Chapter 11: Data Link Control

Outline

11.1 DLC SERVICES

11.2 DATA-LINK LAYER PROTOCOLS

11-1 DLC SERVICES

Recall that the data-link layer is divided into two sublayers: data link control (DLC) and media access control (MAC).

DLC deals with procedures for communication between two adjacent nodes, i.e., node-to-node communication, regardless whether the link is point-to-point or broadcast. DLC functions include framing and flow and error control.

MAC deals with procedures to handle access to a shared link (next chapter).

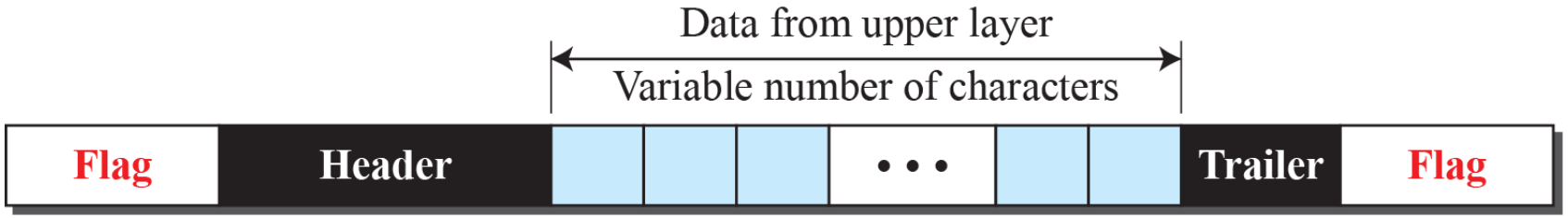
11.1.1 Framing

The data-link layer pack bits into frames, so that each frame is distinguishable from another. Framing (fixed-size or variable-size) separates messages from one source to a destination by adding a sender address and a destination address. The destination address defines where the packet is to go and the sender address helps the recipient acknowledge the receipt.

In variable-size framing, the end of one frame and the beginning of the next frame needs to be defined. We look at character-oriented framing and bit-oriented framing.

Figure 11.1: Character-Oriented Framing

In character-oriented framing, data to be carried are 8-bit characters from a coding system such as ASCII. To separate one frame from the next, an 8-bit (1 character) flag composed of protocol-dependent special characters is added at the beginning and the end of a frame to signal the start or end of a frame.



Character-oriented framing was popular when only text was exchanged by the data-link layers. However, for information types such as audio or video, any character used for the flag could also be part of the information.

Figure 11.2: Byte stuffing and unstuffing

A byte-stuffing (or character-stuffing) strategy was added to character-oriented framing: a special character (e.g., ESC) is added to the data portion of the frame when there is a character with the same pattern as the flag. Whenever the receiver encounters the special character, it removes it from the data portion and treats the next character as data.

