# Computer Networks and Communication

**Lecture 10**

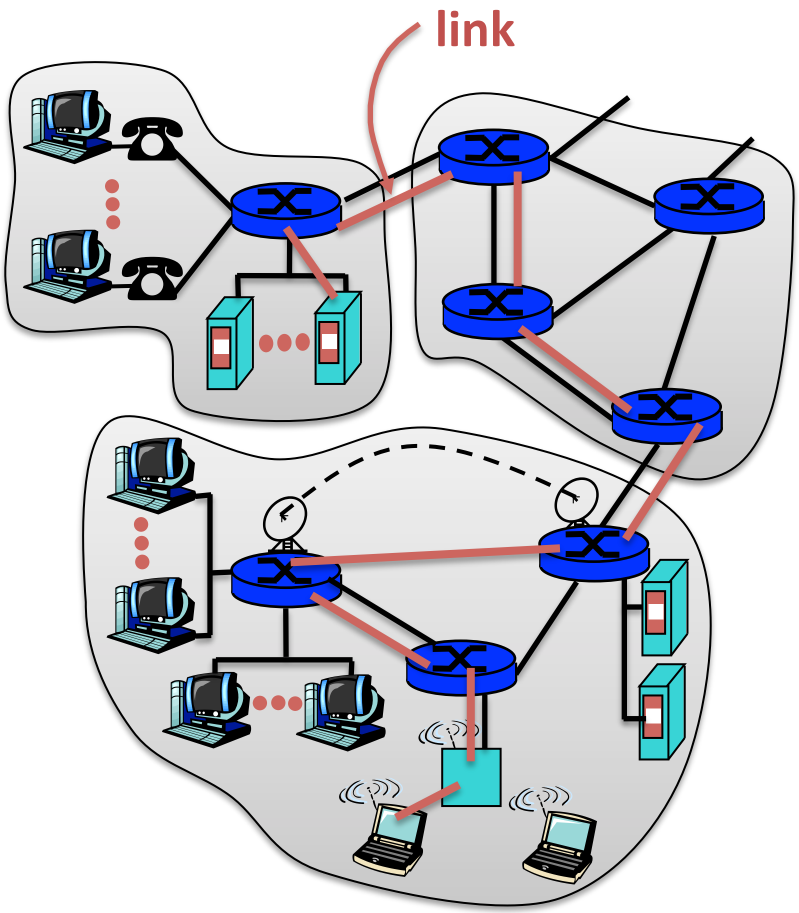
Link Layer

# Link Layer

- In this course, we will learn the following link layer services:
  - Error detection and correction
  - Multiple access: sharing a broadcast channel
  - Link-layer addressing
  - Flow control
  - Reliable data transfer:
    - Acknowledgement and retransmission
    - Essential to some network links (e.g. WLAN, UMTS) are prone to error
  - A number of link layer technologies and protocols

# Link Layer (2)

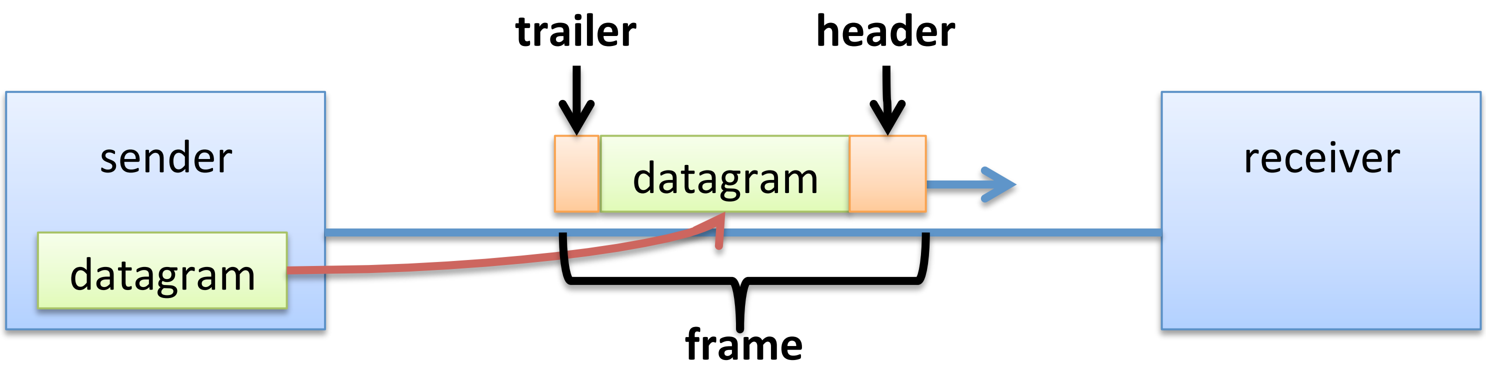
- **Nodes**: Hosts and routers
- **Links**: Communication channels that connect adjacent nodes along communication path
  - Wired links (LANs)
  - Wireless links (WLANs)
- **Frame**: Layer-2 packet that encapsulates datagram
- **Data-link layer** is responsible for transferring frame from one node to **adjacent** node over a link



