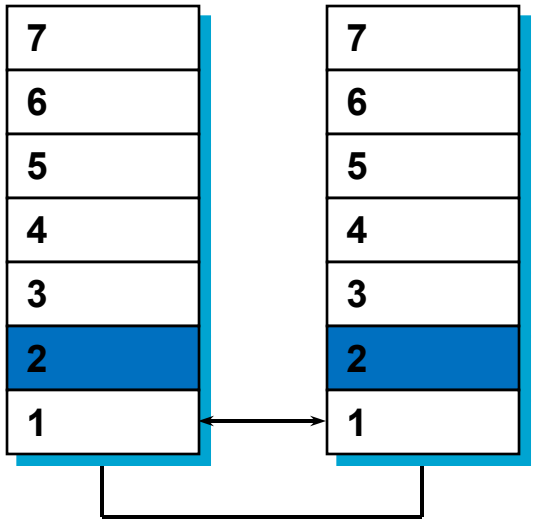
FREIE UNIVERSITÄT BERLIN

Operating Systems & Computer Networks
10. Network Access II – Data Link Layer

Dr. Larissa Groth
Computer Systems & Telematics (CST)



Roadmap

8. Networked Computer & Internet
9. Network Access Layer I – Physical Layer
10. **Network Access Layer II – Data Link Layer**
11. Internet Layer – Network Layer
12. Transport Layer
13. Applications

Lernziele I

- Sie nennen:
  - die wesentlichen Aufgaben dieser Schicht und die Ein- und Ausgabe über die Service Access Points
  - die wesentlichen Aspekte der Error Control
  - wesentliche Netzwerk-Topologien

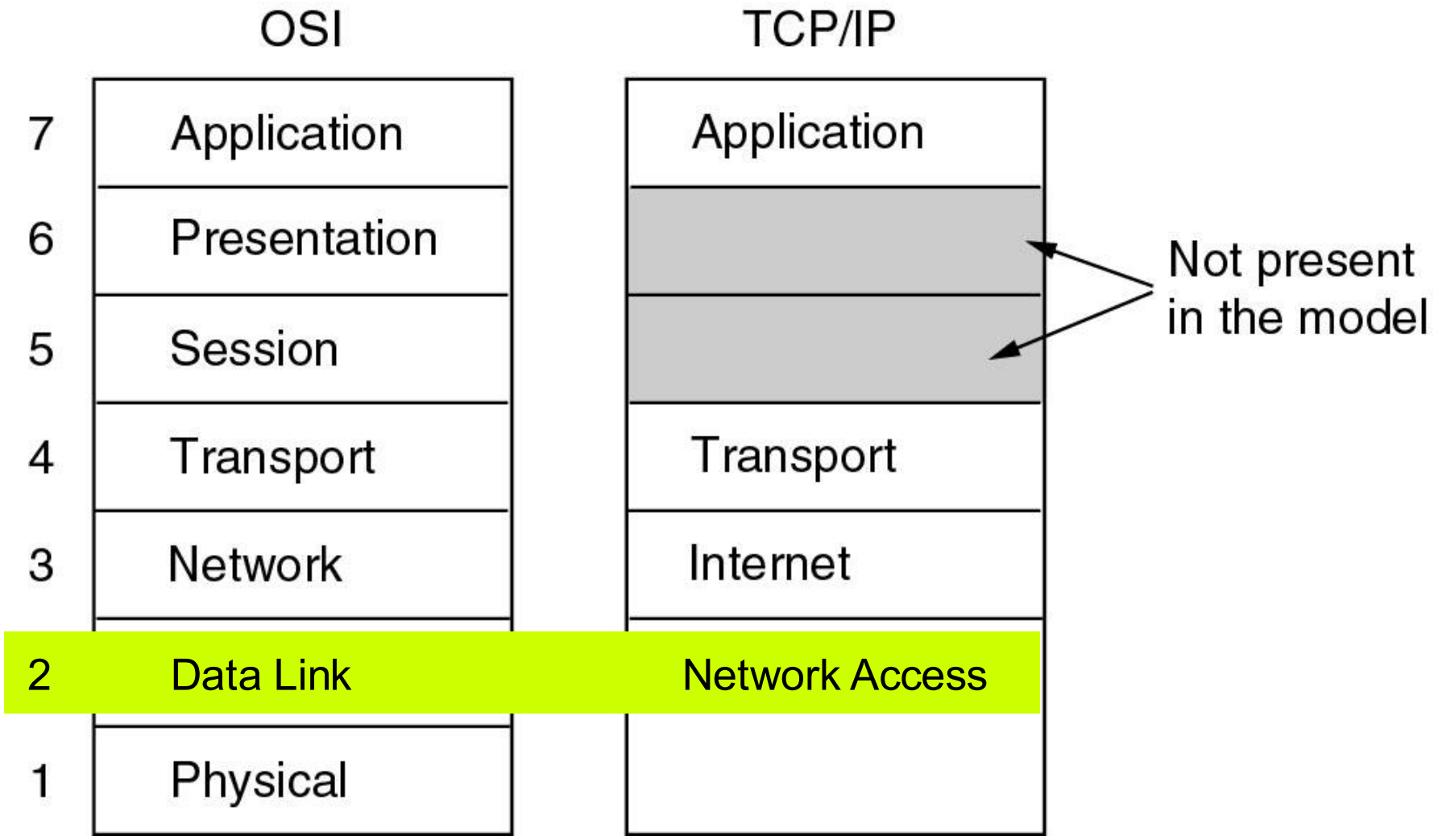
- Sie stellen dar:
  - welche Eigenschaften und Grenzen die Berechnung einer CRC-Prüfsumme hat
  - wie Fluss-Steuerung mittels Stop-and-Wait-Verfahren umgesetzt werden kann und wie dadurch eine Überlastung des Empfängers verhindert wird
  - warum Medien-Zugriffsverfahren notwendig sind
  - wie Medien-Zugriff mittels ALOHA und Slotted ALOHA umgesetzt werden kann

Lernziele II

- Sie wenden Verfahren auf konkrete Eingaben an:
  - Framing durch Flag Bit Pattern & Bitstuffing
  - Berechnung einer CRC-Prüfsumme

- Sie wägen die Vor- und Nachteile ab:
  - von Framing durch Flag Bit Pattern & Bitstuffing gegenüber naiveren Ansätzen (Framing by Byte Count) ab
  - von Slotted ALOHA gegenüber ALOHA ab

