CS120: Computer Networks

Lecture 4. Framing and Error Detection

Zhice Yang

Quantifying Error

- Shannon capacity gives a throughput upper bound.
  - Actual throughput << channel capacity
    - => underutilized
  - Actual throughput >= channel capacity
    - => error
- Bit error rate (BER): error bits/transmitted bits
  - 10^{-1}, 10^{-2}, 10^{-3},…
- Trade off
  - High rate -> low reliability

Handling Error

- Error Detection
  - Parity Check
  - Checksum
  - Cyclic Redundancy Check (CRC)
- Error Correction
  - Hamming code
  - Others: convolutional code, fountain code, etc.

Parity Check

- Method: adding one extra bit to a 7-bit code to balance the number of 1s in the byte.

Even Parity

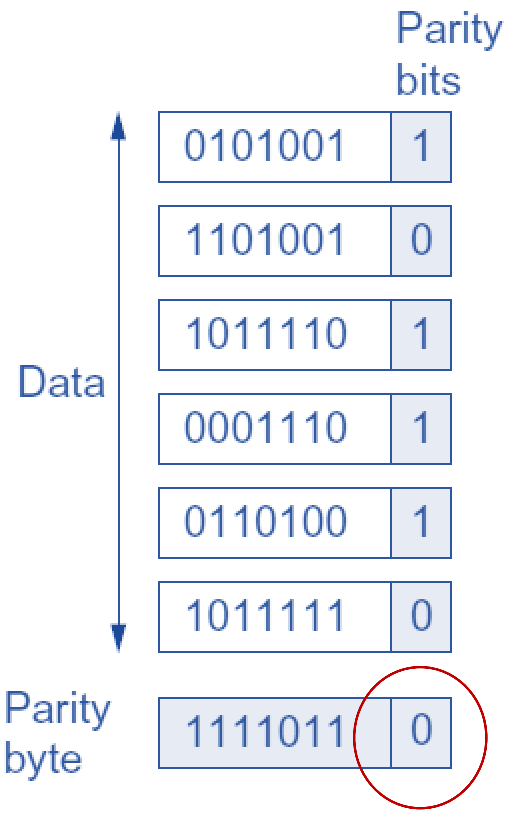
0101001	1
1101001	0

Odd Parity

0101001	0
1101001	1

2D Parity Check

- Add one byte to check the "columns"
  - 1-bit error
  - 2-bit error
  - 3-bit error
  - 4-bit error
    - not all



Checksum

- Method: add all the bytes up use ones’ complement arithmetic; then take ones’ complement of the result.

Ip header

```
0000  01 00 5e 00 00 fb 00 28  f8 57 76 c3 08 00 45 00   ..^....( .Wv...E.
0010  01 2b 19 0a 00 00 ff 11  71 a0 0a 14 45 08 e0 00   .+...... q...E...
0020  00 fb 14 e9 14 e9 01 17  21 b9 00 00 84 00 00 00   ........ !.......
0030  00 04 00 00 00 04 01 38  02 36 39 02 32 30 02 31   .......8 .69.20.1
0040  30 07 69 6e 2d 61 64 64  72 04 61 72 70 61 00 00   0.in-add r.arpa..
0050  0c 80 01 00 00 00 78 00  13 0b 54 49 41 4e 59 49   ......x. ..TIANYI
0060  47 53 2d 58 31 05 6c 6f  63 61 6c 00 01 43 01 36   GS-X1.lo cal..C.6
0070  01 46 01 41 01 32 01 33  01 38 01 39 01 45 01 42   .F.A.2.3 .8.9.E.B
```

Ones' Complement

Bits	Unsigned	Signed	Ones' Complement
1111	15	-1	-0
1110	14	-2	-1
1101	13	-3	-2
1100	12	-4	-3
1011	11	-5	-4
1010	10	-6	-5
1001	9	-7	-6
1000	8	-8	-7
0000	0	0	+0

Ones' Complement

- Calculation Examples
  - Signed -4 + (-2)
    - 1100+1110=11010
    - ignore carry
    - 1010 (i.e., -6)
  - Ones' Complement -4 + (-2)
    - 1101+1011=11000
    - shift and add carry
    - 1000+1=1001 (i.e., -6)

Checksum

- Method: add all the bytes up use ones' complement arithmetic; then take ones' complement of the result.

