IF2230 Jaringan Komputer

Data Link Layer

Robithoh Annur
Andreas Bara Timur
Monterico Andrian

- Prinsip dasar
  - Peran data link layer
  - Error handling
  - Flow control
  - Ethernet and Medium Access

Prinsip dasar (1)

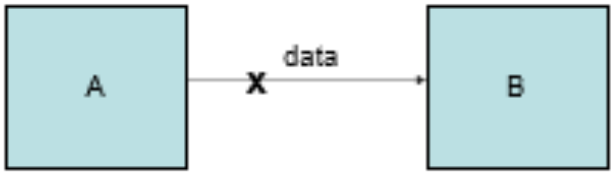
- Masalah utama dalam komunikasi data: reliability. Sinyal yang dikirim melalui medium tertentu dapat mengalami pelemahan, distorsi, keterbatasan bandwidth
- Data yang dikirim dapat menjadi rusak, hilang, berubah, terduplikasi
- Tugas data link layer adalah menangani kerusakan dan hilangnya data antar 2 titik komunikasi yang terhubung oleh satu medium transmisi fisik

Basic Principle (1)

- The main problem in data communication: reliability. Signals sent through a certain medium can experience attenuation, distortion, bandwidth limitations.
- The data sent can be damaged, lost, changed, duplicated
- The task of the data link layer is to handle damage and loss of data between 2 communication points connected by one physical transmission medium

Prinsip dasar (2)

- A mengirim data ke B. Jalur antara A dan B tidak reliable, sehingga mungkin ada data yang rusak/hilang.
- Bagaimana menjamin transmisi data A ke B tetap reliable?

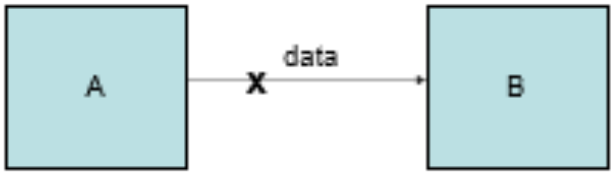


- A mengirim data yang panjang ke B. Data dibagi menjadi frame, sehingga kerusakan sebuah frame tidak merusak keseluruhan data.
- Bagaimana B dapat mendeteksi bahwa frame yang dikirim A mengalami kerusakan?
  - A menambahkan error check bits ke frame, sehingga B dapat memeriksa frame dan menentukan apakah telah terjadi perubahan

Basic Principle (2)

- A sends data to B. The path between A and B is unreliable, so there may be data corruption/loss.
- How to ensure that data transmission from A to B remains reliable?

di message yg si A kirim ada error check bitsnya

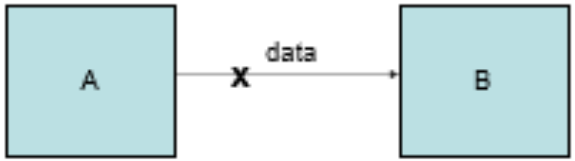


- A sends a long data to B. The data is divided into frames, so that the corruption of one frame does not corrupt the entire data.
- How can B detect that the frame sent by A is corrupted?
  - A adds error check bits to the frame, so that B can examine the frame and determine if any changes have occurred.

meski yg error cuman 1 bit, the whole data is considered corrupted.

Prinsip dasar (3)

- Bagaimana A mengetahui data yang dikirimnya telah diterima B?
  - B dapat mengirimkan ACK/pemberitahuan jika data diterima dengan benar, dan NAK/pemberitahuan data salah jika data rusak
  - A dapat mengirimkan ulang frame yang rusak

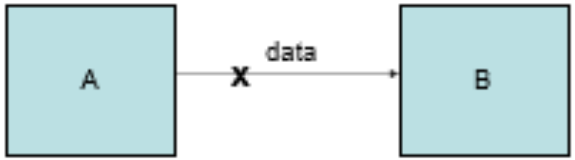


- Mengapa frame dapat hilang?
  - Bagian alamat/id/header mengalami kerusakan, sehingga frame tidak dikenali
  - Temporer disconnection
  - Apa yg terjadi jika frame dapat hilang?
  - B tidak mengetahui ada pengiriman dari A, A menunggu ACK dari B
  - B mengirimkan ACK namun hilang di jalan. A menunggu ACK dari B

Basic Principle (3)

- How does A know that the data he sent has been received by B?
  - B can send ACK/acknowledgement if the data is received correctly, and NAK/error data acknowledgement if the data is corrupted.
  - A can retransmit the corrupted frame.

kalau corrupted, A harus mengirim kembali

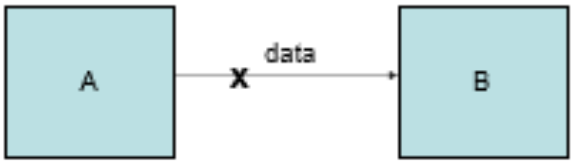


- Why can frames be lost?
  - The part of the frame containing the address/id/header is damaged, so the frame is not recognized
  - Temporary disconnection
  - What happens if a frame can be lost?
  - B is unaware of any transmission from A, A waits for an ACK from B
  - B sends an ACK but it gets lost on the way. A waits for an ACK from B

nunggu² terus nanti si A

Prinsip dasar (4)

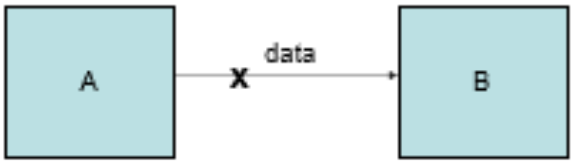
- A harus memiliki timer, yang akan mengirim ulang jika tidak menerima kabar dari B
- Jangka waktu timeout harus diatur.
  - Jika timeout terlalu cepat, A akan mengirimkan ulang sebelum ACK dari B tiba.
  - Jika timeout terlalu lama, A akan menunggu terlalu lama jika ada frame yg hilang



- A mengirim frame 1
- A mengalami timeout, dan mengirimkan ulang frame 1
- A menerima ACK, melanjutkan mengirim frame 2
- A menerima ACK kedua untuk frame 1, namun dianggap sebagai ACK untuk frame 2 (error)
  - A harus memberikan frame number, sehingga B dapat memberikan ACK spesifik untuk frame number tertentu

Basic Principle (4)

- A must have a timer, which will resend if it does not receive news from B.
- The timeout period must be set.
  - If the timeout is too fast, A will retransmit before the ACK from B arrives.
  - If the timeout is too long, A will wait too long if there is a lost

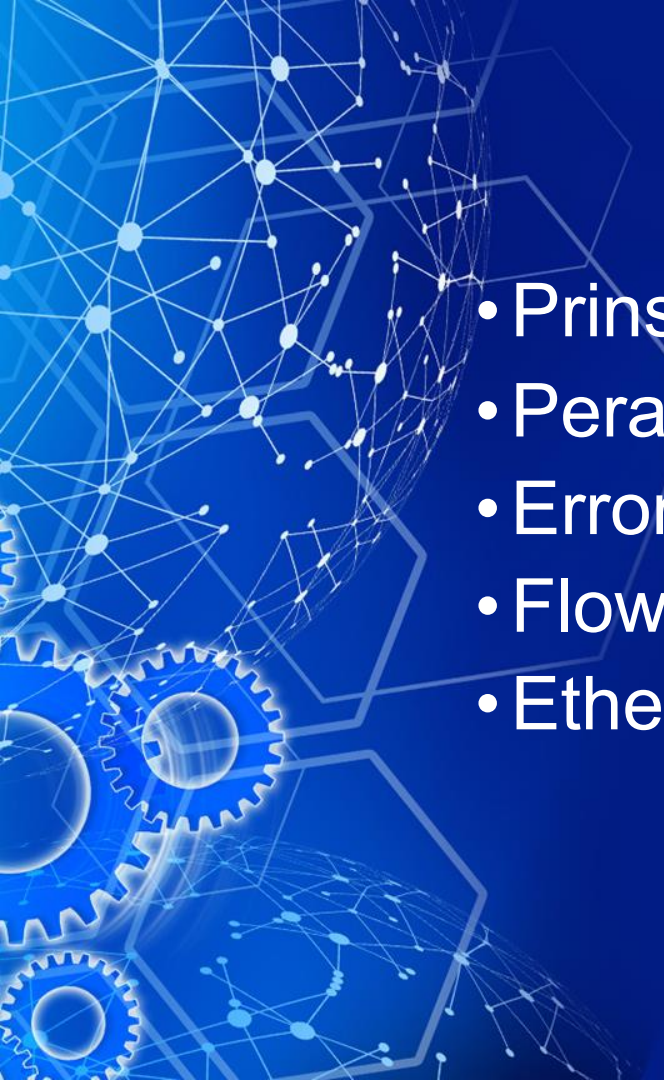


- A sends frame 1
- A times out, and resends frame 1
- A receives ACK, continues sending frame 2
- A receives second ACK for frame 1, but treats it as ACK for frame 2 (error)
  - A must provide frame number, so B can provide specific ACK for a particular frame number

- Prinsip dasar
- Peran data link layer
- Framing
- Error handling
- Ethernet and Medium Access

Role of Data link layer

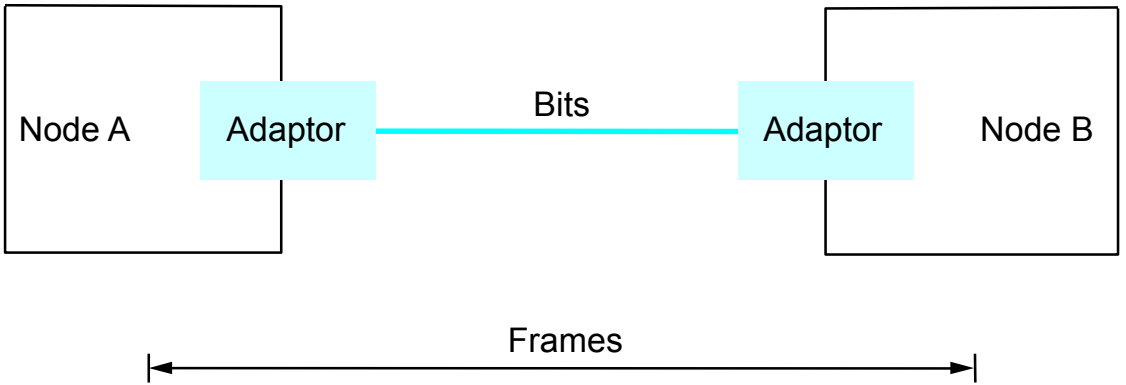
- Physical layer determines how bits are encoded to signals sent through the transmission medium
- DL handles transmission errors, and provides services to the network layer including:
  - Error control, error detection
  - Flow control
  - Link management
  - Medium access



- Prinsip dasar
- Peran data link layer
- Error handling
- Flow control
- Ethernet and Medium Access

Framing

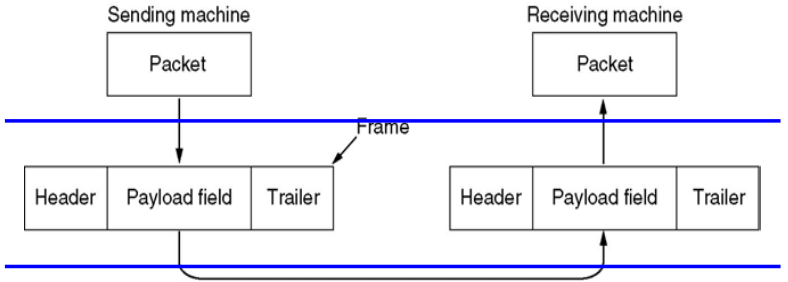
- Break sequence of bits into a frame – why? determine where a frame starts and ends
- Typically implemented by network adaptor



Framing

- Data-link layer takes the packets from the Network Layer and encapsulates them into frames with adding control information like headers and trailers for transmission over the physical medium.
- DL performs framing:
  - Reduce the possibility of errors
  - Adjust to the physical layer
  - Limited receiver buffer capacity

↳ reduce the length of buffer capacity.



