THE UNIVERSITY OF SYDNEY

Lecture 3

Data Link Layer

ELEC 3506/9506
Communication Networks

Dr Wibowo Hardjawana
School of Electrical and Information Engineering

THE UNIVERSITY OF SYDNEY

Topics of the day

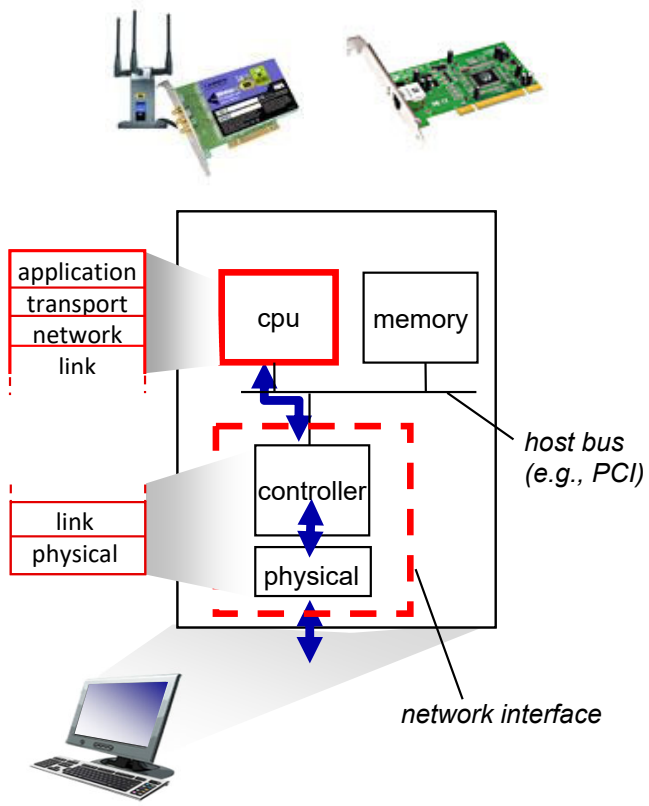
- Data Link Layer Services
- Framing
- Error Detection
- Error Correction
- Protocols
- Data Link Layer networking devices:
  - Bridges
  - Switches
  - Routers

Data Link Layer Services

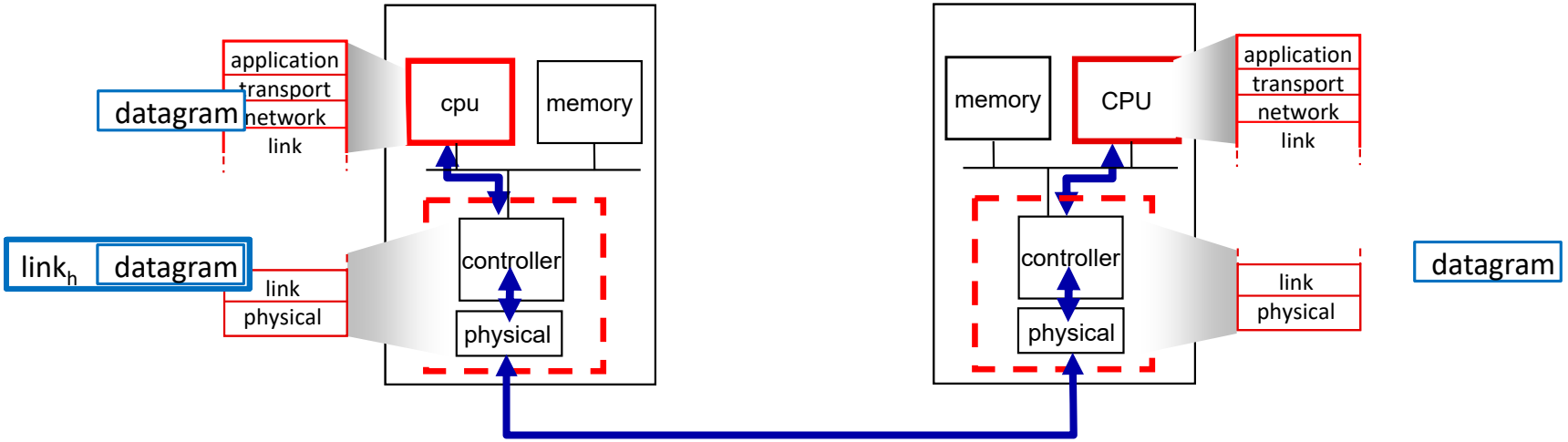
- Data Link layer ensures **reliable datagram** delivery between two physically connected devices
- Framing
  - encapsulate datagram into frame, adding header, trailer
  - 'physical addresses' used in frame headers to identify source, destination - different from IP address!
- Access Control
  - Determining who should have access to the physical layer
- Error Detection
  - errors caused by signal attenuation, noise
  - receiver detects presence of errors, signals sender for retransmission or drops frame
- Error Correction
  - receiver identifies *and corrects* bit error(s)

Where is the link layer implemented?

- in each-and-every host
- link layer implemented in *network interface card* (NIC) or on a chip
  - Ethernet, WiFi card or chip
  - implements link, physical layer
- attaches into host's system buses
- combination of hardware, software, firmware



Interfaces communicating



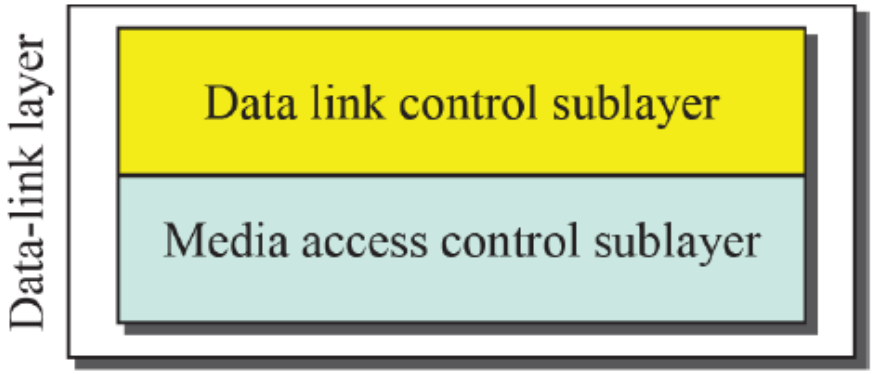
sending side:

- encapsulates datagram in frame
- adds error checking bits, reliable data transfer, flow control, etc.

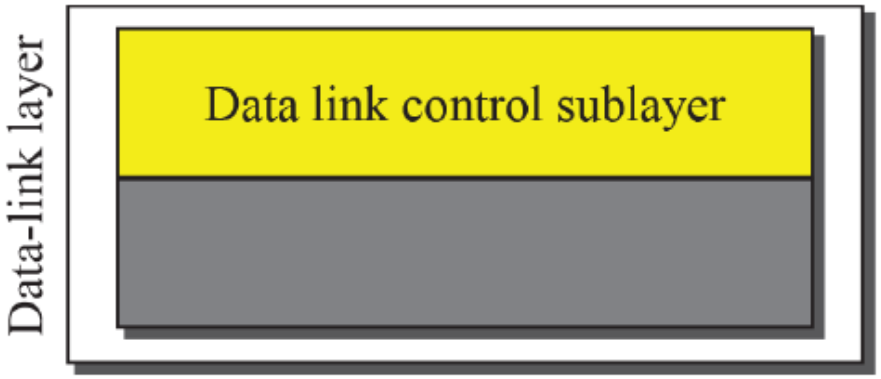
receiving side:

- looks for errors, reliable data transfer, flow control, etc.
- extracts datagram, passes to upper layer at receiving side

Two sublayers

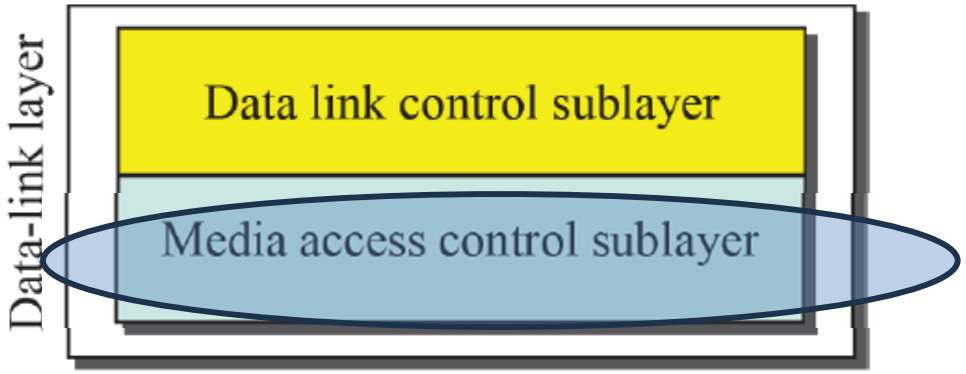


a. Data-link layer of a broadcast link



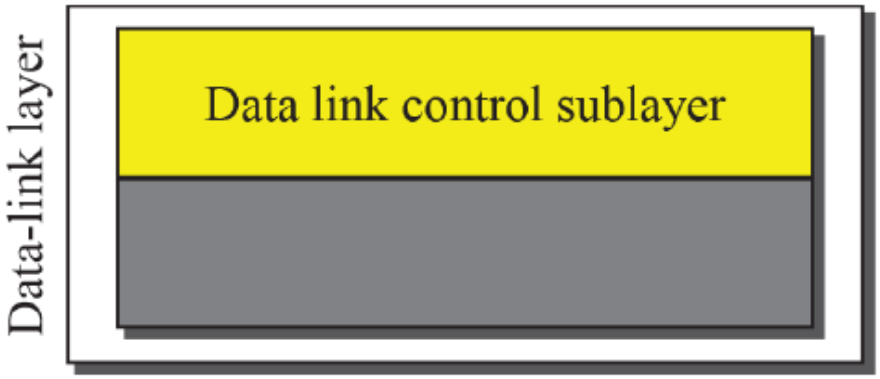
b. Data-link layer of a point-to-point link

Two sublayers



next week

a. Data-link layer of a broadcast link



b. Data-link layer of a point-to-point link

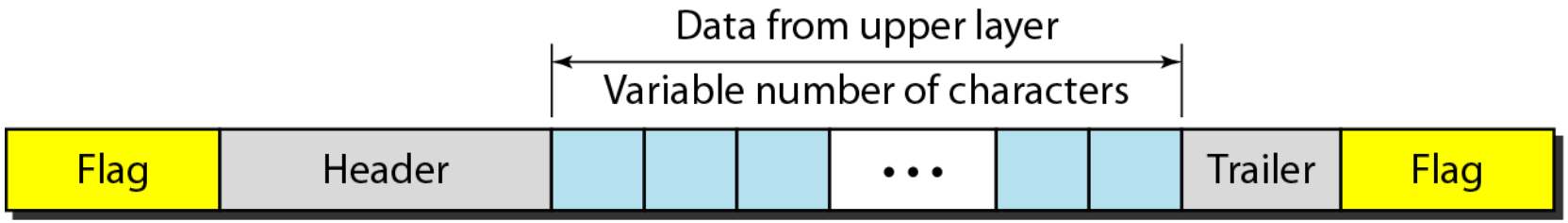
THE UNIVERSITY OF SYDNEY

DLC: Framing

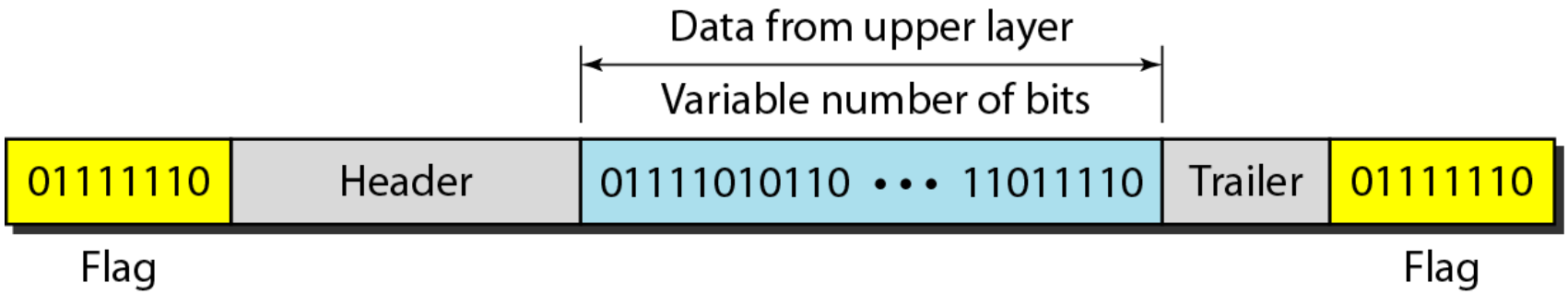
- The data link layer needs to pack bits into frames, so that each frame is distinguishable from another.
- Sender and destination addresses are added to the frame
- Usually a message is packetized into smaller frames (rather than packetizing into a single frame)
- Two types of framing:
  - Fixed-size framing (e.g., ATM)
  - Variable-size framing (e.g., LANs)

