

COMP 3721

Introduction to Data Communications

07b - Week 7 - Part 2

Learning Outcomes

- By the end of this lecture, you will be able to
 - Explain the DLC (Data-Link Control) services.
 - Explain DLC protocols and how they function.

Introduction

- The **data link control (DLC)** deals with procedures for communication between two adjacent nodes (**node-to-node communication**).
- No matter whether the link is dedicated or broadcast.
- We discussed that data link control functions include framing and flow and error control.
- The other sublayer in data-link layer is MAC.
- DLC functions/services:
 - **Framing**
 - **Flow control**
 - **Error control**

Framing Analogy: Postal System

- The data-link layer needs to pack bits into **frames**, so that each frame is distinguishable from another.

Framing Analogy: Postal System

- The data-link layer needs to pack bits into **frames**, so that each frame is distinguishable from another.
- The message is usually packed into **multiple frames, why?**

Framing Analogy: Postal System

- The data-link layer needs to pack bits into **frames**, so that each frame is distinguishable from another.
- The message is usually packed into **multiple frames, why?**
 - Inefficient flow and error control if the frame is very large → a single-bit error requires retransmission of the whole frame

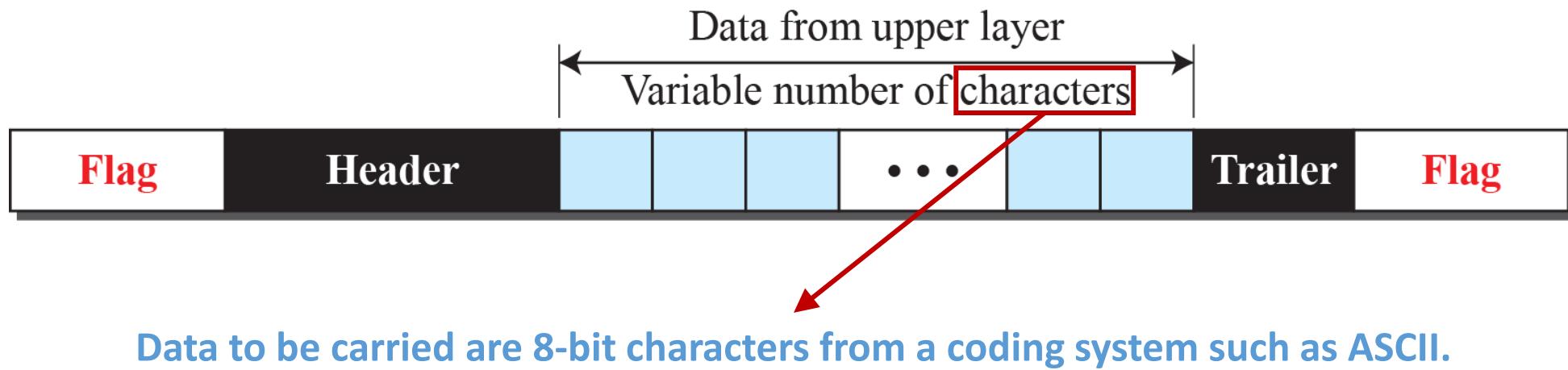
Framing Analogy: Postal System

- The data-link layer needs to pack bits into **frames**, so that each frame is distinguishable from another.
- The message is usually packed into **multiple frames, why?**
 - Inefficient flow and error control if the frame is very large → a single-bit error requires retransmission of the whole frame
- Framing in the data-link layer separates a message from one source to a destination by adding a **sender address** and a **destination address**.

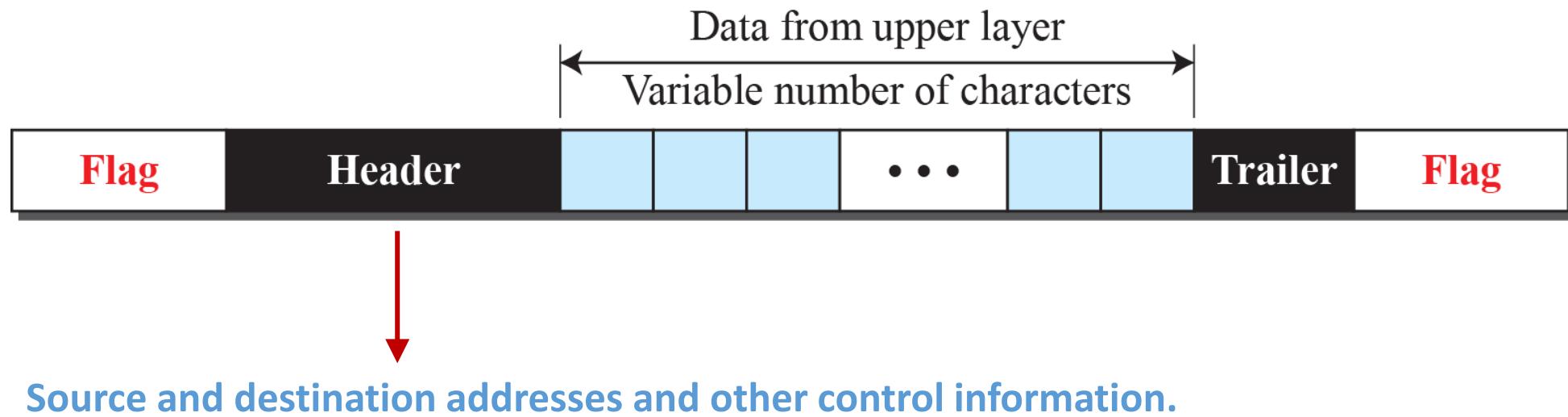
Frame Size

- **Frame size**
 - **Fixed** → no need to define boundaries, frame size is fixed!
 - **Variable** → prevalent in LANs (we need a way to define the end of a frame and the beginning of the next)
- To define the end of the frame we have two approaches:
 1. **Character (byte)-oriented approach**
 2. **Bit-oriented approach**