Relationship between packets and frames.

Framing

how to perform framing ?

- Approaches
  - Byte oriented protocols
    - **Binary Synchronous Communication (BISYNC)**
    - **Digital Data Communication Message Protocol (DDCMP)**
    - **Point-to-Point Protocol (PPP)**
  - Bit oriented protocols
    - High-Level Data Link Control (HDLC)
  - Clock based protocols
    - Synchronous Optical Network (SONET)

tp user gabisa tau device pakai approach apa. yg ditau cuman transmission success / no

Byte-Oriented: Sentinel Approach

- Add **START** and **END** sentinels to the data
- Problem: what if **END** appears in the data?
  - Add a special **DLE** (Data Link Escape) character before **END**
  - What if **DLE** appears in the data? Add **DLE** before it.
  - Similar to escape sequences in C
    - printf("You must \"escape\" quotes in strings");
    - printf("You must \\escape\\ forward slashes as well");
- Used by Point-to-Point protocol, e.g. modem, DSL, cellular

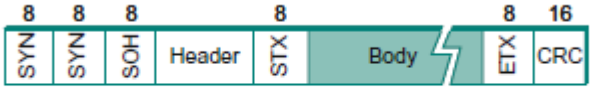


FIGURE 2.7 BISYNC frame format.

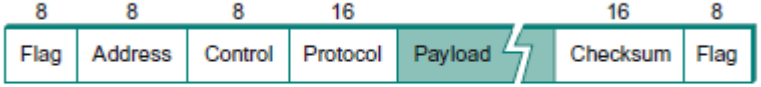


FIGURE 2.8 PPP frame format.

Byte Oriented: Byte Counting

awal → measure the length of the data.

- Sender: insert length of the data in bytes at the beginning of each frame ☿: "basically datanya nanti 132"
- Receiver: extract the length and read that many bytes ☿: "OK"
- What happens if there is an error transmitting the count field?

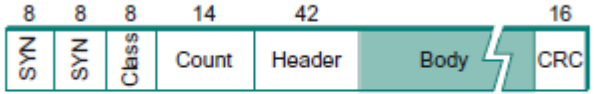


FIGURE 2.9 DDCMP frame format.

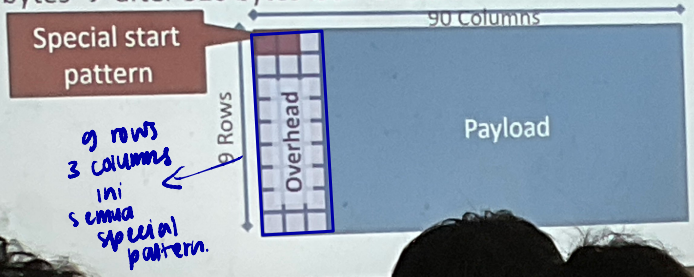
Bit Oriented: Bit Stuffing

- Add sentinels to the start and end of data
  - Both sentinels are the same
  - Example: 01111110 in High-level Data Link Protocol (HDLC)
- Sender: insert a 0 after each 11111 in data
  - Known as “bit stuffing”
- Receiver: after seeing 11111 in the data…
  - 11111**0** → remove the 0 (it was stuffed)
  - 11111**1** → look at one more bit
    - 1111110 → end of frame
    - 1111111 → error! Discard the frame
- Disadvantage: 20% overhead at worst
- What happens if error in sentinel transmission?

FIGURE 2.10 HDLC frame format.

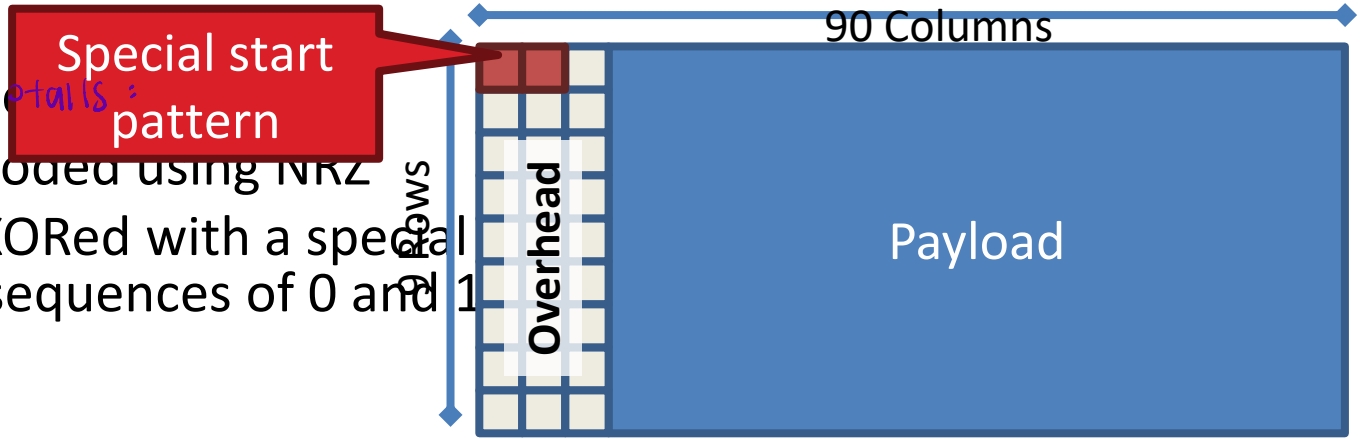
Clock-based Framing: SONET

- **S**ynchronous **O**ptical **Net**work
  - Transmission over very fast optical links
  - STS-*n*, e.g. STS-1: 51.84 Mbps, STS-768: 36.7 Gbps
- STS-1 frames based on fixed sized frames
  - 9*90 = 810 bytes → after 810 bytes look for start pattern



Clock-based Framing: SONET

- **S**ynchronous **O**ptical **Net**work
  - Transmission over very fast optical links
  - STS-*n*, e.g. STS-1: 51.84 Mbps, STS-768: 36.7 Gbps
- STS-1 frames based on fixed sized frames
  - 9*90 = 810 bytes → after 810 bytes look for start pattern
- Physical layer details:
  - Bits are encoded using NRZ
  - Payload is XORed with a special pattern to avoid long sequences of 0 and 1



Clock-based Framing: SONET

- **S**ynchronous **O**ptical **Net**work
  - Transmission over very fast optical links
  - STS-*n*, e.g. STS-1: 51.84 Mbps, STS-768: 36.7 Gbps
- STS-1 frames based on fixed sized frames
  - 9*90 = 810 bytes → after 810 bytes look for start pattern
- Physical layer details
  - Bits are encoded using NRZ
  - Payload is XORed with a special 127-bit pattern to avoid long sequences of 0 and 1

- Prinsip dasar
- Peran data link layer
- Framing
- Error handling
- Ethernet and Medium Access

Error Detection and Correction

Dealing with Noise

- The physical world is inherently noisy
  - Interference from electrical cables
  - Cross-talk from radio transmissions, microwave ovens
  - Solar storms
- How to detect bit-errors in transmissions?
- How to recover from errors?

Error Detection and Correction

- Umumnya penanganan error transmisi dilakukan pada data link layer. Error control dapat pula dilakukan pada layer lain (physical/higher layer)
- Error dapat diperbaiki dan dideteksi dengan menggunakan data redundant/tambahan pada setiap pengiriman data
- Jenis error:
  - Single bit error: hanya sebuah bit yang berubah. Disebabkan oleh white noise
  - Burst error: sederetan bit-bit mengalami error. Disebabkan oleh impulse noise
- Makin tinggi data rate, makin besar efeknya

Naïve Error Detection

- Idea: send two copies of each frame
  - if (memcmp(frame1, frame2) != 0) { OH NOES, AN ERROR! }
- Why is this a bad idea?
  - Extremely high overhead
  - Poor protection against errors
    - Twice the data means twice the chance for bit errors

Channel Coding

- Channel coding is most often applied to communications links in order to improve the reliability of the information being transferred.
- By adding additional bits to the transmitted data stream, it is possible to detect and even correct for errors in the receiver.
- The added coding bits lower the raw data transmission rate through the channel (Coding expands the occupied bandwidth for a particular message data rate).
- Channel codes that are used to detect errors are called error detection codes, while codes that can detect and correct errors are called error correction codes. There are three general types of channel codes:- Block codes, Convolutional codes and Concatenated Codes

Channel Coding – Error Detection & Correction

- Error Detection
  - In its most elementary form this involves recognizing which part of the received information is in error.
- Error Detection and Correction
  - With added complexity, it is possible not only to detect errors, but also to build in the ability to correct errors without recourse to retransmission.
  - This is particularly useful where there is no feedback path to the transmission source with which to request a resend. This process is known as FEC (Forward Error Correction).

Error Detection Schemes – Parity

- One of the simplest yet most frequently used techniques for detecting errors is the parity check bit.
- The parity check bit is usually a single bit (1 or 0) appended to the end of a data word such that the number of 1s in the new data word is even for even parity, or odd for odd parity.
- On reception, each data word, with appended parity bit, is checked to see how many 1s are present. For an even parity design, the number must be even. If it is found to be odd, it can be concluded that at least one error has occurred during transmission and the Automatic Repeat Request (ARQ) process can begin. Of course, if two bits are in error, the parity check will pass, and the errors will go undetected.

7 bits of data (count of 1 bits)	8 bits including parity	
	even	odd
0000000 (0)	**0**0000000 (0)	**1**0000000 (1)
1010001 (3)	**1**1010001 (4)	**0**1010001 (3)
1101001 (4)	**0**1101001 (4)	**1**1101001 (5)
1111111 (7)	**1**1111111 (8)	**0**1111111 (7)

Exercise 1

- Data given: “AZ15” (Use Parity Check Bit – even parity)
- A -> 0x41 -> 1000001 -> 01000001
- Z -> 0x5A -> 1011010 -> 01011010
- 1 -> 0x31 -> 0110001 -> 00110001
- 5 -> 0x35 -> 0110101 -> 00110101

Checksums

- Idea:
  - Add up the bytes in the data
  - Include the sum in the frame

- Use ones-complement arithmetic
- Lower overhead than parity: 16 bits per frame
- But, not resilient to errors
  - Why? 1 101001 + 0 101001= 10010010
- Used in UDP, TCP, and IP

Error Detection Schemes – Checksum

The Sender follows these steps:

- data is divided into k segments, each of n bits.
- All segments are added using one's complement toget the sum.
- The sum is complemented and becomes the checksum.
- The checksum is sent with the data

The Receiver follows these steps:

- data is divided into k segments, each of n bits.
- All sections are added using one's complement toget the sum.
- The sum is complemented.
- If the result is zero, the data are accepted:otherwise, rejected.

Error Detection Schemes – Checksum

- Suppose the following block of 16 bits is to be sent using a checksum of 8 bits. 10101001 00111001. The numbers are added using one's complement:

	10101001
	00111001
	00000000

Sum	11100010
Checksum	00011101

- **The pattern sent is 10101001 00111001 00011101**
- **Now suppose the receiver receives the pattern with no error.**

  10101001 00111001 00011101
- **When the receiver adds the three blocks, it will get all 1s, which, after complementing, is all 0s and shows that there is no error.**

	10101001
	00111001
	00011101
Sum	11111111
Complement	00000000 means that the pattern is OK.

Error Detection Schemes – CRC

- One of the most common, and one of the most powerful, error-detecting codes is the Cyclic Redundancy Check (CRC). CRC is well suited for detecting burst errors and is particularly easy to implement in hardware, and is therefore commonly used in digital networks and storage devices such as hard disk drives.
- Implementation of CRC can be described as follow:
  - Given a k bit block of bits, or message, the transmitter generates an (n - k)-bit sequence, known as a Frame Check Sequence (FCS), such that the resulting frame, consisting of n bits, is exactly divisible by some predetermined number.
  - The receiver then divides the incoming frame by that number and, if there is no remainder, assumes there was no error.