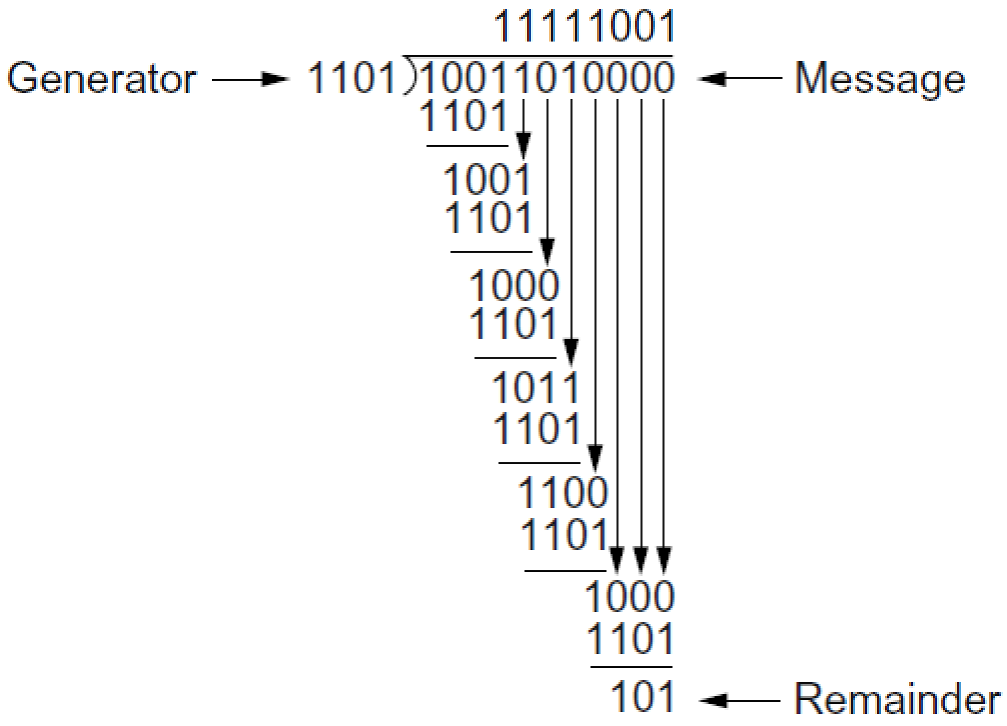
```
0x4500+0x012b+0x190a+0x0000+0xff11+0x0a14+0x4508+0xe000+0x00fb=0x28e5d
=>0x8e5f=>0x71a0
```

```
0000  01 00 5e 00 00 fb 00 28  f8 57 76 c3 08 00 45 00   ..^....( .Wv...E.
0010  01 2b 19 0a 00 00 ff 11  71 a0 0a 14 45 08 e0 00   .+...... q...E...
0020  00 fb 14 e9 14 e9 01 17  21 b9 00 00 84 00 00 00   ........ !.......
0030  00 04 00 00 00 04 01 38  02 36 39 02 32 30 02 31   .......8 .69.20.1
0040  30 07 69 6e 2d 61 64 64  72 04 61 72 70 61 00 00   0.in-add r.arpa..
0050  0c 80 01 00 00 00 78 00  13 0b 54 49 41 4e 59 49   ......x. ..TIANYI
0060  47 53 2d 58 31 05 6c 6f  63 61 6c 00 01 43 01 36   GS-X1.lo cal..C.6
0070  01 46 01 41 01 32 01 33  01 38 01 39 01 45 01 42   .F.A.2.3 .8.9.E.B
```

Checksum

- Method: add all the bytes up use ones' complement arithmetic; then take ones' complement of the result.
  - Fast calculation
  - Weak protection

Cyclic Redundancy Check (CRC)



CRC Performance

- CRC v.s. Checksum
  - CRC protects more bits
  - CRC takes more time/resource to calculate
- CRC v.s. Hash
  - CRC does not protect data integration
  - Hash functions are more complex than CRC
- CRC Polynomials
  - http://users.ece.cmu.edu/~koopman/crc/

Table 2.3 Common CRC Polynomials

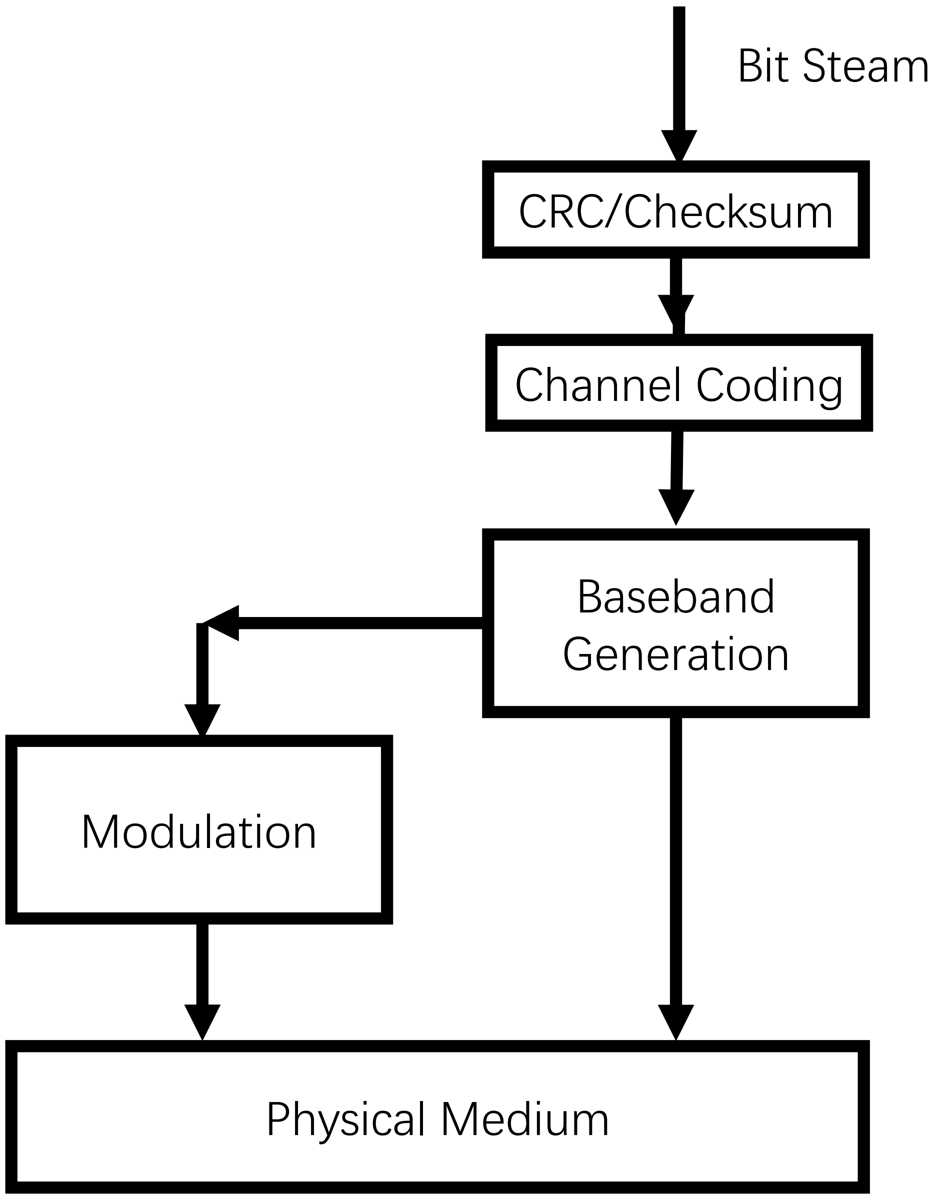
CRC	$C(x)$
CRC-8	$x^8+x^2+x^1+1$
CRC-10	$x^{10}+x^9+x^5+x^4+x^1+1$
CRC-12	$x^{12}+x^{11}+x^3+x^2+x+1$
CRC-16	$x^{16}+x^{15}+x^2+1$
CRC-CCITT	$x^{16}+x^{12}+x^5+1$
CRC-32	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}$ $+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