Physical Layer



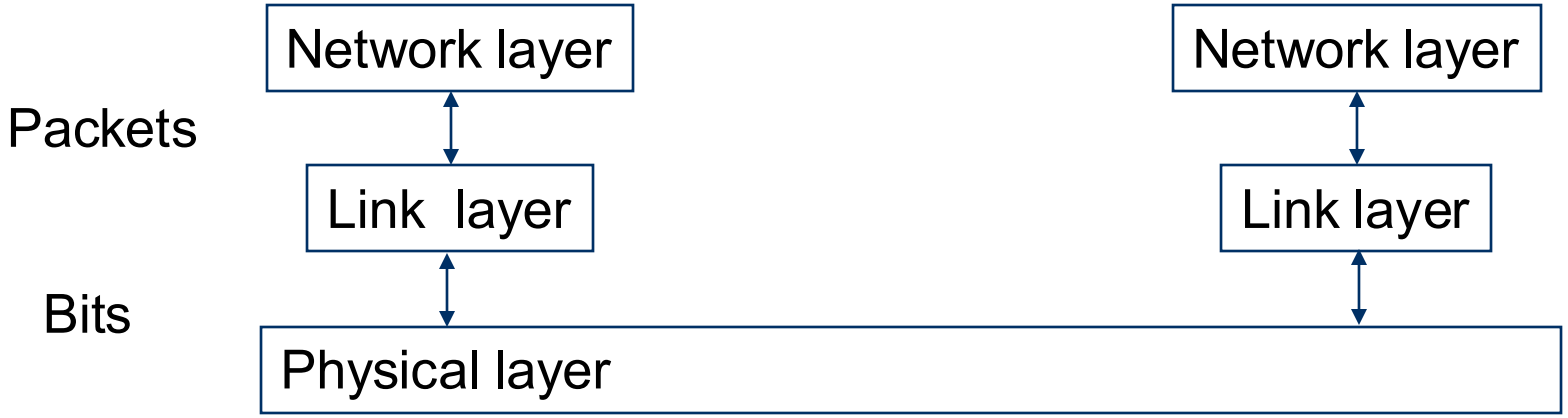
Setting for Data Link Layer

Link layer sits on top of physical layer

- Can thus use a bit stream transmission service
- But: Service might have incorrect bits

Expectations of the higher layer (networking layer)

- Wants to use either a packet service
  - Rarely, a bit stream service
- Does not really want to be bothered by errors
- Does not really want to care about issues at the other end



Services of Data Link Layer

Given setting and goals, the following services are required:

- Transparent communication between two directly connected nodes
- Framing of a physical bit stream into a structure of frames/packets
  - Frames can be retransmitted, scheduled, ordered, ...
- Error control
  - Error Detection and Error correction
- Connection setup and release
  - Signaling and resource management on hosts
- Acknowledgement-based protocols
  - Make sure that a frame has been transmitted
- Flow control
  - Arrange for appropriate transmission speed between hosts

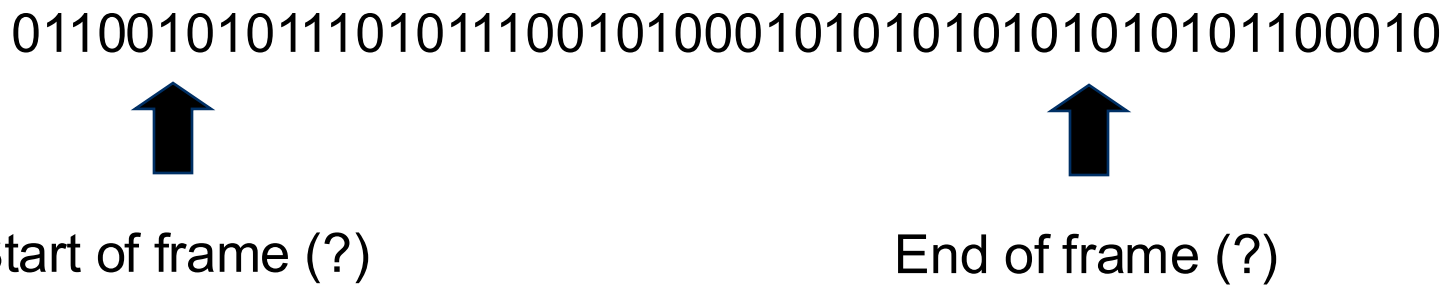
Framing

Framing

How to turn a bit stream into a sequence of frames?

- More precisely: how does a receiver know when a frame starts and when it ends?

Delivered by physical layer:



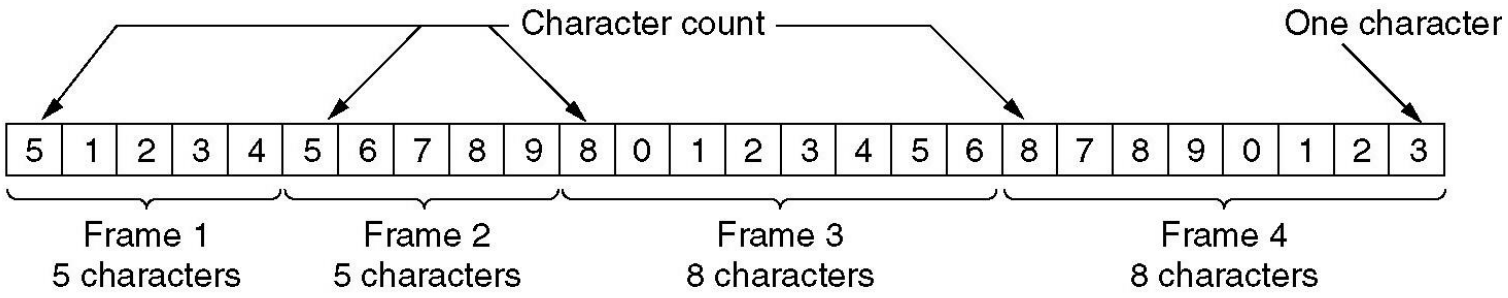
Note: Physical layer might try to detect and deliver bits when the sender is not actually transmitting anything

- Receiver tries to get any information from the physical medium

Framing by Character / Byte Count

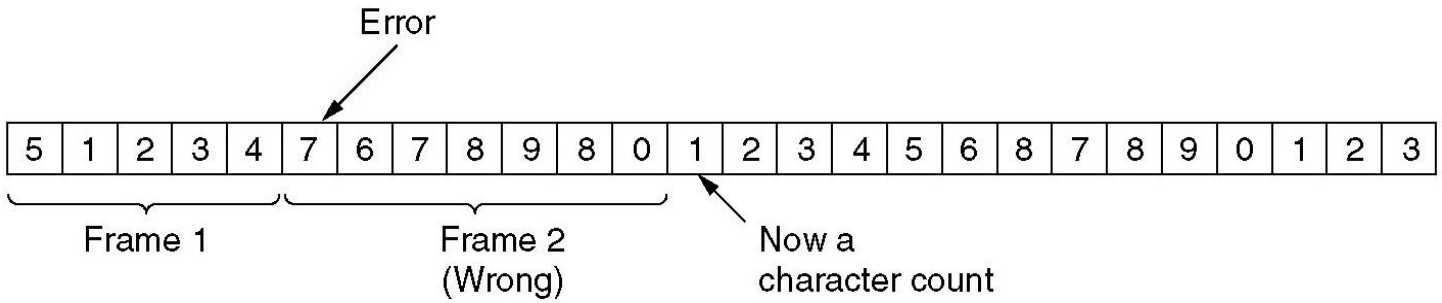
Idea: Announce number of bits (bytes, characters) in a frame to the receiver

- Put this information at beginning of a frame – *frame header*



Problem: What happens if *count* information itself is damaged during transmission?