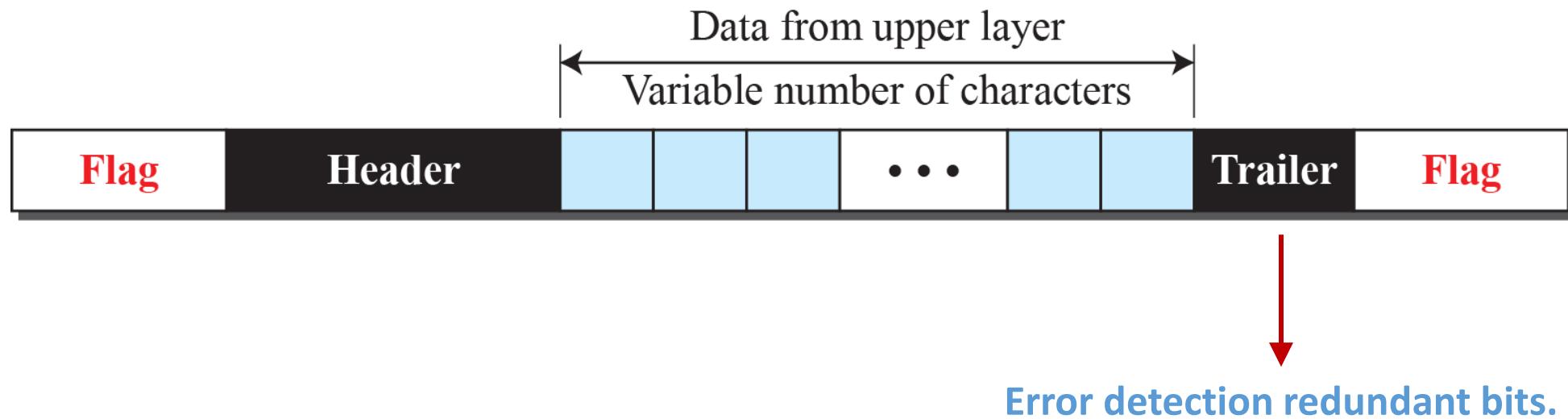
Character-Oriented Approach – Format of a Frame



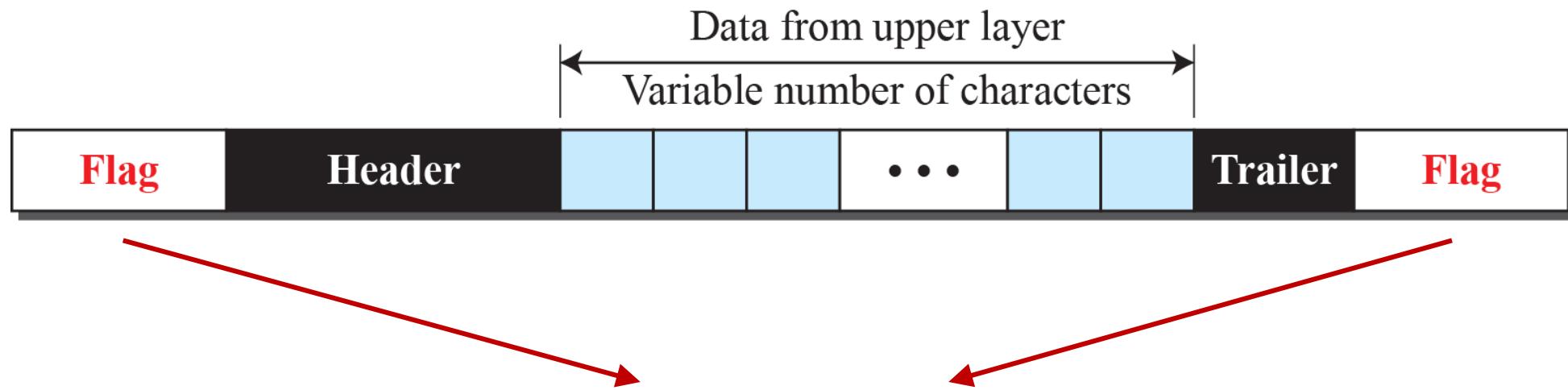
Character-Oriented Approach – Format of a Frame



Character-Oriented Approach – Format of a Frame



Character-Oriented Approach – Format of a Frame



To separate one frame from the next. It is composed of protocol-dependent special characters (8-bit (1-byte)).