CRC Size (bits)	CRC Polynomial		HD	Hamming weights for number of bits corrupted:					
				1 bit	2 bits	3 bits	4 bits	5 bits	6 bits
16	CCITT-16	0x8810	4	0	0	0	**84**	**0**	**2 430**
16	[Baicheva00]	0xC86C	6	0	0	0	0	0	**2 191**
15	CAN	0x62CC	6	0	0	0	0	0	**4 314**
12	CRC-12	0xC07	4	0	0	0	**575**	**0**	**28809**
12		0x8F8	5	0	0	0	0	**1 452**	**13 258**
8	DARC-8	0x9C	2	0	**66**	**0**	**2 039**	**13 122**	**124 248**
8	CRC-8	0xEA	4	0	0	0	**2 984**	**0**	**253 084**
7	CRC-7	0x48	3	0	0	**216**	**2 690**	**27 051**	**226 856**
7		0x5B	4	0	0	0	**5 589**	**0**	**451 125**

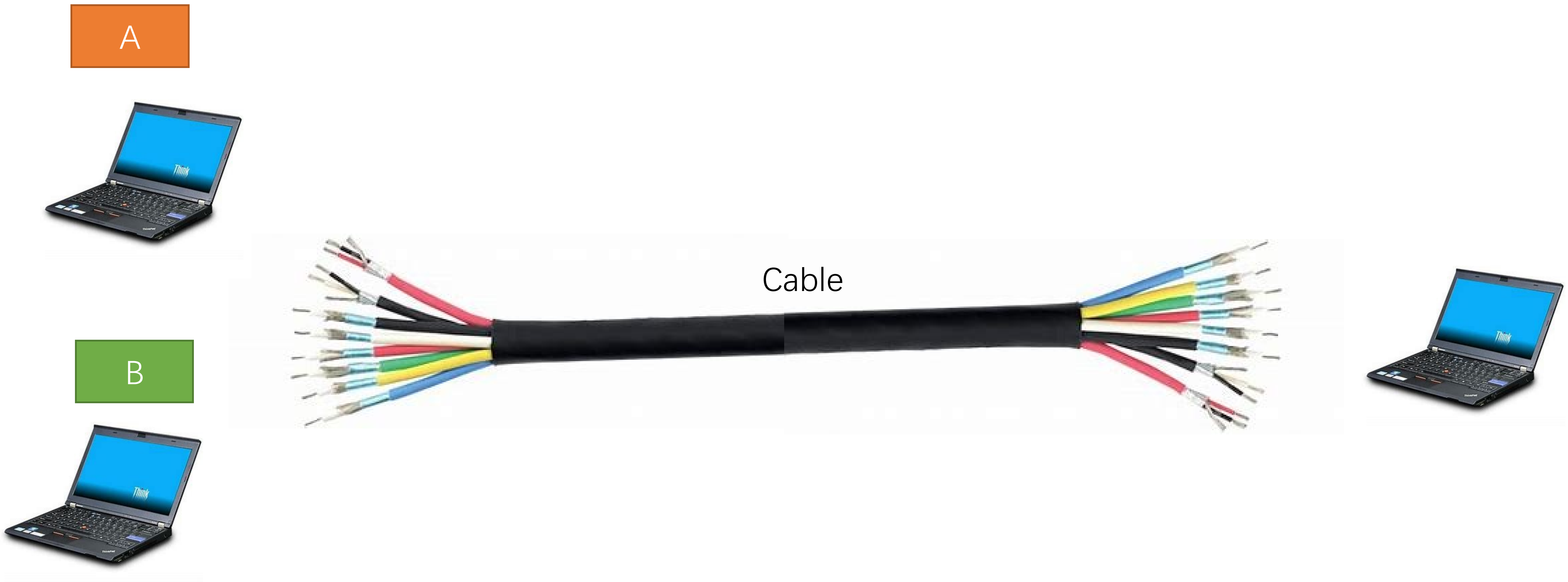
Handling Errors

- Error Detection
  - Parity Check
  - Checksum
  - Cyclic Redundancy Check (CRC)
- Error Correction
  - Hamming code
  - Others: convolutional code, fountain code, etc.
  - Trade off: efficiency and reliability
    - Retransmission is expensive
      - Storage, satellite, etc.
    - Errors are probable
      - Wi-Fi, cellular (channel is unstable, interference)