Error Detection Schemes – CRC

- CRC is a block code which uses a shift register to perform encoding and decoding.
- The code word with n bits is expressed as

$$c(x)=c_1x^{n-1}+c_2x^{n-2}\ldots\ldots+c_nx^0$$

where each c_i is either a 1 or 0.

$$c(x) = m(x)\, x^{n-k} + c_p(x)$$

where $c_p(x)$ = remainder from dividing $m(x)\, x^{n-k}$ by generator $g(x)$

- If the received signal is $c(x) + e(x)$, where $e(x)$ is the error,
  - To check if received signal is error free, the remainder (syndrome) from dividing $c(x) + e(x)$ by $g(x)$ is obtained.
  - If this is 0 then the received signal is considered error free else error pattern is detected from known error syndromes.

CRC – Code Creation & Detection

- The CRC creation process is defined as follows:
  - Get the block of raw message (k bits).
  - Left shift the raw message by (n-k) bits and then divide it by G (r bits).
  - Get the remainder R as FCS (number of bits, r=n-k bits)
  - Append the R to the raw message . The result (n bits) is the frame to be transmitted.
- CRC is checked using the following process:
  - Receive the frame (n bits)
  - Divide it by G (r bits)
  - Check the remainder. If the remainder is not zero, then there is an error in the frame.
- These procedures can be further clarified in three ways: modulo 2 arithmetic, polynomials, and digital logic.

CRC – Modulo 2 Arithmetic

- Modulo-2 arithmetic is in fact just the exclusive-OR (XOR) operation.

$$\begin{array}{rr} 1111 & 1111 \\ +\ 1010 & -\ 0101 \\ \hline 0101 & 1010 \end{array}$$

- Now define

  T = n-bit frame to be transmitted

  D = k-bit block of data, or message, the first k bits of T

  F = (n - k)-bit FCS, the last (n - k) bits of T

  P = pattern of n - k + 1 bits; this is the predetermined divisor

- $T_{(n\ bits)} = D_{(k\ bits)} \mid F_{(n-k\ bits)}$

  => $T = 2^{(n-k)}D + F$

  By multiplying D by $2^{(n-k)}$, we have in effect shifted it to the left by (n – k) bits and padded out the result with zeroes. Adding F yields the concatenation of D and F, which is T.

CRC – Modulo 2 Arithmetic Example

1. Given

Message D = 1010001101 (10 bits)

Pattern P = 110101 (6 bits)

FCS R = to be calculated (5 bits)

Thus, $n = 15$, $k = 10$, and $(n - k) = 5$.

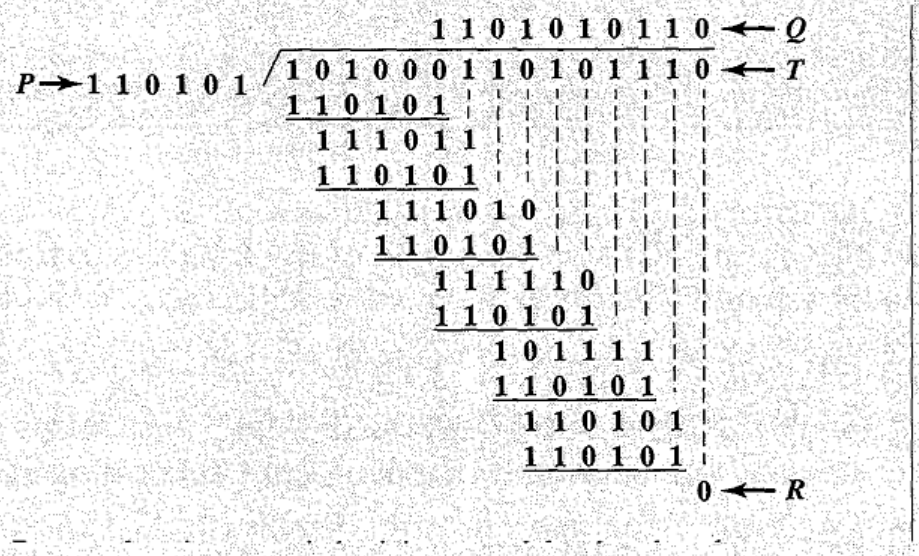
2. The message is multiplied by 2^5, yielding 101000110100000.

3. This product is divided by P:

```
                  1 1 0 1 0 1 0 1 1 0  ← Q
            ┌─────────────────────────
P → 1 1 0 1 0 1 │ 1 0 1 0 0 0 1 1 0 1 0 0 0 0 0  ← 2^(n−k)D
                  1 1 0 1 0 1
                  ─────────
                    1 1 1 0 1 1
                    1 1 0 1 0 1
                    ─────────
                      1 1 1 0 1 0
                      1 1 0 1 0 1
                      ─────────
                        1 1 1 1 1 0
                        1 1 0 1 0 1
                        ─────────
                          1 0 1 1 0 0
                          1 1 0 1 0 1
                          ─────────
                            1 1 0 0 1 0
                            1 1 0 1 0 1
                            ─────────
                              0 1 1 1 0  ← R
```

CRC – Modulo 2 Arithmetic Example (cont)

4. The remainder is added to 2^5D to give $T = 101000110101110$, which is transmitted.
5. If there are no errors, the receiver receives T intact. The received frame is divided by P:



Because there is no remainder, it is assumed that there have been no errors.

CRC – Polynomials

- In polynomials approach, the CRC process can be viewed by expressing all values as polynomials in a dummy variable X, with binary coefficients. The coefficients correspond to the bits in the binary number.
    - A generator polynomial with constant coefficients, is used as the divisor in a polynomial long division over a finite field.
    - Taking the input data (blocks of input bits called message polynomial) as the dividend.
    - Where the remainder (the FCS) becomes the result.
- Even parity is a special case of a single-bit CRC, where the single-bit FCS is generated by the generator polynomial (divisor) x+1.

CRC - Polynomials

- Message polynomial is divided by Generator polynomial giving quotient and remainder, the coefficients of the remainder form the bits of final CRC.
- Define:

  M – The original frame (k bits) to be transmitted before adding the Frame Check Sequence (FCS).

  F – The resulting FCS of r bits to be added to M (usually r=8, 16, 32).

  T – The cascading of M and F.

  G – The predefined CRC generating polynomial with pattern of r+1 bits..
- The main idea in CRC algorithm is that the FCS is generated so that the remainder of T/G is zero.

CRC – Polynomials Example

0 1 1 1 1 0 1 1

Example : Find the codewords c(x) if $m(x)=x^2+x+1$ and $g(x)=x^3+x+1$ for (7,4) CRC.

We have n = Total number of bits = 7, k = Number of information bits = 4,
r = Number of parity bits = n - k = 3.

$$\because \quad c_p(x) = rem\left[\frac{m(x)x^{n-k}}{g(x)}\right] = rem\left[\frac{(x^2+x+1)x^3}{x^3+x+1}\right]$$

$$= rem\left[\frac{x^5+x^4+x^3}{x^3+x+1}\right] = x$$

m(x) has been shifted left for r bits => 0 1 1 1 0 0 0

Then,

$$c(x) = m(x)x^{n-k} + c_p(x) = x^5 + x^4 + x^3 + x$$

0 1 1 1 0 1 0

If the received bits = transmitted bits = 0 1 1 1 0 1 0,

$$\frac{T}{G} = rem\left[\frac{x^5+x^4+x^3+x}{x^3+x+1}\right] = 0$$

=> No Error !

Error Detection Schemes – CRC Capabilities

- An error E(X) will only be undetectable if it is divisible by G(X). It can be shown that all of the following errors are not divisible by a suitably chosen G(X) and hence are detectable:
  - All single-bit errors, if G(X) has more than one non-zero term.
  - All double-bit errors, as long as G(X) has a factor with at least three terms.
  - Any odd number of errors, as long as P(X) contains a factor (X + 1)
  - Any burst error for which the length of the burst is less than or equal to (n – k); that is, less than or equal to the length of the FCS
  - A fraction of error bursts of length n - k + 1; the fraction equals $(1 - 2^{-(n-k-1)})$
  - A fraction of error bursts of length greater than n - k + 1; the fraction equals $(1 - 2^{-(n-k)})$

Error Detection Schemes – CRC

- Common CRC Codes

Code	Generator polynomial g(x)	Parity check bits
CRC-12	$1+x+x^2+x^3+x^{11}+x^{12}$	12
CRC-16	$1+x^2+x^{15}+x^{16}$	16
CRC-CCITT	$1+x^5+x^{15}+x^{16}$	16

- The CRC-12 system is used for transmission of streams of 6-bit characters and generates a 12-bit FCS.
- Both CRC-16 and CRC-CCITI are popular for 8-bit characters, in the United States and Europe, respectively, and both result in a 16-bit FCS.

Exercise 3

- Data given: “AZ15” (Use CRC-16)
- A -> 0x41 -> 01000001
- Z -> 0x5A -> 01011010
- 1 -> 0x31 -> 00110001
- 5 -> 0x35 -> 00110101

x^{31}	x^{30}	x^{29}	x^{28}	x^{27}	x^{26}	x^{25}	x^{24}		x^{0}
0	1	0	0	0	0	0	1	…	1

$$R(x) = rem\left[\frac{m(x)x^{n-k}}{g(x)}\right] = rem\left[\frac{(x^{30}+x^{24}+\ldots+x^{2}+1)x^{16}}{x^{16}+x^{15}+x^{2}+1}\right]$$

$$= rem\left[\frac{x^{46}+x^{40}+\ldots+x^{18}+x^{16}}{x^{16}+x^{15}+x^{2}+1}\right] = x^{13}+x^{12}+x^{10}+x^{9}+x^{6}+x^{5}+x^{3}+x+1$$

$$R(x) = x^{13} + x^{12} + x^{10} + x^9 + x^6 + x^5 + x^3 + x + 1$$

x^{15}	x^{14}	x^{13}	x^{12}	x^{11}	x^{10}	x^9	x^8	x^7	x^6	x^5	x^4	x^3	x^2	x^1	x^0
0	0	1	1	0	1	1	0	0	1	1	0	1	0	1	1

- CRC-16: 0x366B
- Codeword (in HEX): 0x415A3135366B

- Kode Hamming merupakan kode non-trivial untuk koreksi kesalahan yang pertama kali diperkenalkan.
- Kode ini dan variansinya telah lama digunakan untuk kontrol kesalahan pada sistem komunikasi digital.
- Kode Hamming biner dapat direpresentasikan dalam bentuk persamaan:

  $(n,k) = (2^m-1, 2^m-1-m)$

- Contoh:

  jika m = jumlah paritas = 3

  k = jumlah data = 4

  n = jumlah bit informasi yang membentuk n sandi = 7

  maka kode Hamming nya adalah C (7,4) dengan dmin =3

Forward Error Correction

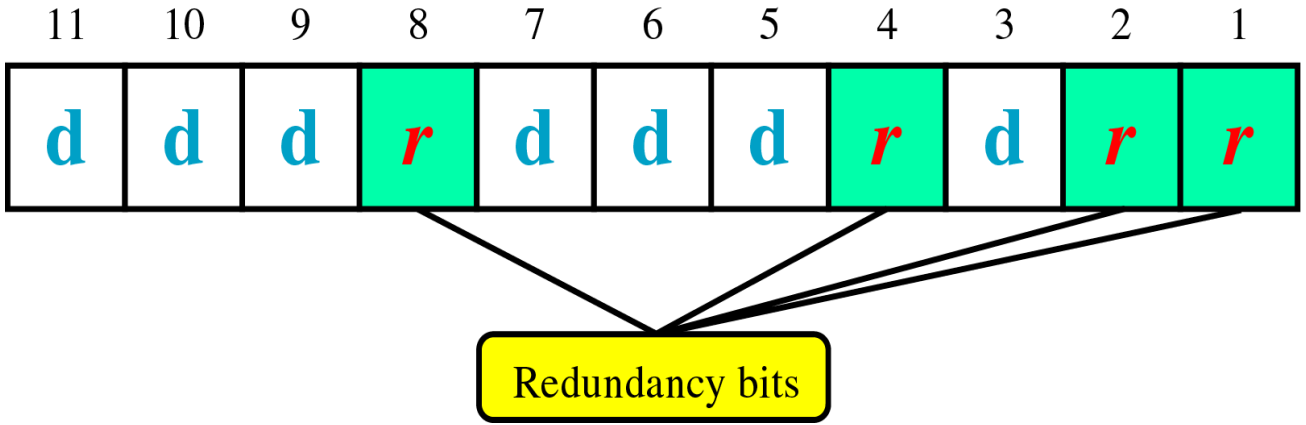
- Error Correcting codes dinyatakan sebagai penerusan koreksi kesalahan untuk mengindikasikan bahwa pesawat penerima sedang mengoreksi kesalahan.
- Kode pendeteksi yang paling banyak digunakan merupakan kode Hamming.
- Posisi bit-bit Hamming dinyatakan dalam 2^n dengan n bilangan bulat sehingga bit-bit Hamming akan berada dalam posisi 1, 2, 4, 8, 16, dst..