# Link Layer: Concept

- Datagrams are transferred by different link-layer protocols over different links
- Transportation analogy: A tourist books a trip from Rayong to Venice
  - First link: A **bus** from Rayong to Suvarnabhumi airport
  - Second link: A **plane** from Suvarnabhumi to Fiumicio airport
  - Third link: A **train** from Fiumicio to Venice
  - Tourist = datagram
  - Travel agent = routing algorithm
- **Remember**: link layer concerns only transferring data within a single link

# Network Adaptors Communication



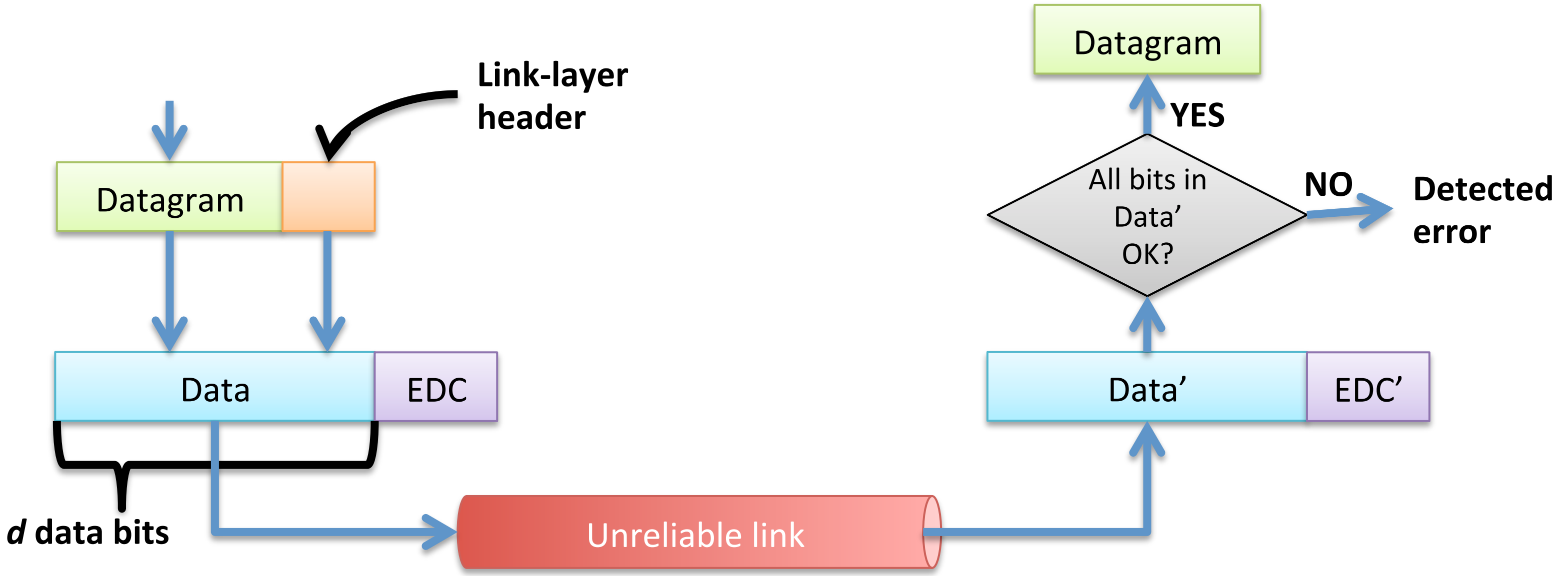

- Link layer software and hardware are implemented **network adaptor** or **network interface card (NIC)**
  - Ethernet card
  - WiFi card
  - Bluetooth adaptor
- Datagram from network layer are encapsulated in a **frame**
  - **Header** is added in the **front** of the datagram
  - Some protocols also add a **trailer** at the **end** of the datagram

# Error Detection and Correction

- Since a physical links are not perfect, the link layer has to provide error detection (and sometimes correction) mechanisms
- Error detection techniques can be categorized into three groups:
  - **Parity checking**
  - **Checksum**
  - **Cyclic Redundancy Check (CRC)**

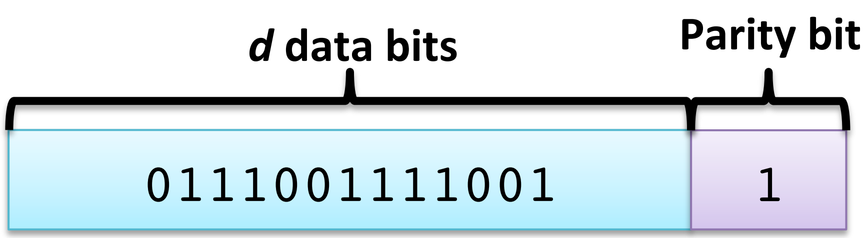
# Error Detection and Correction (2)

