

Computer Networks X_400487

Lecture 2

Chapter 3: The Data Link Layer Part 1



Lecturer: Jesse Donkervliet



Vrije Universiteit Amsterdam

Recap of the Physical Layer

Responsible for transferring *bits* over a *wire-like* medium.

Maximum data rate determined by *bandwidth* and *signal-to-noise ratio*. $R = B \times \log_2 \left(1 + \frac{S}{N} \right)$

Physical layer responsible for translating bits to and from waves; sharing channel with multiple users



0100111010101100110101100110101011011011010000110101001100
 0101101011001011101010010100111101010110011010110011010101110
 0011101010100101010101101101100001101010100110010101110101001
 010010100011101010100110101100111010110110110100001101010100
 1010101101101100001101010100110011001101010101010101101100110
 010101011011001101010100101011101010010100011101010101100111
 010011001100110101101010101100110101000101110101010100110
 011011001101010100101110101010011010101011011001101010101001010

Where are the frames?

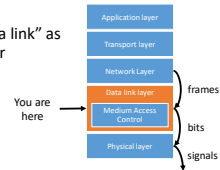
Q: Why do we make frames?

010101101010101101100110101010010101101101101100001101010011
 0001100100110101011010101011011001101010100101101101010110110
 001101011001101010110110110110000101110101001010011101010110
 010101101010101101100110101010010101101101101100001101010011
 000101011010100101001110101011001101011001101010101011011011
 0110011010101001010110101001010011101010101100110101100110101
 0111010101011001010110101001010011101010101100110101100110101

The Data Link layer

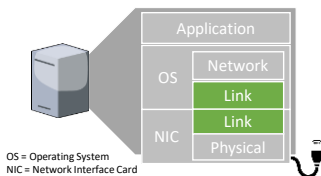
Responsible for transferring *frames* over a single link

1. Groups bits into frames
2. Offers “sending frames over a link” as a *service* to the network layer
3. Handles transmission errors
4. Regulates data flow



Link layer environment

Commonly implemented as NICs and OS drivers; network layer (IP) is often OS software.



OS = Operating System
NIC = Network Interface Card

© 2006 by Tanenbaum & Walden, © Pearson Education, Prentice Hall, © Walden, 2011

9

Data Link Layer — Roadmap

Part 1

1. Framing → From single bits to discrete data units: *frames*
2. Flow Control → Prevent overloading the receiver
3. Guaranteed Delivery → Because the network can drop frames
4. Sliding Window Protocols → Using network resources efficiently

Part 2

1. Error detection
2. Error correction

10

Framing

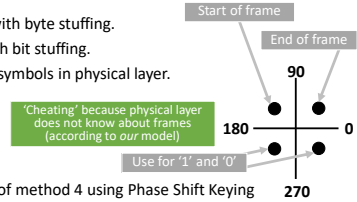
From Bit Stream to Discrete Units of Information



11

Framing Methods

1. Byte count.
2. Flag bytes with byte stuffing.
3. Flag bits with bit stuffing.
4. Use special symbols in physical layer.

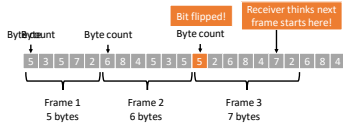


Example of method 4 using Phase Shift Keying

12

Framing Byte count

Q: Advantage?
Disadvantage?



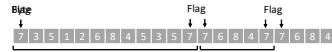
Click by Donitescu & Hillbrand, © Pearson Education—Prentice Hall and G. Walker, 2011.

13

Framing Byte stuffing

Q: Disadvantage?

Use a 'flag' byte to indicate start and end of frame.
Let's say our flag byte is 0000111₂ (7₁₀).



14

Framing Byte stuffing

Character	Escape Character
%	%25

What if the data contains a flag byte?
Use an 'escape' byte to ignore certain flag bytes.
Let's say our escape byte is 0000100₁₂ (9₁₀).



Q: Algorithm on the receiving side?

Q: Are we done?

15

Framing Byte stuffing

Q: What is the overhead of this approach?

Escape bytes can also occur in data!
Let's use letters for generality.
Flag byte = F, Escape byte = E.