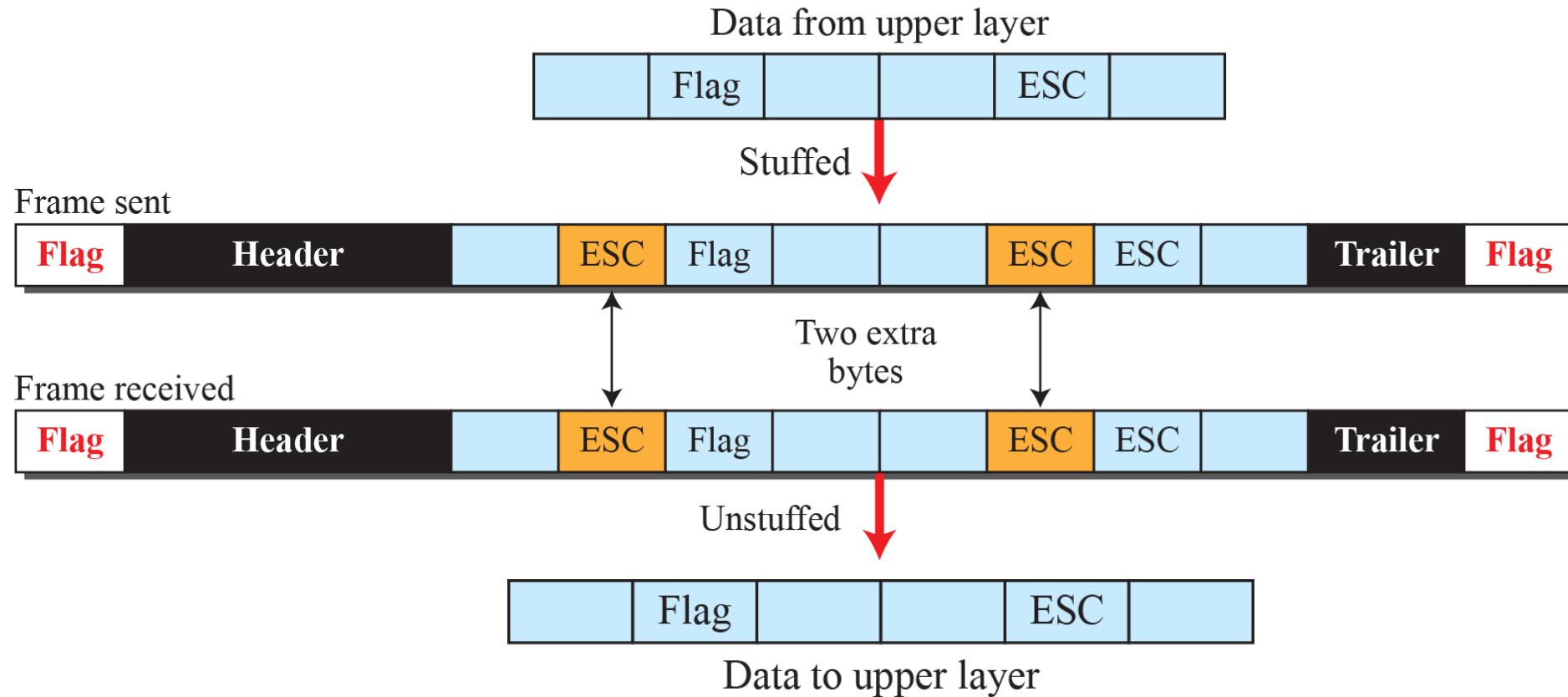
Character-Oriented Approach – Problem

- Character-oriented framing was popular when only **text** was exchanged.
- In addition to text, we send other types of information such as graphs, audio, and video; **any character used for the flag could also be part of the information**. If this happens, the receiver, when it encounters this pattern in the middle of the data, **thinks** it has reached the end of the frame.

Character-Oriented Approach – Byte Stuffing and Unstuffing

- **Byte stuffing (character stuffing)** is the process of adding one extra byte (with a predefined bit pattern), which is called the escape character (ESC), whenever there is a flag or ESC in the text.
- Whenever the receiver encounters the **ESC character**, it removes it from the data section and treats the **next character** as **data**, not as a delimiting flag.

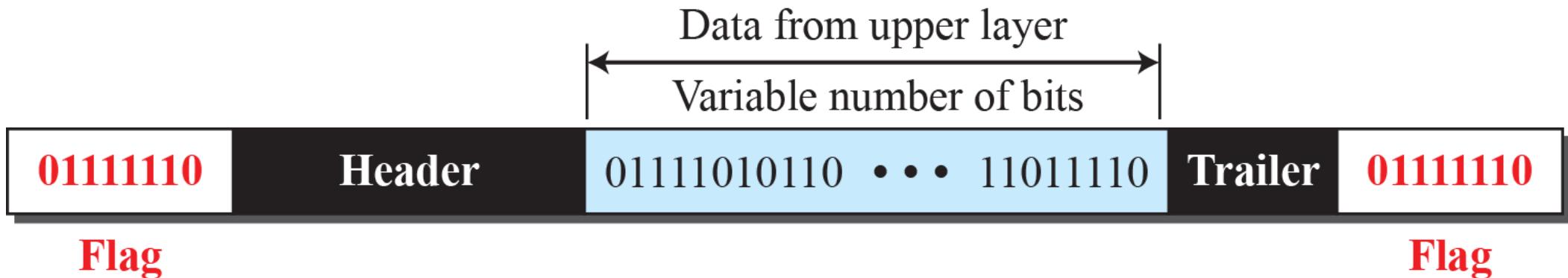
Byte Stuffing and Unstuffing



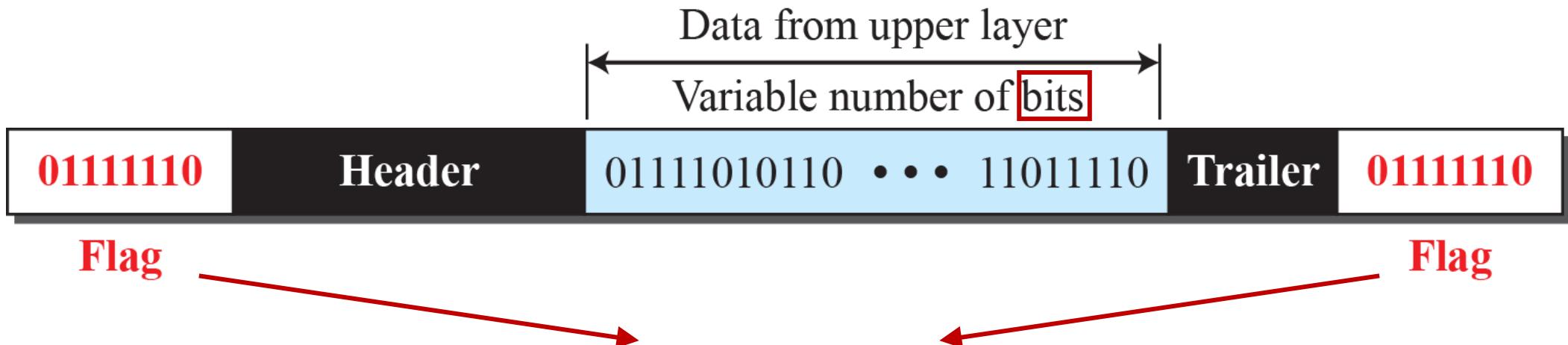
Character-Oriented Approach – The Other Problem!

- The universal coding systems in use today, such as Unicode, have 16-bit and 32-bit. Characters that conflict with 8-bit characters.
- In general, the tendency is moving toward the **bit-oriented** protocols.

Bit-Oriented Approach



Bit-Oriented Approach



A special 8-bit pattern as the delimiter to define the beginning and the end of the frame – **01111110** is the common flag between most protocols.