By Now

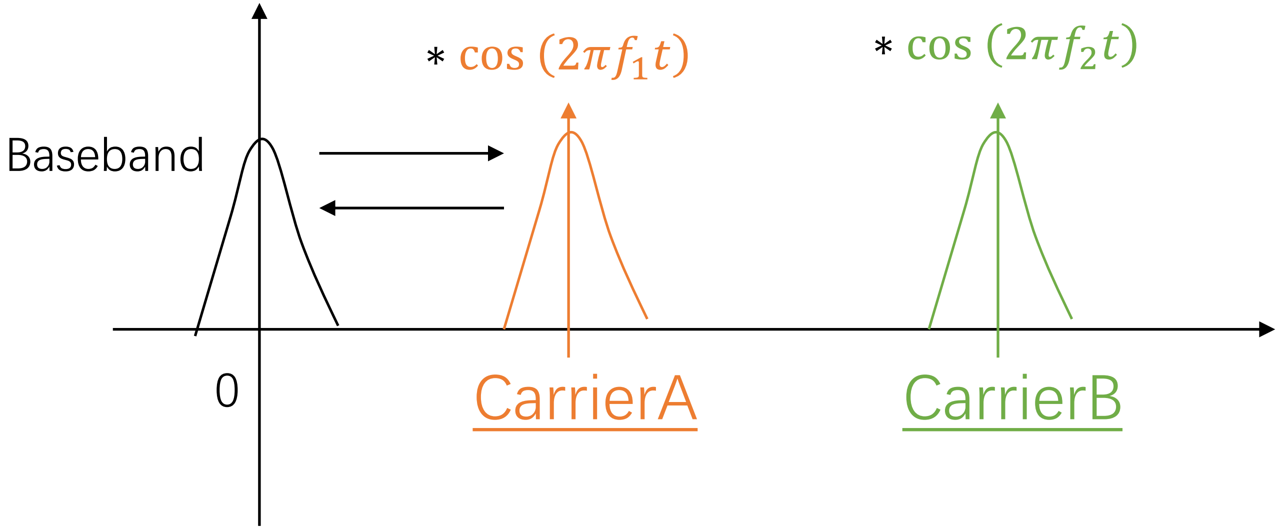

Bit Steam
CRC/Checksum
Channel Coding
Baseband Generation
Modulation
Physical Medium

The Multiplexing Problem



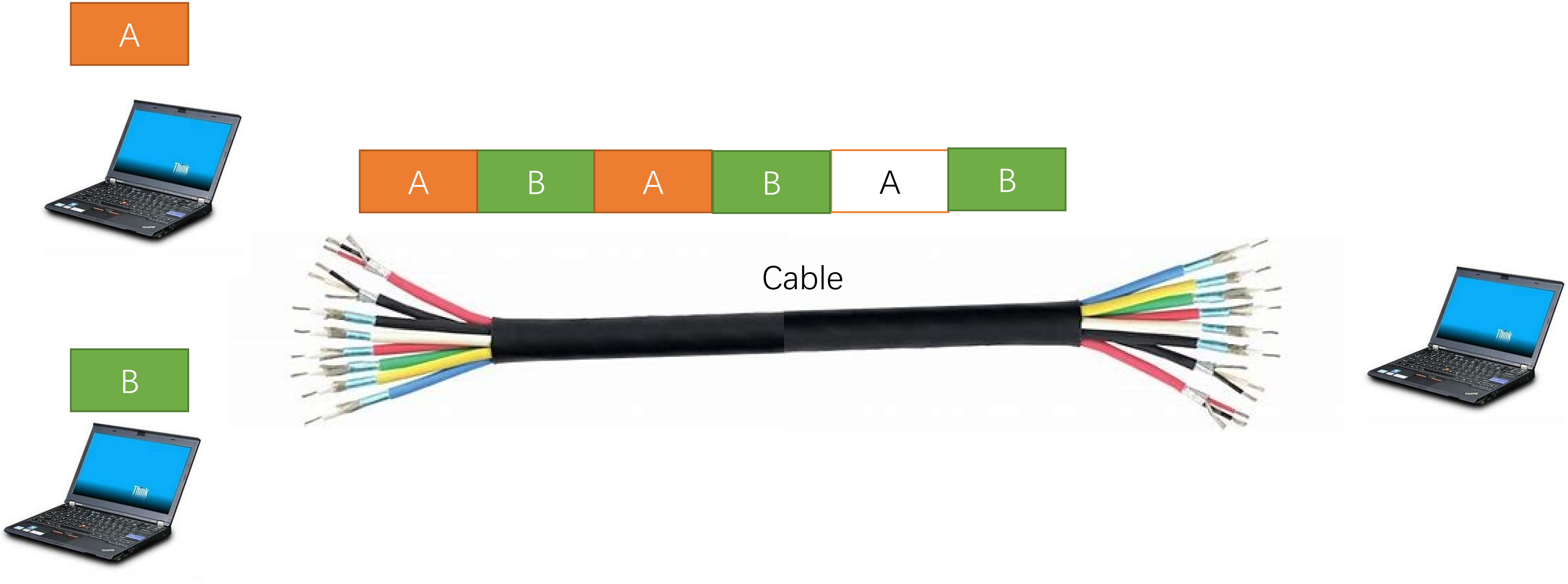
Multiplexing Approaches

- Frequency-division multiplexing (FDM)