DLC: Variable-size framing

- Character-oriented



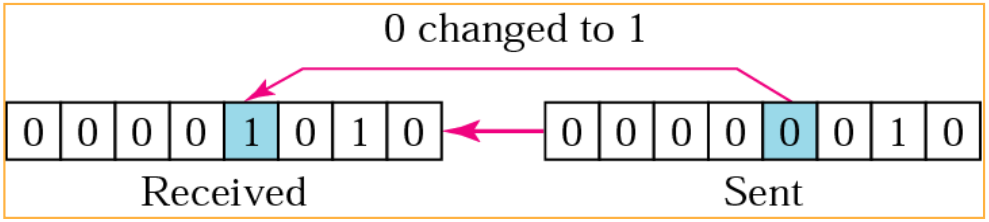
- Bit-oriented



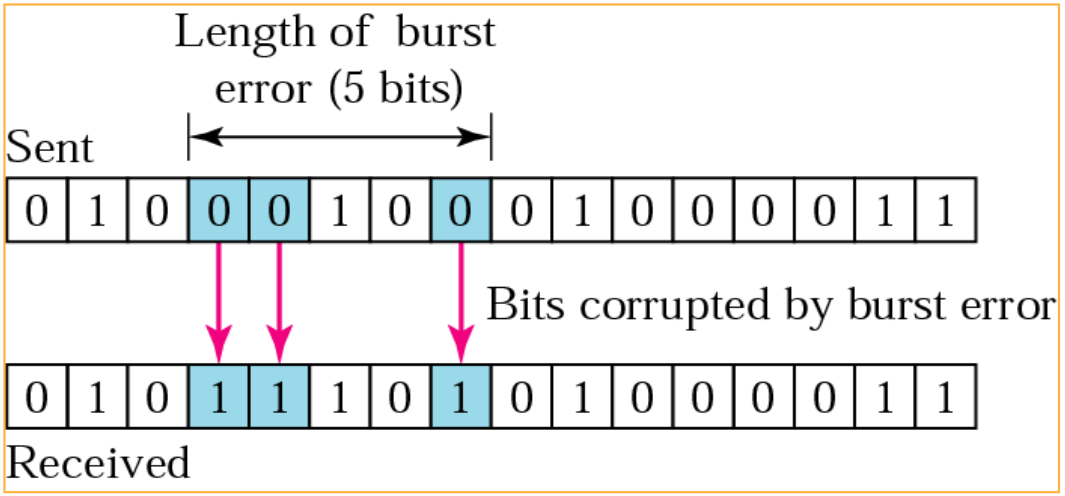
DLC: Error Detection

Types of errors

- Single-bit Errors – Only one bit has changed



- Burst Error – Two or more bits have changed



THE UNIVERSITY OF SYDNEY

DLC: Error Detection Methods

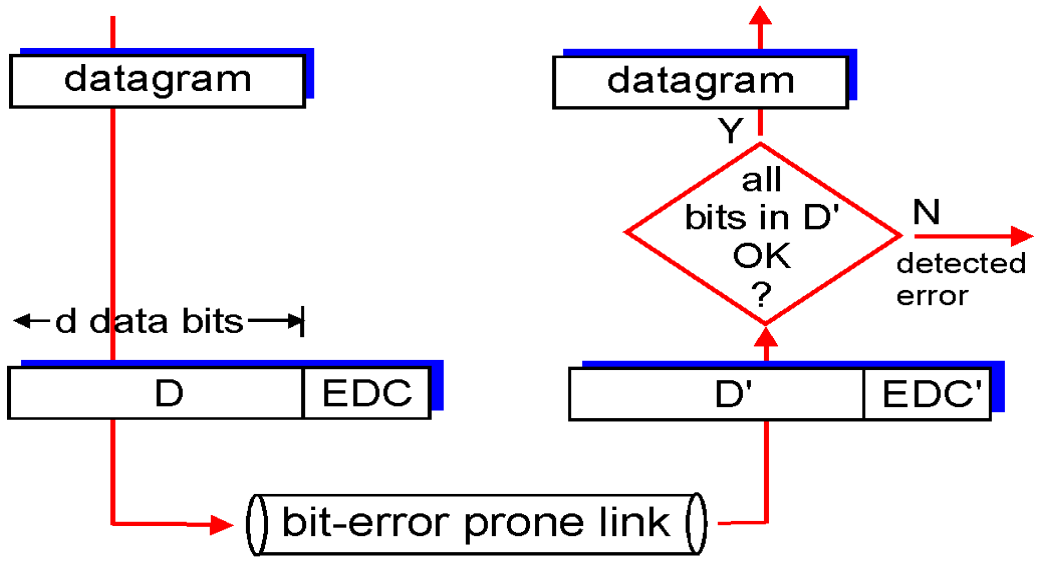
- Three most common error detection methods
  - Parity Checking (Vertical Redundancy Checking - VRC)
  - Longitudinal Redundancy Checking - LRC
  - Polynomial checking
    - Checksum
    - Cyclic Redundancy Checking (CRC)

DLC: Error Detection

- Error detection not 100% reliable!
  - protocol may miss some errors, but rarely
  - larger EDC field yields better detection and correction

EDC = Error Detection and Correction bits (redundancy)

D = Data protected by error checking, may include header fields

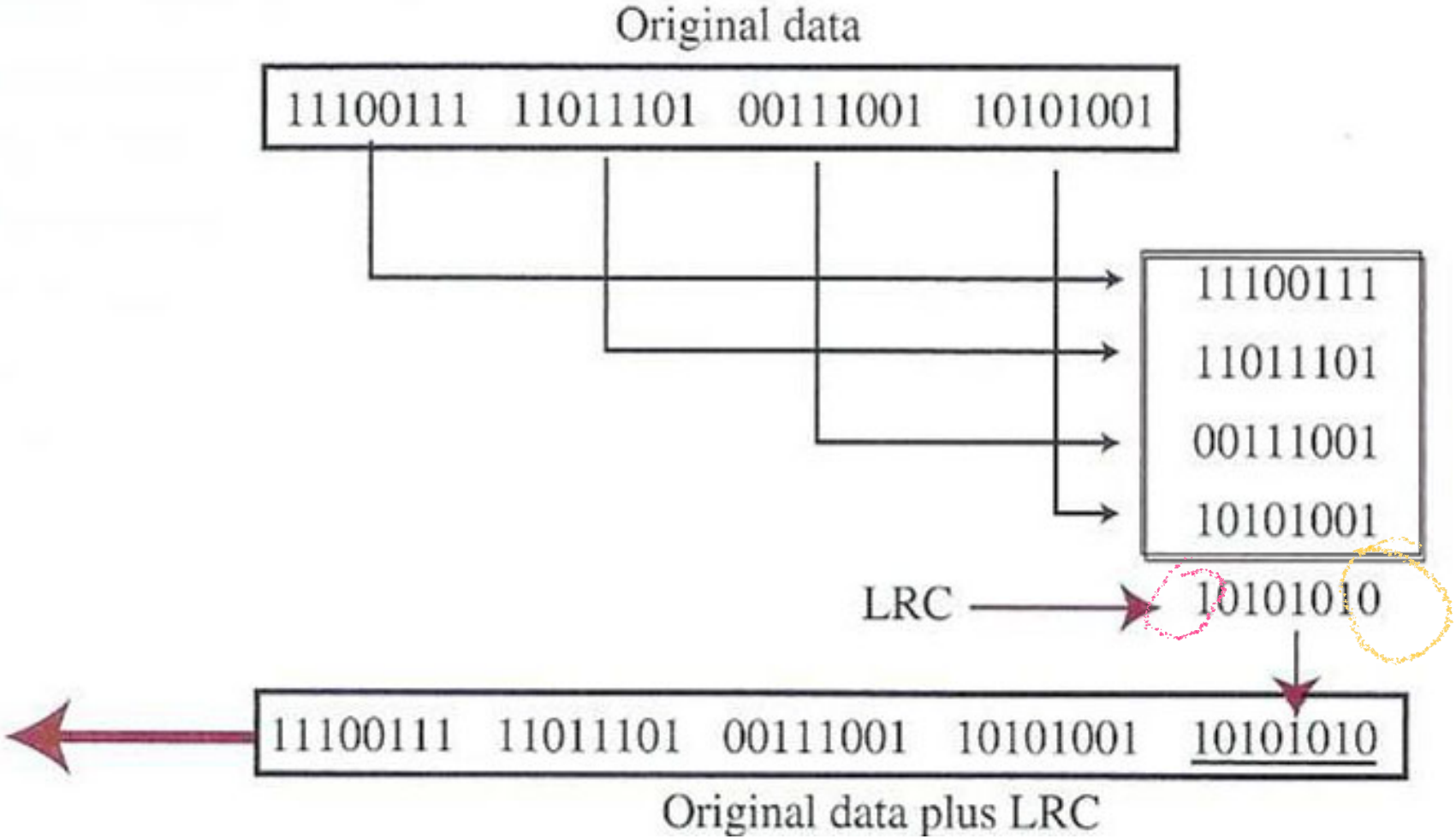


DLC: Parity Checking (Vertical Redundancy Checking – VRC)

- One of the oldest and simplest methods
- Adds 1 additional bit to each byte in the message
- The value of this parity bit is dependent on the number of 1's in each byte transmitted.
  - Even parity causes the sum of all bits to be even
  - Odd parity causes the sum of all bits to be odd
- Unfortunately if two bits are erroneous, the parity checking will fail.
- Parity checking results in about a 50% reliability rate.
- Example:
  - Assume we are using even parity with 7-bit ASCII.
  - The letter V in 7-bit ASCII is encoded as 0110101.
  - Because there are four 1s (an even number), parity is set to zero.
  - This would be transmitted as (codeword) : 01101010.

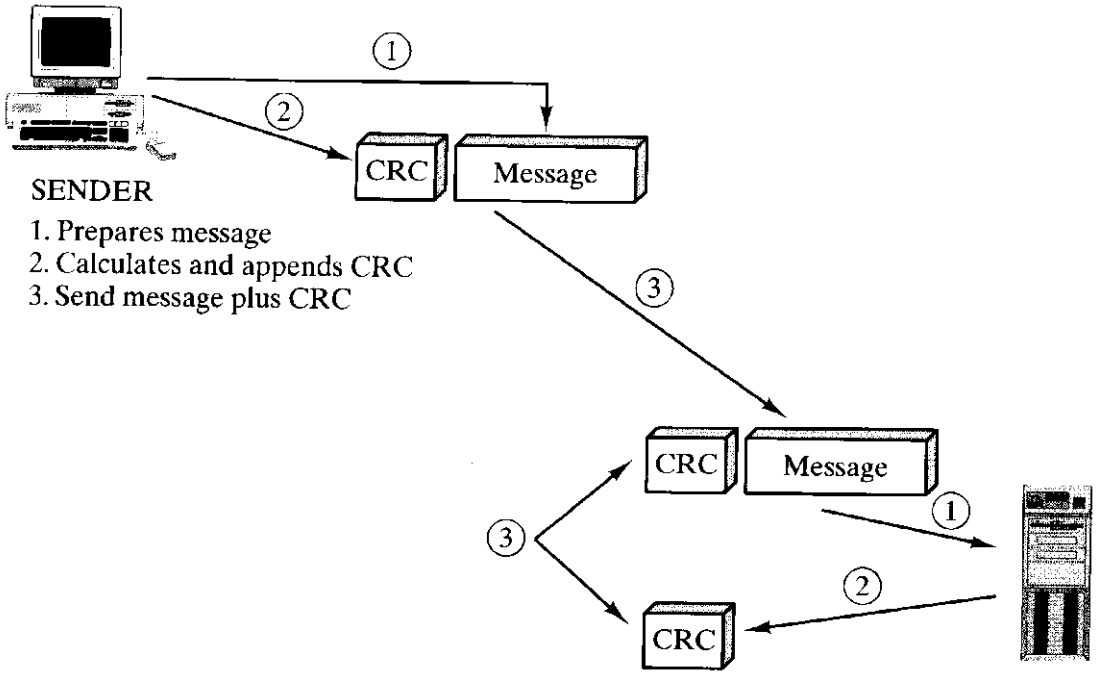
DLC: Longitudinal Redundancy Checking – LRC (Even Parity)

- A block of bits is divided into rows and a redundant row of bits is added to the whole block.



DLC: The Cyclic Redundancy Check (CRC)

- The CRC is based on binary division
- The redundancy bits are derived by dividing the data unit by a predetermined divisor and the remainder is the CRC



DLC: Error Detection

We have a message of *k* bits, we add (how?) *(n-k)* parity bits, to get a codeword of length *n* bits.

Objective: if one or more bits are received with errors, we can still recover the original *k* bits

The code rate is defined as $R = k/n$

The lower the rate, the better error protection you can get but you will get lower throughput.

DLC: Cyclic Redundancy Check Codes

Cyclic redundancy check (CRC) codes are cyclic which means that any circular shift of a codeword is a valid codeword too!