- Receiver will loose frame synchronization and produce different sequence of frames than original one

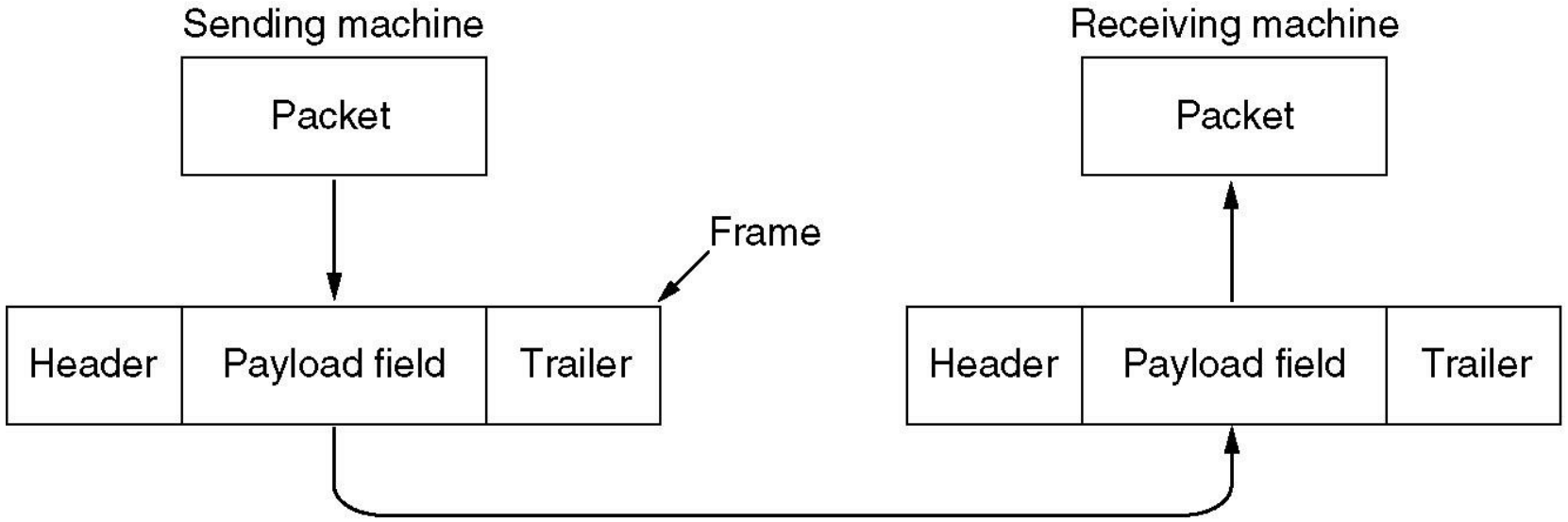


Basic Technique: Control Header

Although “character count” is not a good framing technique, it illustrates an important technique: *headers*

- If sender has to communicate administrative or control data to receiver, it can be added to actual packet content (“payload”)
- Usually at the start of the packet; sometimes at the end (“trailer”)
- Receiver uses headers to learn about sender’s intention

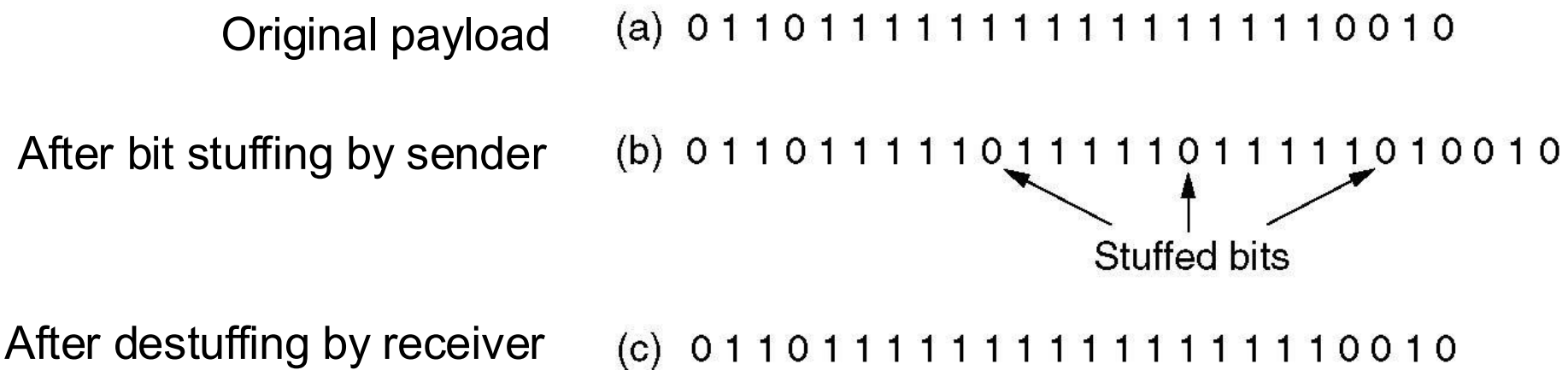
➢ Same principle applicable to all packet-switched communication



Framing by Flag Bit Patterns / Bit Stuffing

Use dedicated flag bits to demarcate start/stop of frame

- Use bit pattern – often, 0111 1110
- Bit stuffing process:
  - Whenever sender sends five 1s in a row, it automatically adds a 0 into bit stream – except in flag pattern
  - Receiver throws away (“destuffs”) any 0 after five 1s