- Generally, error detection and correction are done by:
  - Adding **Error-Detection and –Correction bits (EDC)** along with the original data
  - The longer EDC, the better detection and correction
  - Nevertheless, error detection is not 100% reliable



# Parity Checking

- Simplest form of error detection
- Use a **parity bit** as EDC bit
  - **Even parity scheme**: total number of 1s in data and the parity bits is **even**
  - **Odd parity scheme**: total number of 1s in data and the parity bits is **odd**

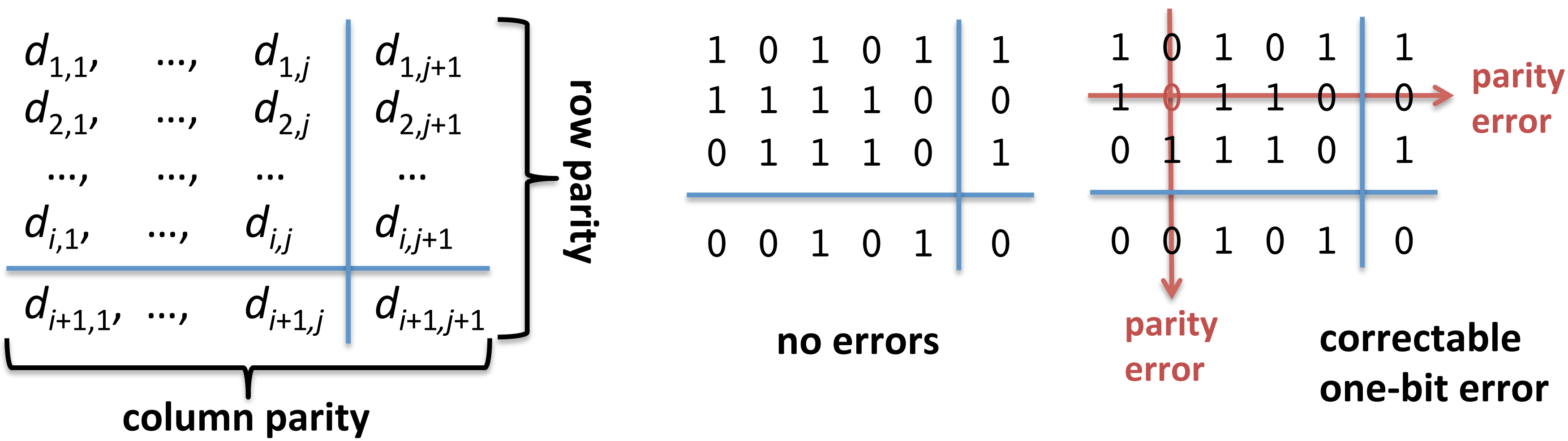



**One-bit even parity**

Do you think one-bit parity scheme is safe?

# 2D Parity Checking

- Enhanced variation of parity checking
  - Detect and correct one-bit error at **receiver side**
  - Can detect (but not correct) two-bit error (Why?)
- Ability of detecting and correcting the error at the receiver side without retransmission is called **Forward Error Correction (FEC)**



# Internet Checksum

- Detecting errors (flipped bits) in transmitted segment
- It is used at transport layer only
  - Because it is faster than CRC check
- Sender computes checksum sequence and put it in the protocol header
- Receiver computes the checksum of the received segment and verify with the checksum value in the header
- We have discussed this already few weeks ago

# Cyclic Redundancy Check

- **Cyclic Redundancy Check (CRC)**
- Most popular error detection
  - Sometimes called **CRC check**
  - It is also called **Polynomial** code because the coded bits can be represented by polynomial function

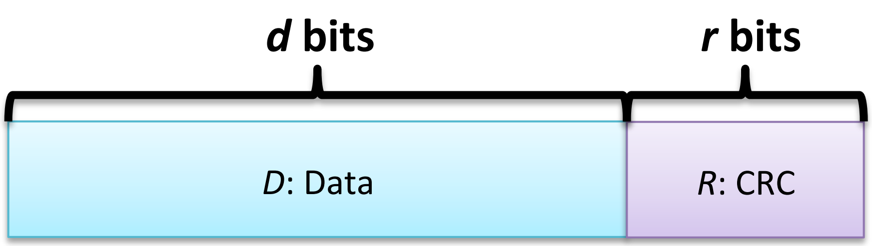
$$G(X) = X^n + X^{n-1} + \ldots + X^2 + X^1 + X^0$$