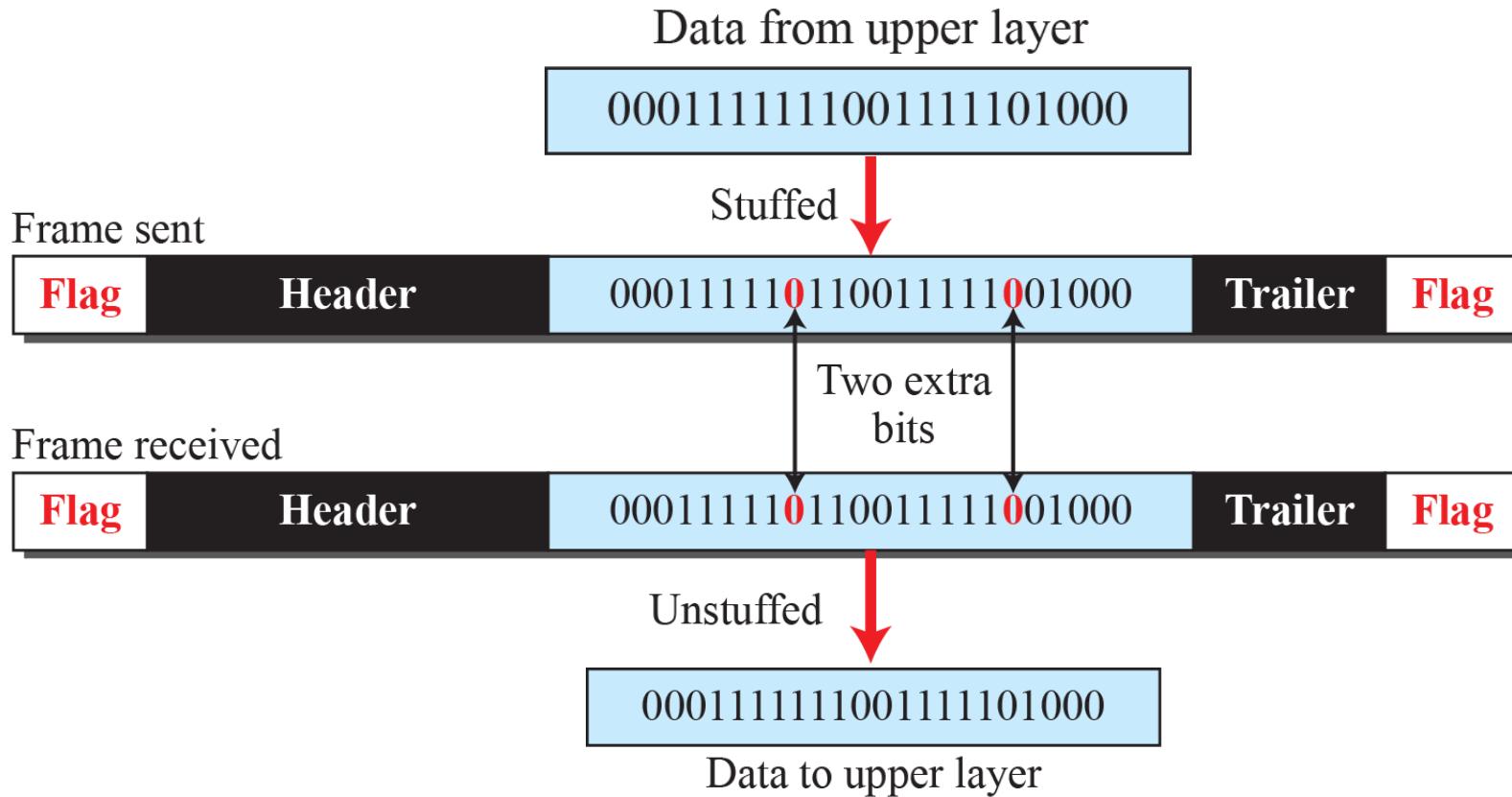
Bit-Oriented Approach – Bit Stuffing and Unstuffing

- **Bit stuffing** is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistake the pattern **01111110** for a flag.

Bit-Oriented Approach – Bit Stuffing and Unstuffing

- **Bit stuffing** is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistake the pattern **01111110** for a flag.

Bit-Oriented Approach – Bit Stuffing and Unstuffing

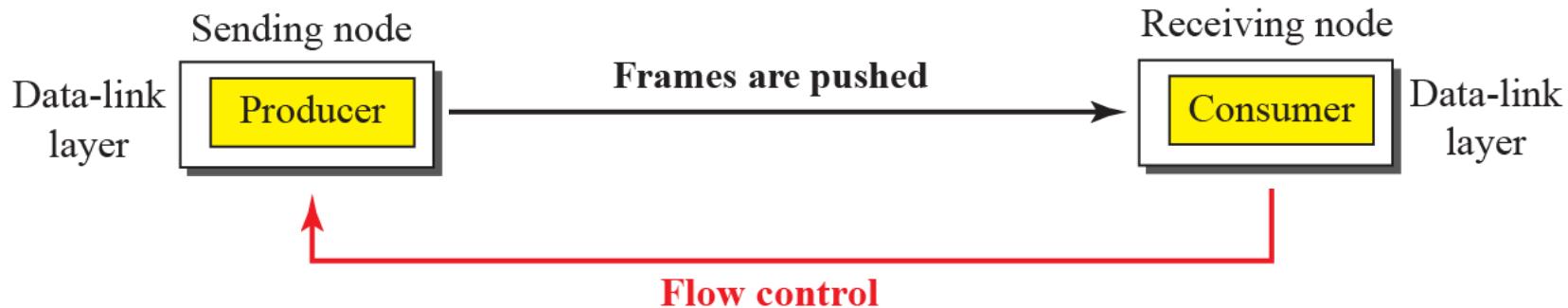


Flow Control

- Whenever an entity produces items and another entity consumes them, there should be a **balance** between **production** and **consumption rates**.
- If the **rate** of **produced** frames is higher than the **rate** of **consumed** frames, frames at the **receiving end** need to be **buffered** while waiting to be consumed (processed).
- A **buffer** is a set of memory locations that can hold packets at the sender and receiver.

Flow Control (Cont.)

- Different strategies to implement flow control
 1. The receiving data-link layer **drops the frames** if its buffer is full.
 2. Receiving data-link layer sends a **feedback** to the sending data-link layer to ask it to stop or slow down.