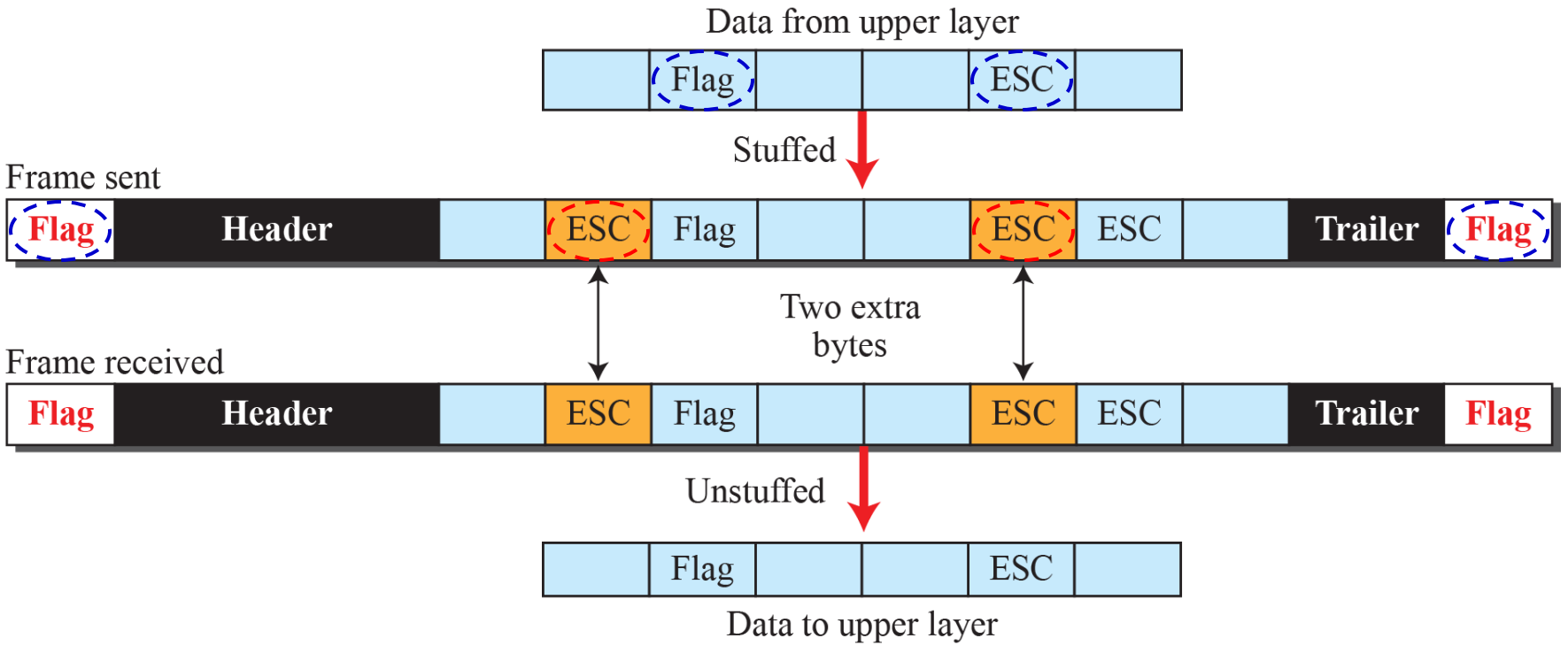


Figure 11.3: *A frame in a bit-oriented protocol*

However, universal coding systems, such as Unicode, have 16-bit and 32-bit characters that conflict with 8-bit characters used in byte-stuffing, hence, the tendency is moving toward bit-oriented framing.

In bit-oriented framing, the data section of a frame is a sequence of bits. Most protocols use a special 8-bit pattern flag, e.g., 01111110, as a delimiter to define the beginning and end of the frame.

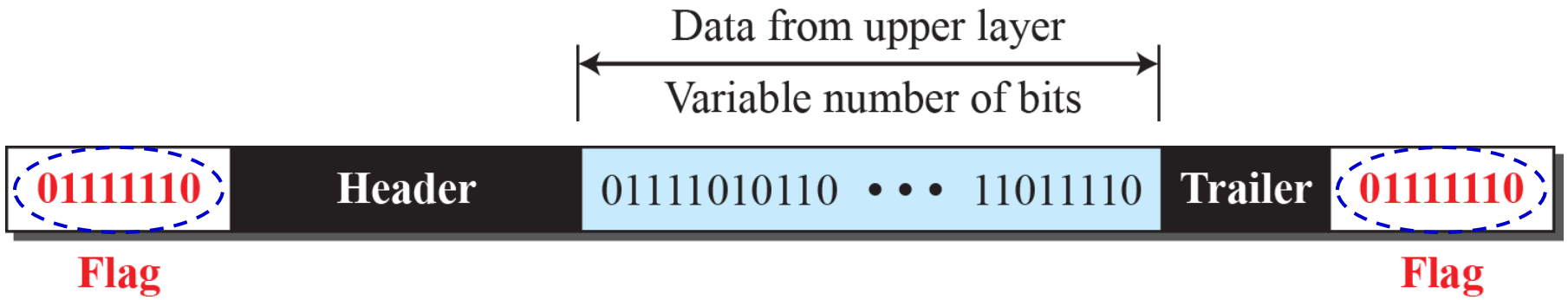
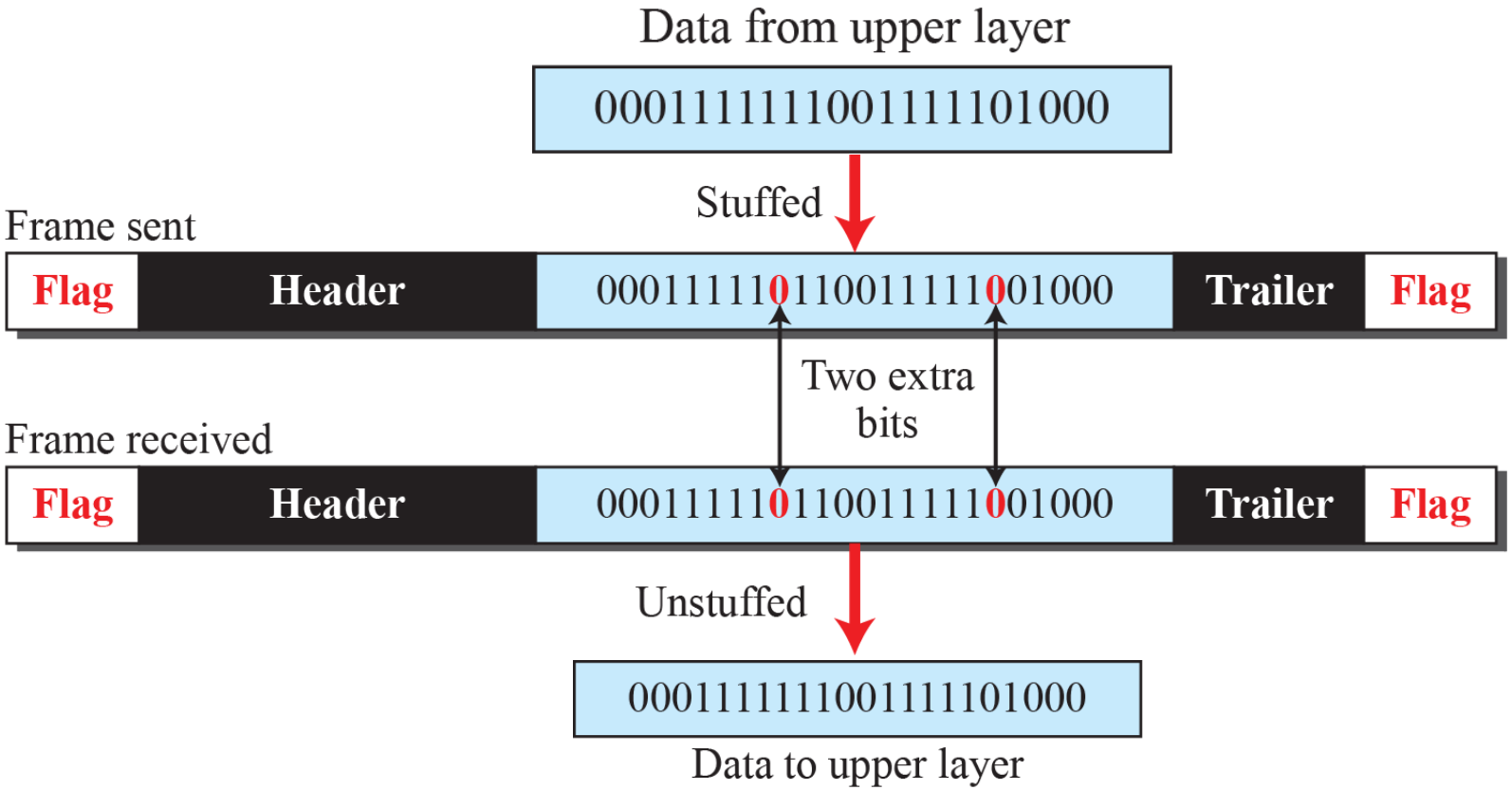