  - Example:

$$G(X) = X^3 + X^0 = 1001$$

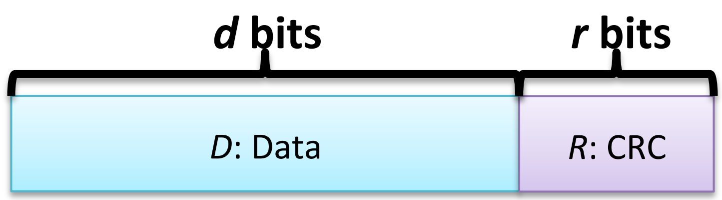
# CRC Computation

- The sender wants to send the data ***D*** of length ***d***
- The CRC code ***R*** of length ***r*** must be generated and append to the data
  - This code will be used to check if the data is correct (similar to parity bit)
- The CRC code *R* is computed using a generator ***G***
  - *G* is a sequence of predefined, static binary patterns
  - *G* has **r+1** bits
  - Most significance bit (MSB) of *G* must be 1
  - Both sender and receiver knows *G* before the communication
  - Receiver uses *G* to verify the data



# CRC Computation (2)

- CRC key concepts
  - We want a sequence of bits DR such that it is exactly divisible by G




  - The division is done using modulo-2 arithmetic
  - The sender uses D and G to compute R which yields such sequence
  - Receiver divides the sequence by G
  - If there is no remainder, then the data is correct

# CRC Computation (3)

- For the sender, the main computation is to compute *R*

$$R = \text{remainder of } \frac{D * 2^r}{G}$$

- $D * 2^r$ means appending *r* bits of 0 to *D*

Ex: $D$ = 101110 and
$r$ = 3, we got
$D * 2^r$ = 101110000

- Now, as we have $D * 2^r$, let's see how division is done
- The modulo-arithmetic used in CRC requires XOR division
  - Addition or subtraction without carries or borrow

# Dividing Binary

*D* = 101110
*G* = 1001
*r* = 3
Find *R*

```
                          1 0 1 0 1 1
              ┌─────────────────────
1 0 0 1  XOR  │1 0 1 1 1 0 0 0 0
               1 0 0 1
               ───────
          XOR    1 0 1
                 0 0 0
                 ─────
          XOR    1 0 1 0
                 1 0 0 1
                 ───────
          XOR      1 1 0
                   0 0 0
                   ─────
          XOR      1 1 0 0
                   1 0 0 1
                   ───────
          XOR        1 0 1 0
                     1 0 0 1
                     ───────
                       0 1 1  ← R
```

Final Sequence: 101110 011

**XOR Truth Table**

| | A=0 | A=1 |
|---|---|---|
| B=0 | 0 | 1 |
| B=1 | 1 | 0 |

# Receiver Test

- Dividing 101110011 with G
  - G = 1001
  - If the remainder is zero, then the data are correct
- Anyone want to try?

# CRC Considerations

- CRC is a very popular error detection method
- The main issue is *G* must be known by both sender and receiver
- There are standards for *G*s with different sizes
  - They are used for different purposes
  - 16 bits: 10000000000000101
  - 32 bits: 100000100110000010001110110110111

# CRC Considerations (2)

- CRC can detect
  - All single-bit errors
  - All double-bit errors
  - All odd number of errors
  - All bursts error of length up to *r* bits
    - CRC-32 can detect up to 32-bit errors
- Burst errors larger than *r* can be detected with probability $1\text{-}0.5^r$

# Multiple Access Links

- Point-to-point links:
  - One pair of communication parties per link
  - Example: PPP for dial-up access,
    tin can telephone

- Broadcast link
  - **Multiple access**: Shared wired or wireless medium
  - Example:
    - Walkie-Talkie
    - Traditional Ethernet
    - 802.11 Wireless LAN
    - 2G/3G/4G mobile networks

  - **Multiple access problem**: How to coordinate multiple use shared broadcast channel

# Multiple Access Protocol

- Single broadcast channel shared by many nodes
- If more than one transmit (broadcast) frames at the same time, a **collision** occur
  - The receivers receive the collided frames
  - The frame involved in the collision are lost and the broadcast channel is wasted during the collision
- This is a very serious issue and a lot of multiple access protocols have been proposed to solve it
- These protocols are distributed algorithms that determine how node share channel (i.e., when the nodes can transmit)

# Multiple Access Analogy

- In the real world, we always use multiple access protocols, e.g. in a class room
  - Students have to raise hands before the can speak
  - Teacher coordinates the medium access
  - You should not keep talking alone. Let others take turns to talk too
  - In a group discussion, you might not need teacher or any coordinator, just follow the protocol

# Ideal Multiple Access Protocol

- If the broadcast channel has transmission rate of *R* bps
  - If **one** node wants to transmit, it can send the data at rate ***R***
  - If ***M*** nodes want to transmit at the same time, each node can send at rate ***R/M***
  - Fully decentralized
    - No central node that coordinate transmission