We can represent any bit sequence using a polynomial

$$(v_{k-1}, \ldots, v_1, v_0) \equiv v_{k-1}x^{k-1} + \ldots . + v_1 x + v_0$$

Any codeword of a cyclic code is a multiple of a unique polynomial called the generator polynomial, called *g(x).*

DLC: Cyclic Redundancy Check Codes

- Message length *k*, codeword length *n*
- To encode a message *u(x)*
- 1- Multiply *u(x)* by x^{n-k}
- 2- divide $u(x)x^{n-k}$ by *g(x)* and obtain the remainder *b(x)*
- 3- forming the codeword b(x)+ $u(x)x^{n-k}$

- Example: $g(x) = x^3 + x + 1$ and u$(x) = x^3 + 1 = (1\ 0\ 0\ 1)$, *k = 4,* and *n = 7.*

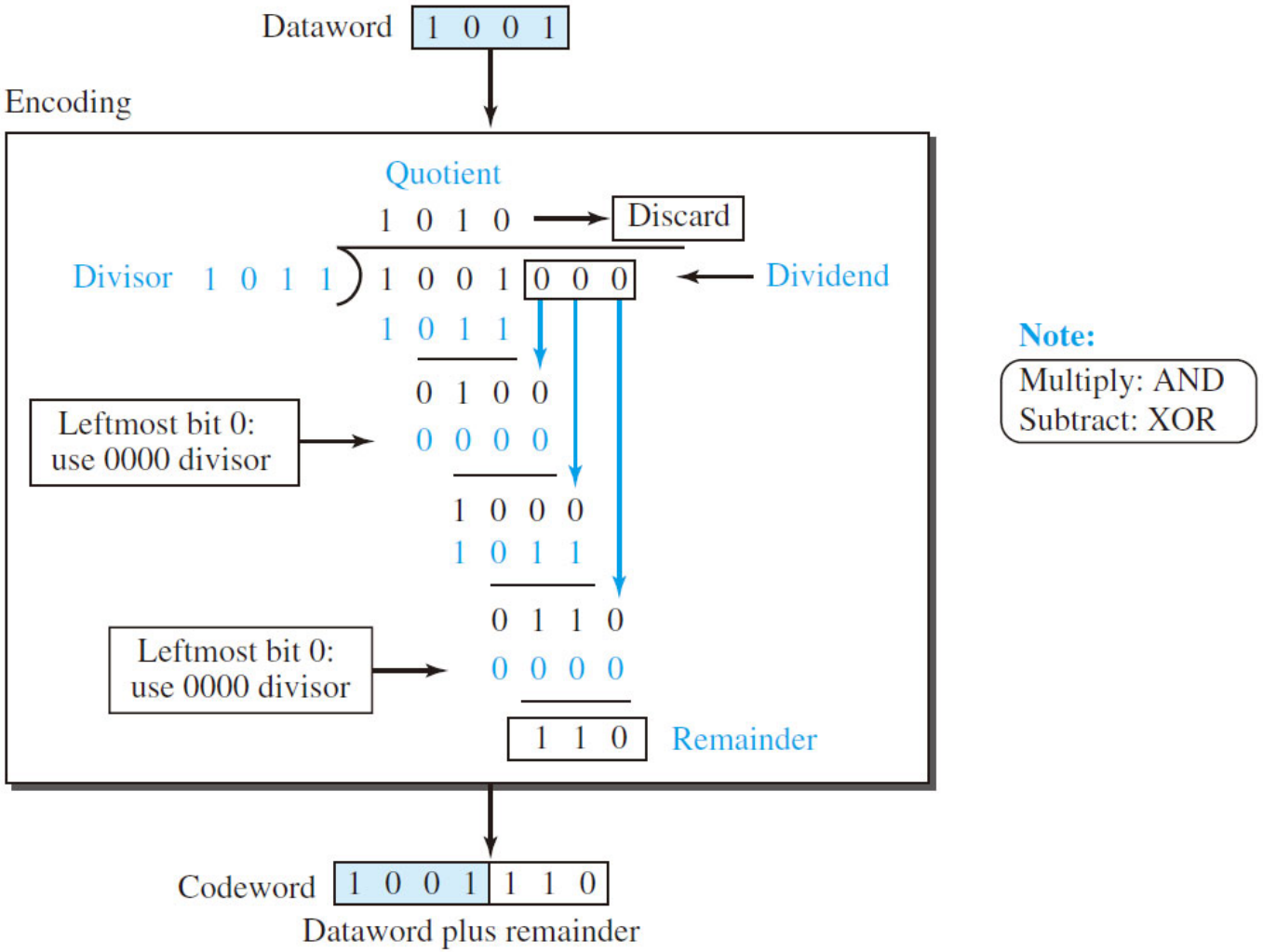
$$\frac{x^3\,u(x)}{g(x)} = \frac{x^6 + x^3}{x^3 + x + 1}\ and\ the\ remainder\ is\ b(x) = x^2 + x$$

The codeword is the

$$v(x) = b(x) + u(x)x^{n-k}$$
$$= x^2 + x + x^3\left(x^3 + 1\right) = x^6 + x^3 + x^2 + x = (1\ 0\ 0\ 1\ 1\ 1\ 0)$$

- $g(x)$ is a generator polynomial with a length of n-k+1 which is known by sender and receiver

DLC: CRC encoder



CRC decoder

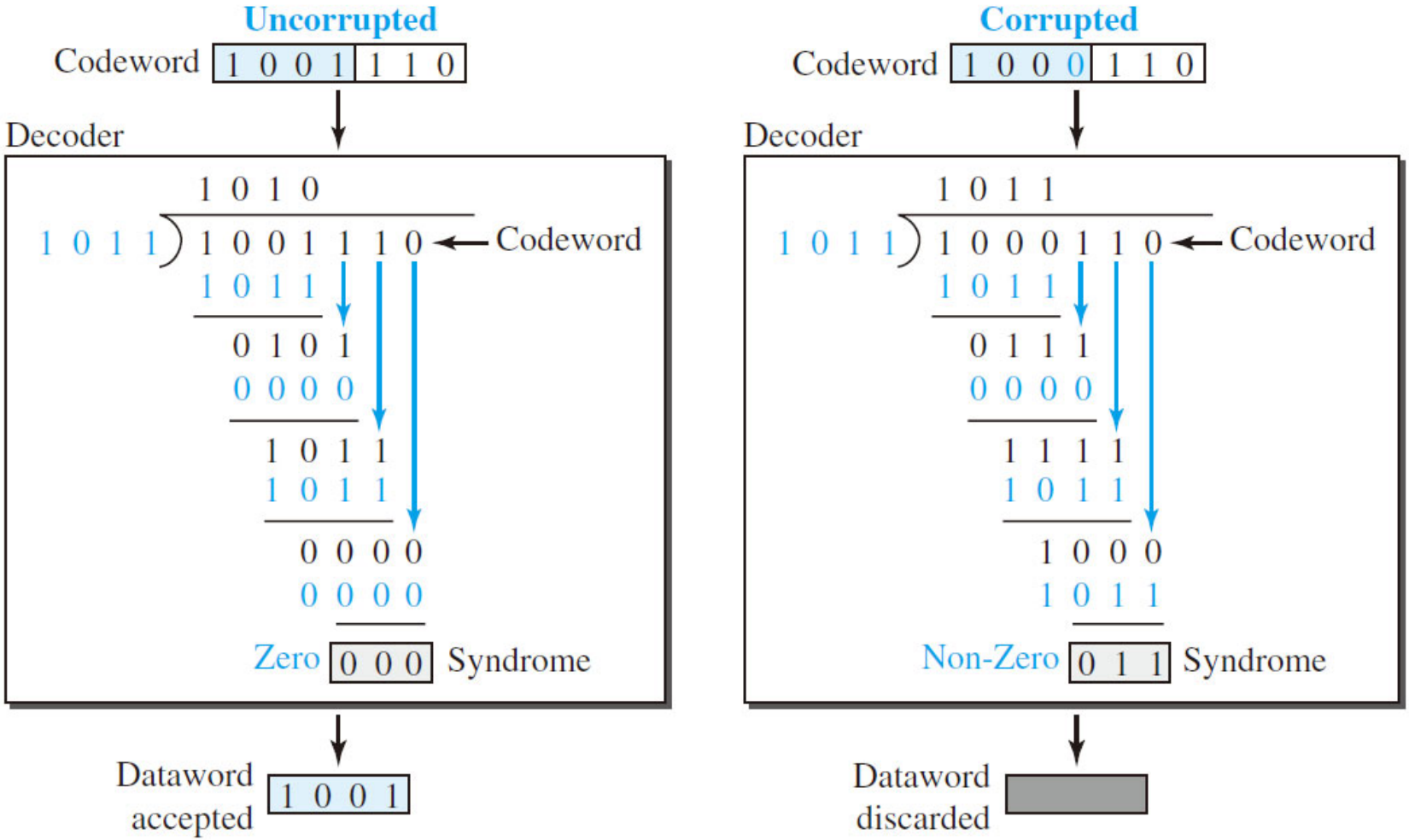
- Message length *k*, codeword length *n*
- To decode a received message *v(x)*
- 1- divide *v(x)* by g(x)
- If the remainder is zero then the received message is correct
- *Otherwise there are some errors*

- Example: $g(x) = x^3 + x + 1$ and v$(x) = (1\ 0\ 0\ 0\ 1\ 1\ 0)$, *k = 4,* and *n = 7.*

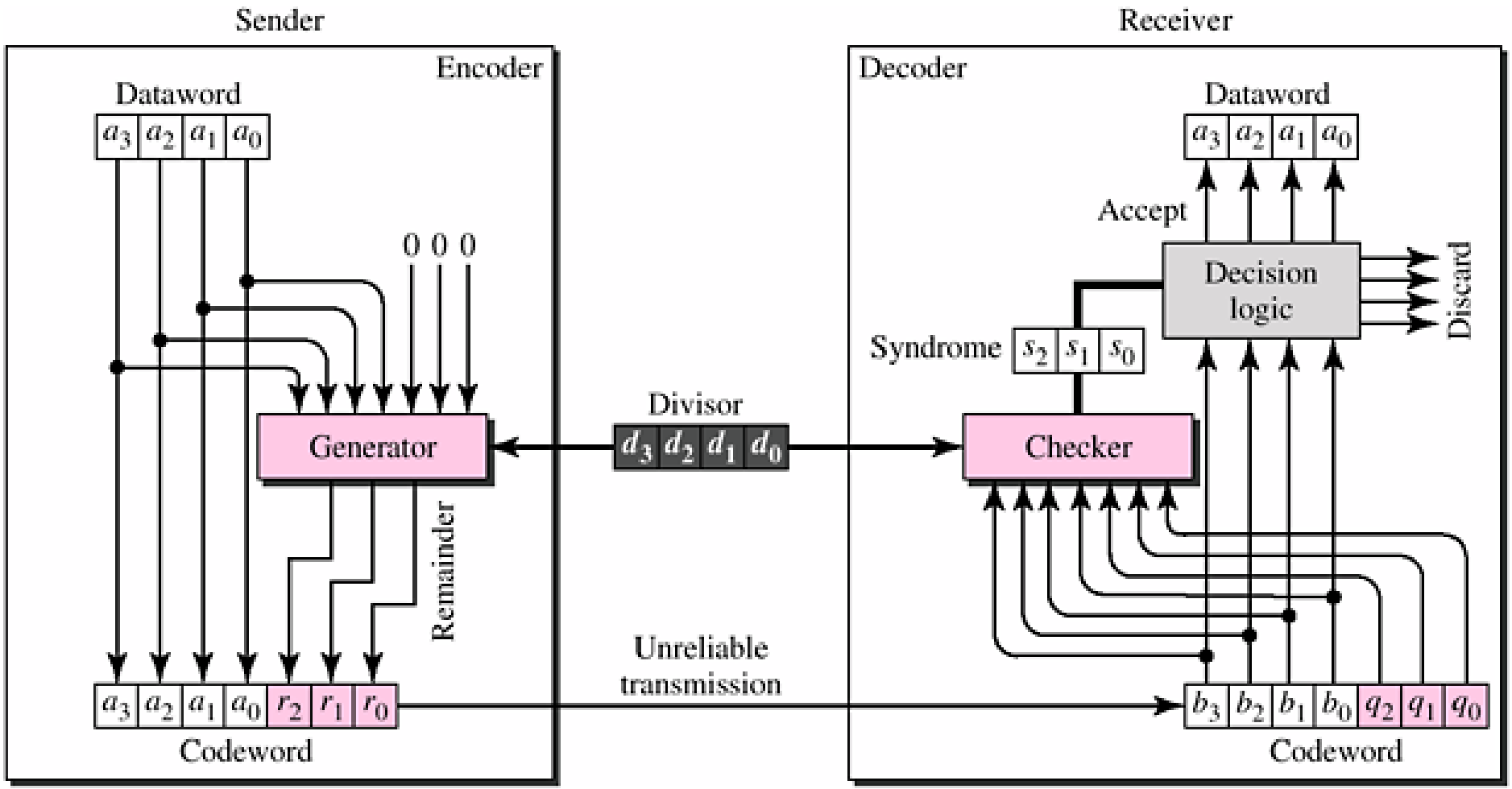
$$\frac{x^6 + x^2 + x}{x^3 + x + 1} \; and\ the\ remainder\ is\ x + 1$$

There are some errors!

DLC: CRC decoder



DLC: CRC Encoder and Decoder Process



DLC: Forward Error Correction (FEC)

We have a message of *k* bits, we add (how?) *(n-k)* parity bits, to get a codeword of length *n* bits.

Objective: if one or more bits are received with errors, we can still recover the original *k* bits

The code rate is defined as $R = k/n$

The lower the rate, the better error protection you can get but you will get lower throughput.

In 5G NR, we have LDPC (for data transmissions) and Polar Codes (for control channels)

Why? Control channel has very short length in which Polar code excels.