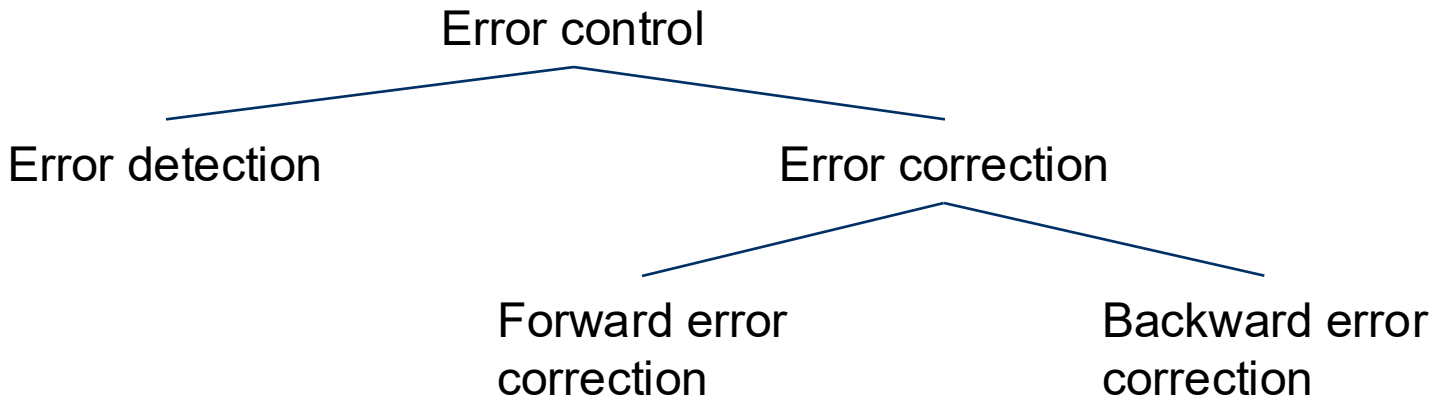


Error Control

Link Layer Functions – Error Control

Error detection – Check for incorrect bits
Error correction – Correct erroneous bits

- Forward error correction (FEC) – invest effort *before* error happened; avoid delays in dealing with it
  - ➢ Redundancy / overhead
- Backward error correction – invest effort *after* error happened; try to repair it ➔ ARQ (Automatic Repeat reQuest)
  - ➢ Delays



➢Usually build on top of framing

Error Detection: Cyclic Redundancy Check (CRC)

CRC can check arbitrary, unstructured sequence of bits
A check sequence (CRC code) is appended to checked data

- Typically calculated by a feedback shift register in hardware

When calculating the CRC code a generator polynomial is used which is known to both sender and receiver

Calculation of CRC code

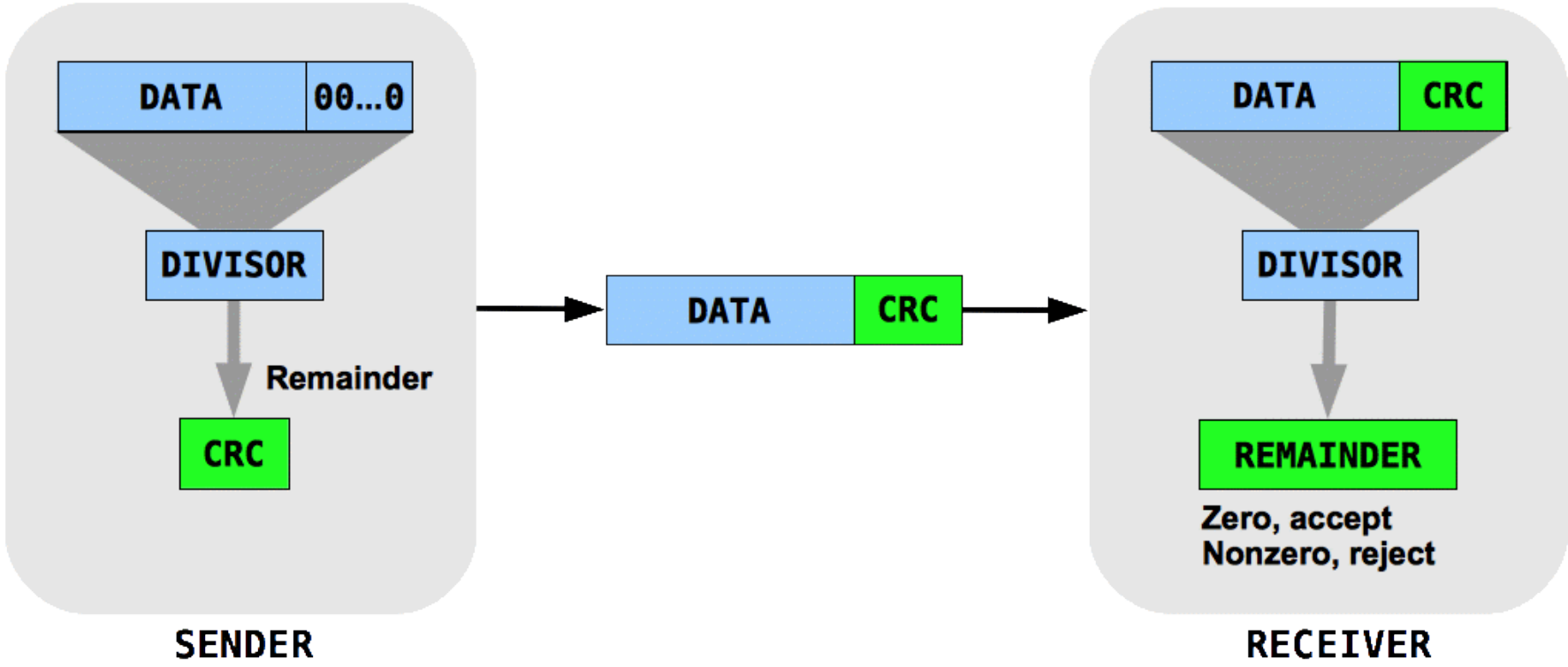
1. View bit sequence as polynomial with binary coefficients:
   - **100010** is viewed as $\mathbf{1}*x^5+\mathbf{0}*x^4+\mathbf{0}*x^3+\mathbf{0}*x^2+\mathbf{1}*x^1+\mathbf{0}*x^0$
2. Expand polynomial with n 0s, n is the degree of the generator polynomial
3. Divide expanded bit sequence (i.e. polynomial) by generator polynomial

➢ CRC code is the remainder of the division, result is discarded

4. Receiver again divides received bit sequence (including the CRC code) by generator polynomial

➢ If no error occurred the remainder is 0

Illustration of CRC



CRC Example (Sender)

Transmitted payload: 110011

Generator polynomial: $x^4 + x^3 + 1 = 1*x^4 + 1*x^3 + 0*x^2 + 0*x^1 + 1*x^0$

CRC Example (Sender)

Transmitted payload: 110011

Generator polynomial: $x^4 + x^3 + 1 = 1*x^4 + 1*x^3 + 0*x^2 + 0*x^1 + 1*x^0$

- Translates into sequence of coefficients: 11001

CRC Example (Sender)

