(1-P_S)}$

– efficiency $\eta = \frac{U}{R} = \frac{DP_S}{(D+H)\{P_S+N(1-P_S)\}}$

33

Comparison

Example for $H = A = 50$ bit, $p = 10^{-5}$, $R = 1$ Mbit/s, $\tau = 10$ ms



– efficiency falls with longer delay, larger N
  - small fall if probability of success high…
  - jagged curve due to step changes in N

– still better than stop and wait…

34

Summary

- Stop & Wait
  – simple, can use "one way at a time" channel
  – OK if waiting time short, relative to frame
    - poor with long link, high bit rate
- Sliding Window, Go Back N
  – need storage at sender, but simple receiver
  – need bi-directional channel
  – OK with short-medium delay
    - or longer delay if probability of success is high
- Sliding Window, Selective Reject
  – need storage at sender and receiver
  – if well designed, windows large enough…
    - performance ~independent of delay or bit rate

35