DLC: Error Correction/Control

- Once an error is detected, what is the receiver going to do?
  - Do nothing, or
  - Return an error message to the transmitter, or
  - Fix the error with no further help from the transmitter

- Error Control consists of the following steps (not all are necessary)
  - Error detection (Parity, CRC, etc. are used)
  - Positive Acknowledgment (ACK) - no detected error
  - Retransmission after time-out -- a frame or an ACK may have been lost
  - Negative acknowledgment (NAK) and retransmission

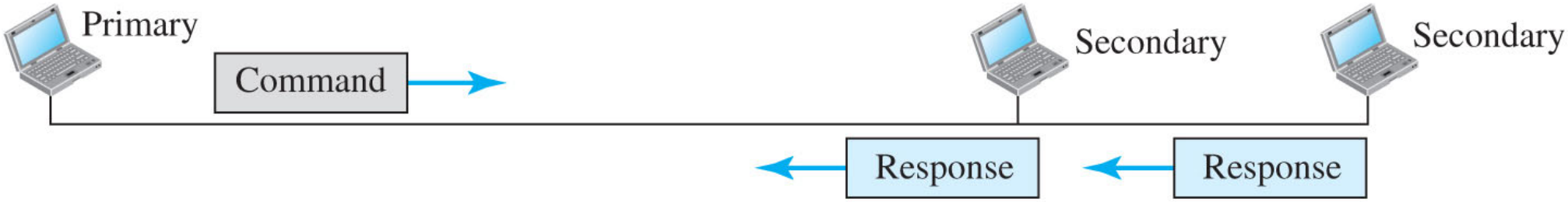
- All these ACK mechanisms are called Automatic Repeat reQuests – ARQs

- ARQ is a flow control mechanism for datagram, run in Transport Layer (TCP) and will be revisited later in the course

DLC Protocols: High-Level DLC

Normal Response Mode

a. Point-to-point



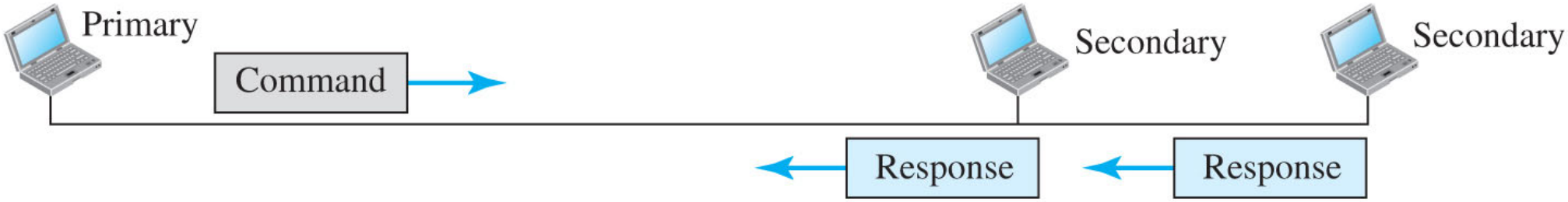
b. Multipoint

Asynchronous Balanced Mode

DLC Protocols: High-Level DLC

Normal Response Mode

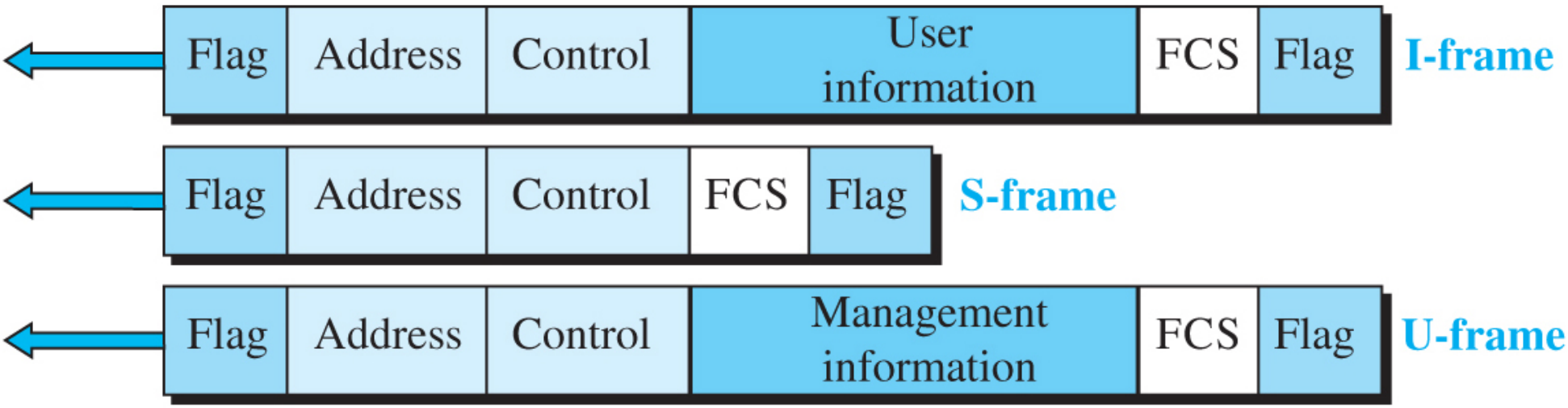
a. Point-to-point



b. Multipoint

Asynchronous Balanced Mode

DLC Protocols: HDLC Frame

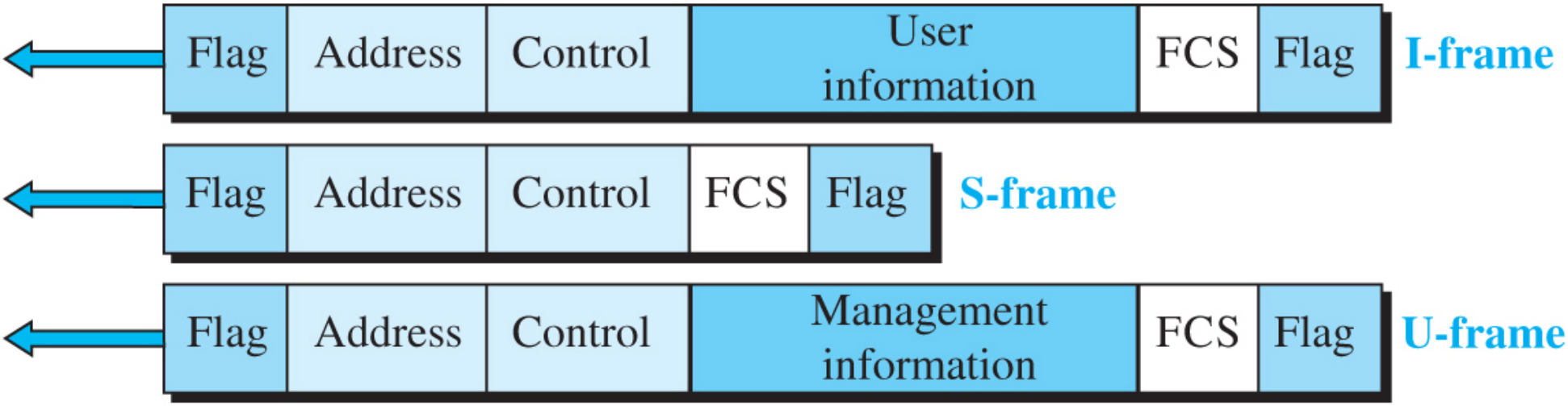


I-frame: for transporting user data and piggyback control information

S-frame: transport control information, used for flow and error control whenever piggybacking is impossible or inappropriate, such as when a station does not have data to send.

U-frame: link management, used for session management

DLC Protocols: HDLC Frame



Flag field: Synchronisation pattern

Address field: Destination address

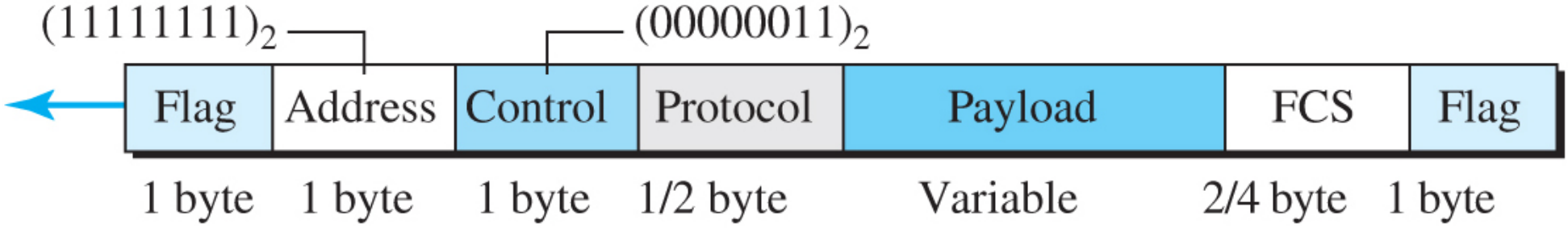
Control field: flow control by frame sequence identification

Information field: user data from higher layer

FCS field: CRC correction

DLC Protocols: PPP

Derived from DLC



Flag field: 1-byte start and end indicators 01111110

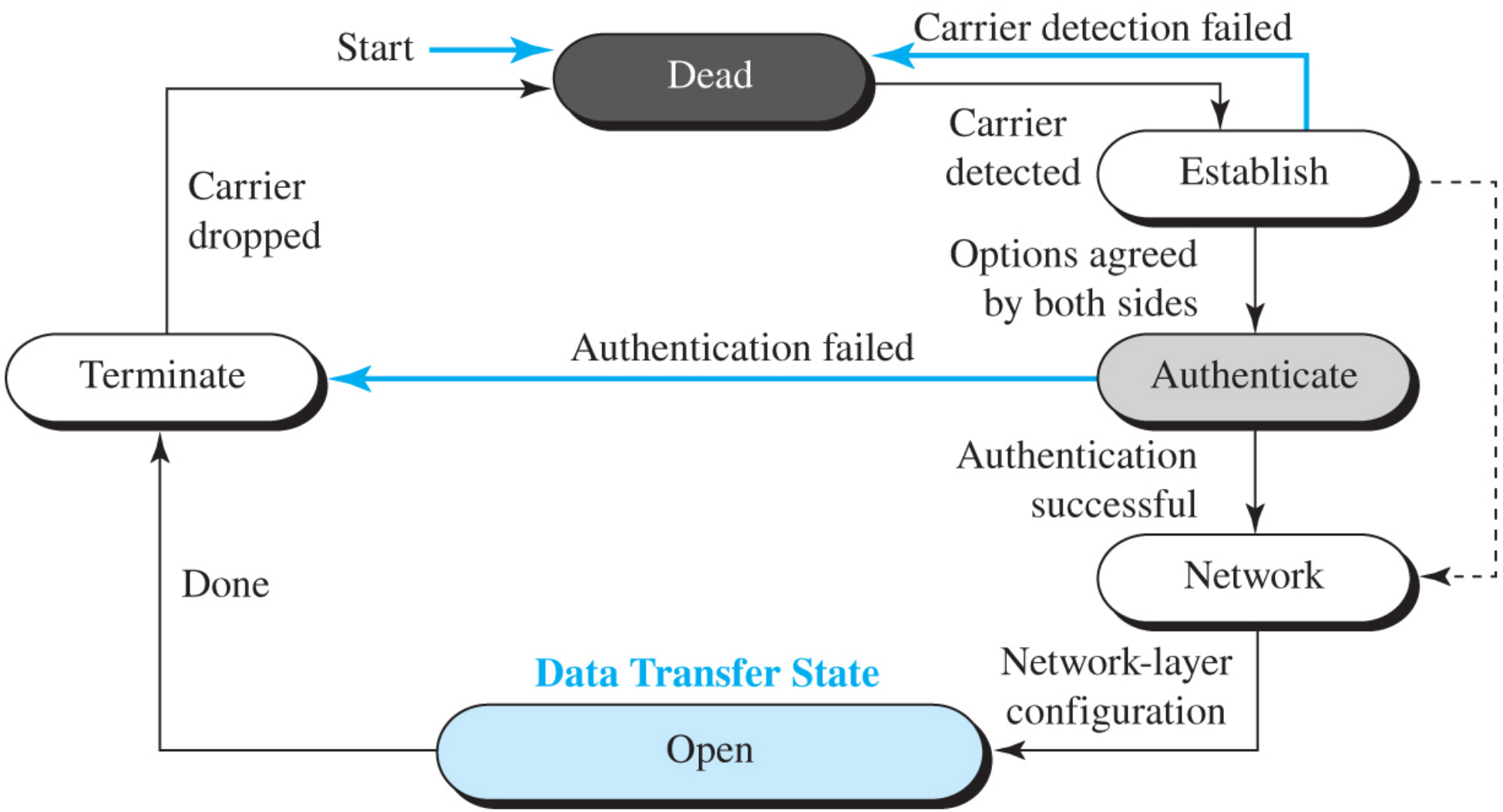
Address field: Broadcast address 11111111

Control field: constant values, no flow control

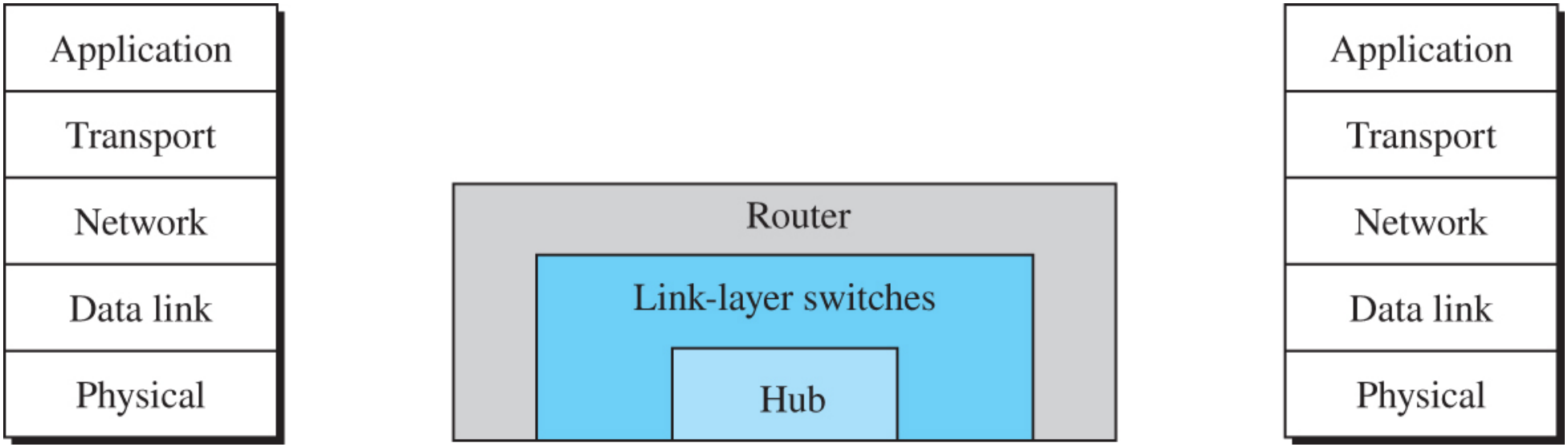
Protocol field: defined what is carried, user data or other info