Error Control

- A CRC is added to the frame header by the sender and checked by the receiver.
- Two approaches
 1. Corrupted frames are **discarded**, and the uncorrupted ones are delivered to the network layer.
 - Mostly used in wired LANs such as Ethernet
 2. Corrupted frames are **discarded**, and an **acknowledgment** is sent (for the purpose of both flow and error control) to the sender for the **uncorrupted frames**.

Combination of Flow and Error Control

- The **acknowledgment** that is sent for **flow control** can also be used for **error control** to tell the sender the packet has arrived **uncorrupted**.
- The lack of acknowledgment means that there is a problem in the sent frame.

Data-Link Layer Protocols

- Two most commonly used DLC protocols to deal with flow and error control
 - 1. Simple
 - 2. Stop-and-Wait
- The behavior of a data-link-layer protocol can be better shown as a finite state machine (FSM).

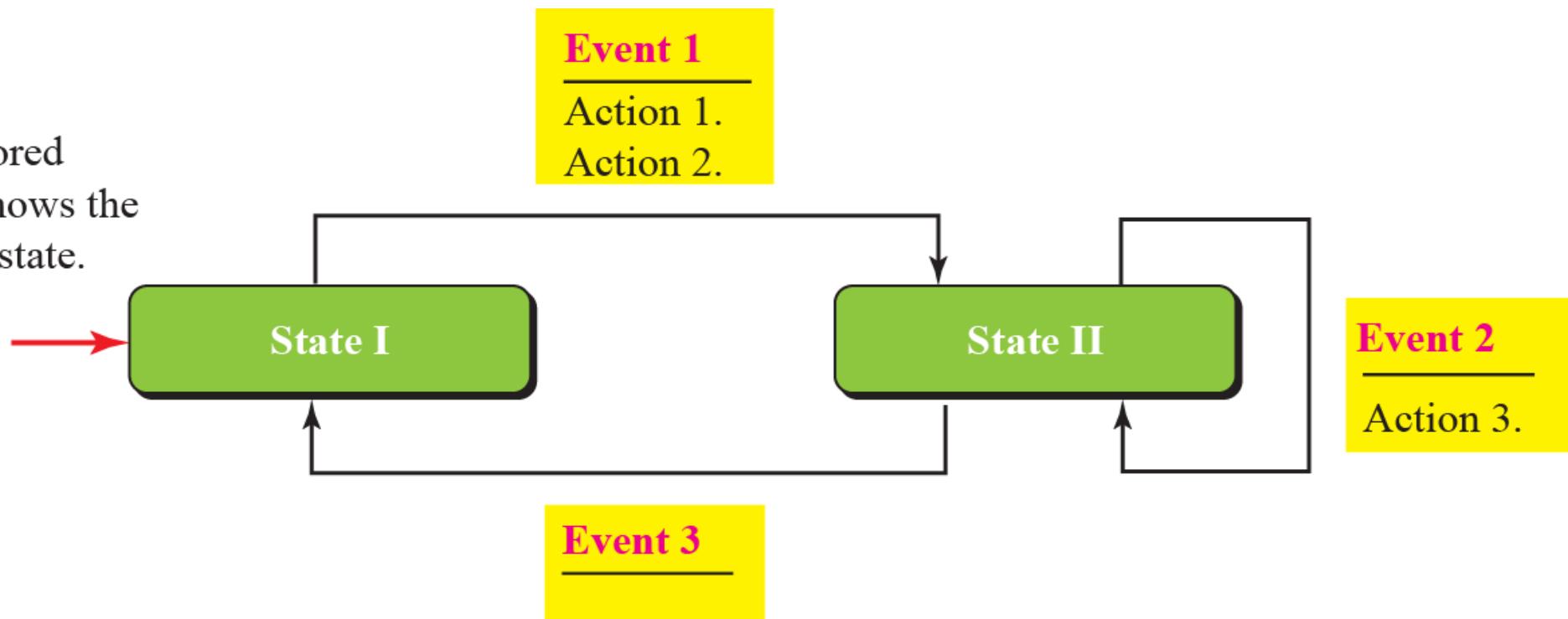
Finite State Machines (FSMs)

- FSM is used to simulate sequential logic
- An FSM is thought of as a machine (system) with a finite number of states.
- The machine is always in one of the states until an event occurs.
 - **Rounded-corner rectangles** or **circles** to show **states**.
 - **Colored text** to show **events**.
 - **Regular black text** to show **actions**.

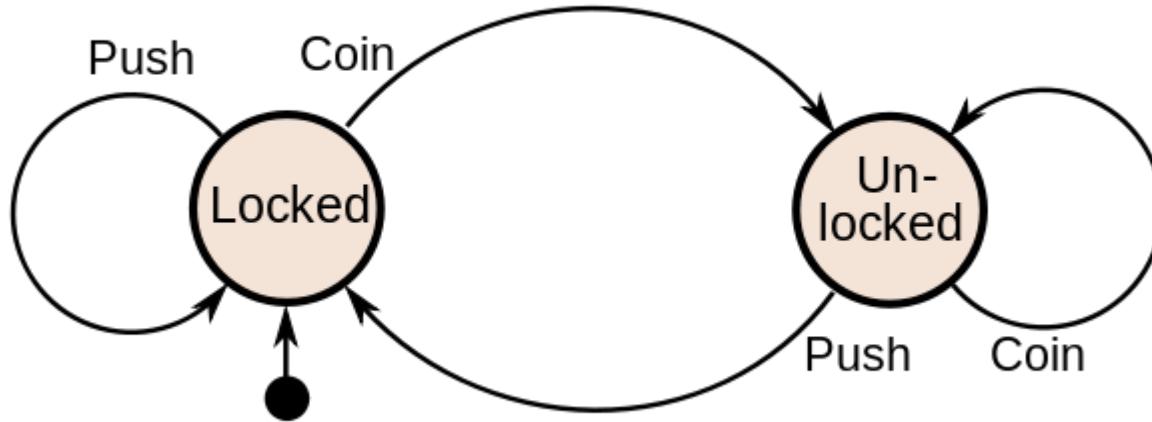
Finite State Machines (FSMs)

Note:

The colored
arrow shows the
starting state.