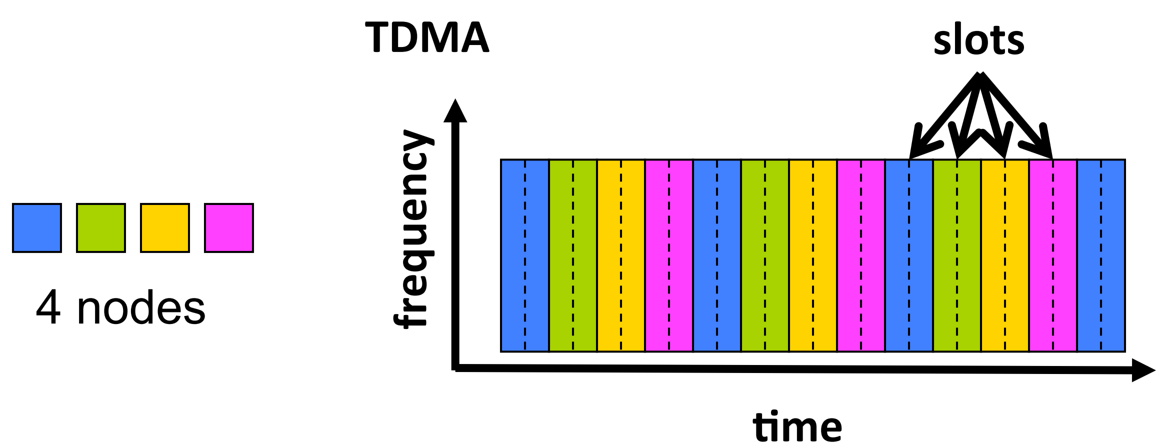
      e.g., Bluetooth uses a central node
    - No synchronization among nodes
  - Simple

# Medium Access Protocol Categories

- Medium access protocols can be divided into:
  - Channel partitioning protocols
    - Divide a channel into smaller pieces (e.g., TDMA, FDMA)
    - Allocate each piece to a node
    - No collision
  - Random access protocols
    - Channel is not divided, collision is allowed
    - Sending nodes **recover** from collision
  - Taking-turns protocols
    - Nodes take turns to send
    - Nodes with more data to send might have to take longer turns

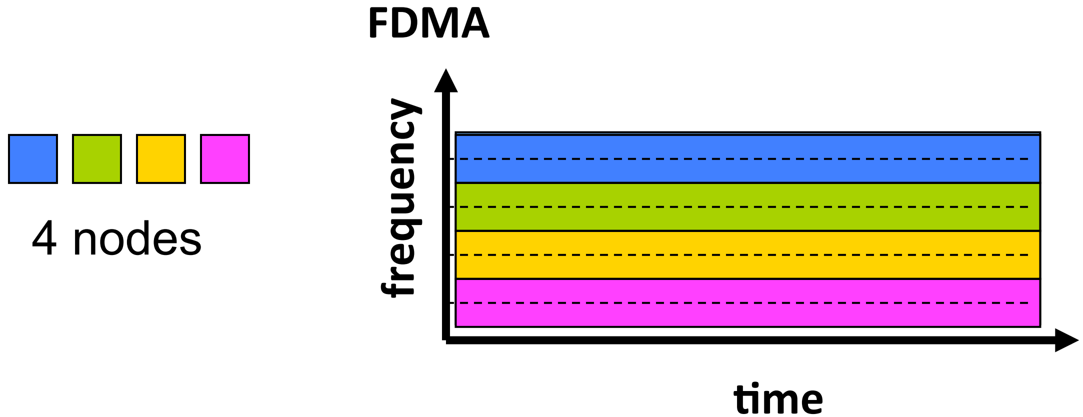
# Channel Partitioning: TDMA

- **TDMA: Time Division Multiple Access**
  - Access channels in rounds
  - Each node is assigned with a fixed-length slot
  - Unused slots are wasted
  - No collision



