

Tutorial 1: Physical & Link Layer



Lukas Kompatscher
Computer Networks 2025-2026

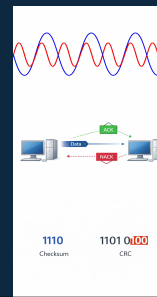
Menu of The Day

Physical Layer

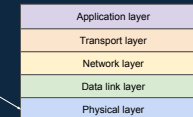
1. Nyquist vs Shannon

Link Layer

1. Package Framing
 - A. Framing Protocols
 - B. Acknowledgement handling
2. Error Correction and Detection
 - C. Checksum
 - D. Cyclic Redundancy check



Physical Layer



3

Communication Limits

Noiseless (Nyquist Theorem)

Shows how data rate increases with bandwidth and signal levels.

Finds: Data Rate

Given: number of signal levels

$$C = 2 \times B \times \log_2(V)$$

C Maximum Data Rate

B Bandwidth

V Number of discrete signals

Noisy (Shannon Theorem)

Give realistic maximum data rate with noise.

Finds: Data Rate

Given: signal to noise ratio

$$C = B \times \log_2 \left(1 + \frac{S}{N} \right)$$

C Capacity

B Bandwidth

$\frac{S}{N}$ Signal-to-noise ratio

4

Communication Limits - Exercise

Consider a physical-layer protocol that uses **256** signal levels on a channel with **100 MHz** bandwidth and an SNR of **30 dB**.

What is the maximum data rate that we can obtain using this protocol and channel?

Nyquist: $C = 2 \times B \times \log_2(V)$

Shannon: $C = B \times \log_2 \left(1 + \frac{S}{N} \right)$

$$\frac{S}{N} = 10^{\left(\frac{dB}{10}\right)}$$

5

Communication Limits - Solution

256 signal levels, 100 MHz bandwidth, SNR of 30 dB

What is the maximum data rate?

$$\frac{S}{N} = 10^{\left(\frac{dB}{10}\right)}$$

Nyquist

$$2 \times B \times \log_2(V)$$

$$2 \times 100 \times 10^6 \times \log_2(256)$$

$$2 \times 100 \times 10^6 \times 8$$

$$1.6 \times 10^9 \Rightarrow R = 1.6 \text{ Gbps}$$

Shannon

$$B \times \log_2 \left(1 + \frac{S}{N} \right)$$

$$100 \times 10^6 \times \log_2(1 + 10^3)$$

$$\approx 10^9 \Rightarrow 1 \text{ Gbps}$$

$$R = \min(1.6 \text{ Gbps}, 1 \text{ Gbps}) = 1 \text{ Gbps}$$

6

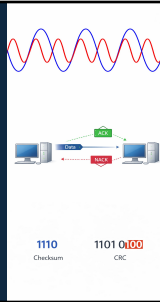
Menu of The Day

Physical Layer

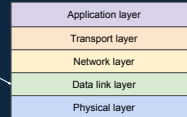
4. Nyquist vs Shannon

Link Layer

1. Package Framing
 - A. Framing Protocols
 - B. Acknowledgement handling
2. Error Correction and Detection
 - C. Checksum
 - D. Cyclic Redundancy check



Link Layer

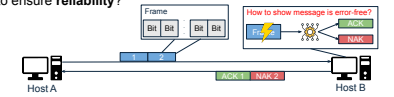


Why link layer ?

How to divide data into handle units?



How to ensure reliability?



Bits and Bytes

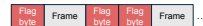
$1 Kb = 10^3 bits$
 $1 KB = 10^3 bytes$
 $= 1000 bytes$
 $1 MB = 10^6 bytes$
 $1 GB = 10^9 bytes$
 $1 TB = 10^{12} bytes$

$1 Kib = 2^{10} bits$
 $1 KiB = 2^{10} bytes$
 $= 1024 bytes$
 $1 MiB = 2^{20} bytes$
 $1 GiB = 2^{30} bytes$
 $1 TiB = 2^{40} bytes$

Framing

Byte Stuffing

Insert escape bytes before control bytes



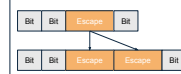
Bit Stuffing (very similar)

Insert 0s to prevent confusion with control flags

Problem: frame can contain flag

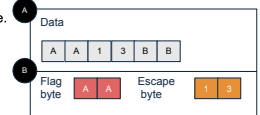


Problem 2: frame can contain escape



Framing - Exercise

Frame in (A) is transmitted over the wire.



Show what the frame will look like after byte stuffing. Flag and escape symbols are shown respectively (B).

Do not forget to add flags both at the beginning and end of frame.

Calculate transmission overhead in bytes. Since symbols are represented in hexadecimal, each symbol is half a byte.

$$\text{Overhead} = \frac{\text{flag} + \text{esc.}}{\text{data}}$$

Framing - Solution

Data: A A 1 3 B B

Operation	Resulting frame
Escape escape	A A 1 3 1 3 B B
Escape flag	1 3 A A 1 3 1 3 B B
Add flags to frame	A A 1 3 A A 1 3 1 3 B B A A

Overhead
Byte stuffing: $4 + 3 = 133\%$

13

Stop-and-Wait protocol

Disadvantage: low link utilization

14

Sliding window protocol

Key Idea
Do not wait for ack. Pipeline!