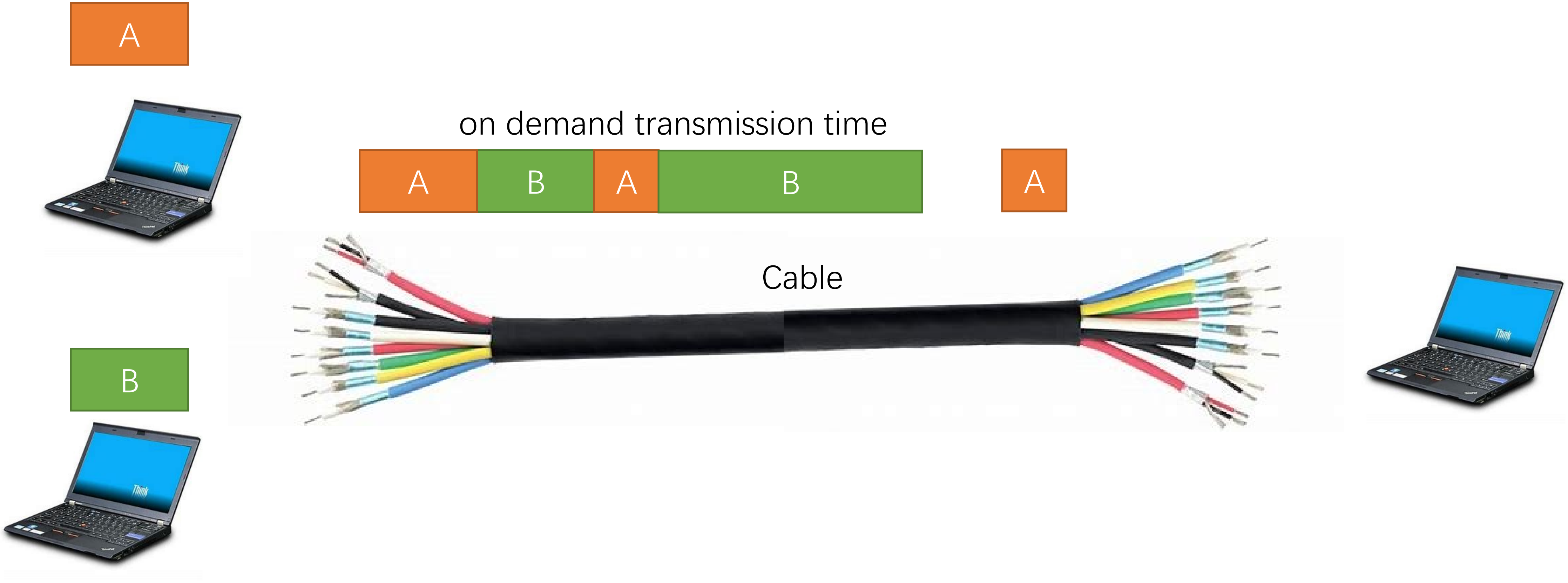
Multiplexing Approaches

- Synchronous Time-division Multiplexing (STDM)



Multiplexing Approaches

- Packet Switching



Multiplexing Approaches

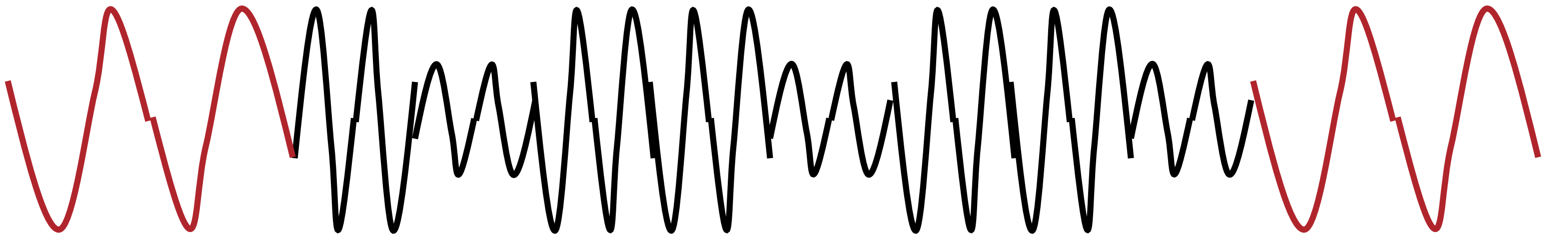
- Fixed Multiplexing
  - STDM, FDM (e.g. telecommunication network, WAN, etc.)
  - Predictable performance
  - Expensive
- Statistical Multiplexing
  - Packet Switching (e.g. Internet, etc.)
  - Performance is random
  - Cheap

Framing

- Why ?
  - One transmitter should not occupy the communication resource forever
- Functionality
  - Help receiver recognize the start and the end of the transmission
- Framing Design
  - Frame detection
  - Frame termination

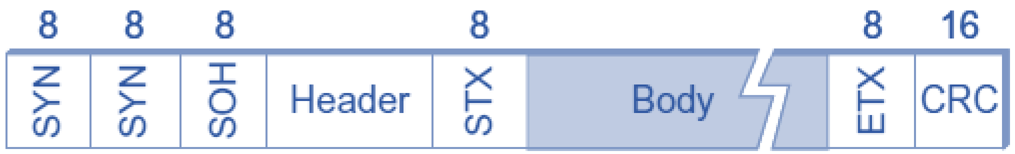
Framing - Detection

- Sync signal: find accurate start through special pattern (e.g., Preamble, Pilot, etc.)
  - Add a special pattern before or after each frame
  - Correlate the special pattern …