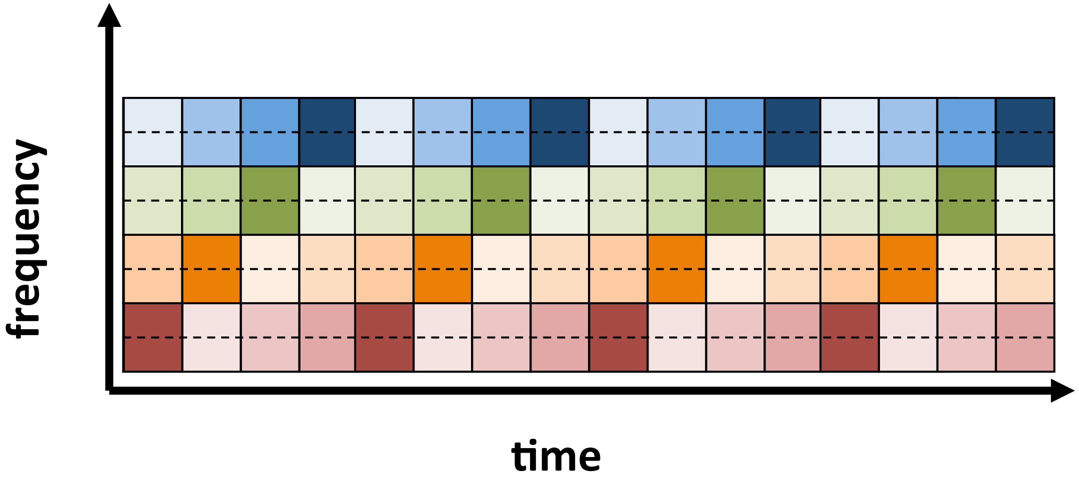
# Channel Partitioning: FDMA

- **FDMA: Frequency Division Multiple Access**
  - Channel spectrum is divided into frequency bands
  - Each node is assigned to a fixed band
  - Unused frequency band is wasted
  - No collision



# TDM and FDM

- Some network technologies, e.g., GSM networks, use both TDM and FDM together
  - The frequency bands are divided into time slots
  - The frequency bands regulation is a national issue



# Channel Partitioning: CDMA

- **CDMA: Code Division Multiple Access**
  - Each node is assigned with different unique **code**
  - It uses the assigned code to encode the data bit to send
  - In CDMA, different nodes can send data simultaneously
  - Receivers can distinguish their data if they know the senders' codes
  - Widely used in recent mobile networks (e.g. 3G)

# Random Access Protocols

- When a node has packet to send
  - transmit at full channel data rate R.
  - no a priori coordination among nodes
- If two or more nodes transmit at the same time, **collision** occurs
- Random access protocol specifies:
  - How to detect collisions
  - How to recover from collisions (e.g., via delayed retransmissions)
- Examples of random access protocols:
  - Slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

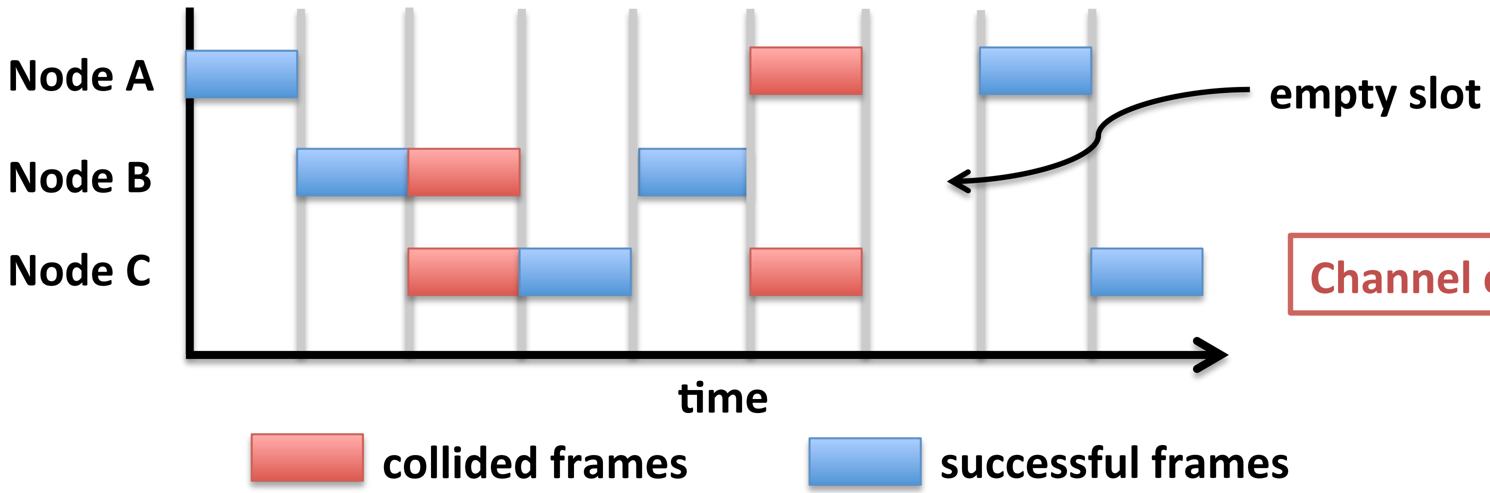
# Slotted ALOHA

- Improved variant of **ALOHA protocol**
- Setting:
  - Like TDMA, channel time is divided into slots
  - Each slot is equal to the time to transmit one frame
  - Node can send frames only at the beginning of slots
  - Nodes are synchronized: Determine when the slot begin
  - If two or mode nodes transmit in the same slot, all node detect collision

# Slotted ALOHA (2)

- Sending operation:
  - When a node has a frame to send, it transmits in next slot
  - If there is no collision, the node can send new frame in next slot
  - If a collision occurs, node tries to retransmit the failed frame in each subsequent slot with probability *p* until success
  - That is, the node decides at the next slot, whether it should resend the frame with probability *p*

# Slotted ALOHA (3)




Channel efficiency: 37%

**Advantage**

- Single active node transmit at full channel capacity
- Decentralized: Although require slots to be synchronized
- Simple