Advantage: higher link utilization (LU)

Disadvantage: harder to implement

Formula:
$$LU \leq \frac{w}{1 + 2 * B_f * D}$$

15

Sliding Window Protocol - Exercise

You use a sliding-window protocol with a window size of 1 (effectively stop and wait protocol) and frame size of 10,000 bits over a channel with propagation delay of 8 ms. The max. data rate of the physical channel is 1 Gb/s. What is the maximum link utilization (LU) you can achieve? Now, what if you use a window size of 100? What is the maximum LU you can achieve?

Link utilization:

$$LU \leq \frac{w}{1 + 2 * B_f * D} \quad B_f = \frac{B_p}{f}$$

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

16

Sliding Window Protocol - Solution

$$B_f = \frac{B_p}{f} = \frac{1 \text{ Gb/s}}{10,000 \text{ bits/frame}} = 100,000 \text{ frame/s}$$

Window Size = 1

$$LU \leq \frac{w}{1 + B_f * D} = \frac{1}{1 + 2 * (100,000 \text{ frame/s}) * (8\text{ms})} = \frac{1}{1601} = 0.0006$$

Window Size = 100

$$LU \leq \frac{w}{1 + B_f * D} = \frac{100}{1 + 2 * (100,000 \text{ frame/s}) * (8\text{ms})} = \frac{100}{1601} = 0.06$$

17

Link Layer Transmission - Exercise

The figure below is a model of the networked system. In this case, retransmission is the responsibility of the link layer, every router between the hosts detects loss and issues retransmission. That is whenever data is lost only the nearest router retransmits it. Hop traversal (HT) is 0.2 sec. The retransmission timeout (RT) between hops is set to 0.4 sec. How long does it take to successfully send a message across if the reliability of the individual link is 1.0 and 0.5?

Expected transmission count (EC) = $\frac{1}{LR}$

Expected transmission time = $(EC - 1) * RT + HT$

18

Link Layer Transmission - Solution (Reference Model)

Router caches the frame and can issue retransmission between the routers

19

Link Layer Transmission - Solution

Expected transmission count (EC) = $\frac{1}{\text{link reliability}}$

a) Case 1.0: $\frac{1}{1} = 1$
 b) Case 0.5: $\frac{1}{0.5} = 2$

Total time for retransmissions

Time to 1 hop = (EC - 1) × retransmission timeout + hop traversal

a) Case 1.0: $(1 - 1) \times 0.4s + 0.2s = 0.2s$
 b) Case 0.5: $(2 - 1) \times 0.4s + 0.2s = 0.6s$

Link Layer: time to destination = hop count × time for 1 hop

a) Case 1.0: $4 \times 0.2s = 0.8s$
 b) Case 0.5: $4 \times 0.6s = 2.4s$

20

Transport Layer Transmission - Exercise

The figure below is a model of the networked system. In this case, retransmission is the responsibility of the **transport layer**, only final hosts detect loss, and issues retransmission. Hop travels takes **0.2 s**. The end-to-end retransmission timeout is set to **1.6 s**. How long does it take to successfully send a message across if link reliability is **1.0** and **0.5**?

Which solution do you think is used in modern networks and why?

Transport layer scope

End to end reliability = $\text{link reliability}^{\text{hop count}}$

21

Transport Layer Transmission - Solution (Reference Model)

Route does not cache the frame and can issue retransmission between the routers

22

Transport Layer Transmission - Solution

End-to-end reliability

a) Case 1.0: $1.0^4 = 1.0$
 b) Case 0.5: $0.5^4 = 0.06$

Link layer
 Case 1.0: 0.8 s
 Case 0.5: 2.4 s

Expected transmission count (EC) = $\frac{1}{\text{endtoend reliability}}$

a) Case 1.0: $\frac{1}{1.0} = 1$
 b) Case 0.5: $\frac{1}{0.06} = 16$

What is better in practice link or transport layer?

Time to destination = (EC - 1) × retransmission timeout + time to receiver

a) Case 1.0: $(1 - 1) \times 1.6s + 0.8s = 0.8s$
 b) Case 0.5: $(16 - 1) \times 1.6s + 0.8s = 24.8s$

Nr. hops × hop traversal time

23

BREAK

24

Cyclic Redundancy Check (CRC) - Example

For example, the generator is $x^3 + x^2 + 1$ and message is 1111. $1x^3 + 1x^2 + 0x + 1 = 1101$

1. Add N (three) 0s to the end of the message

2. Using mod 2 arithmetic keep dividing until can no more.

3. Substitute remainder at the end of the message.

How does final host check for correctness?

Final Message: 1111 111

Cyclic Redundancy Check (CRC) - Exercise

You want to send the data bits 1011 0001 0101 1001.

You use CRC to enable the receiver to detect errors in the data. You use the generator polynomial 1011 (already in binary form).

What is the code word that you transmit?