Framing - Termination

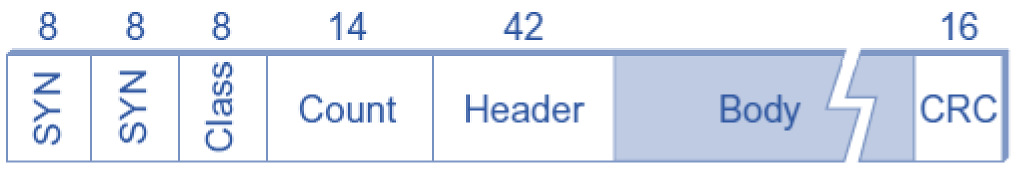
- Sentinel-Based Approach
  - Use special patterns (sentinel characters) to indicate the start and end of the frame
  - Similar patterns may exist in payload
    - Solution: character stuffing
    - e.g., `disp ('abc''')` in matlab shows `abc'`; `'` is the escape character



Example: BISYNC Frame

Framing - Termination

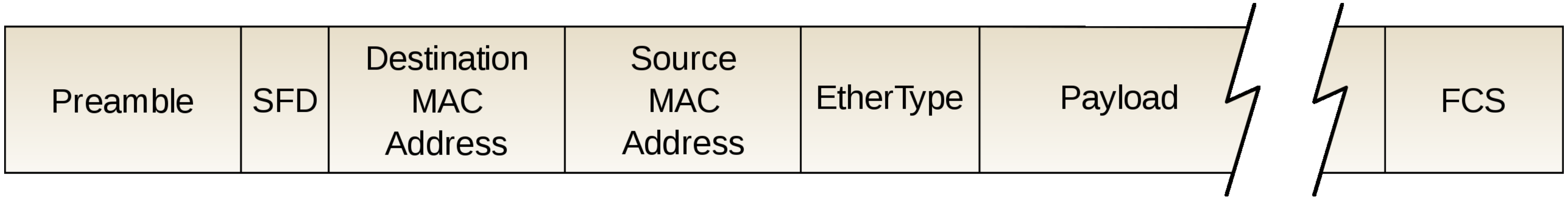
- Counting-based Approach
  - Use special patterns to indicate the start of the frame
  - Use number to indicate the length of the frame
    - Count may contain error
      - Solution: limit the maximum frame length



Example: DDCMP Frame

Examples

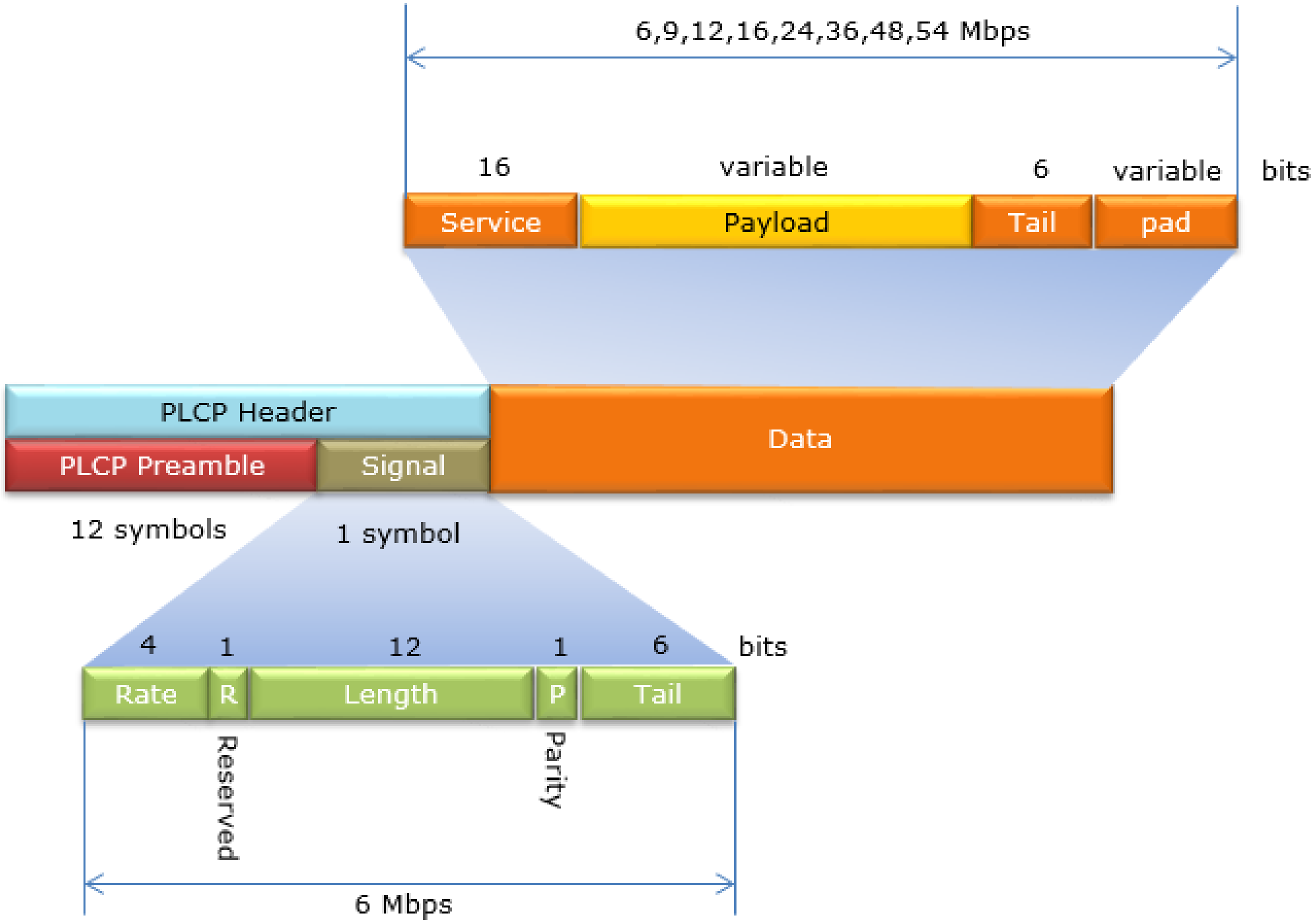
- Ethernet



source: https://en.wikipedia.org/wiki/Ethernet_frame#/media/File:Ethernet_frame.svg