**Disadvantage**

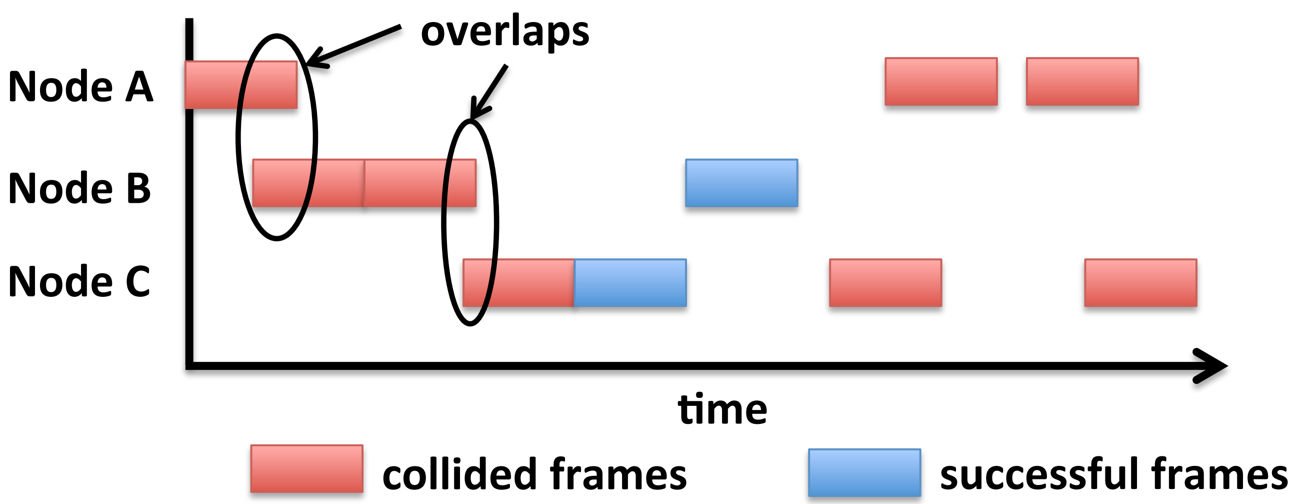
- Collision slots are wasted
- Slots could also be wasted after collision
- Require synchronization

# Pure ALOHA

- Developed in 1968 at the University of Hawaii
  - To be used in ALOHAnet
  - Aimed to connect several radio nodes scattered over Hawaiian Islands
  - It was the first radio packet network
- Sending operation:
  - If a node has a frame to send, send the frame
  - If the frames collide, back off and send later with probability *p* (similar to Slotted ALOHA)
  - Unlike Slotted ALOHA, no slot synchronization is done

# Pure ALOHA (2)

- Pure ALOHA requires no synchronization
  - Simpler than slotted variance
  - Purely decentralized
- However, collision probability is worse than Slotted ALOHA
  - Caused by overlapping frames
  - Efficiency is only 50% compared to slotted ALOHA