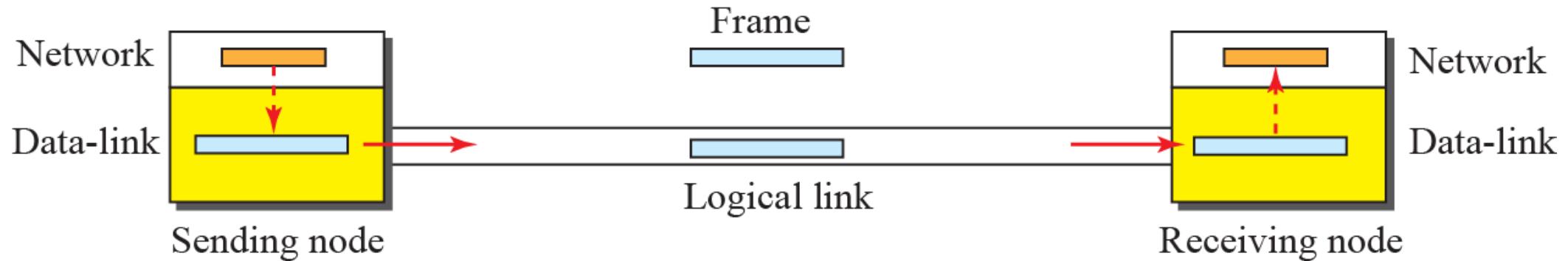
Finite State Machines (FSMs) – Example



Current State	Input	Next State	Output
Locked	coin	Unlocked	Unlocks the turnstile so that the customer can push through.
	push	Locked	None
Unlocked	coin	Unlocked	None
	push	Locked	When the customer has pushed through, locks the turnstile.

Simple Protocol

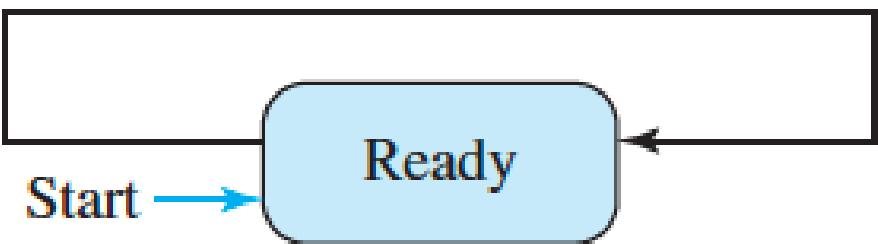
- **No flow and error control.**
- Assumption is that the receiver can never be overwhelmed with incoming frames.



Simple Protocol – FSM

Packet came from network layer.

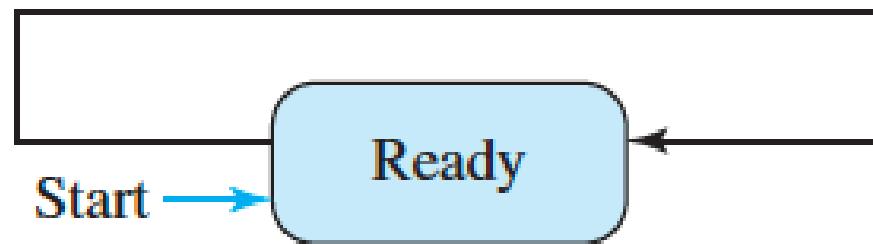
Make a frame and send it.



Sending node

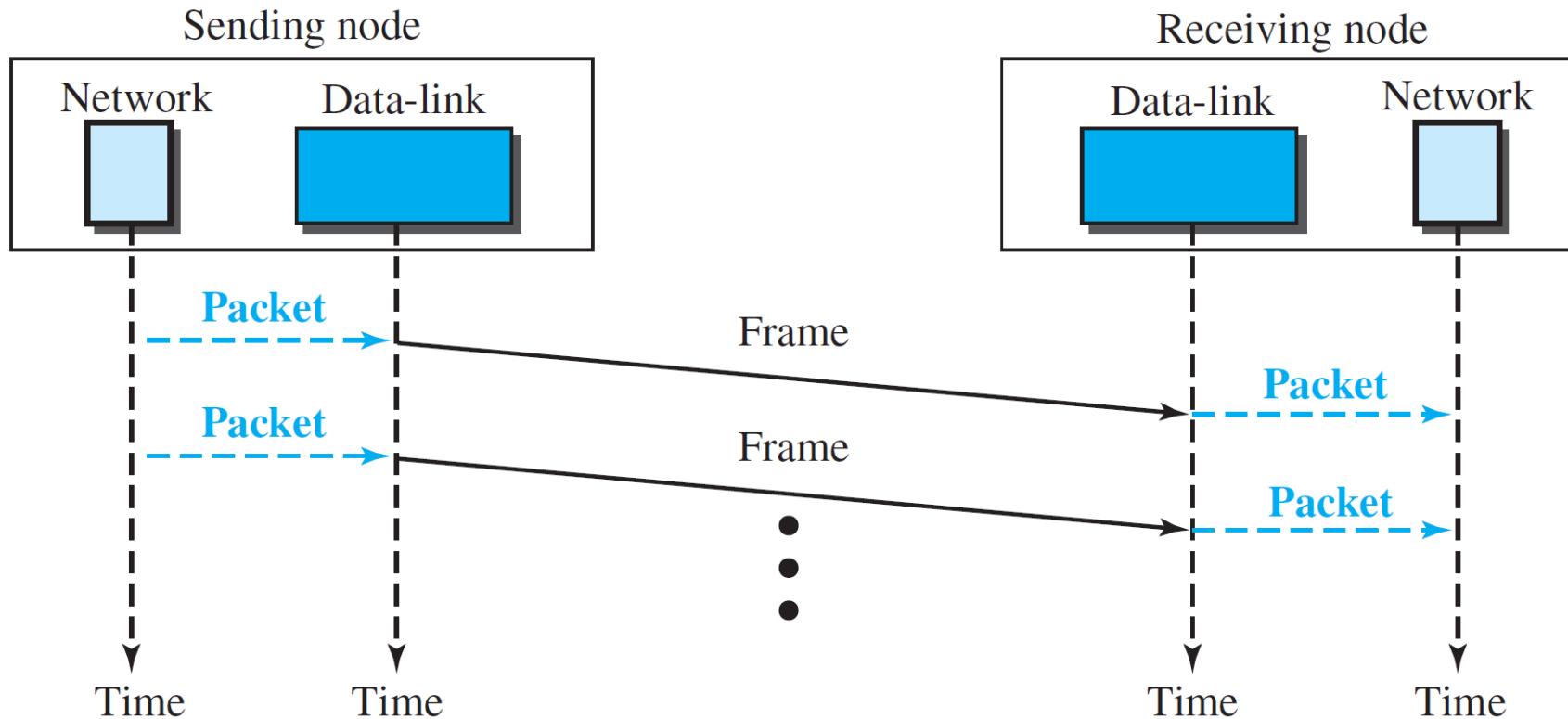
Frame arrived.