Examples

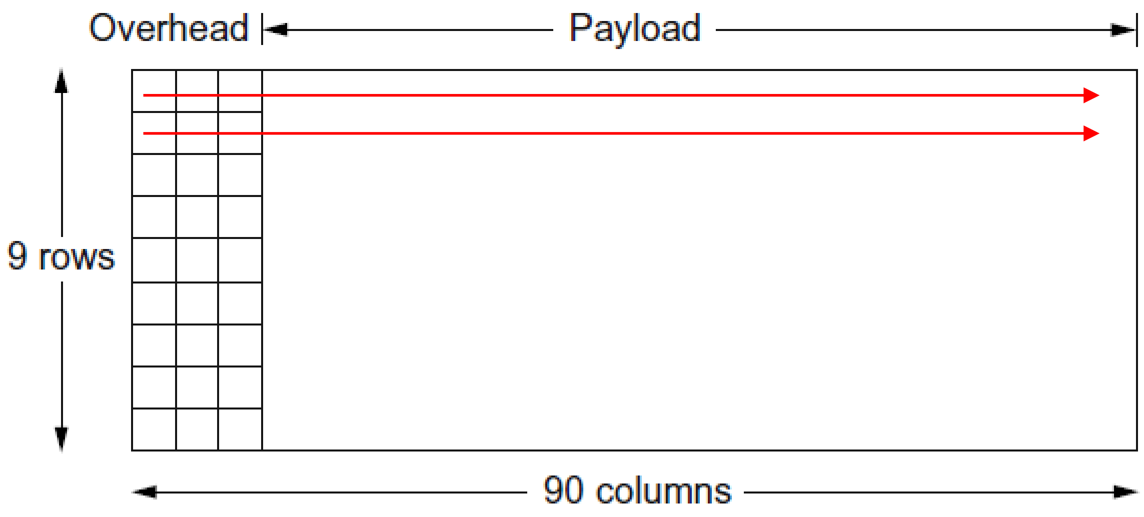
- Wi-Fi



source: https://www.sharetechnote.com/html/WLAN_PHY_Frames.html

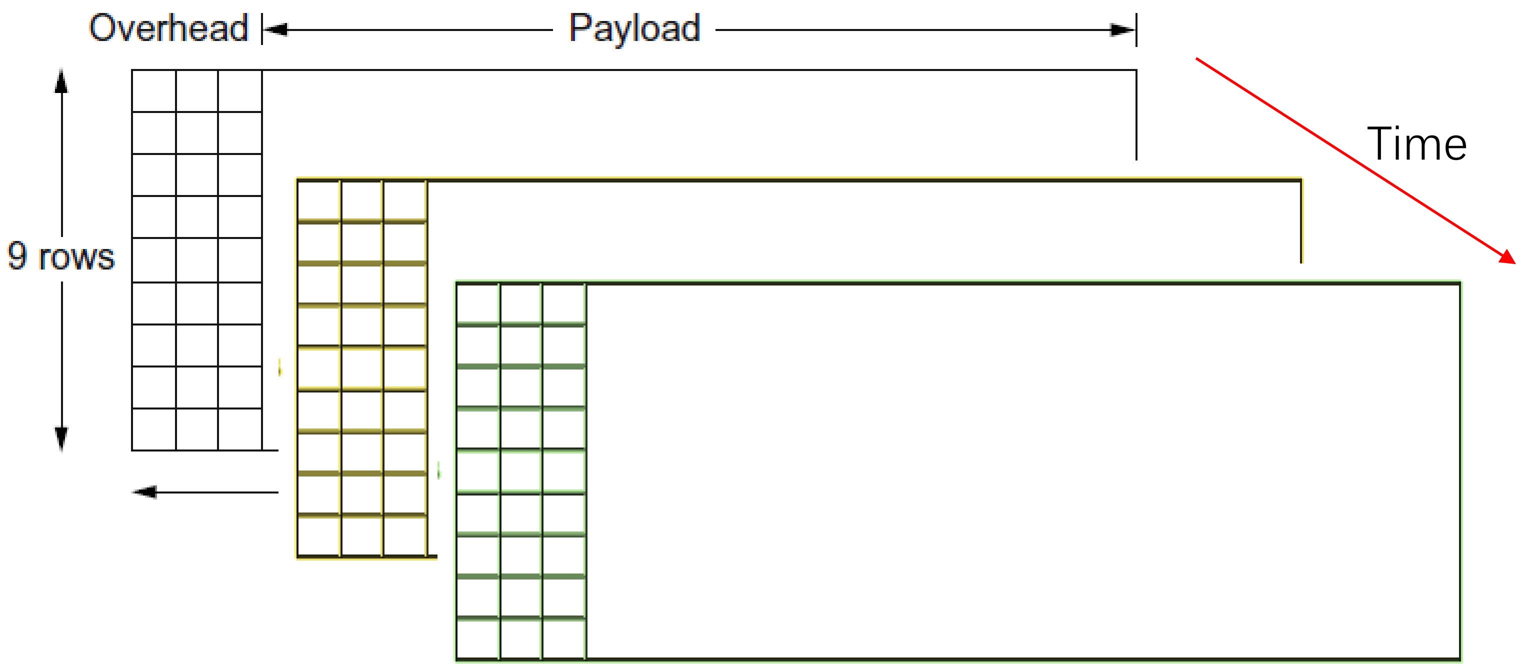
Examples

- Synchronous Optical Network (SONET)/Synchronous Digital Hierarchy (SDH)