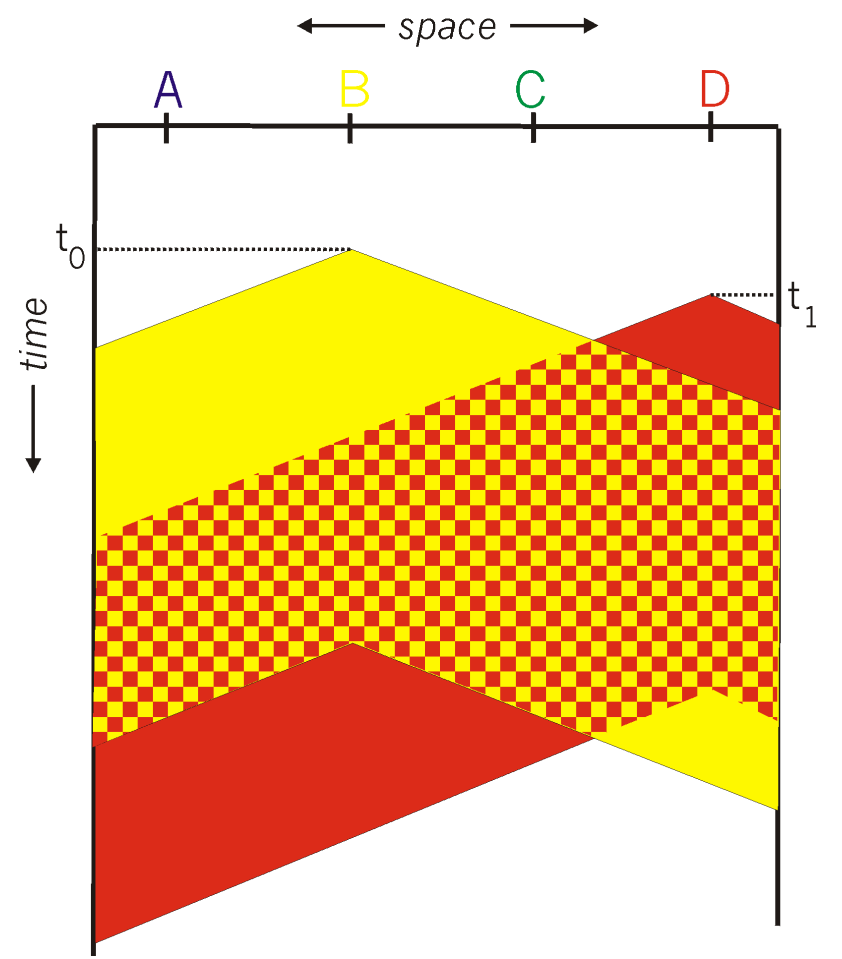

**Channel efficiency: 18.5%**

# Carrier Sense Multiple Access

- **CSMA: Carrier Sense Multiple Access**
- Both pure and slotted ALOHA transmit frames without checking if another node is sending
  – Why waste slots like that?
- Channel efficiency can be improved by
  – Listen to (**sense**) the **carrier** before transmitting
  – If collision is detected, stop sending
- Human analogy: Do not interrupt others when they are talking

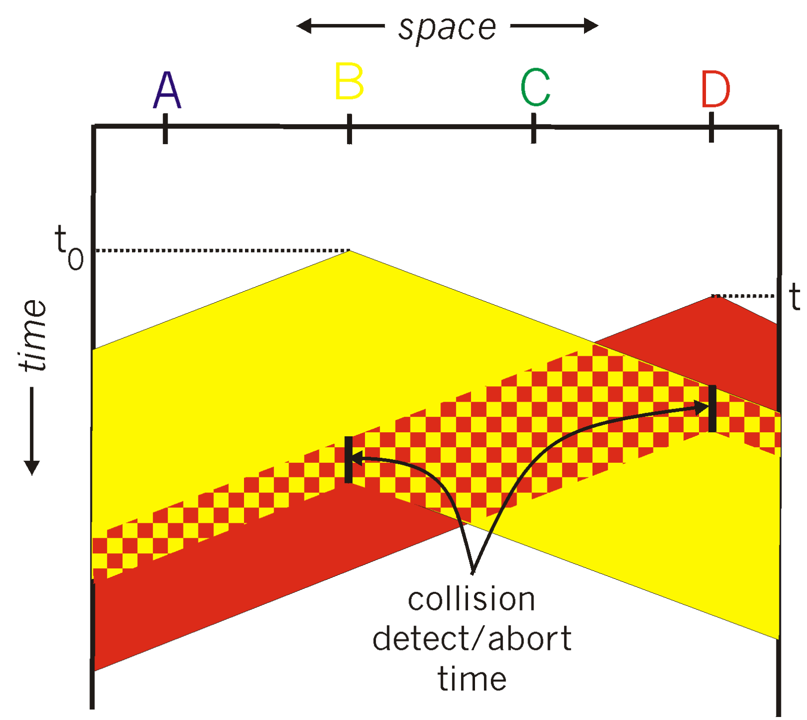
# CSMA Collision

- Even with carrier-sensing, collision can still occur
  - Due to propagation delay
  - The transmitting frame has not arrived other node who also has a frame to send
- Collision causes the entire packet transmission to to be wasted
- Collision probability depends on distance and propagation delay



# CSMA with Collision Detection

- **CSMA/CD: CSMA with Collision Detection**
- Detect collision within a very short time
- Abort transmission when a collision is detected to reduce channel waste



# Taking-Turns Protocols

- Channel partitioning protocols:
  - Efficient at high load: Share the channel fairly
  - Inefficient at low load: Time slots or frequency band are wasted
- Random access protocols:
  - Efficient at low load: Single node can fully utilize the channel
  - Inefficient at high load: Collision overhead
- Taking-turns protocols:
  - Find the best of both worlds

# Taking-Turns Protocols Categories

- **Polling protocols**
  - A **master** node polls each **slave** node to transmit data in round-robin fashion
  - **Advantage**: No collision
  - **Disadvantage**: Polling causes overhead and if master node fails, the entire network fails too
- **Token passing protocols**
  - Special token is passed from one node to the next
  - A node can send the data only if it has the token
  - **Advantage**: No collision and no master node
  - **Disadvantage**: We have to be sure that the token is passed throughout the nodes

# Media Access Protocol: Summary

- We have discussed
  - Channel partitioning by time, frequency and code
  - Random partitioning
    - ALOHA, Slotted ALOHA
    - CSMA
    - CSMA/CD: Ethernet
  - Taking turns
    - Polling protocols: Bluetooth
    - Token passing: Fiber Distributed Data Interface (FDDI)
- We will discuss:
  - Link-layer addressing
  - Wired and wireless network technologies and protocols