Transmitted payload: 110011

Generator polynomial: $x^4 + x^3 + 1 = 1*x^4 + 1*x^3 + 0*x^2 + 0*x^1 + 1*x^0$

- ➢ Translates into sequence of coefficients: 11001
- Addition or subtraction equal **simple bitwise XOR**
  - ➢ Special arithmetic for polynomials modulo 2

Length of CRC = degree of generator polynomial = 4

Calculation of CRC:

```
1100110000 ÷ 11001 = 100001 (mod 2)
11001
0000010000
    11001
     1001  = remainder
```

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

➢ Transmitted bit sequence: 1100111001

CRC Example (Receiver)

Reception of a correct bit sequence:

```
1100111001 ÷ 11001 = 100001 (mod 2)
11001
0000011001
     11001
      0000 = remainder
```

➢ No remainder, thus the received bits *should* be error free

Reception of a erroneous bit sequence:

```
1111111000 ÷ 11001 = 101001 (mod 2)
11001
0011011
  11001
  00010000
     11001
      1001 = remainder ≠ 0
```

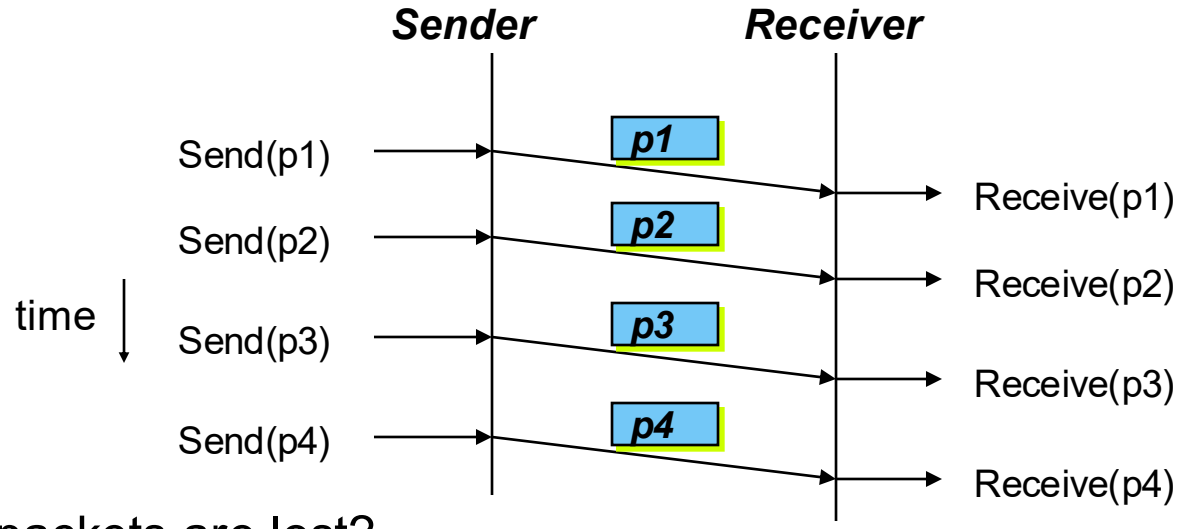
➢ There is a remainder unequal 0, thus there was *definitely* a transmission error

Flow Control

Link Layer Functions – Flow Control

Assumptions in an ideal world:

- Sender/receiver are always ready to send/receive
- Receiver can handle amount of incoming data
- No errors occur that cannot be handled by Forward Error Correction (FEC)



➢What happens if packets are lost?

➢What happens if the sender floods the receiver?

- Imagine a web server sending data to a mobile phone…

Very Simple Solution: Stop-and-Wait

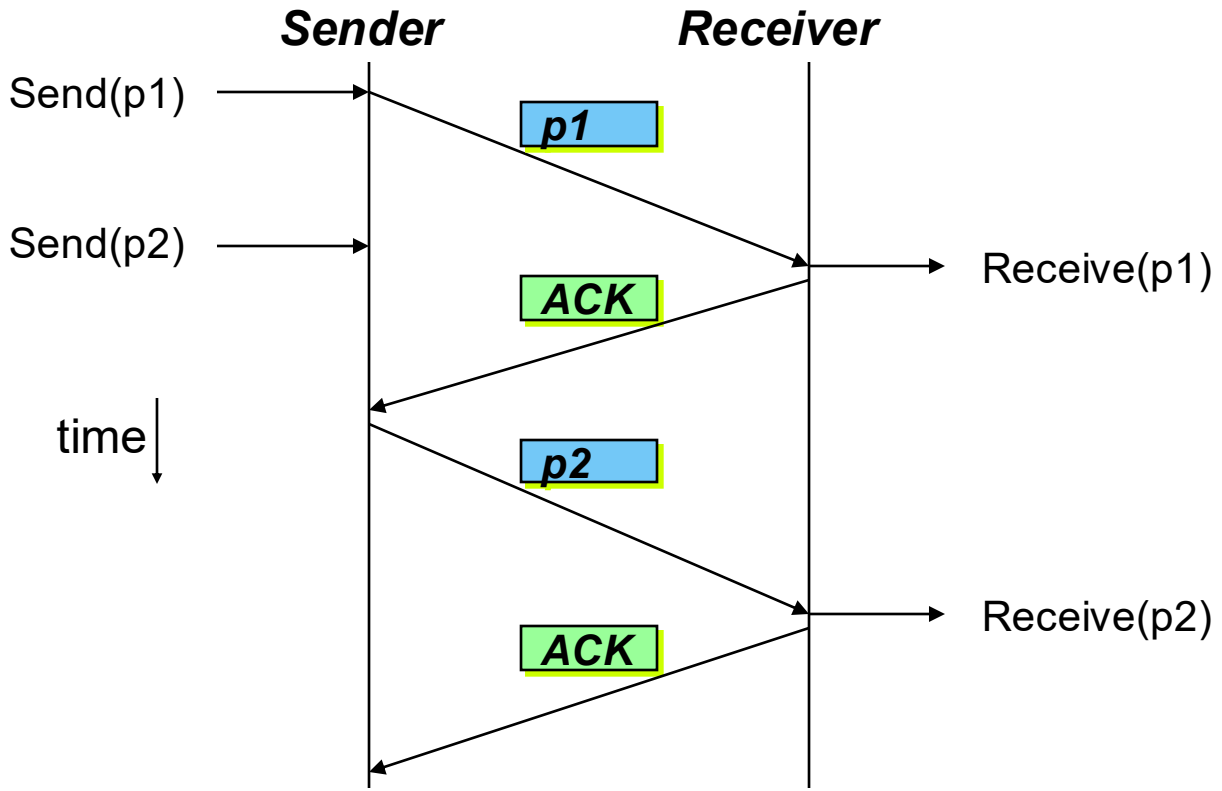
Concentrate on one single packet

Receiver acknowledges correct reception of the packet

Sender has to wait for that acknowledgement before continuing with next packet

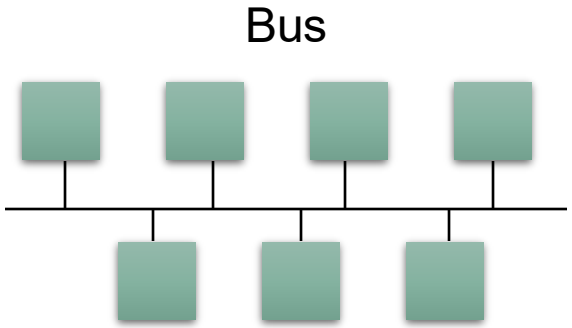
No overloading
of the receiver
possible!