Error Correction(cont'd)

- Hamming Code

  ~ developed by R.W.Hamming

- positions of redundancy bits in



The key to the Hamming Code is the use of extra parity bits to allow the identification of a single error. Create the code word as follows:

- **Mark all bit positions that are powers of two as parity bits. (positions 1, 2, 4, 8, 16, 32, 64, etc.)**
- **All other bit positions are for the data to be encoded. (positions 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, etc.)**
- **Each parity bit calculates the parity for some of the bits in the code word. The position of the parity bit determines the sequence of bits that it alternately checks and skips.**

  Position 1: check 1 bit, skip 1 bit, check 1 bit, skip 1 bit, etc. (1,3,5,7,9,11,13,15,...)
  Position 2: check 2 bits, skip 2 bits, check 2 bits, skip 2 bits, etc. (2,3,6,7,10,11,14,15,...)
  Position 4: check 4 bits, skip 4 bits, check 4 bits, skip 4 bits, etc. (4,5,6,7,12,13,14,15,20,21,22,23,...)
  Position 8: check 8 bits, skip 8 bits, check 8 bits, skip 8 bits, etc. (8-15,24-31,40-47,...)
  Position 16: check 16 bits, skip 16 bits, check 16 bits, skip 16 bits, etc. (16-31,48-63,80-95,...)
  Position 32: check 32 bits, skip 32 bits, check 32 bits, skip 32 bits, etc. (32-63,96-127,160-191,...)
  etc.
- **Set a parity bit to 1 if the total number of ones in the positions it checks is odd. Set a parity bit to 0 if the total number of ones in the positions it checks is even.**

Bit position		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Encoded data bits		p1	p2	d1	p3	d2	d3	d4	p4	d5	d6	d7	d8	d9	d10	d11	p5	d12	d13	d14	d15	
Parity bit coverage	p1	X		X		X		X		X		X		X		X		X		X		
	p2		X	X			X	X			X	X			X	X			X	X		...
	p3				X	X	X	X					X	X	X	X					X	
	p4								X	X	X	X	X	X	X	X						
	p5																X	X	X	X	X	

Example:

A byte of data: **10011010**

Create the data word, leaving spaces for the parity bits: _ _ **1** _ 0 0 **1** _ **1 0 1 0**

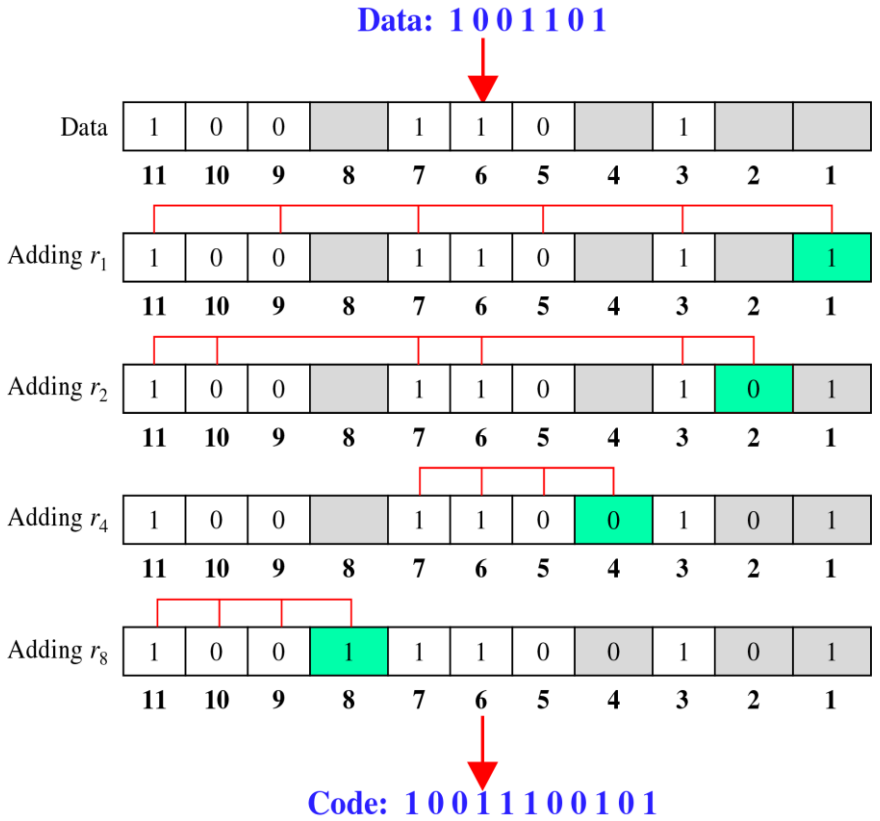
Calculate the parity for each parity bit (a ? represents the bit position being set):

- ☐ Position 1 checks bits 1,3,5,7,9,11:
  ? _ **1** _ **0** 0 **1** _ **1** 0 **1** 0. Even parity so set position 1 to a 0: **0** _ **1** _ **0** 0 **1** _ **1** 0 **1** 0
- ☐ Position 2 checks bits 2,3,6,7,10,11:
  0 **? 1** _ 0 **0 1** _ 1 **0 1** 0. Odd parity so set position 2 to a 1: 0 **1 1** _ 0 **0 1** _ 1 **0 1** 0
- ☐ Position 4 checks bits 4,5,6,7,12:
  0 1 1 **? 0 0 1** _ 1 0 1 **0**. Odd parity so set position 4 to a 1: 0 1 1 **1 0 0 1** _ 1 0 1 **0**
- ☐ Position 8 checks bits 8,9,10,11,12:
  0 1 1 1 0 0 1 **? 1 0 1 0**. Even parity so set position 8 to a 0: 0 1 1 1 0 0 1 **0 1 0 1 0**

- ☐ Code word: 011100101010.

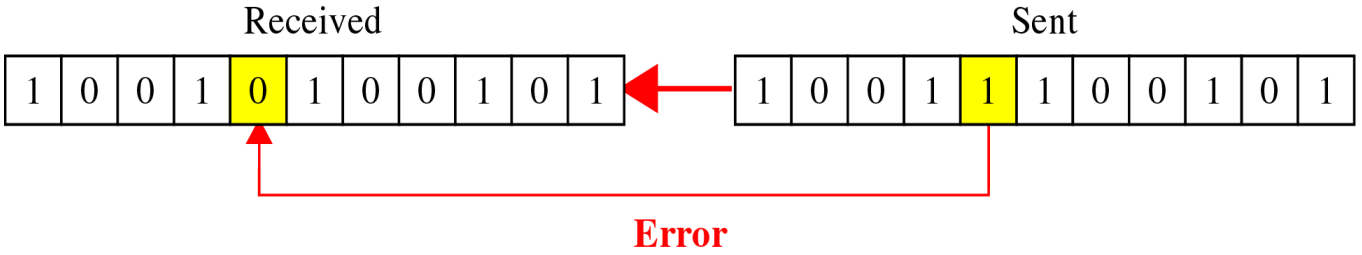
Error Correction(cont'd)

- Calculating the r values



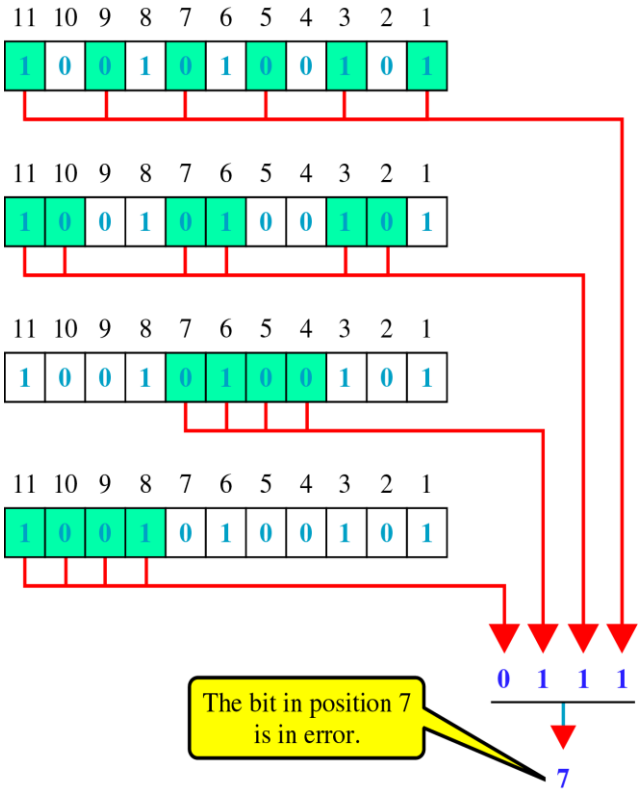
Contoh Error Correction(cont'd)

- Error Detection and Correction



Error Correction(cont'd)

- Error detection using Hamming Code



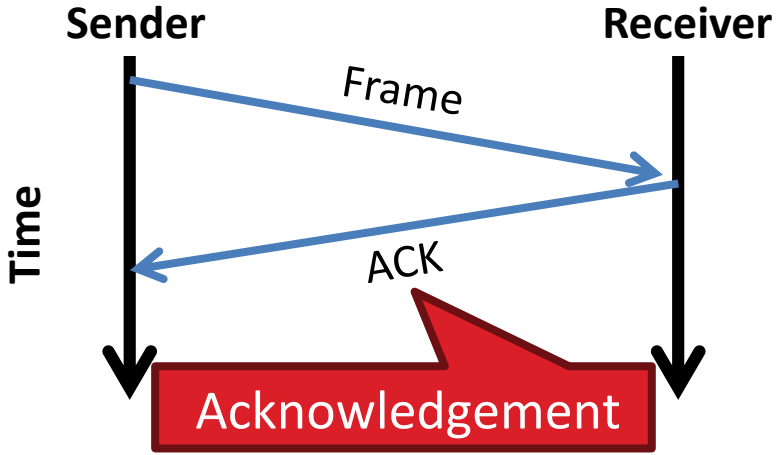
Error Correction(cont'd)

- Multiple-Bit Error Correction
  - redundancy bits calculated on overlapping sets of data units can also be used to correct multiple-bit errors.