FCS field: 2-byte or 4-byte CRC correction

DLC Protocols: PPP



Connecting Devices



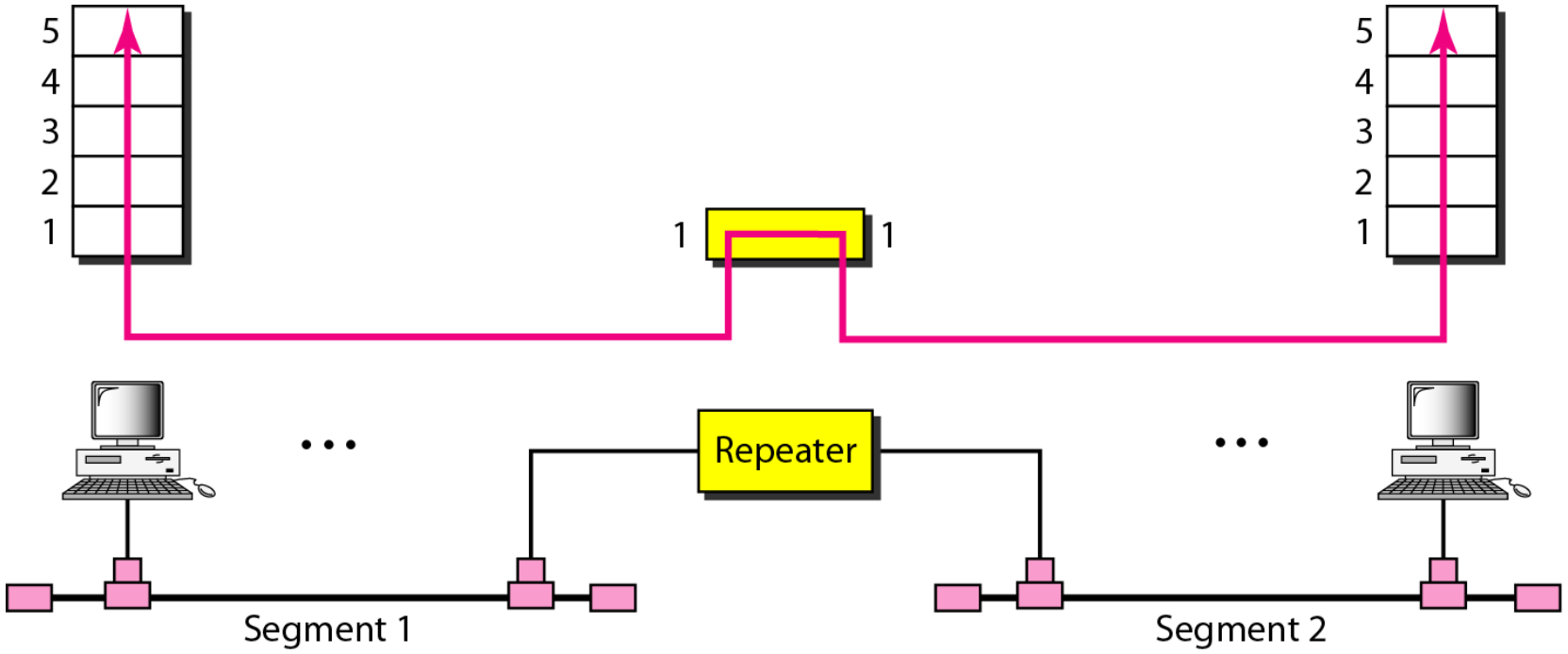
THE UNIVERSITY OF SYDNEY

Passive Hub

- A physical layer device that is used to provide connections to communication devices
- Is equivalent to a bus. However much more convenient for attaching a device
- Operation
  - The source NIC sends a frame
  - The hub receives the frame
  - The hub broadcasts the frame across its internal bus to all other NICs
  - The hub is the collision point

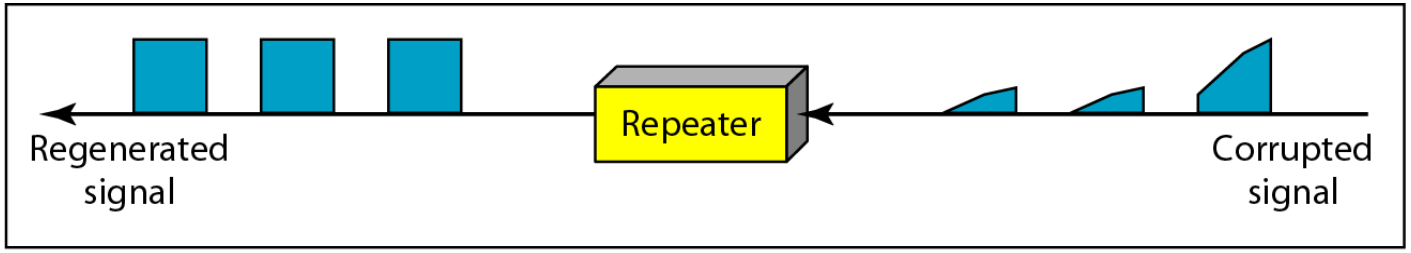
Repeater

- A repeater connects two segments of a LAN.
- A repeater forwards every frame; it has no filtering capability.

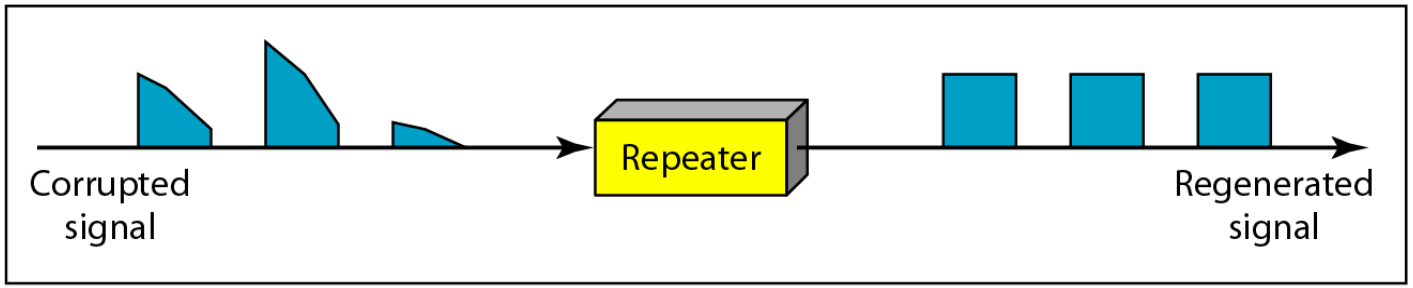


Function of a repeater

- Difference between an amplifier and a repeater
  - An amplifier cannot discriminate between intended signal and noise; it amplifies everything fed into it
  - A repeater does not amplify the signal, it regenerates the signal
    - Noise is eliminated during this regeneration



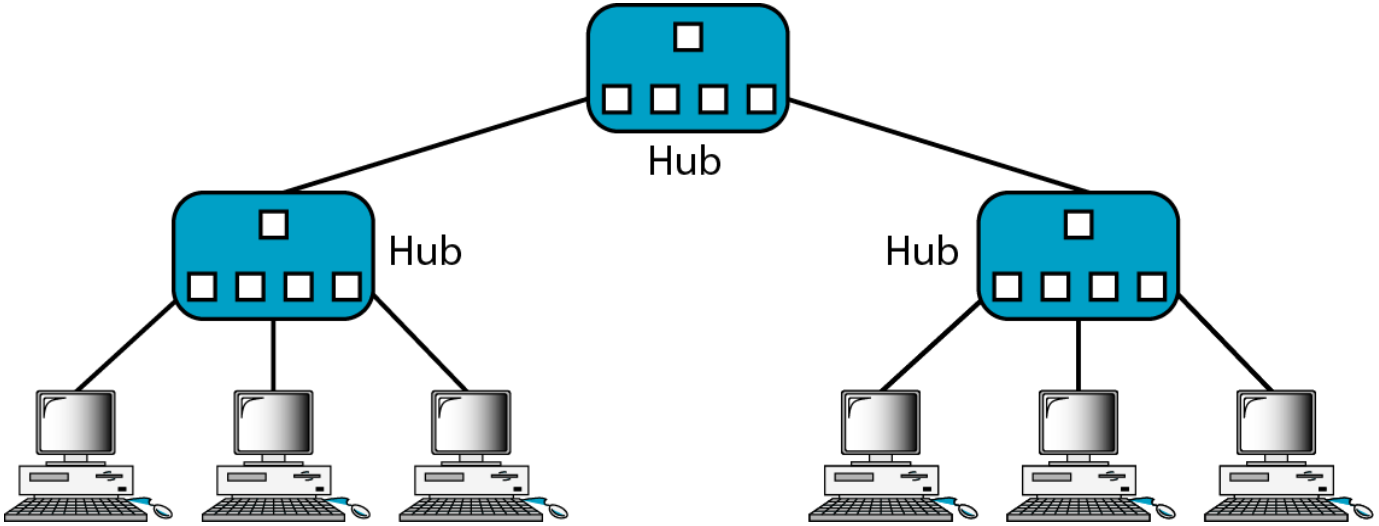
a. Right-to-left transmission.



b. Left-to-right transmission.

Active Hub

- An active Hub is a multipoint repeater.
- Normally used to create connections between stations in a physical star topology.
- Hubs can also be used to create multiple levels of hierarchy (e.g., 10BaseT).



THE UNIVERSITY OF SYDNEY

Bridges

- They operate at both the Physical and Data Link layers.
- As a physical layer device it provides the services of a repeater
- As a data link layer device it can check the source/destination physical (MAC Address) contained in the frame and forward.
- A bridge has a table used for filtering forwarding decisions.

Transparent Bridge

- When a frame arrives, a bridge must decide whether to discard or forward it.
- If to forward, then on which LAN to put the frame.
- This decision is made by looking up the destination address in a table inside the bridge.
- When the bridge is first plugged in, all the tables are empty.
- First every incoming frame for an unknown destination is flooded to all destinations connected except for the one it arrived from.

Transparent Bridge

- As time goes on the bridges know where their destinations by using an backward learning algorithm.
- Once a destination is known, frames destined for it are put on only the proper LAN and are not flooded.
- Bridging Procedure
  - If destination and source LANs are the same, discard the frame.
  - If destination and source LANs are different, forward the frame.
  - If the destination LAN is unknown, use flooding.