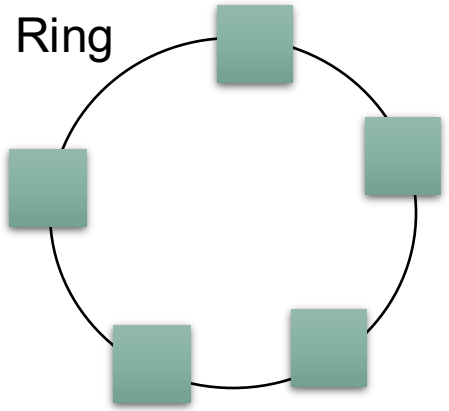
➢Basic flow control

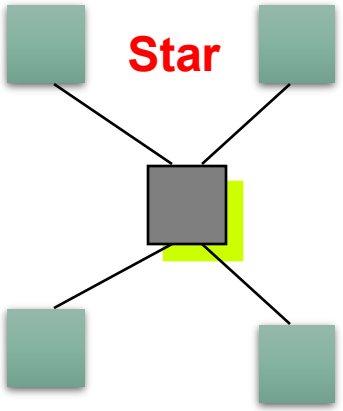


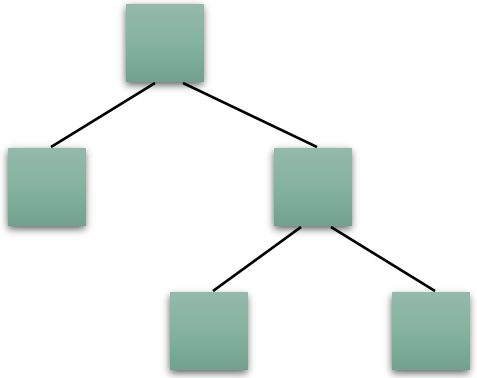
Network Topologies

Topologies by Network Structure

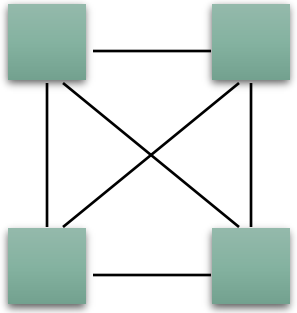




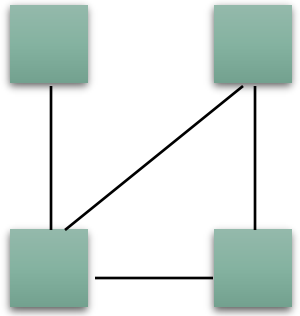




Tree



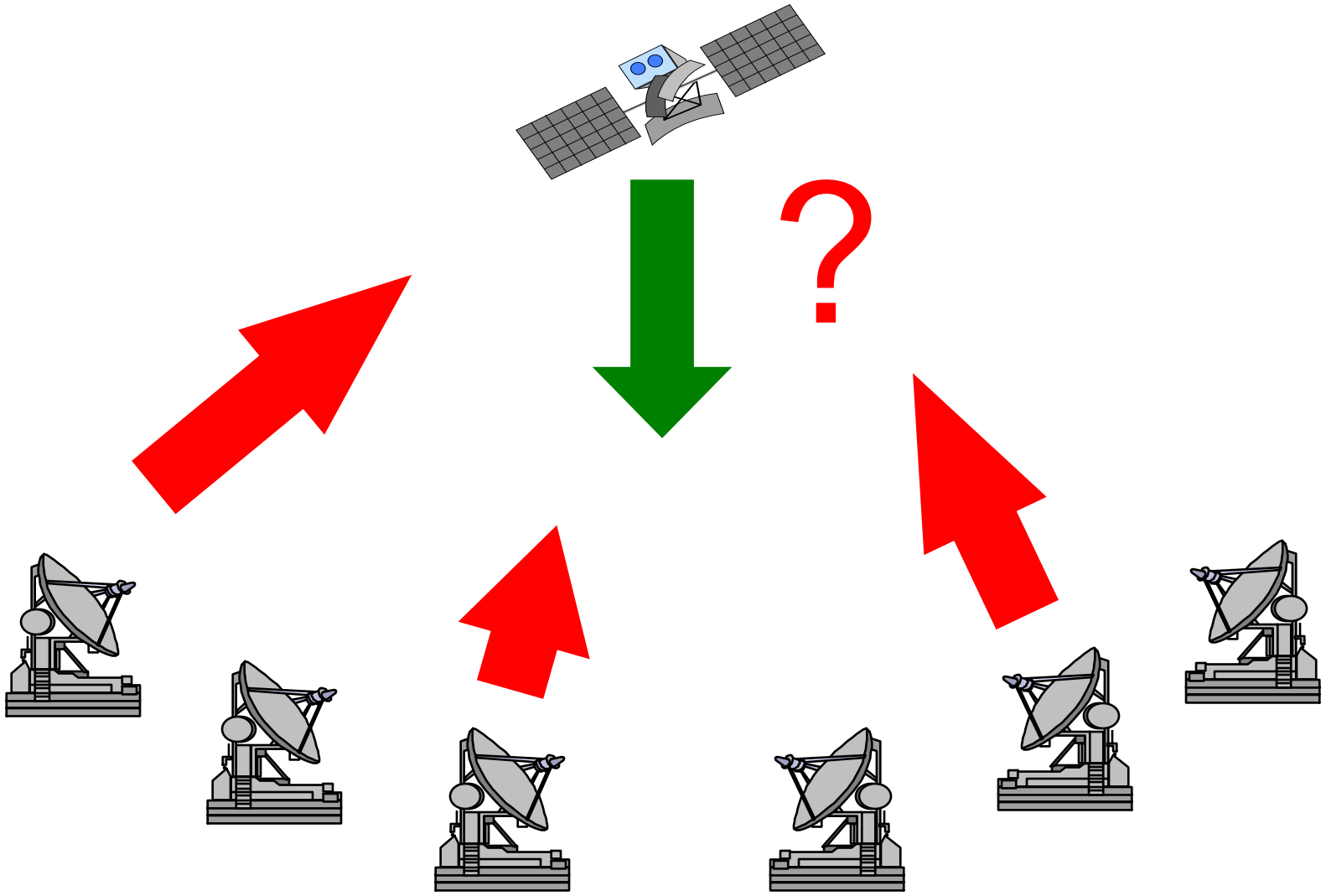