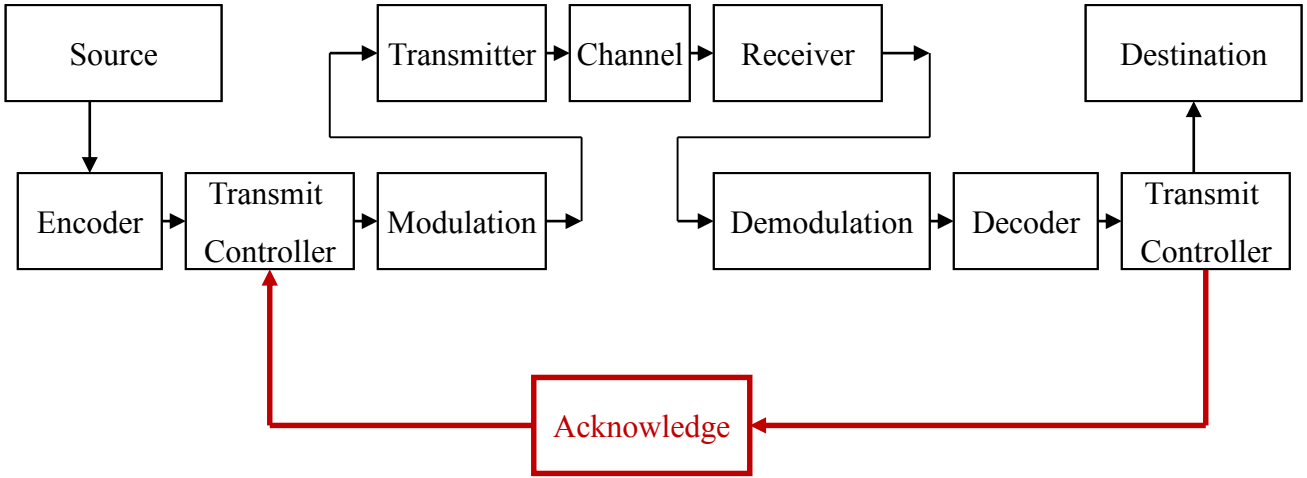
  Ex) to correct double-bit errors, we must take into consideration that two bits can be a combination of any two bits in the entire sequence

What About Reliability?

- How does a sender know that a frame was received?
  - What if it has errors?
  - What if it never arrives at all?



Channel Coding – Automatic Repeat Request Systems (ARQ)



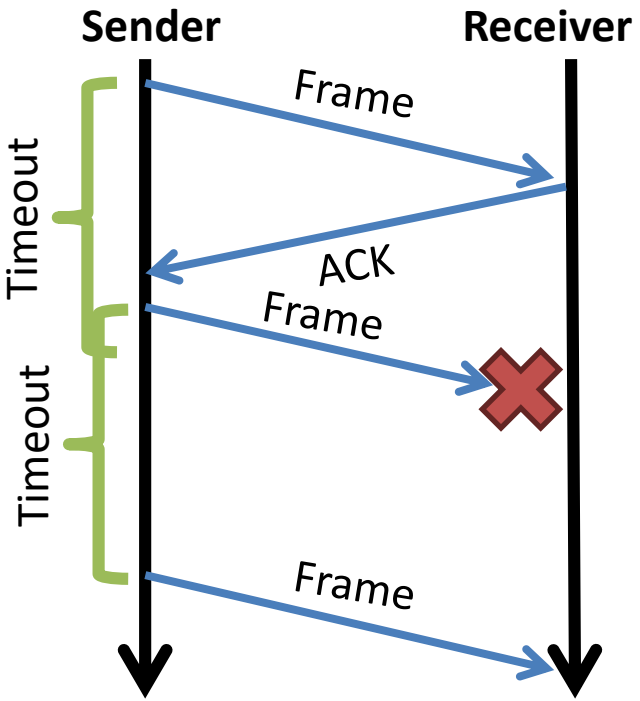
- Stop and Wait ARQ
- Sliding Window

Stop and Wait

- Simplest form of reliability
- Example: Bluetooth
- Problems?
  - Utilization
  - Can only have one frame in flight at any time
- 10Gbps link and 10ms delay
  - Assume packets are 1500B

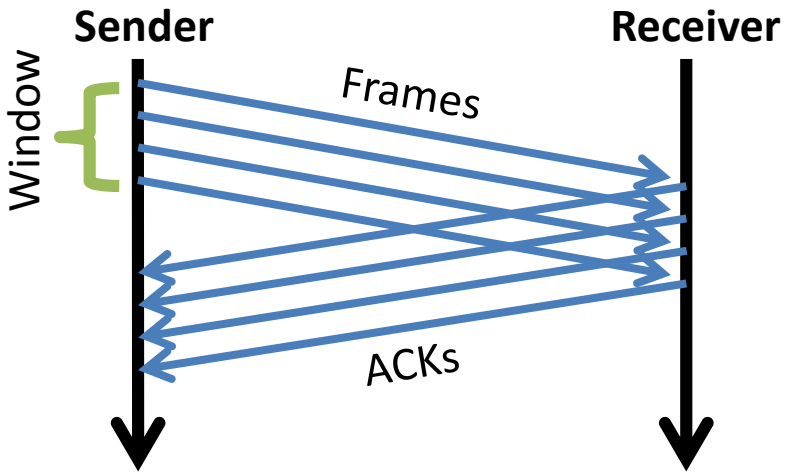
1500B*8bit/(2*10ms) = 600Kbps

Utilization is 0.006%



Sliding Window

- Allow multiple outstanding, un-ACKed frames
- Number of un-ACKed frames is called the window



- Made famous by TCP
  - We'll look at this in more detail later

Should We Error Check in the Data Link?

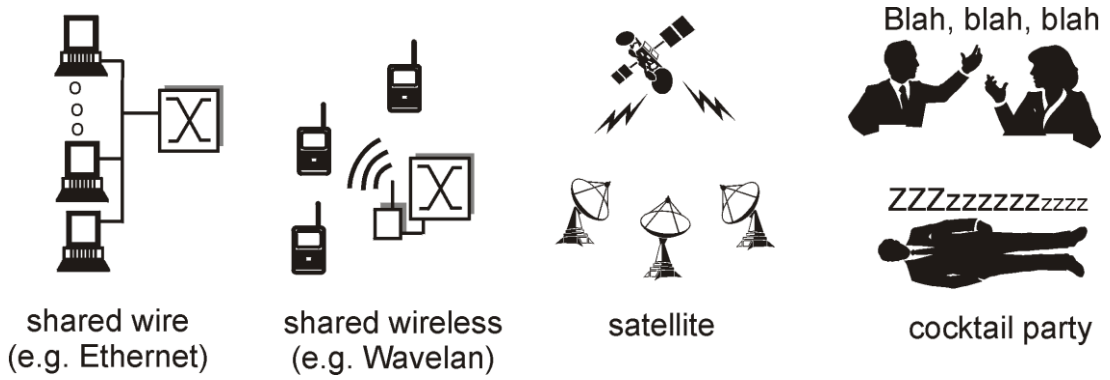
- Recall the End-to-End Argument
- Cons:
  - Error free transmission cannot be guaranteed
  - Not all applications want this functionality
  - Error checking adds CPU and packet size overhead
  - Error recovery requires buffering
- Pros:
  - Potentially better performance than app-level error checking
- Data link error checking in practice
  - Most useful over lossy links
  - Wifi, cellular, satellite

- Prinsip dasar
- Peran data link layer
- Framing
- Error handling
- Ethernet and Medium Access

Multiple Access Links and LANs

Two types of “links”:

- point-to-point, e.g.,
  - PPP for dial-up access, or over optical fibers
- broadcast (shared wire or medium), e.g.
  - traditional Ethernet
  - 802.11 wireless LAN



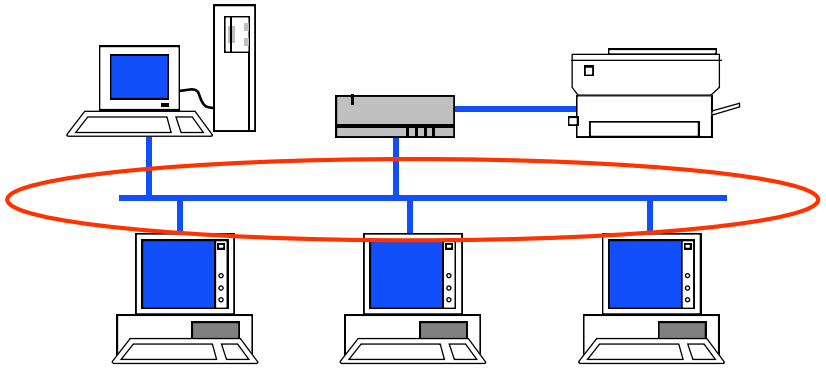
Broadcast Links: Multiple Access

Single shared communication channel

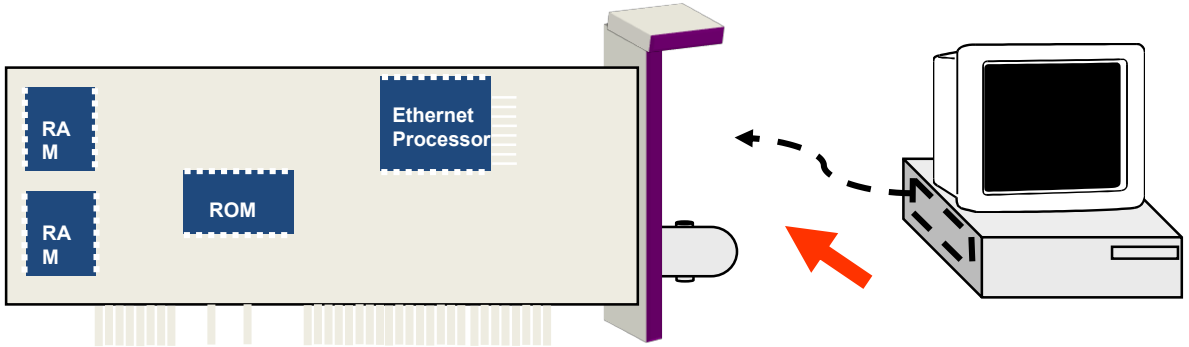
- Only one can send successfully at a time
- Two or more simultaneous transmissions
  - interference!
- How to share a broadcast channel
  - media access control uses same shared media

- Humans use multi-access protocols all the time

Typical LAN Structure

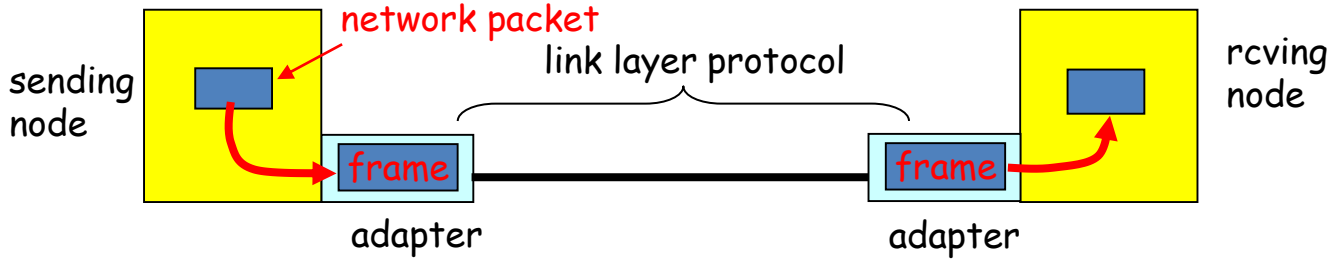


- Transmission Medium
- Network Interface Card (NIC)
- *Unique MAC "physical" address*



Adaptors Communicating

- link layer implemented in “adaptor” (aka NIC), with “transceiver” in it
  - Ethernet card, dial-up modem, 802.11 wireless card
- sending side:
  - encapsulates packet in frame
  - adds error checking bits, flow control, reliable data transmission, etc.
- receiving side
  - looks for errors, flow control, reliable data transmission, etc
  - extracts packet, passes to receiving node
- data link & physical layers are closely coupled!