Escape both 'escape' and 'flag' bytes.

16

Flow Control

Could you speak more slowly, please?

A Resource Management problem

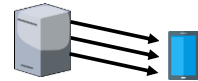


23

Utopian simplex protocol The ideal case

```

...
while True:
    packet = from_network_layer()
    frame.payload = packet
    to_physical_layer(frame)
    
```

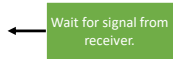


24

Stop-and-wait for error-free channel

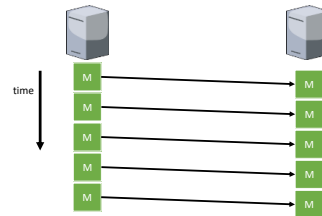
```

...
while True:
    packet = from_network_layer()
    frame.payload = packet
    to_physical_layer(frame)
    event = wait_for_event()
    
```



25

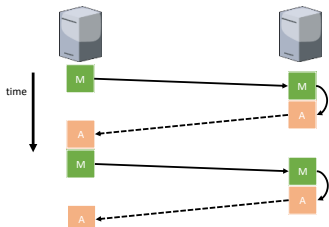
Example of Utopian Simplex Protocol



27

Example of Stop-and-Wait Protocol

Q: What if a frame gets lost?



28

Data Link Layer — Roadmap

Part 1

1. Framing → From single bits to discrete data units: *frames*
2. Flow Control → Prevent overloading the receiver
3. **Guaranteed Delivery** → Because the network can drop frames
4. Sliding Window Protocols → Using network resources efficiently

Part 2

1. Error detection
2. Error correction

29

Guaranteed Delivery

Acknowledgments, Sequence Numbers, and Retransmissions



30

How Can We Know If a Frame Gets Lost?

Ask a different question

Q: How can we know if a frame arrives?

Send a message back: "I got your message!"

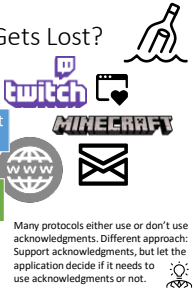
Q: When do we want to retransmit data?

Q: What if the acknowledgment gets lost?

We assume our original message did not arrive

It depends on ... the application!

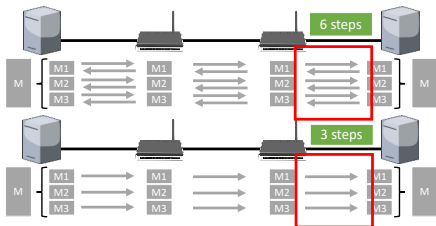
Acknowledgments let the sender know it does **not** need to retransmit data.



Many protocols either use or don't use acknowledgments. Different approach: Support acknowledgments, but let the application decide if it needs to use acknowledgments or not.

31

To acknowledge, or not to acknowledge



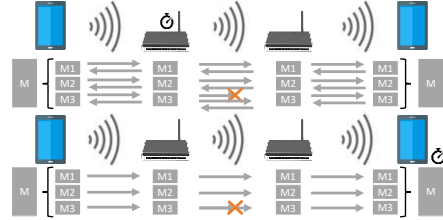
33

To acknowledge, or not to acknowledge

It depends on ... the physical medium!

It depends on ... the application!

This problem will return later (in the transport layer)



34

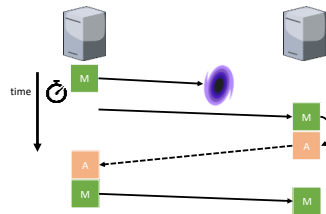
Automatic Repeat reQuest (ARQ) Guaranteed Delivery over Unreliable Channel

Same as stop-and-wait, except:

1. Keep track of frames using sequence numbers.
2. Wait until previous frame has been accepted.

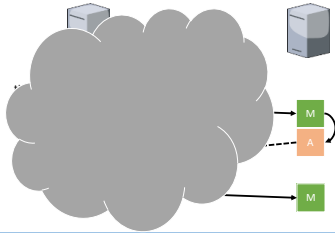
35

ARQ Example



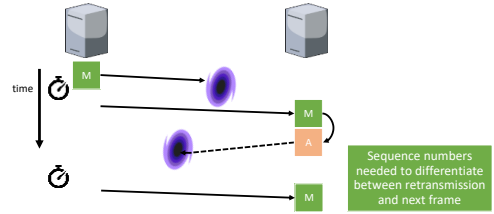
36

ARQ Example



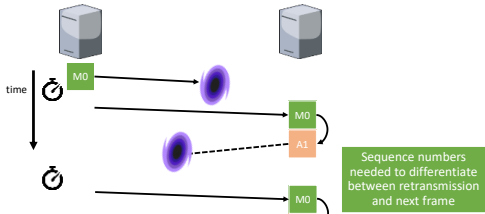
37

ARQ Example



38

ARQ Example



39

Automatic Repeat ReQuest (ARQ) Guaranteed Delivery over Unreliable Channel

ARQ adds error control

Receiver acks frames that are correctly delivered.

Sender sets timer and resends frame if no ack.

Q: How long should we wait?

Q: What can go wrong?

Frames and acks must be identifiable (e.g., with sequence number)

Else receiver cannot tell retransmission (due to lost ack or early timer) from new frame.

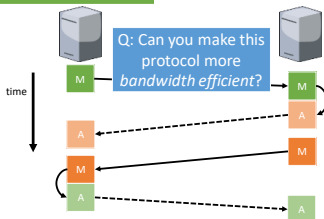
For stop-and-wait, 2 numbers (1 bit) are sufficient.

Q: Why sufficient?

41

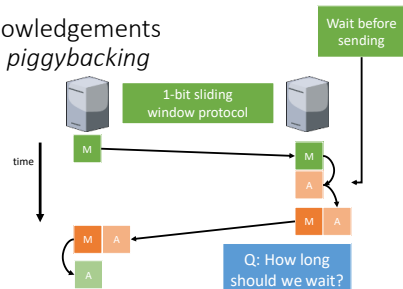
Acknowledgements

Bidirectional communication



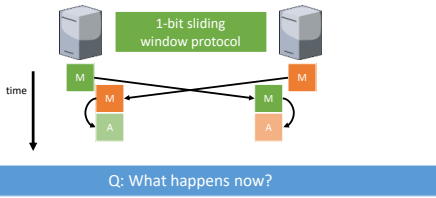
42

Acknowledgements With piggybacking



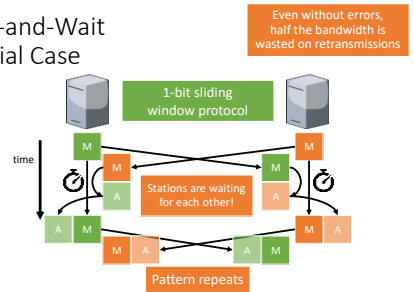
43

Stop-and-Wait



44

Stop-and-Wait Special Case



45

Data Link Layer — Roadmap

Part 1

1. Framing → From single bits to discrete data units: *frames*
2. Flow Control → Prevent overloading the receiver
3. Guaranteed Delivery → Because the network can drop frames

4. Sliding Window Protocols → Using network resources efficiently

Part 2

1. Error detection
2. Error correction

46

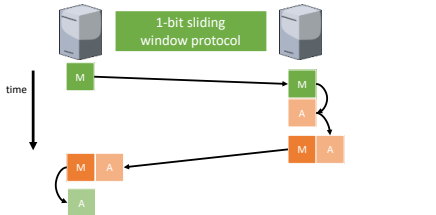
Sliding Window Protocols

Improving Performance using *Pipelining*



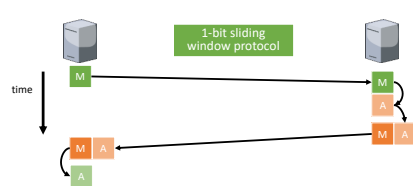
47

Stop-and-Wait: A 1-Bit Sliding Window Protocol



48

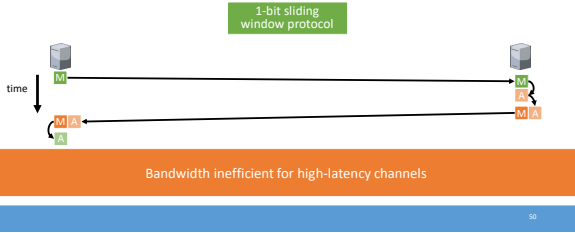
Stop-and-Wait: A 1-Bit Sliding Window Protocol



49

Stop-and-Wait: A 1-Bit Sliding Window Protocol

Q: Which properties cause performance to decrease?