Fully meshed network



Partially meshed network

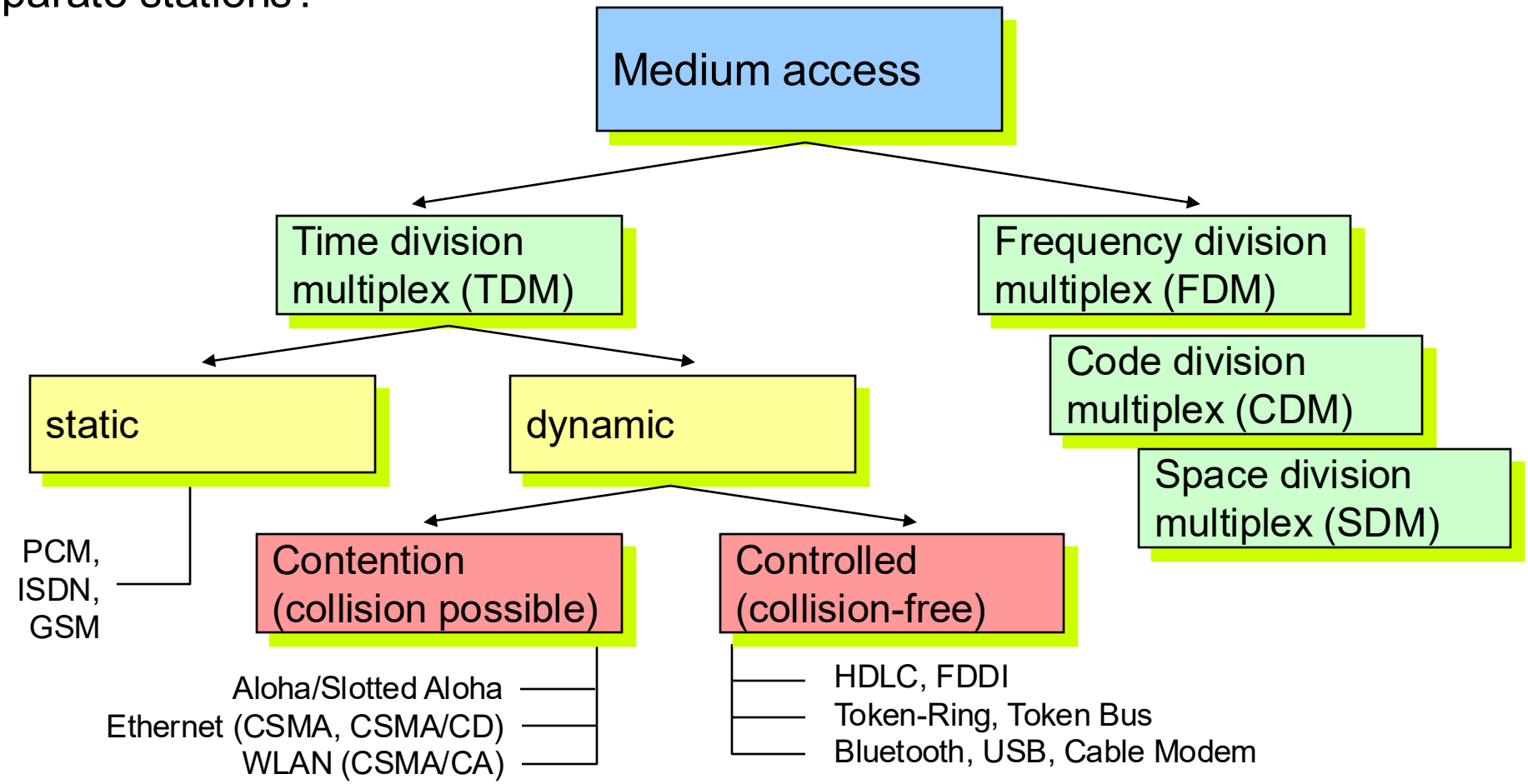
Medium Access Control

Motivation: Satellite Medium Access



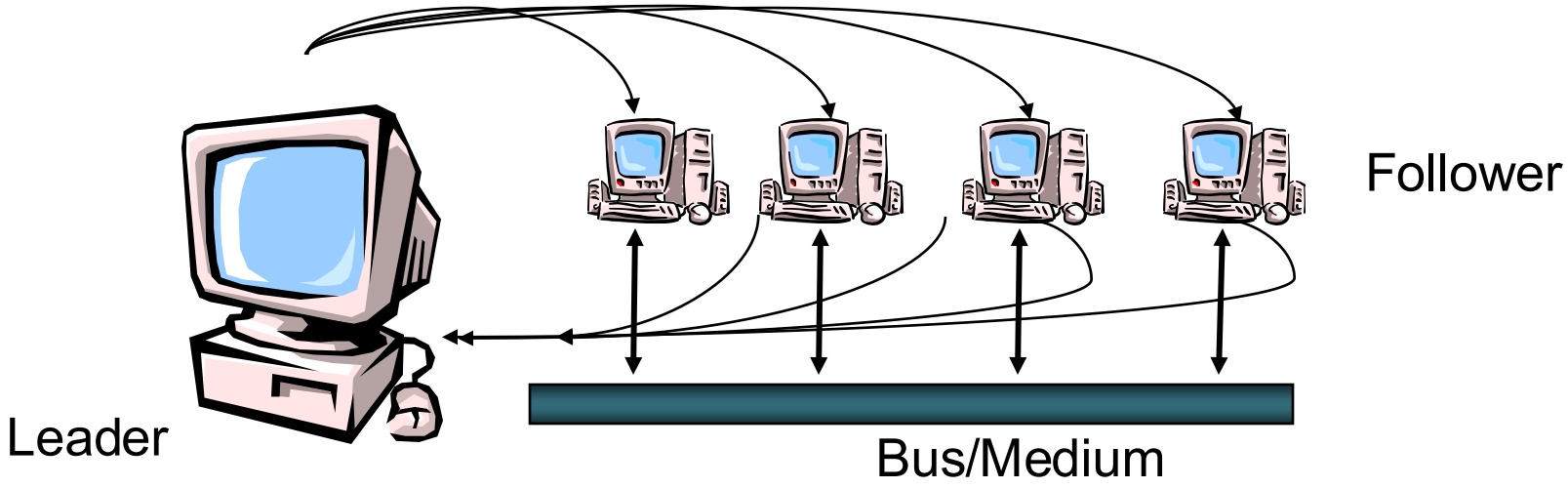
Approaches to Medium Access

Several stations want to access the same medium
– how to separate stations?