Include the whole codeword (data bits + CRC check bits)

Cyclic Redundancy Check (CRC) - Solution

Data – 1011 0001 0101 1001

Polynomial – 1011

Add three 0s to the message and divide until you can't no more.

Final message to divide: 1011 0001 0101 1001 000

We also leave 0s out for clarity!

Final Message: 1011 0001 0101 1001 101

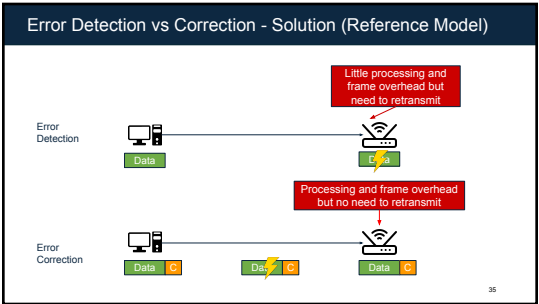
Error Detection vs Correction - Exercise

In this exercise, we work with a wired setup. Assume that, for every Mb, the probability of at least 1 bit-flip is 10^{-6} (PBF). We want to choose whether to use error correction or error detection for our setup. Assume that both always correct and detect successfully. For error correction, the code rate is 0.75 and for error detection it is 1.0. Error correction introduces an additional 0.01s processing on the receiver side for every Mb received.

In one frame, we want to send 1 MB of data. Link bandwidth 8 Mb/s, and latency is 0.2s. The retransmission timeout is 1s (RT). How long does it take to successfully send a message across?

$PEF = (1 - PBF)^{Mb \text{ count}}$

transmission time = $\frac{\text{data size}}{\text{code rate} \times \text{bandwidth}}$



Error Detection vs Correction - Solution

Error detection. Probability that a frame arrives error free (PEF): $= (1 - PBF)^{Mb \text{ count}} = (1 - 10^{-6})^8 = 0.9999999^8 \approx 1$

Error detection. Expected time: $= (EC - 1) \times \text{retransmission timeout} + \text{successful transmission time} \rightarrow EC = \frac{1}{PEF} = \frac{1}{1} = 1$

$= (1 - 1) \times 1s + \frac{1}{8 \text{ Mb/s}} + 0.2s = 1.2s$

$1 \text{ MB} = 8 \text{ Mb} \rightarrow \frac{8 \text{ Mb}}{0.75} = 10.6 \text{ Mb}$

Error correction. Time to send: $= \text{transmission time for (frame + overhead)} + \text{latency} + \text{processing overhead of MB}$

$= \frac{10.6 \text{ Mb}}{8 \text{ Mb/s}} + 0.2s + 0.01 \frac{s}{\text{Mb}} \times 10.6 \text{ Mb} \approx 1.6s$

Error Detection vs Correction - Exercise

Now we work with a wireless setup. Assume that, for every Mb, the probability of at least 1 bit-flip is 10^{-1} (PBF). Assume that both error correction and detection always correct and detect successfully. For error correction, the code rate is **0.75**, and for error detection, the code rate is **1.0**. Error correction bears additional **0.01s** processing on the receiver side for every Mb received. In one frame, we want to send **1MB** of data. The link bandwidth is **8 Mb/s**, and the latency is **0.2 s**. The retransmission timeout (RT) is **1s**.

How long does it take to successfully send a message across? Which error handling scheme is more performant and for which setup? How does this and the previous exercises contradict our notion of layered architecture?

!Round expected transmission count to nearest integer.

37

Error Detection vs Correction - Solution

Error detection. Probability that a frame arrives error free (PEF):
 $= (1 - \text{PBF})^{\text{Mb count}} = (1 - 10^{-1})^8 = 0.9^8 \approx 0.5$

Error detection. Expected time:
 $= (\text{EC} - 1) \times \text{retransmission timeout} + \text{successful transmission time} \rightarrow \text{EC} = \frac{1}{\text{PEF}} = \frac{1}{0.5} = 2$
 $= (2 - 1) \times 1s + \frac{1}{8 \text{ Mb/s}} = 1s + 0.125s = 1.125s$

Which setup is better?

Error correction. Time to send:
 $= \text{transmission time for (frame + overhead)} + \text{latency} + \text{processing overhead of MB}$
 $= \frac{1 \text{ MB}}{8 \text{ Mb/s}} + 0.2 \text{ s} + 0.01 \frac{\text{s}}{\text{Mb}} \times 1 \text{ MB} \approx 1.6 \text{ s (same as part 1)}$

38

Does our model hold in practice?

More theoretical model

Our architecture

Network Layer

Link Layer

Physical layer

More practical model

TCP/IP model [2]

Network Layer

Link + Physical layer

Separate but in practice for performance, link layer takes into account application into account.

39

Recap

Physical Layer

- Nyquist vs Shannon

Link Layer

- Package Framing
 - Framing Protocols (Byte & Bit stuffing)
 - Acknowledgement handling (Stop-and-Wait vs Sliding Window)
- Error Correction and Detection
 - Checksum (Error detection with 1's complement addition)
 - Cyclic Redundancy check (Strong error detection with polynomial division)

1110

Checksum

1101 0100

CRC