Deliver the packet to network layer.



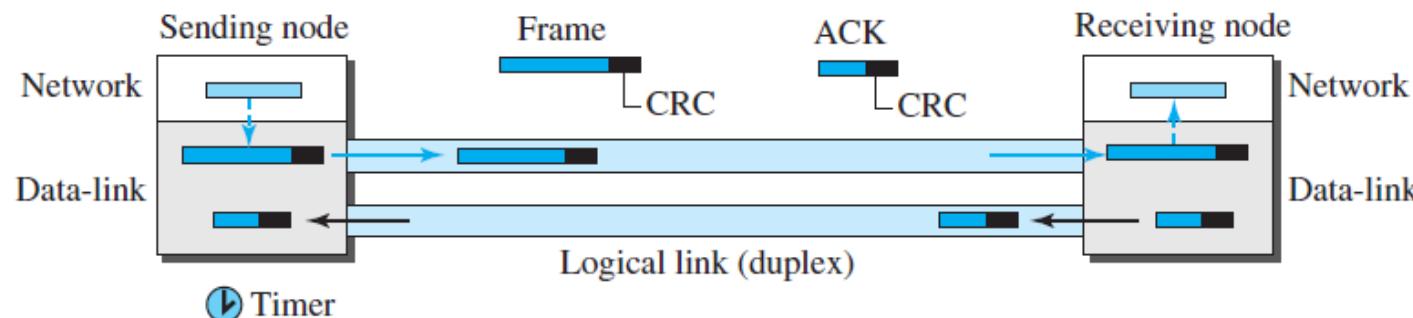
Receiving node

Simple Protocol – Flow Diagram



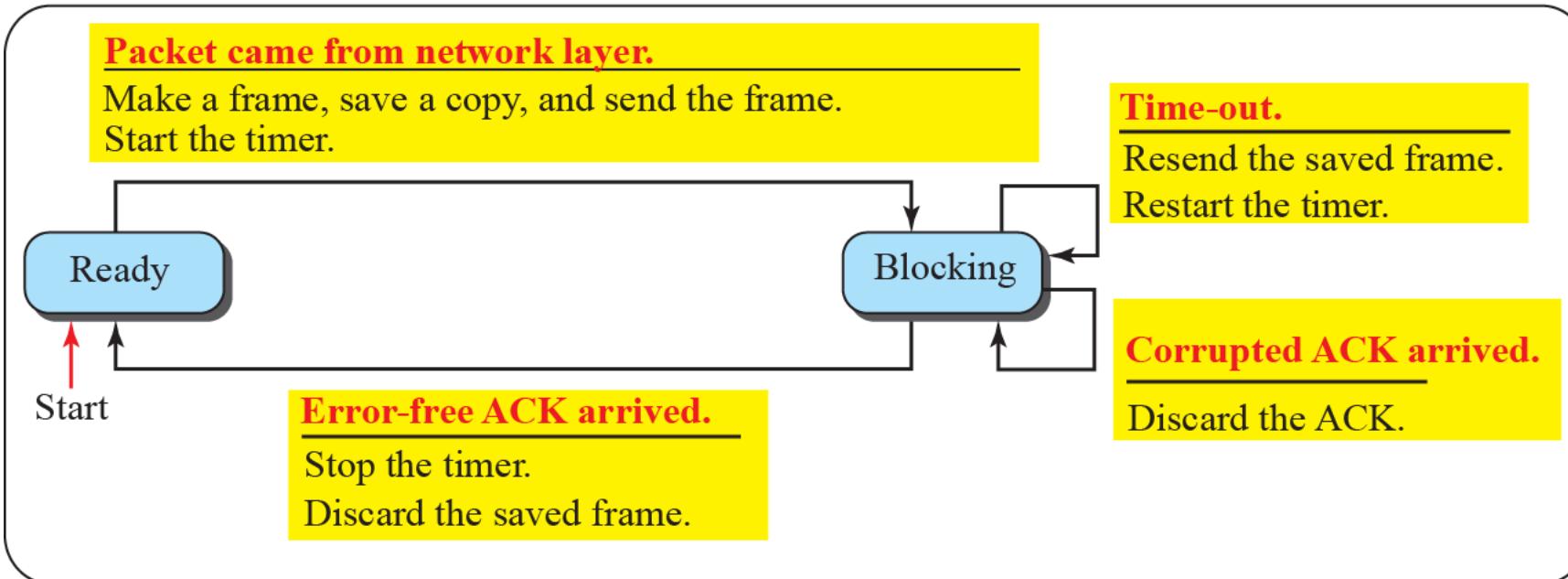
Stop-and-Wait Protocol

- A **connection-based** (connection-oriented) protocol
- **Both flow and error control provided.**
- The sender sends one frame at a time and waits for an acknowledgment before sending the next one.
- A **timer** used by the sender (if the timer expires, the frame is retransmitted).
- If the CRC is correct, the receiver will send an acknowledgement.