Sliding window protocols

When using stop-and-wait, *data rate decreases* when:

- Latency increases
- Frame size decreases

Solution

Send next frame while waiting for acknowledgment of current frame

Sender window specifies how many frames a sender is allowed to send before waiting for an acknowledgement.

Receiver window specifies the range of frames that the receiver is allowed to accept.

Stop-and-Wait / ARQ: A 1-Bit Sliding Window Protocol

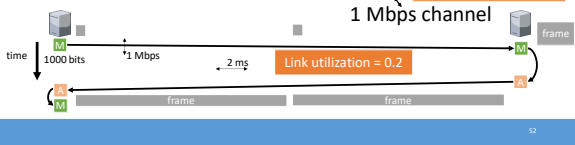
Q: What is the link utilization?

It takes $\frac{1,000}{1,000,000} = 0.001$ seconds to send frame

It takes 2 ms for the frame to arrive at the receiver, takes 2 ms for the (0-bit) acknowledgment to come back at the sender

1 frame per $0.001+0.002+0.002$ seconds = 200 kbps

Small window reduces performance



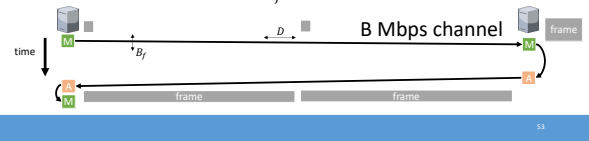
Link Utilization

$$\text{Link utilization} \leq \frac{w}{1+2B_f D}$$

- Frame size (in bits/bytes): f
- Window size (in frames): w
- Bandwidth (max. data rate of physical channel): B_f
- Bandwidth (frames per second): B_f
- Propagation delay (in seconds): D

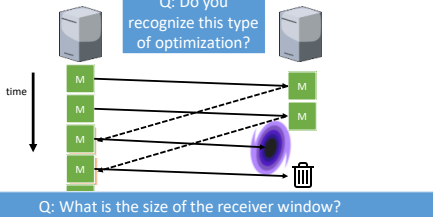
Sliding window protocol with window size of w frames can send w frames before stopping to wait for acks

Link bandwidth of B_f frames per second, and delay of D seconds \rightarrow the link can hold $2B_f D$ frames. The receiver can hold and process 1 additional frame \rightarrow at most $1 + 2B_f D$ frames in transit



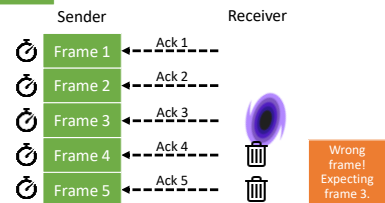
Go-Back-N

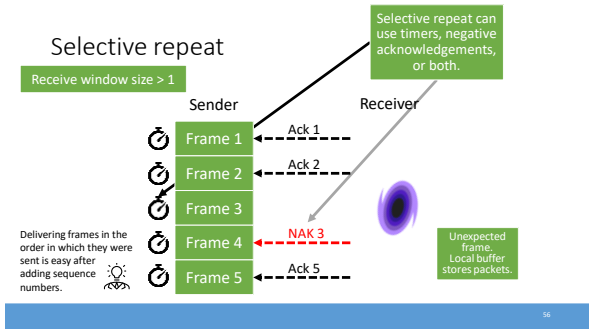
Q: Do you recognize this type of optimization?



Go-Back-N

Receive window size = 1





Data Link Layer — Roadmap

Part 1

- Framing
- Flow Control
- Guaranteed Delivery
- Sliding Window Protocols and Link Utilization

Part 2

- Error detection
- Error correction