MAC (Physical) Addresses

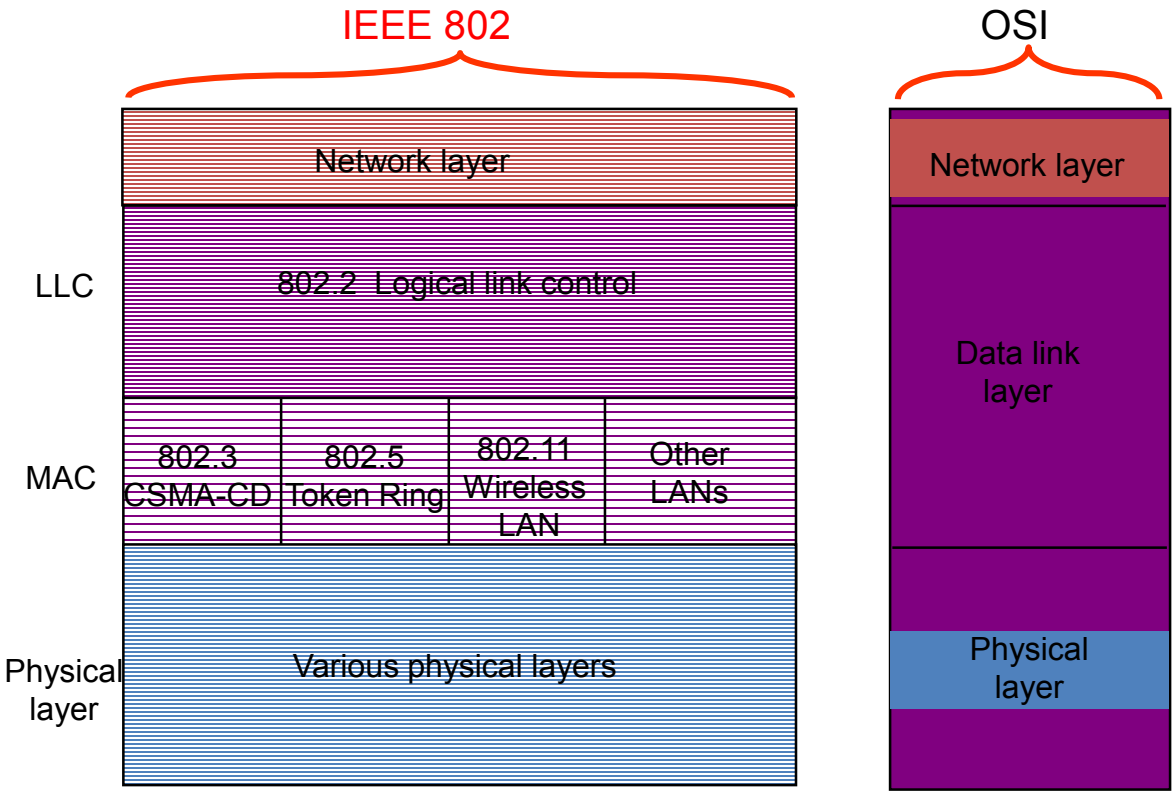
- Addressing needed in shared media
  - MAC (media access control) or physical addresses
  - To identify source and destination interfaces and get frames delivered from one interface to another physically-connected interface (i.e., on same physical local area network!)
- 48 bit MAC address (for most LANs)
  - fixed for each adaptor, burned in the adapter ROM
  - MAC address allocation administered by IEEE
    - 1st bit: 0 unicast, 1 multicast.
    - all 1's : broadcast
- MAC flat address -> portability
  - can move LAN card from one LAN to another

MAC (Physical, or LAN) Addresses

MAC addressing operations on a LAN:

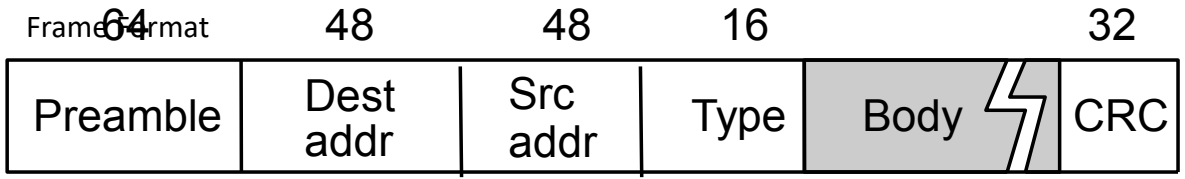
- each adaptor on the LAN "sees" all frames
- accept a frame only if dest. (unicast) MAC address matches its own MAC address
- accept all broadcast (MAC= all 1's) frames
- accept all frames if set in "promiscuous" mode
- can configure to accept certain multicast addresses (first bit = 1)

MAC Sub-layer



Ethernet Overview

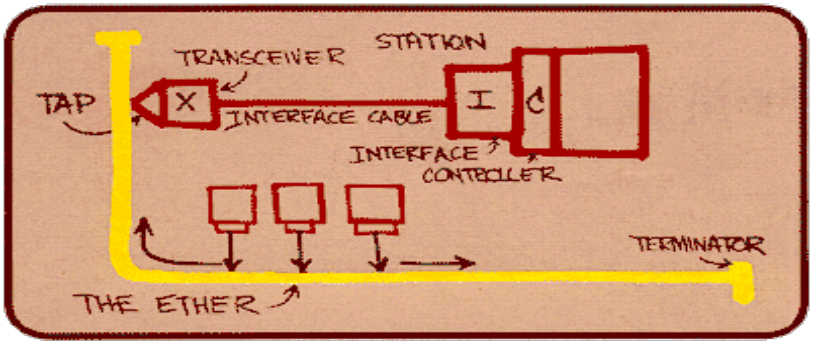
- History
  - developed by Xerox PARC in mid-1970s
  - roots in Aloha packet-radio network
  - standardized by Xerox, DEC, and Intel in 1978
  - similar to IEEE 802.3 standard
- CSMA/CD
  - carrier sense
  - multiple access
  - collision detection
- Frame format



Ethernet

"Dominant" LAN technology:

- cheap $20 for 100Mbs!
- first widely used LAN technology
- Simpler, cheaper than token ring LANs and ATM
- Kept up with speed race: 10, 100, 1000 Mbps

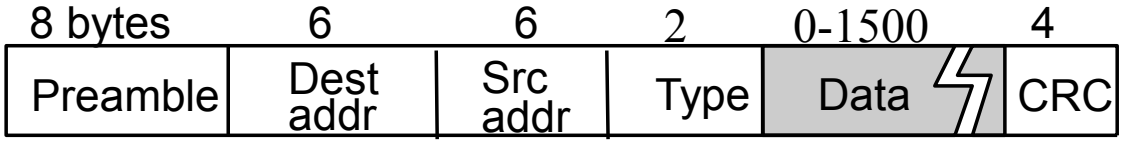


Metcalfe's Ethernet sketch

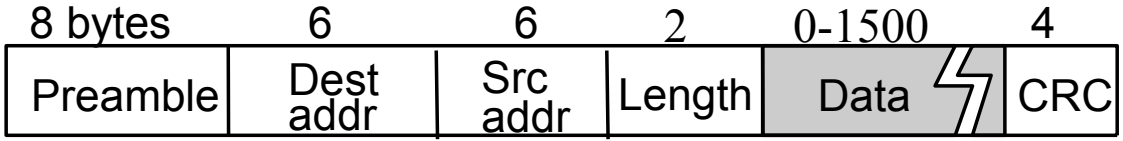
Ethernet Frame Format

Sending adapter encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame

DIX frame format



IEEE 802.3 format



- Ethernet has a maximum frame size: data portion <=1500 bytes
- It imposes a minimum frame size: 64 bytes (excluding preamble)
  If data portion <46 bytes, pad with "junk" to make it 46 bytes

Q: Why minimum frame size in Ethernet?

Fields in Ethernet Frame Format

- Preamble:
  - 7 bytes with pattern 10101010 followed by one byte with pattern 10101011 (SoF: start-of-frame)
  - used to synchronize receiver, sender clock rates, and identify beginning of a frame
- Addresses: 6 bytes
  - if adapter receives frame with matching destination address, or with broadcast address (eg ARP packet), it passes data in frame to network layer protocol (specified by TYPE field)
  - otherwise, adapter discards frame
- Type: indicates the higher layer protocol, mostly IP but others may be supported such as Novell IPX and AppleTalk
  - 802.3: Length gives data size; “protocol type” included in data
- CRC: checked at receiver, if error is detected, the frame is simply dropped

IEEE 802.3 MAC: Ethernet

MAC Protocol:

- CSMA/CD
- Truncated binary exponential backoff
  - for retransmission n: $0 < r < 2^k * \text{ASlotTime}$, where $k=\min(n,10)$
  - give up after 16 retransmissions
- *Slot Time* is the critical system parameter
  - upper bound on time to detect collision
  - upper bound on time to acquire channel
  - upper bound on length of frame segment generated by collision
  - quantum for retransmission scheduling
  - max{round-trip propagation, MAC jam time}

IEEE 802.3 Parameters

- 1 bit time = time to transmit one bit
  - 10 Mbps ➔ 1 bit time = 0.1 μs
- Maximum network diameter ≤ 2.5km
  - Maximum 4 repeaters
- "Collision Domain"
  - Distance within which collision can occur and be detected
  - IEEE 802.3 specifies:
    worst case collision detection time: 51.2 μs
- Slot time
  - 51.2 μs = 512 bits = 64 bytes
- Why minimum frame size?
  - 51.2 μs => minimum # of bits can be transited at 10Mpbs is 512 bits => 64 bytes is required for collision detection

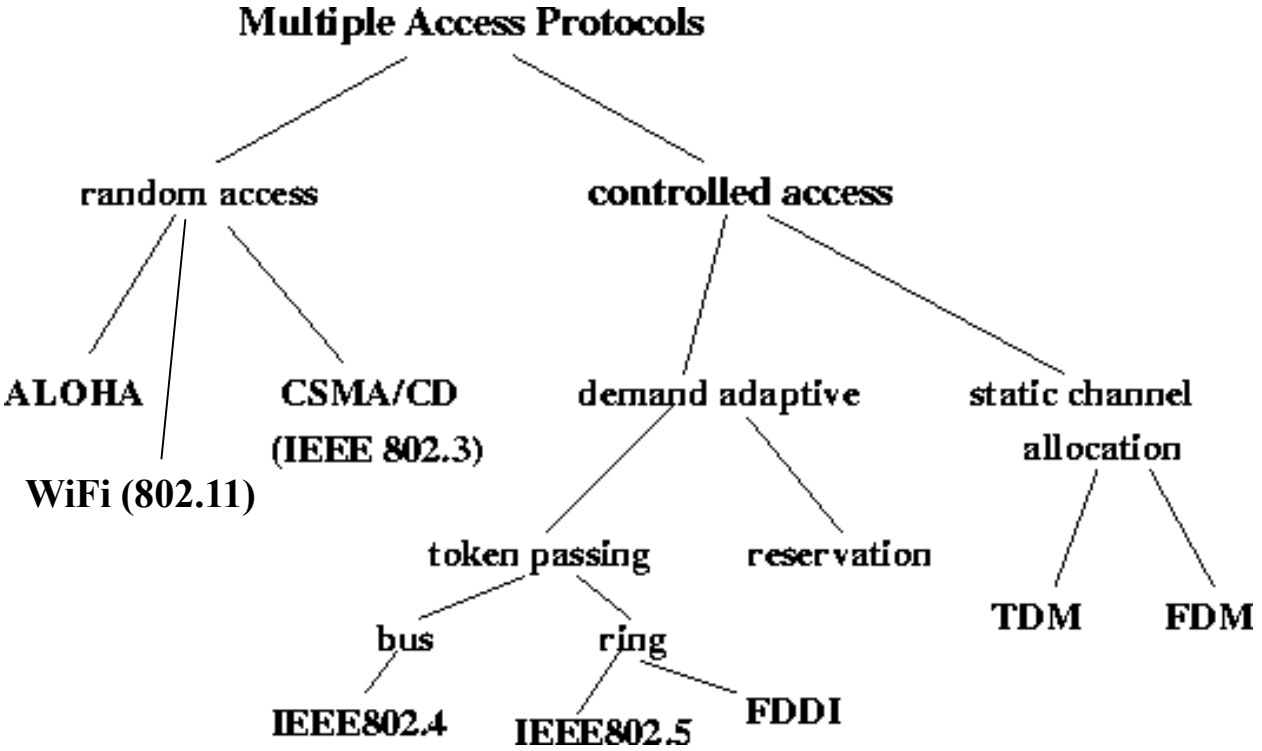
Random Access

- Stations contend for channels
- Overlapping transmissions (collisions) can occur
  - Carrier sensing?
  - Collision detection?
- Protocols
  - Aloha (not covered)
  - Slotted Aloha (not covered)
  - Carrier Sense Multiple Access: Ethernet

Controlled Access

- Stations reserve or are allocated channel
  - No collisions
  - Allocation: static or dynamic
- Protocols
  - Static channel allocation
    - Time division multiple access
  - Demand adaptive channel allocation
    - Reservation protocols
    - Token passing (token bus, token ring)

Taxonomy of MAC Protocols



Carrier Sense Multiple Access

- CSMA: Listen before transmit
  - If channel idle, transmit entire packet
  - If busy, defer transmission
    - How long should we wait?
  - Human analogy: don't interrupt others
- Can carrier sense avoid collisions completely?