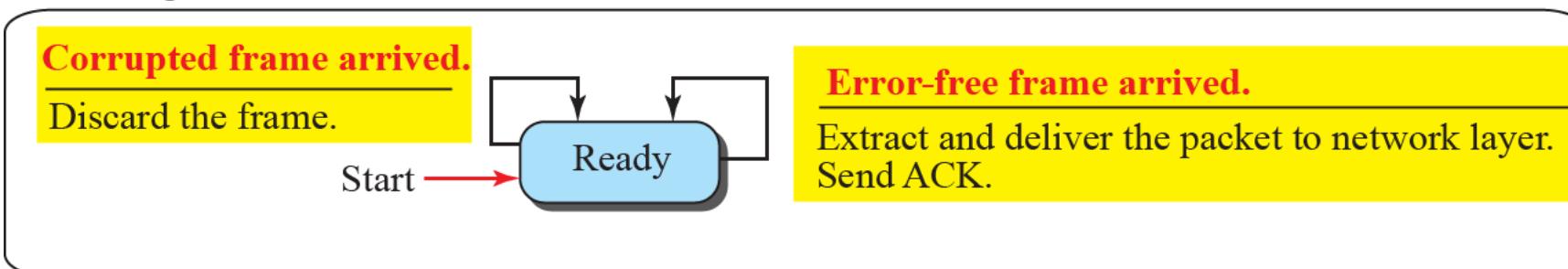


Stop-and-Wait Protocol – FSM

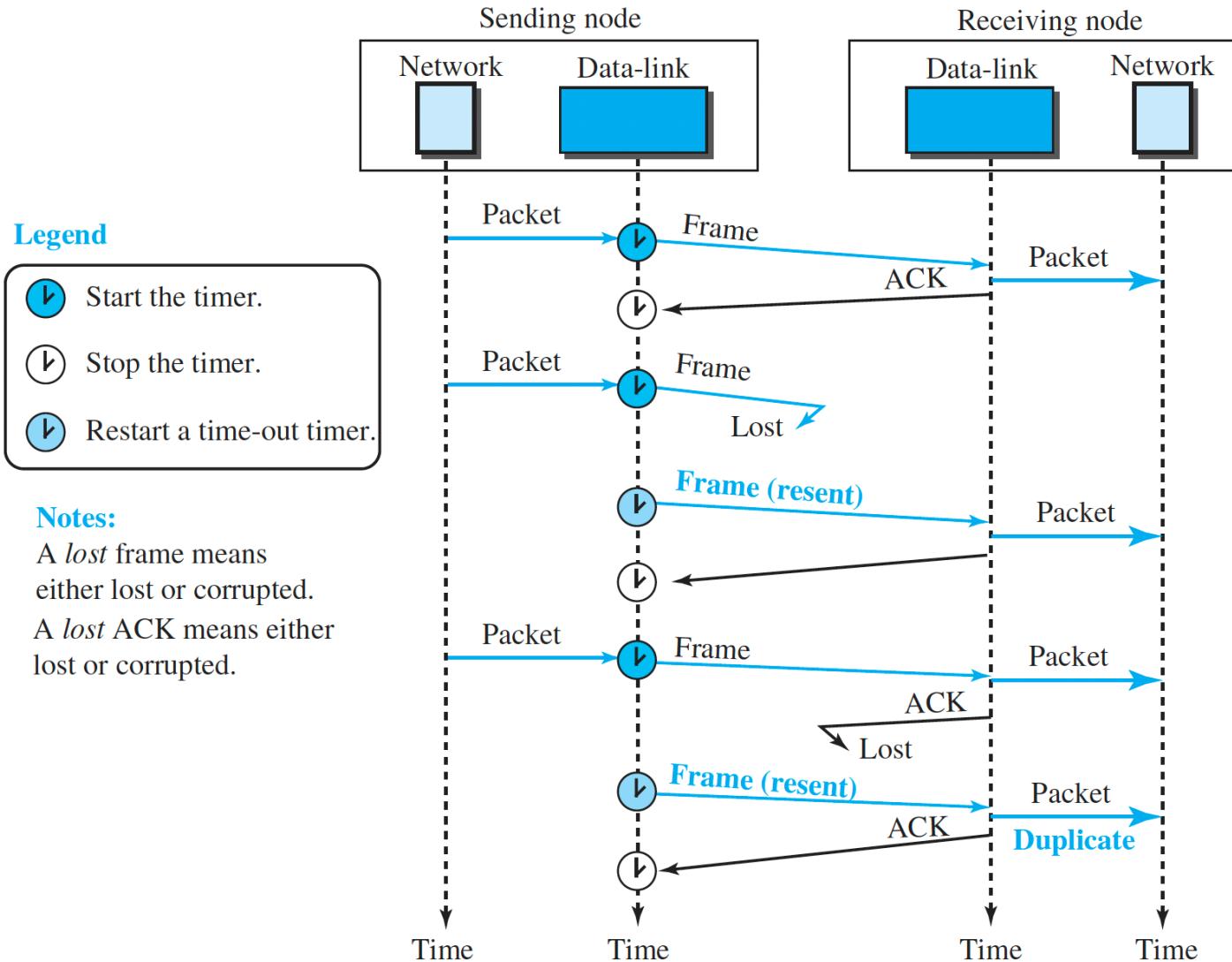
Sending node



Receiving node



Stop-and-Wait Protocol – Flow Diagram



Stop-and-Wait Protocol

- Duplicates and orders can be handled by adding a sequence number and acknowledgment number to the frames.
 - Sequence numbers are 0, 1, 0, 1, 0, 1, ...
 - Acknowledgment numbers are 1, 0, 1, 0, 1, 0, ...
- In other words, the sequence numbers start with 0, the acknowledgment numbers start with 1.
- An acknowledgment number always define the sequence number of the next frame to receive.
- A similar concept in TCP in transport layer → We'll discuss later in the course

Summary

- DLC responsibilities/services
 - Framing → Variable-size frames
 - Flow control
 - Error control
- DLC Protocols
 - Simple protocol
 - Stop-and-wait protocol

References

- [1] Behrouz A. Forouzan, Data Communications & Networking with TCP/IP Protocol Suite, 6th Ed, 2022, McGraw-Hill companies.

Reading

- Chapter 3 of the textbook, sections 3.2.1 and 3.2.3.
- Chapter 3 of the textbook, section 3.6 (Practice Test)