Bridge connecting two LANs

Gradual building of table

Address	Port

a. Original

Address	Port
71:2B:13:45:61:41	1

b. After A sends a frame to D

I know A is connected to Port 1 but NOT D

Address	Port
71:2B:13:45:61:41	1
64:2B:13:45:61:13	4

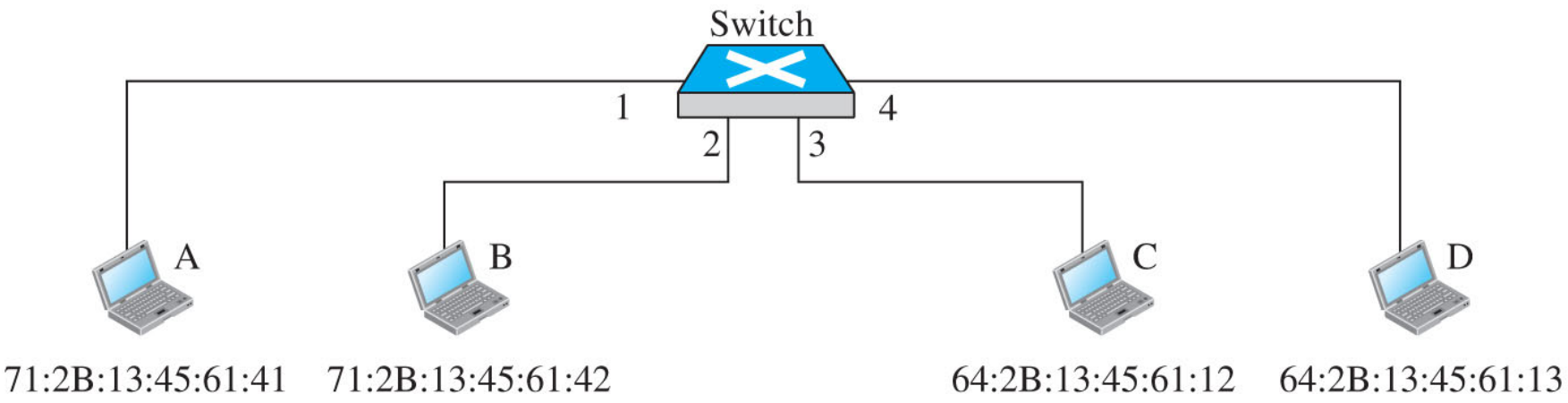
c. After D sends a frame to B

Address	Port
71:2B:13:45:61:41	1
64:2B:13:45:61:13	4
71:2B:13:45:61:42	2

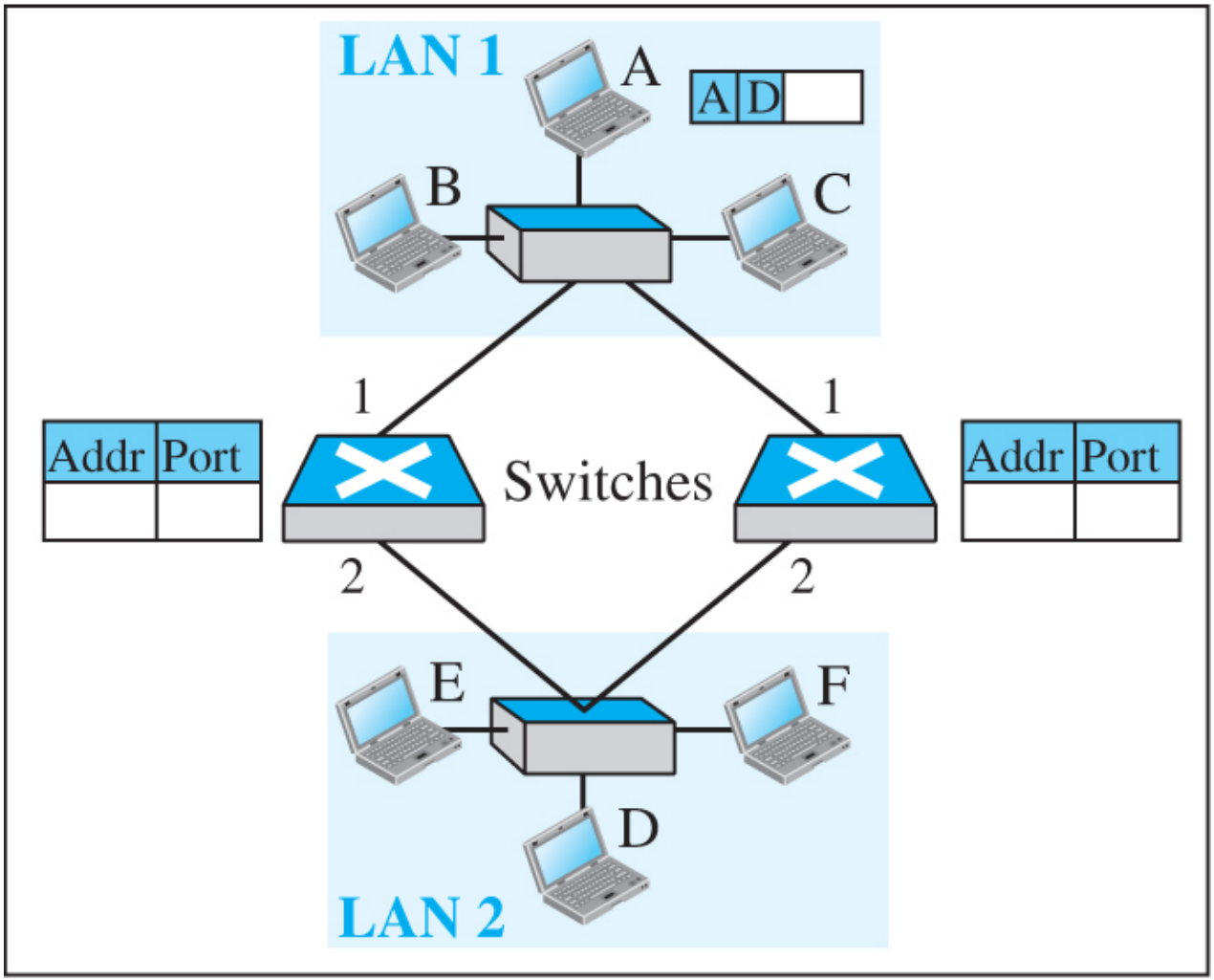
d. After B sends a frame to A

Address	Port
71:2B:13:45:61:41	1
64:2B:13:45:61:13	4
71:2B:13:45:61:42	2
64:2B:13:45:61:12	3

e. After C sends a frame to D

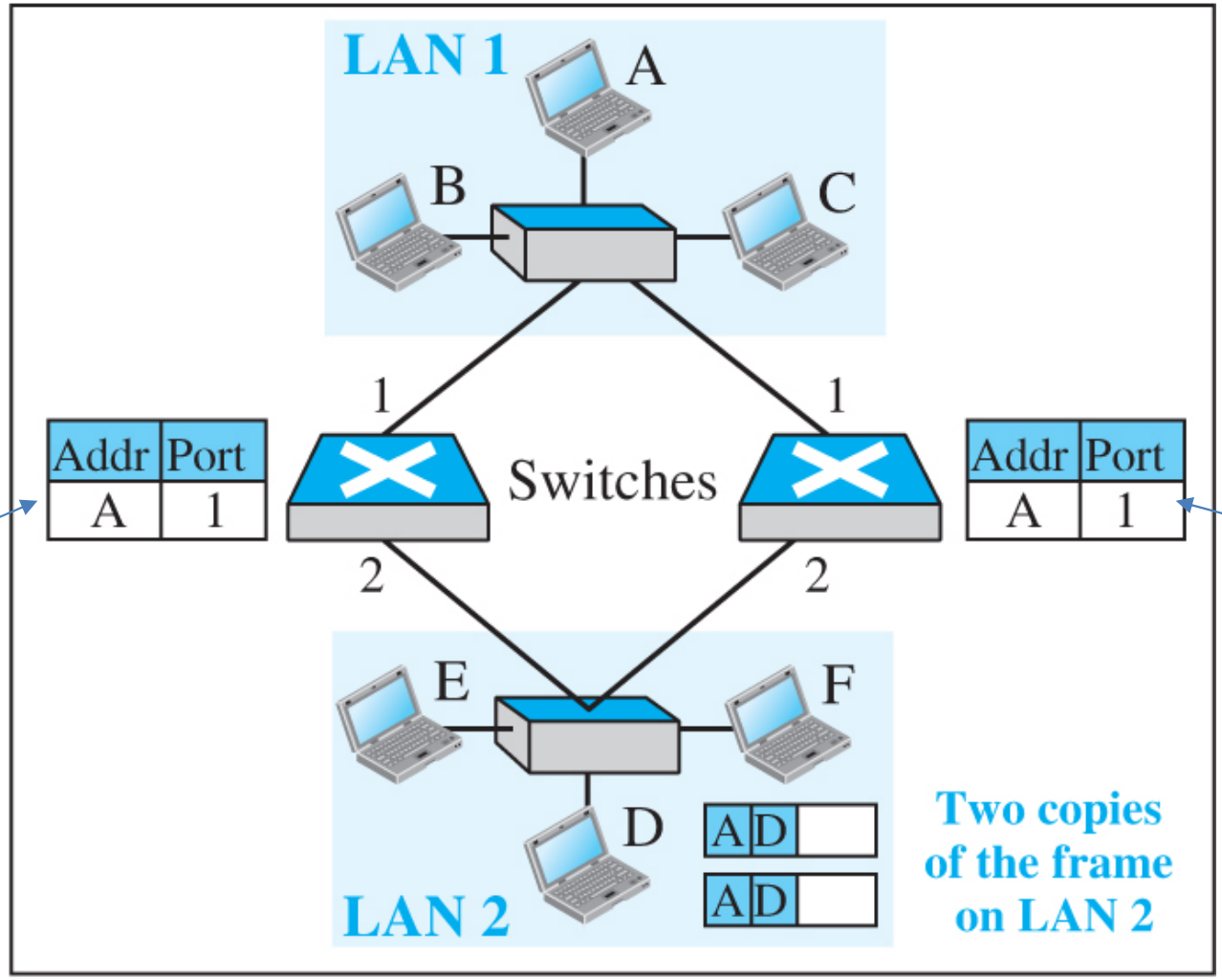


Loop Problem @ time t



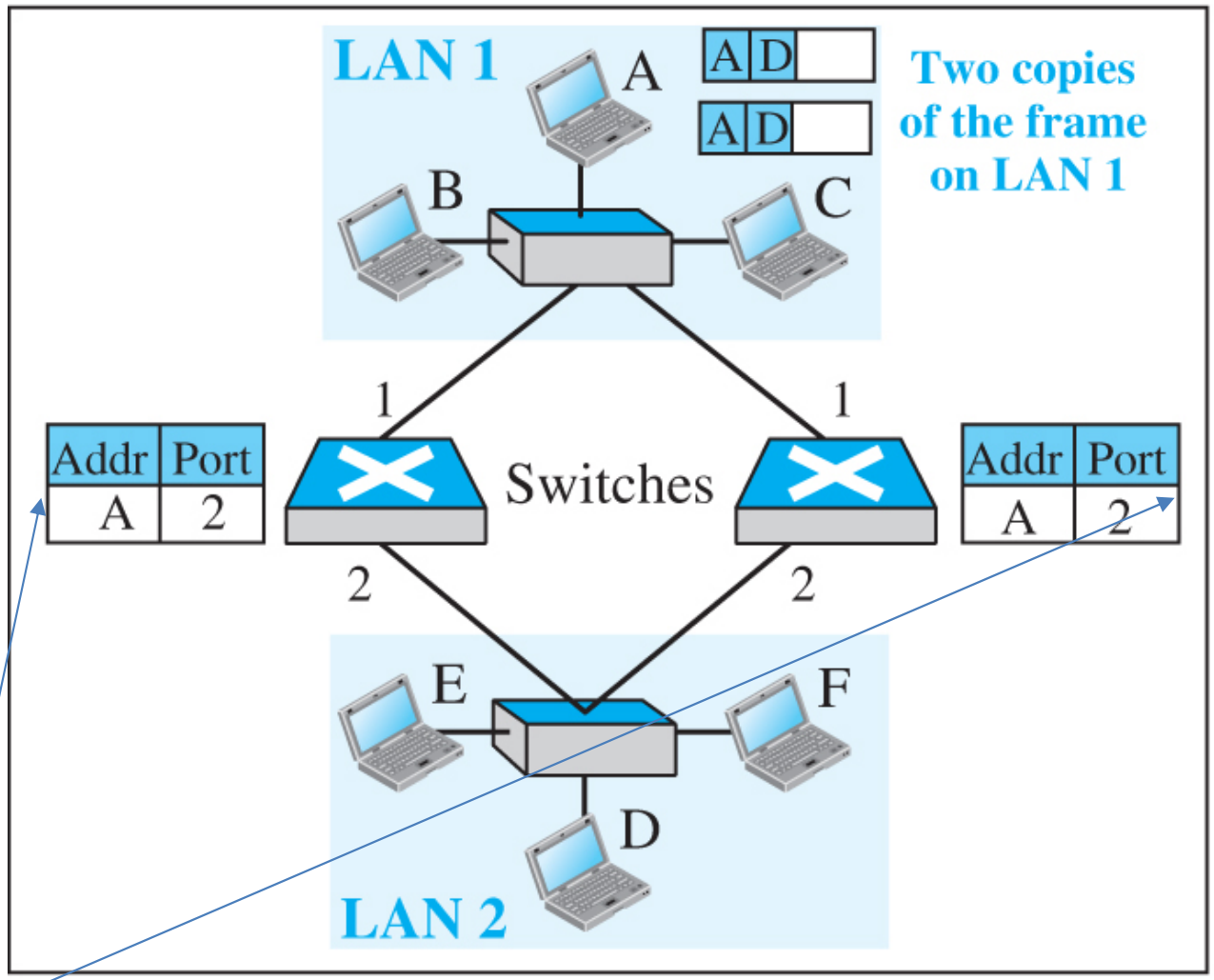
Station A sends a frame to station D

Loop Problem @ time t+1



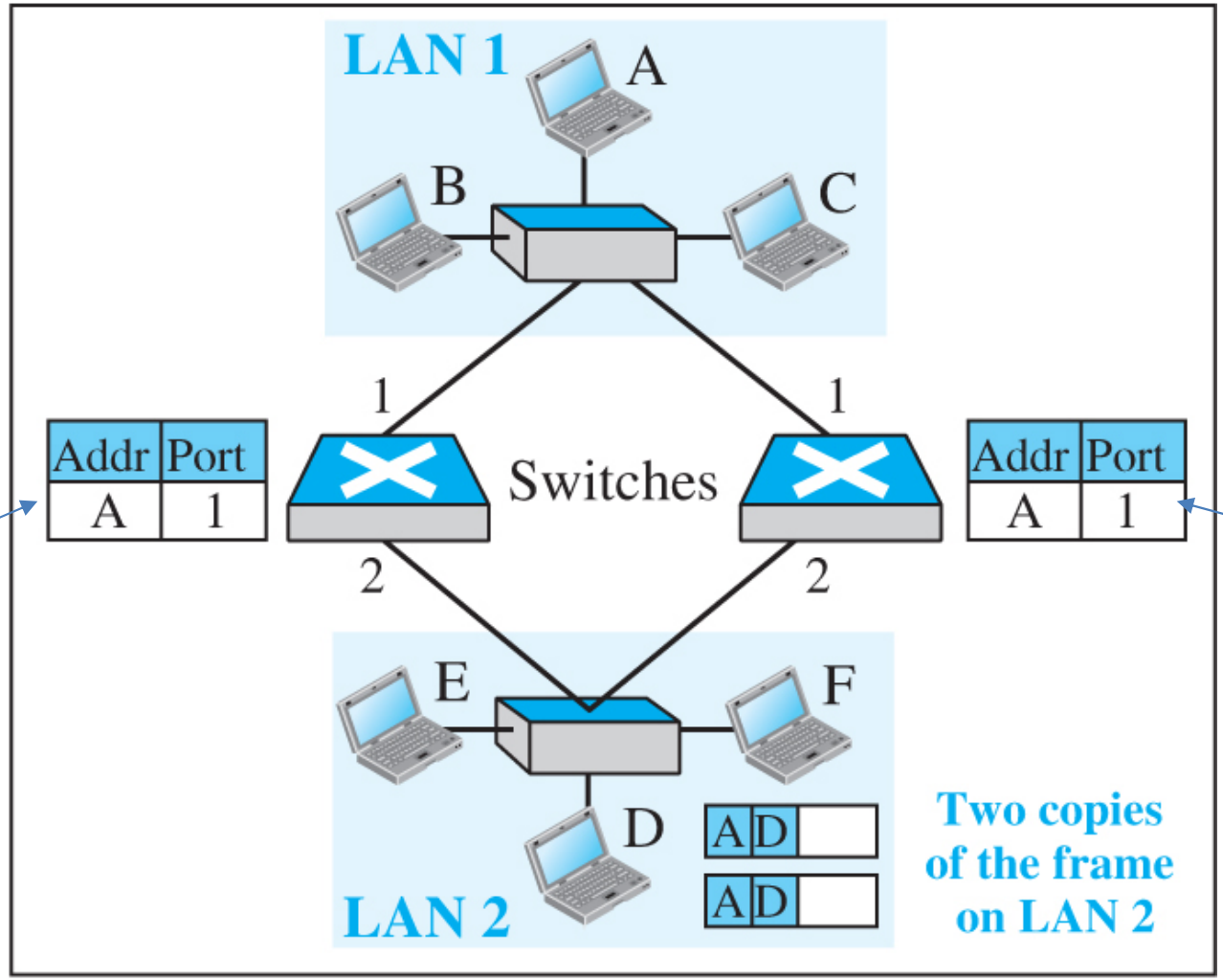
Both Switches simultaneously forward the frame (flooding)