Dynamic/Controlled MAC: Polling

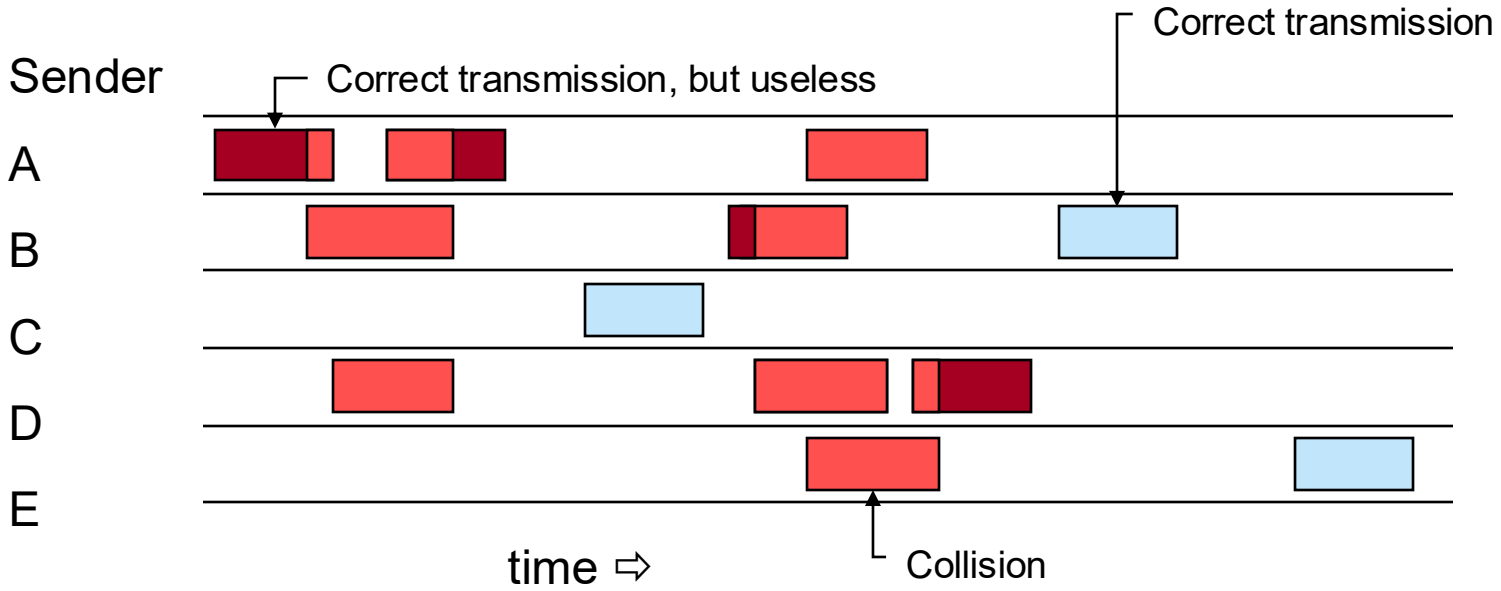
Single leader station
One or more follower stations
Typically bus topology (but also tree)
Leader polls followers according to table or cyclically
Followers may answer only after being polled



➢Examples: Bluetooth, USB, cable modems

Dynamic/Contention MAC: ALOHA

No central control, no coordination between stations
Stations start sending whenever they want to
➢ Collisions destroy frames
Receiver may send acknowledgement after correct reception



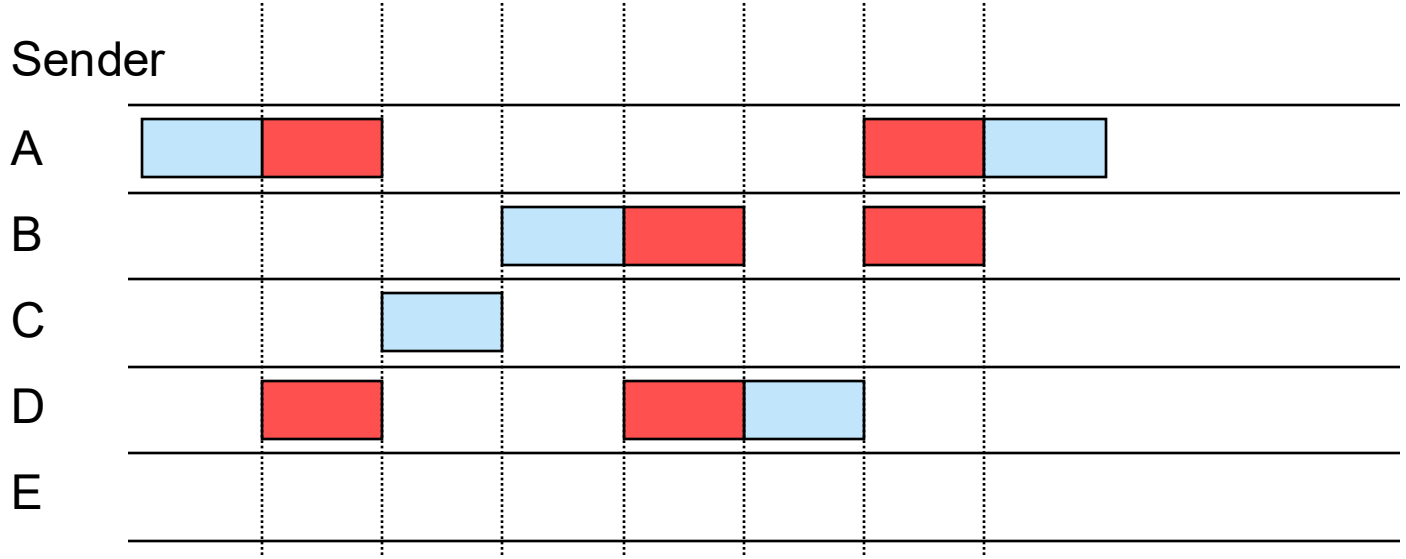
➢Best option without any carrier sensing or central station

Dynamic/Contention MAC: Slotted ALOHA

Send packets of fixed length within fixed time-slots

➢ Requires common time-base for synchronization

Transmission starts at begin of slot only, thus only complete collisions may occur



Delay: < slot time
Throughput: 45%
No fairness

➢Best option without carrier sensing (but sync required)

➢Example: Initial medium access of GSM control channel

Performance – Intuition

Aloha

Slotted Aloha

Roadmap

8. Networked Computer & Internet
9. Network Access Layer I – Physical Layer
10. **Network Access Layer II – Data Link Layer**
11. Internet Layer – Network Layer
12. Transport Layer
13. Applications