Persistent and Non-persistent CSMA

- Non-persistent
  - If idle, transmit
  - If busy, wait random amount of time
  - If collision, wait random amount of time
- p-persistent
  - If idle, transmit with probability p
  - If busy, wait till it becomes idle
  - If collision, wait random amount of time

CSMA/CD

CSMA with collision detection (CD)

- Listen while talking
- Stop transmitting when collision detected
  - Compare transmitted and received signals
  - Save time and bandwidth
  - Improvement over persistent and nonpersistent protocols
- Human analogy
  - Polite conversationalist
- Worst case time to detect a collision?

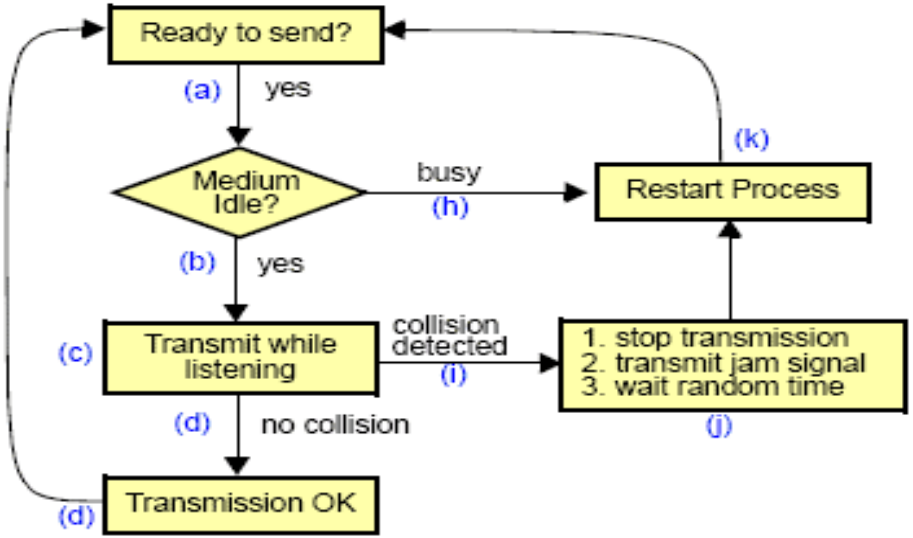
Ethernet MAC Protocol: Basic Ideas

- Ethernet uses CSMA/CD – listens to line before/during sending
- If line is idle (no carrier sensed)
  - send packet immediately
  - upper bound message size of 1500 bytes
  - must wait 9.6us between back-to-back frames
- If line is busy (carrier sensed)
  - wait until idle and transmit packet immediately
    - called *1-persistent* sending
- If collision detected
  - Stop sending and jam signal
  - Try again later

Ethernet CSMA/CD Algorithm

1. Adaptor gets datagram and creates frame
2. If adapter senses channel idle, it starts to transmit frame. If it senses channel busy, waits until channel idle and then transmits
3. If adapter transmits entire frame without detecting another transmission, the adapter is done with frame ! Signal to network layer "transmit OK"
4. If adapter detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, adapter enters exponential backoff: after the m^{th} collision, adapter chooses a K at random from $\{0,1,2,\ldots,2^m-1\}$. Adapter waits K*512 bit times and returns to Step 2
6. Quit after 16 attempts, signal to network layer "transmit error"

Ethernet CSMA/CD Alg. Flow Chart

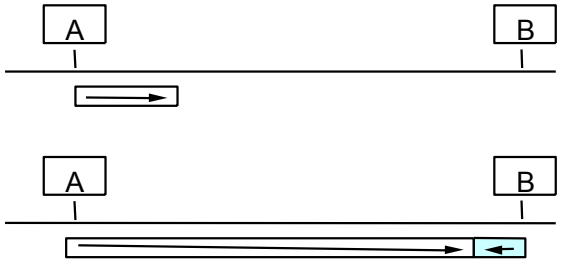


Collisions

Collisions are caused when two adaptors transmit at the same time (adaptors sense collision based on voltage differences)

- Both found line to be idle
- Both had been waiting to for a busy line to become idle

A starts at time 0



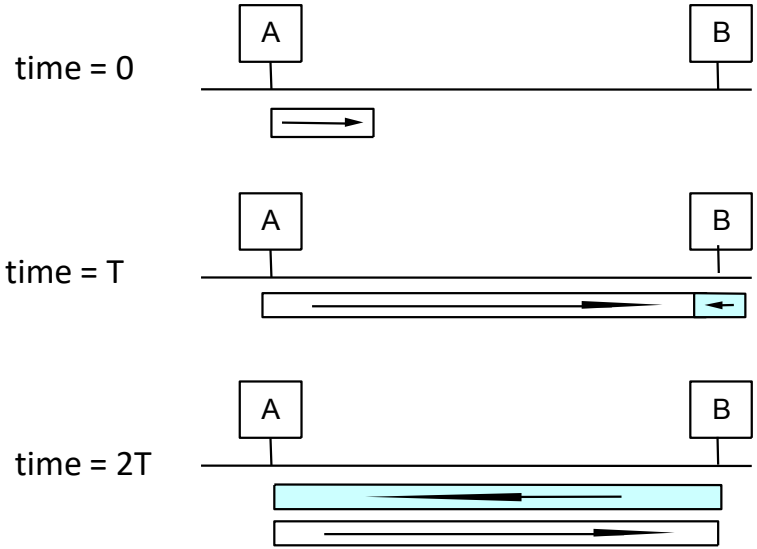
Message almost there at time T when B starts – collision!

How can we be sure A knows about the collision?

Collision Detection

- How can A know that a collision has taken place?
  - There must be a mechanism to insure retransmission on collision
  - A's message reaches B at time T
  - B's message reaches A at time 2T
  - So, A must still be transmitting at 2T
- IEEE 802.3 specifies max value of 2T to be 51.2us
  - This relates to maximum distance of 2500m between hosts
  - At 10Mbps it takes 0.1us to transmit one bit so 512 bits (64B) take 51.2us to send
  - So, Ethernet frames must be at least 64B long
    - 14B header, 46B data, 4B CRC
    - Padding is used if data is less than 46B
- Send jamming signal after collision is detected to insure all hosts see collision
  - 48 bit signal

Collision Detection contd.



Exponential Backoff

- If a collision is detected, delay and try again
- Delay time is selected using binary exponential backoff
  - 1st time: choose K from {0,1} then delay = K * 51.2us
  - 2nd time: choose K from {0,1,2,3} then delay = K * 51.2us
  - *nth* time: delay = *K* x 51.2us, for K=0..2^n – 1
    - Note max value for k = 1023
  - give up after several tries (usually 16)
    - Report transmit error to host
- If delay were not random, then there is a chance that sources would retransmit in lock step
- Why not just choose from small set for K
  - This works fine for a small number of hosts
  - Large number of nodes would result in more collisions

MAC Algorithm from the Receiver Side

- Senders handle all access control
- Receivers simply read frames with acceptable address
  - Address to host
  - Address to broadcast
  - Address to multicast to which host belongs
  - All frames if host is in promiscuous mode

Ethernet's CSMA/CD (more)

Jam Signal: make sure all other transmitters are aware of collision; 32 bits;

Bit time: .1 microsec for 10 Mbps Ethernet ; for K=1023, wait time is about 50 msec

Exponential Backoff:

- *Goal*: adapt retransmission attempts to estimated current load
  - heavy load: random wait will be longer
- first collision: choose K from {0,1}; delay is K x 512 bit transmission times
- after second collision: choose K from {0,1,2,3}…
- after ten collisions, choose K from {0,1,2,3,4,…,1023}

CSMA/CD Efficiency

Relevant parameters

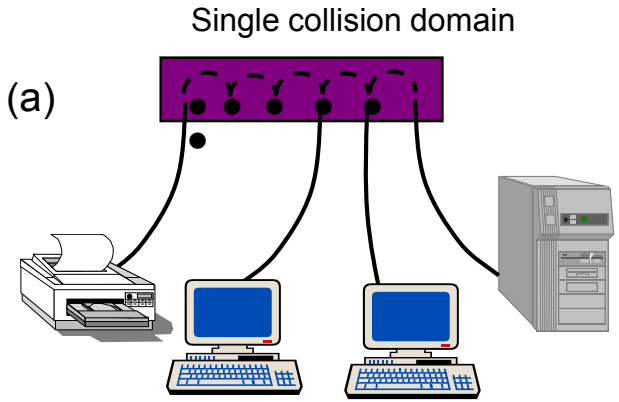
- cable length, signal speed, frame size, bandwidth

- t_{prop} = max prop between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame

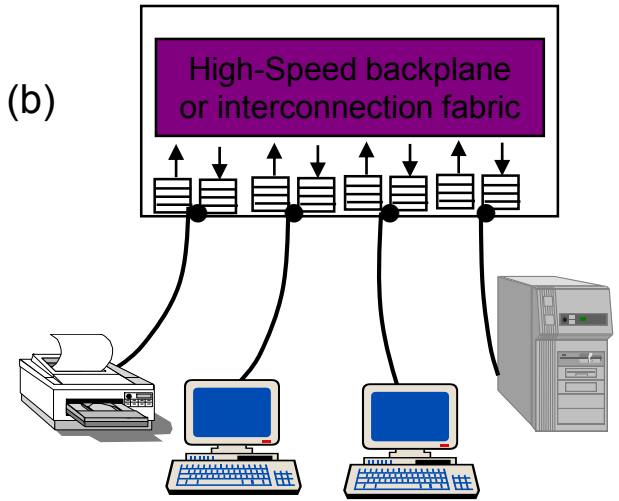
$$\text{efficiency} \approx \frac{1}{1 + 5t_{prop} / t_{trans}}$$

- Efficiency goes to 1 as t_{prop} goes to 0
- Goes to 1 as t_{trans} goes to infinity
- Much better than ALOHA, but still decentralized, simple, and cheap

Ethernet Hubs & Switches



Twisted Pair Cheap
Easy to work with
Reliable
Star-topology CSMA-CD



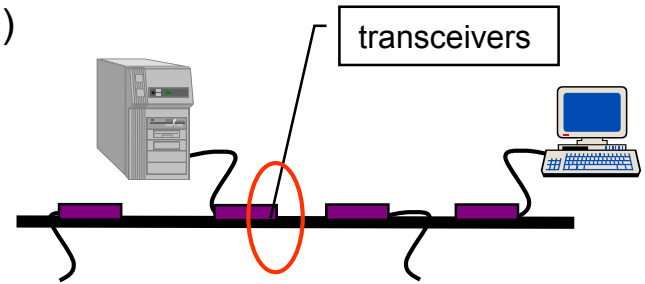
Twisted Pair Cheap
Bridging increases scalability
Separate collision domains
Full duplex operation

IEEE 802.3 Physical Layer

IEEE 802.3 10 Mbps medium alternatives

	10base5	10base2	10baseT	10baseFX
Medium	Thick coax	Thin coax	Twisted pair	Optical fiber
Max. Segment Length	500 m	200 m	100 m	2 km
Topology	Bus	Bus	Star	Point-to-point link

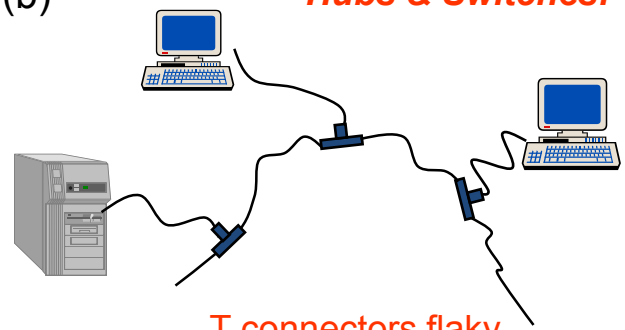
(a)



Thick Coax: Stiff, hard to work with

(b)

Hubs & Switches!