Loop Problem @ time t+2



Both Switches simultaneously receive and forward the frame

Loop Problem @ time t+3



Both Switches simultaneously receive and forward the frame

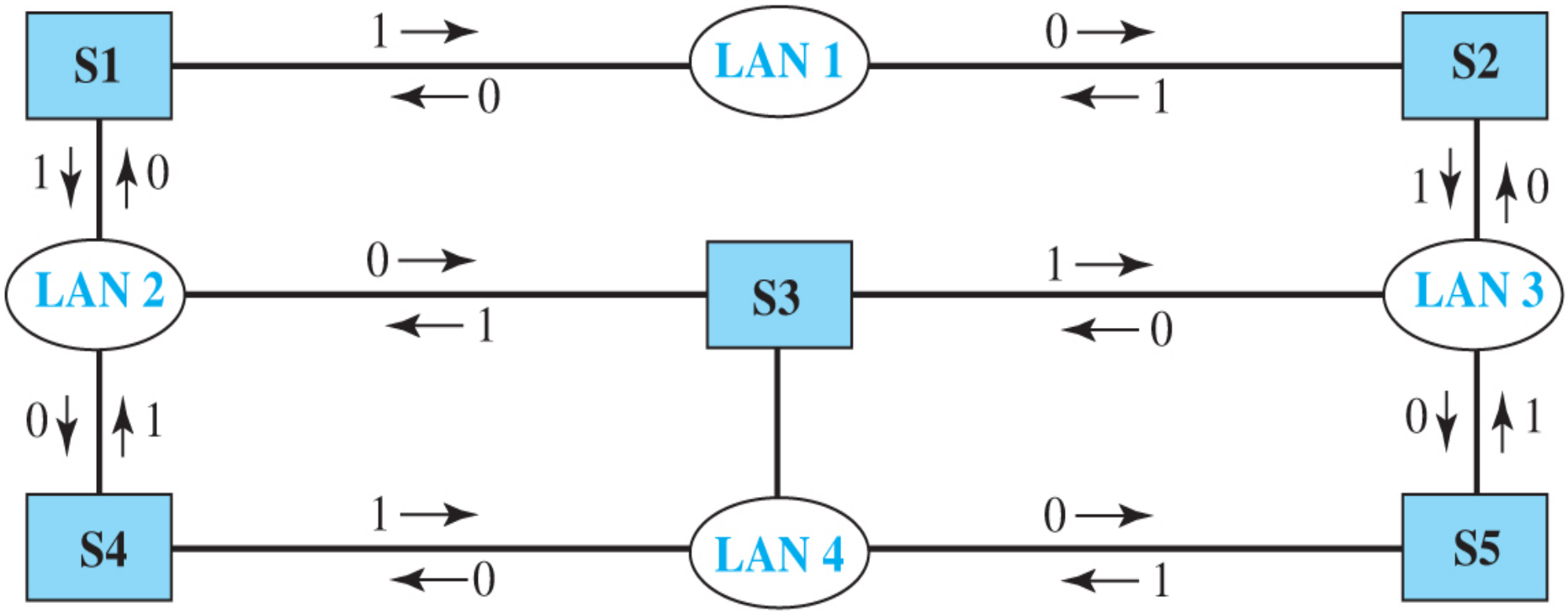
Spanning Tree Algorithm to deal with Loop

- When two or more bridges are used to increase reliability, this sometimes creates loops between LANs.
- Solution is to decide upon having exactly one path from every LAN to every other LAN.

How to Build a Spanning Tree:

- First the bridges have to choose a bridge to be the root of the tree (Usually the one with the lowest serial number)
- Next, a tree of shortest path and cost **from the root** to every bridge and LAN is constructed.
- If a bridge or LAN fails, a new one is computed.
- The result of this technology is that a unique path is established from every LAN to the root, and thus to every other LAN → needs to be done by every bridges

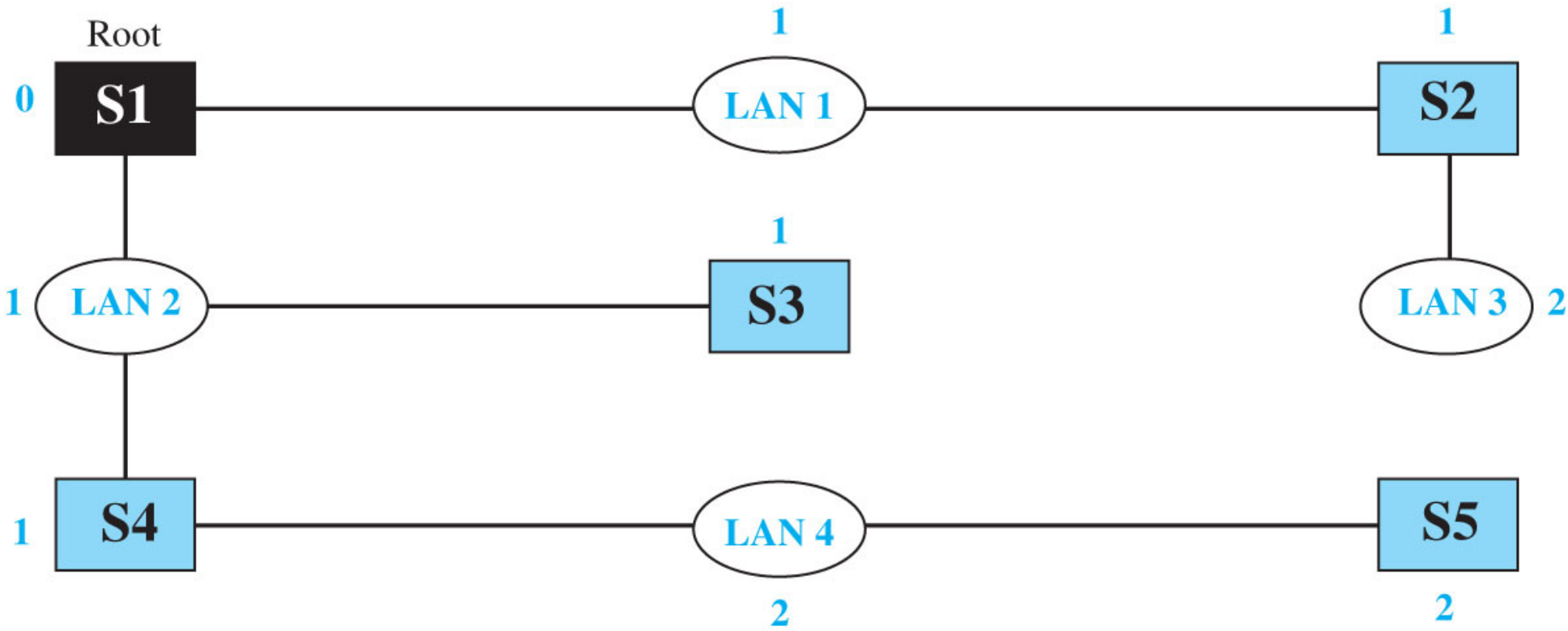
Spanning Tree Algorithm Example



b. Graph representation with cost assigned to each arc

The cost here is defined as minimum hops
e.g., Switch 1 (S1) to LAN 1 needs 1 hop
LAN 1 to Switch 1 (S1) needs 0 hop

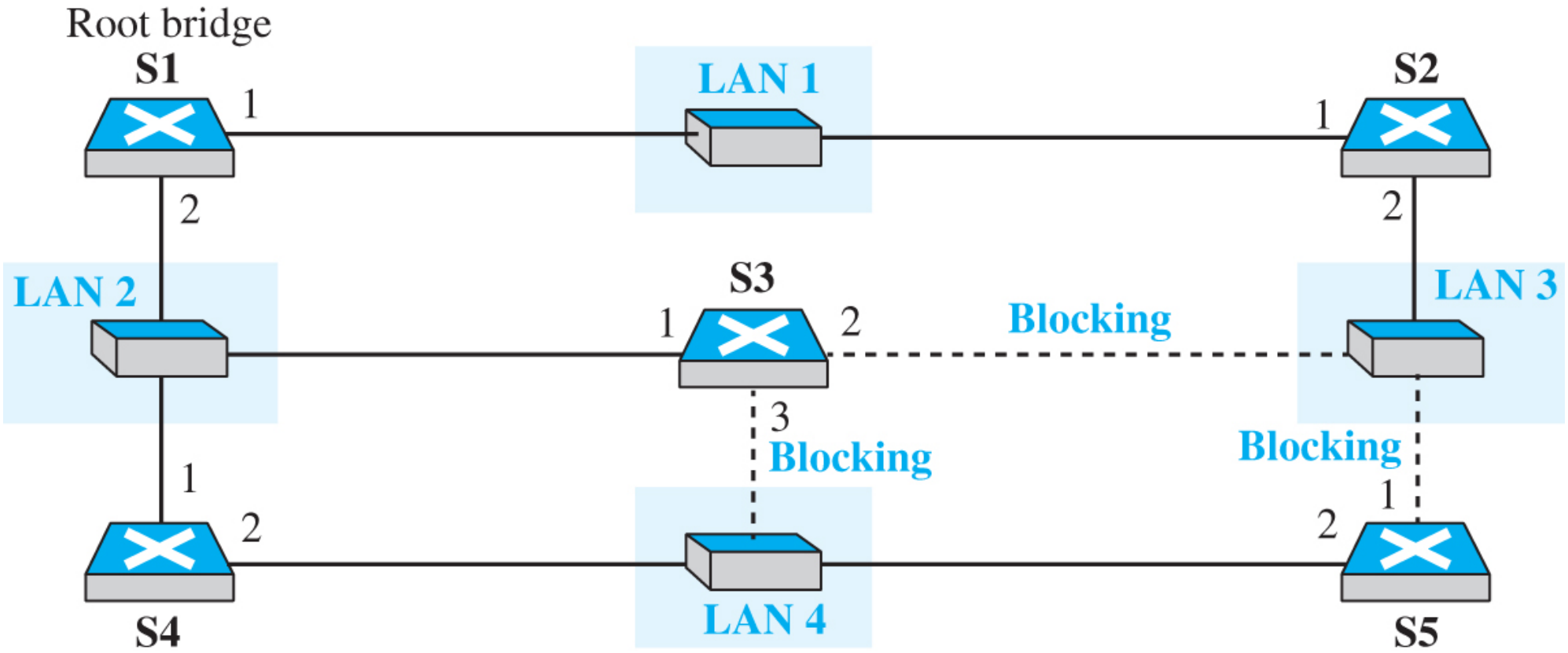
Spanning Tree Algorithm Example



Examine the total cost from the root switch to the destination to create the shortest tree

Spanning Tree Algorithm Example

Ports 2 and 3 of bridge S3 are blocking ports (no frame is sent out of these ports).
Port 1 of bridge S5 is also a blocking port (no frame is sent out of this port).



THE UNIVERSITY OF SYDNEY

Switches

- Device used to link several separate LANs are called switches.
- Switch have multiple ports and each of which can support a single end station or an entire LAN.
- With different LANs connected to each of the switch's ports, it switches packets between LANs as needed.
- Hence it acts as a ***very fast multiport bridge***.
- Packets are filtered by the switch based on the destination MAC address.
- Increases performance by providing each port with a dedicated bandwidth.
- Supports ***multiple transmissions simultaneously***.

THE UNIVERSITY OF SYDNEY

Switches

- In general a switch is a multi-port bridge.
- Deploying a dedicated LAN is a another advantage of using switches.
- Each port on an Ethernet switch supports a dedicated 10/100Mbps Ethernet LAN
- These LANs comprise multiple stations linked to a 10Base-T hub.
- But it is also possible to connect a single high-performance station, such as a server, to a switch port.