Examples

- Synchronous Optical Network (SONET)/Synchronous Digital Hierarchy (SDH)



Examples

- 4G LTE FDD

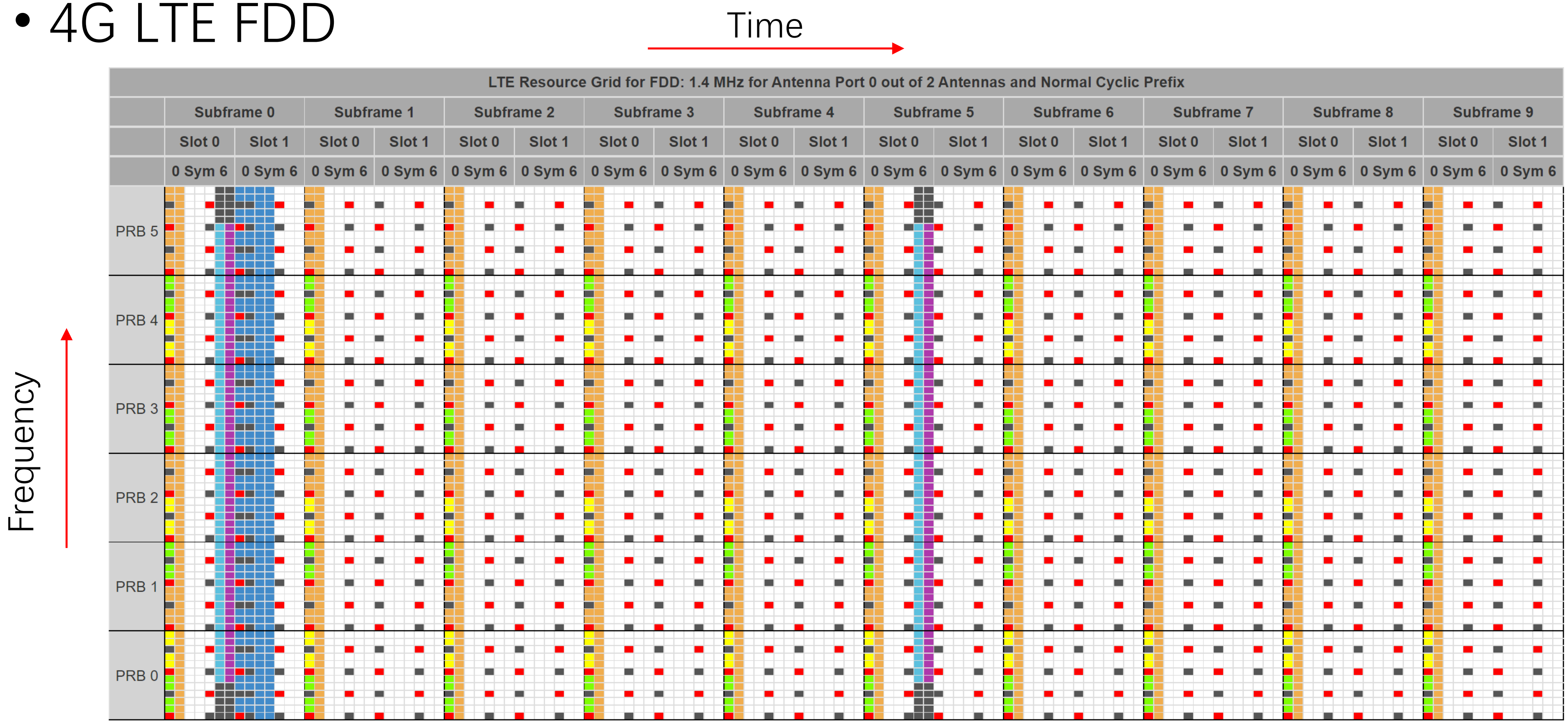
Time

LTE Resource Grid for FDD: 1.4 MHz for Antenna Port 0 out of 2 Antennas and Normal Cyclic Prefix																				
	Subframe 0		Subframe 1		Subframe 2		Subframe 3		Subframe 4		Subframe 5		Subframe 6		Subframe 7		Subframe 8		Subframe 9	
	Slot 0	Slot 1	Slot 0	Slot 1	Slot 0	Slot 1	Slot 0	Slot 1	Slot 0	Slot 1	Slot 0	Slot 1	Slot 0	Slot 1	Slot 0	Slot 1	Slot 0	Slot 1	Slot 0	Slot 1
	0 Sym 6	0 Sym 6	0 Sym 6	0 Sym 6	0 Sym 6	0 Sym 6	0 Sym 6	0 Sym 6	0 Sym 6	0 Sym 6	0 Sym 6	0 Sym 6	0 Sym 6	0 Sym 6	0 Sym 6	0 Sym 6	0 Sym 6	0 Sym 6	0 Sym 6	0 Sym 6

source: https://dhagle.in/LTE.php

Examples

- 4G LTE FDD



source: https://dhagle.in/LTE.php

Reference

- Textbook 1.2.3
- Textbook 2.3
- Textbook 2.4