T connectors flaky

Evolution of Ethernet

From early 80's 10Base Ethernet to 90's 100Base (Fast) Ethernet to today's Gigabit Ethernet to 10 Gigabit Ethernet, ……

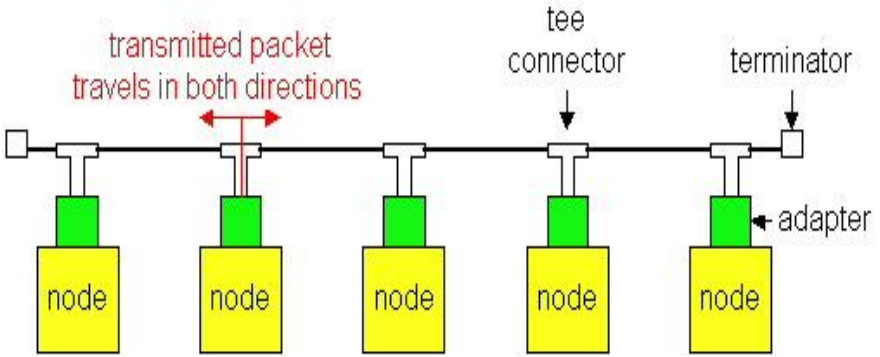
IEEE 802.3 Original Parameters

- transmission Rate: 10 Mbps
- Min Frame: 512 bits = 64 bytes
- slot time: 512 bits/10 Mbps = 51.2 μsec
  - 51.2 μsec x $2x10^5$ km/sec =10.24 km
  - 5.12 km round trip distance
- max Length: 2500 meters + 4 repeaters

- For compatibility, desire to maintain same frame format!
  - *Each x10 increase in bit rate, must be accompanied by x10 decrease in distance ?!*

Ethernet Technologies: 10Base2

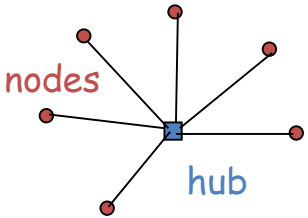
- 10: 10Mbps; 2: under 200 meters max cable length
- thin coaxial cable in a bus topology

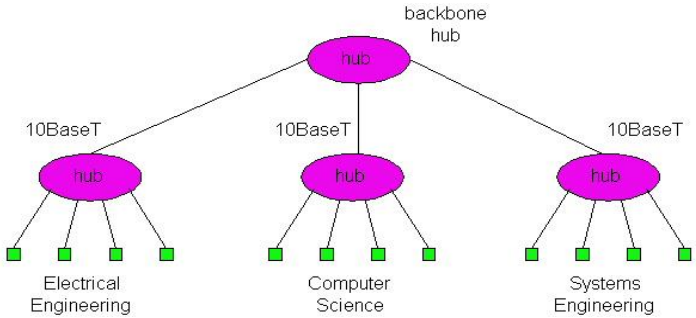


- repeaters used to connect up to multiple segments
- repeater repeats bits it hears on one interface to its other interfaces: physical layer device only!
- has become a legacy technology

10BaseT

- 10 Mbps rate
- T stands for Twisted Pair
- Nodes connect to a hub: “star topology”; 100 m max distance between nodes and hub
- Hubs are essentially physical-layer repeaters:
  - bits coming in one link go out all other links
  - no frame buffering
  - no CSMA/CD at hub: adapters detect collisions
  - provides net management functionality





Fast (100Mbps) Ethernet

- Fast Ethernet (100Mbps) has technology very similar to 10Mbps Ethernet
  - Uses different physical layer encoding (4B5B)
  - Many NIC's are 10/100 capable
    - Can be used at either speed

IEEE 802.3 100 Mbps Ethernet medium alternatives

	100baseT4	100baseT	100baseFX
Medium	Twisted pair category 3 UTP 4 pairs	Twisted pair category 5 UTP two pairs	Optical fiber multimode Two strands
Max. Segment Length	100 m	100 m	2 km
Topology	Star	Star	Star

To preserve compatibility with 10 Mbps Ethernet:

- Same frame format, same interfaces, same protocols
- Hub topology only with twisted pair & fiber
- Bus topology & coaxial cable abandoned
- Category 3 twisted pair (ordinary telephone grade) requires 4 pairs
- Category 5 twisted pair requires 2 pairs (most popular)
- Most prevalent LAN today

Gigabit Ethernet

Gigabit Ethernet Physical Layer Specification

(IEEE 802.3 1 Gigabit Ethernet medium alternatives)

	1000baseSX	1000baseLX	1000baseCX	1000baseT
Medium	Optical fiber multimode Two strands	Optical fiber single mode Two strands	Shielded copper cable	Twisted pair category 5 UTP
Max. Segment Length	550 m	5 km	25 m	100 m
Topology	Star	Star	Star	Star

Gigabit Ethernet

- use standard Ethernet frame format
- allows for point-to-point links and shared broadcast channels
- Compatible with lower speeds
- Uses standard framing and CSMA/CD algorithm
- Distances are severely limited
- Typically used for backbones and inter-router connectivity
- Becoming cost competitive
- Commonly used today: Gigabit switches!
  - Full-Duplex at 1 Gbps for point-to-point links
  - Frame structure preserved but CSMA-CD essentially abandoned
  - Extensive deployment in backbone of enterprise data networks and in server farms

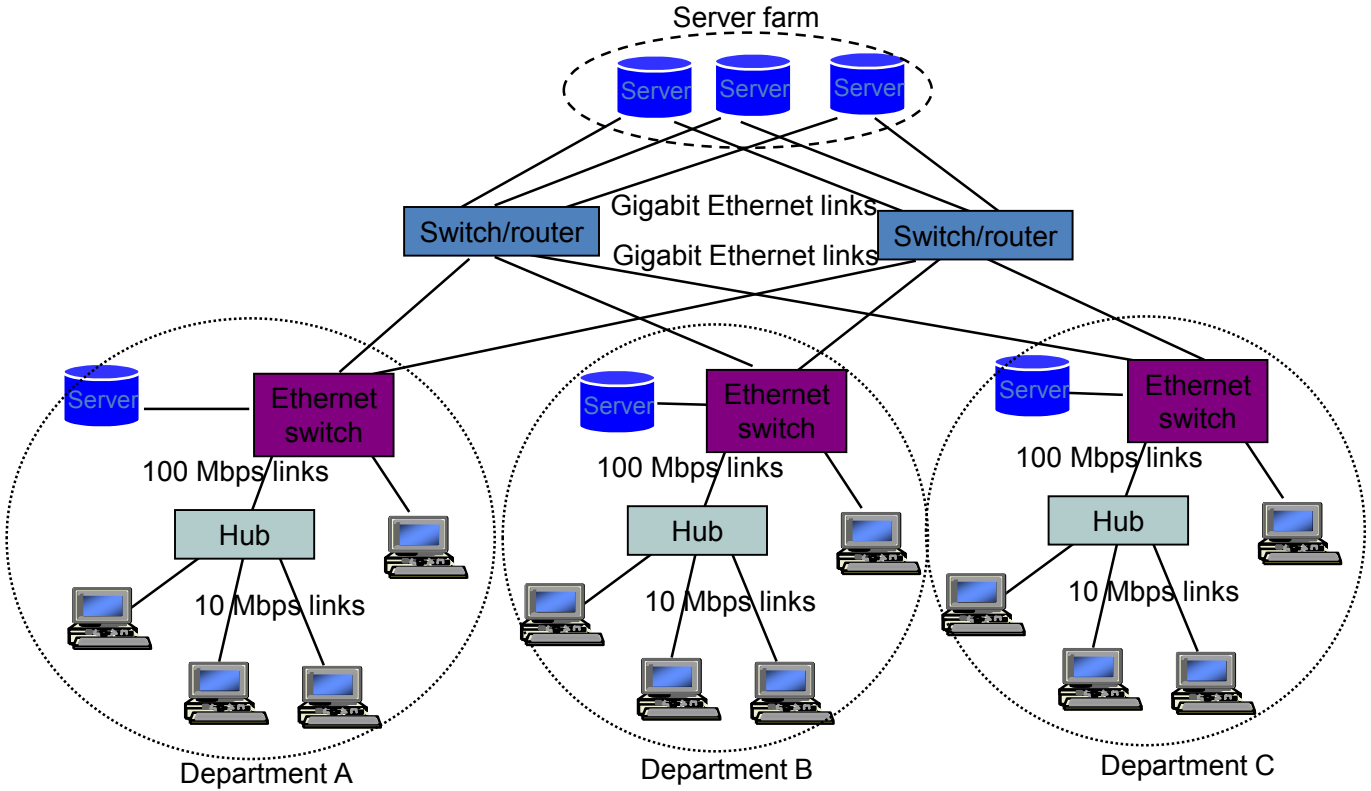
10 Gigabit Ethernet

IEEE 802.3 10 Gbps Ethernet medium alternatives

	10GbaseSR	10GBaseLR	10GbaseEW	10GbaseLX4
Medium	Two optical fibers Multimode at 850 nm 64B66B code	Two optical fibers Single-mode at 1310 nm 64B66B	Two optical fibers Single-mode at 1550 nm SONET compatibility	Two optical fibers multimode/single-mode with four wavelengths at 1310 nm band 8B10B code
Max. Segment Length	300 m	10 km	40 km	300 m – 10 km

- Frame structure preserved
- CSMA-CD protocol officially abandoned
- LAN PHY for local network applications
- WAN PHY for wide area interconnection using SONET OC-192c
- Extensive deployment in metro networks anticipated

Typical Ethernet Deployment



Experiences with Ethernet

- Ethernets work best under light loads
  - Utilization over 30% is considered heavy
    - Network capacity is wasted by collisions
- Most networks are limited to about 200 hosts
  - Specification allows for up to 1024
- Most networks are much shorter
  - 5 to 10 microsecond RTT
- Transport level flow control helps reduce load (number of back to back packets)
- Ethernet is inexpensive, fast and easy to administer!

Ethernet Problems

- Ethernet's peak utilization is pretty low
- Peak throughput worst with
  - More hosts
    - More collisions needed to identify single sender
  - Smaller packet sizes
    - More frequent arbitration
  - Longer links
    - Collisions take longer to observe, more wasted bandwidth
  - Efficiency is improved by avoiding these conditions