Types of Switches

Cut-Through Switches

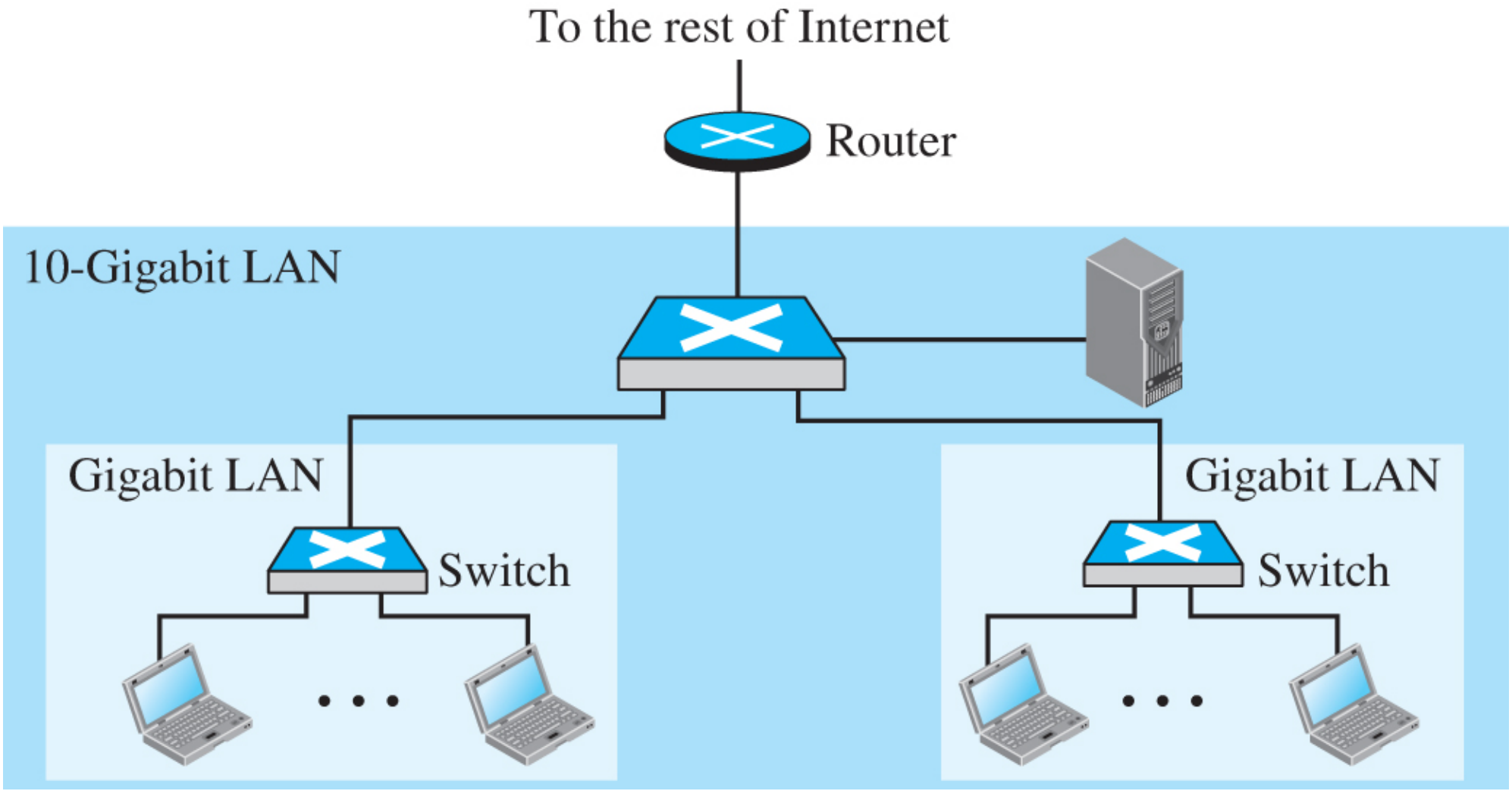
- Only the first few bites of the frame are read to obtain source and destination addresses.
- Then the frame is passed through to the destination segment without checking the rest of the packet for errors.
- Invalid packets are still passed through.
- Minimal delay and high throughput.

Types of Switches

Store-and-Forward Switch

- They examine the entire frame.
- Each incoming frame is buffered and examined.
- Filters any bad frames that it detects.
- Imposes a delay in frame throughput.

Routers (3 Layers Devices)



Routers (3 Layers Devices)

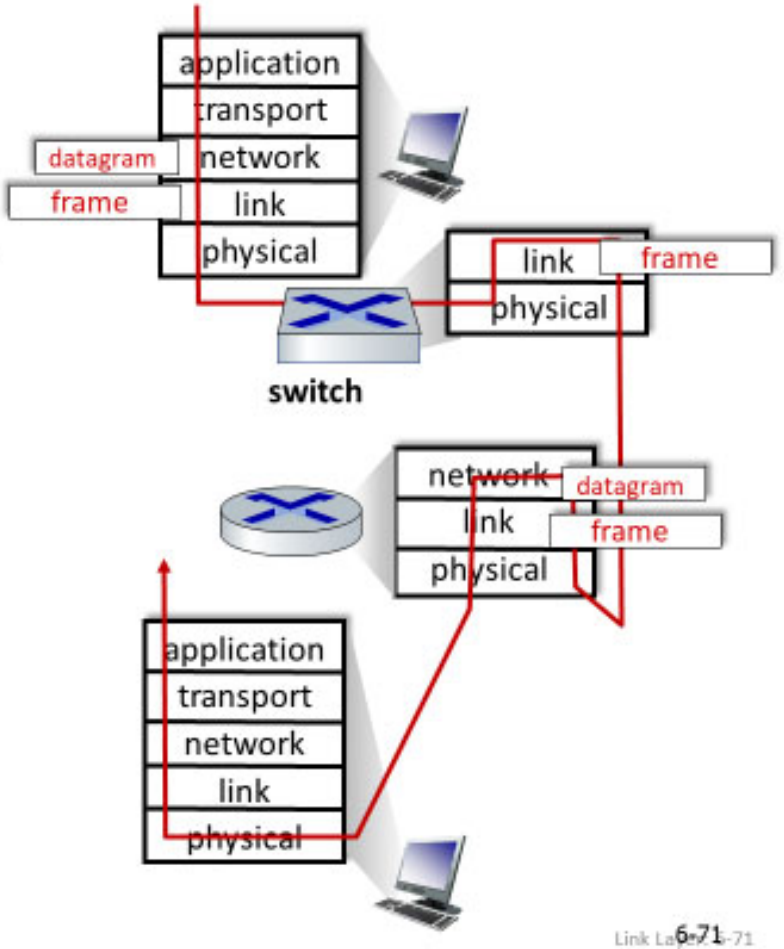
Switches vs. routers

both are store-and-forward:

- *routers*: network-layer devices (examine network-layer headers)
- *switches:* link-layer devices (examine link-layer headers)

both have forwarding tables:

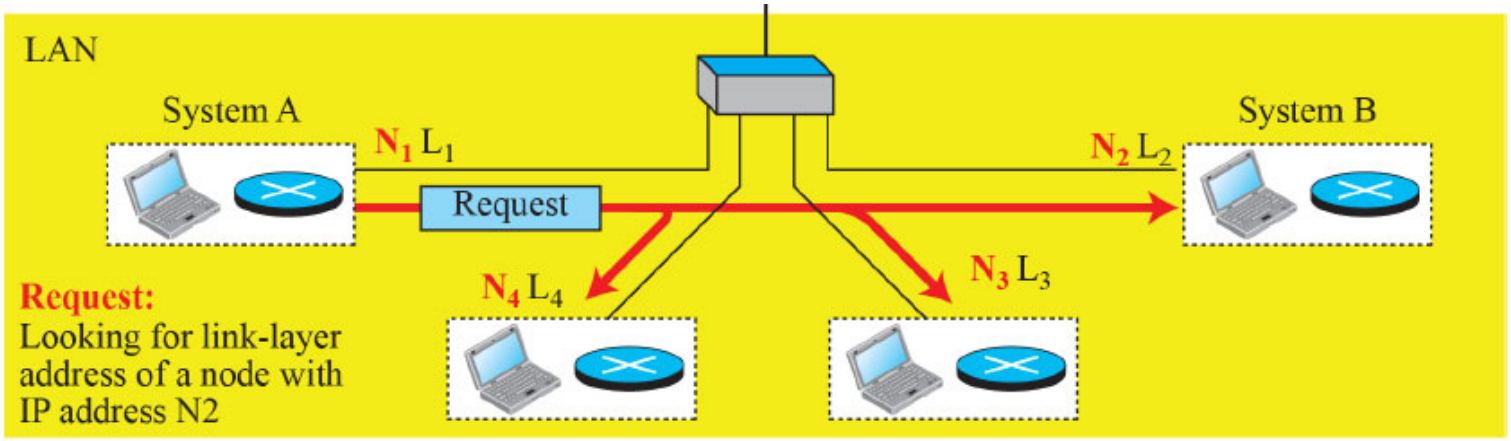
- *routers:* compute tables using routing algorithms, IP addresses
- *switches:* learn forwarding table using flooding, learning, MAC addresses



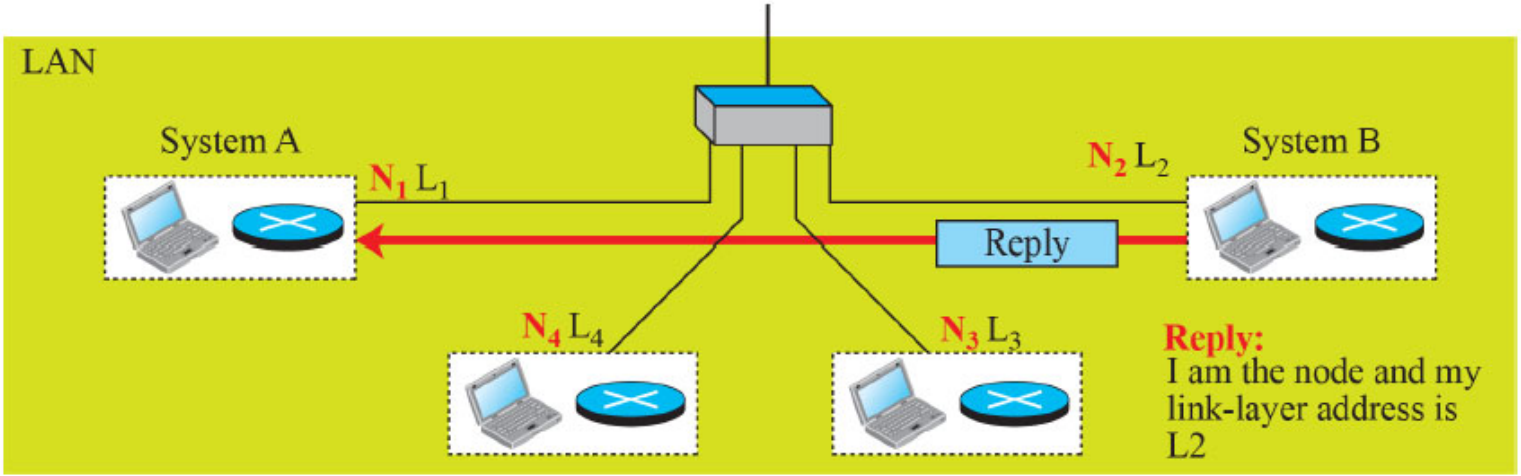
Address Resolution Protocol (ARP)

- ARP is used to find the physical address of a node when it's IP address is known
- A host/route needs to find the physical address of another host on the **same network**.
- It forms a broadcast ARP query packet that includes the IP address and broadcasts it over the network.
- Every host on the network receives and processes it, but only the recipient recognises it and sends back its physical address.

Address Resolution Protocol (ARP)



a. ARP request is broadcast



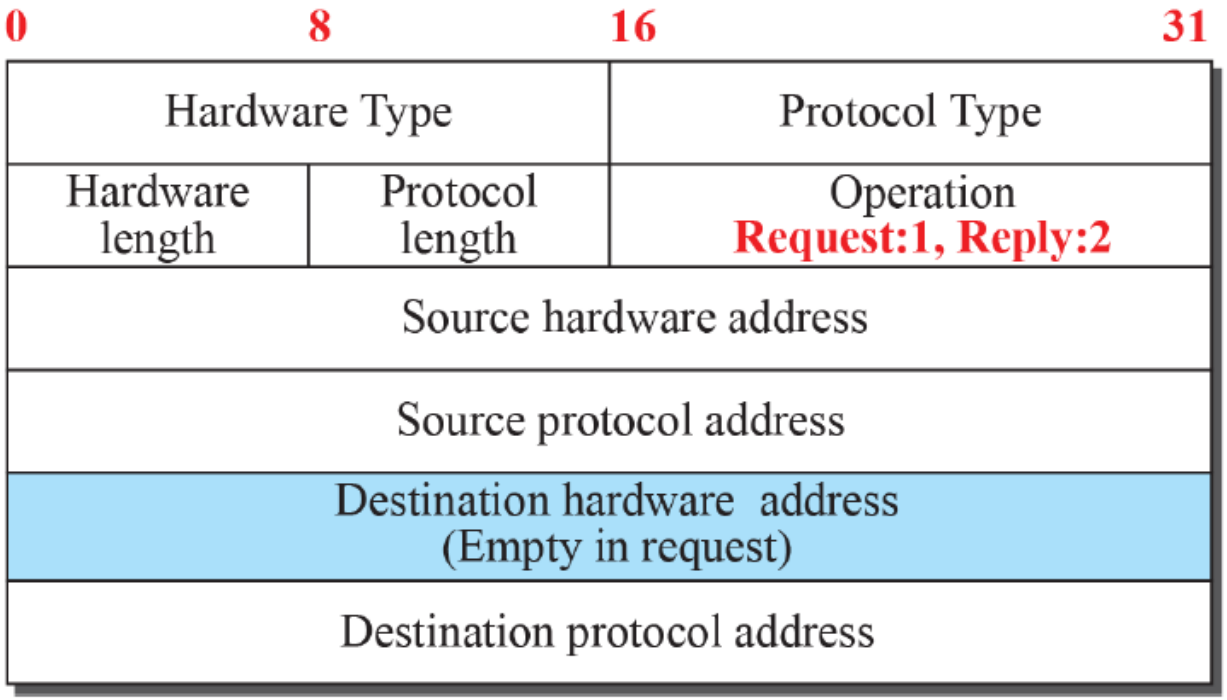
b. ARP reply is unicast

N is an IP Address and L is a Physical/MAC Address

ARP packet

The packet is encapsulated directly into a data-link frame:

Hardware: LAN or WAN protocol
Protocol: Network-layer protocol



ARP Example

```
> Frame 18: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
> Ethernet II, Src: HonHaiPr_c6:96:9d (08:3e:8e:c6:96:9d), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: HonHaiPr_c6:96:9d (08:3e:8e:c6:96:9d)
    Sender IP address: 10.16.91.184
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 10.16.95.254
> Frame 19: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> Ethernet II, Src: CiscoInc_ff:fc:30 (00:08:e3:ff:fc:30), Dst: HonHaiPr_c6:96:9d (08:3e:8e:c6:96:9d)
v Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: CiscoInc_ff:fc:30 (00:08:e3:ff:fc:30)
    Sender IP address: 10.16.95.254
    Target MAC address: HonHaiPr_c6:96:9d (08:3e:8e:c6:96:9d)
    Target IP address: 10.16.91.184
```

Recommended Reading

- Behrouz A. Forouzan, Data Communications and Networking with TCP/IP Protocol Suite, 6th ed., 2022, Chapters 3 and 6

- J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach, 8th ed., 2022, Chapter 6