Figure 11.4: Bit stuffing and unstuffing

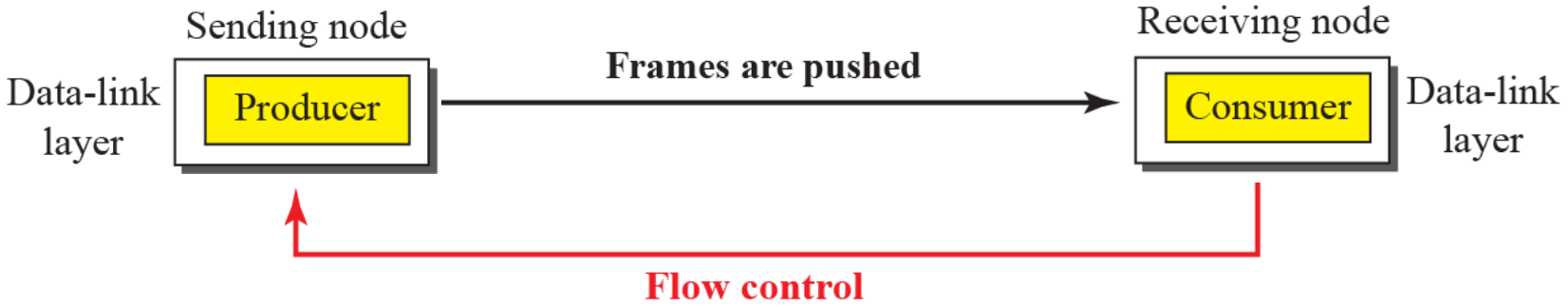
To prevent the issue where the flag pattern also appears in the data, bit stuffing adds a single bit to the data to prevent the data from looking like a flag, e.g., if the delimiter 01111110 is used, an extra 0 is added whenever five consecutive 1s follow a 0 in the data. Note that even if a 0 follows after five 1s, a 0 is still stuffed.



11.1.2 Flow and Error Control

Another responsibility of the data-link control sublayer is flow and error control at the data-link layer:

Flow control: Whenever an entity produces items and another entity consumes them, there should be a balance between production and consumption rates.



Error control: Given that the underlying physical layer is not fully reliable, error control prevents the receiving node from delivering corrupted packets to its network layer. This is usually implemented by including a CRC in the frame header.

11.1.3 Connection

A DLC protocol can be either connectionless or connection-oriented:

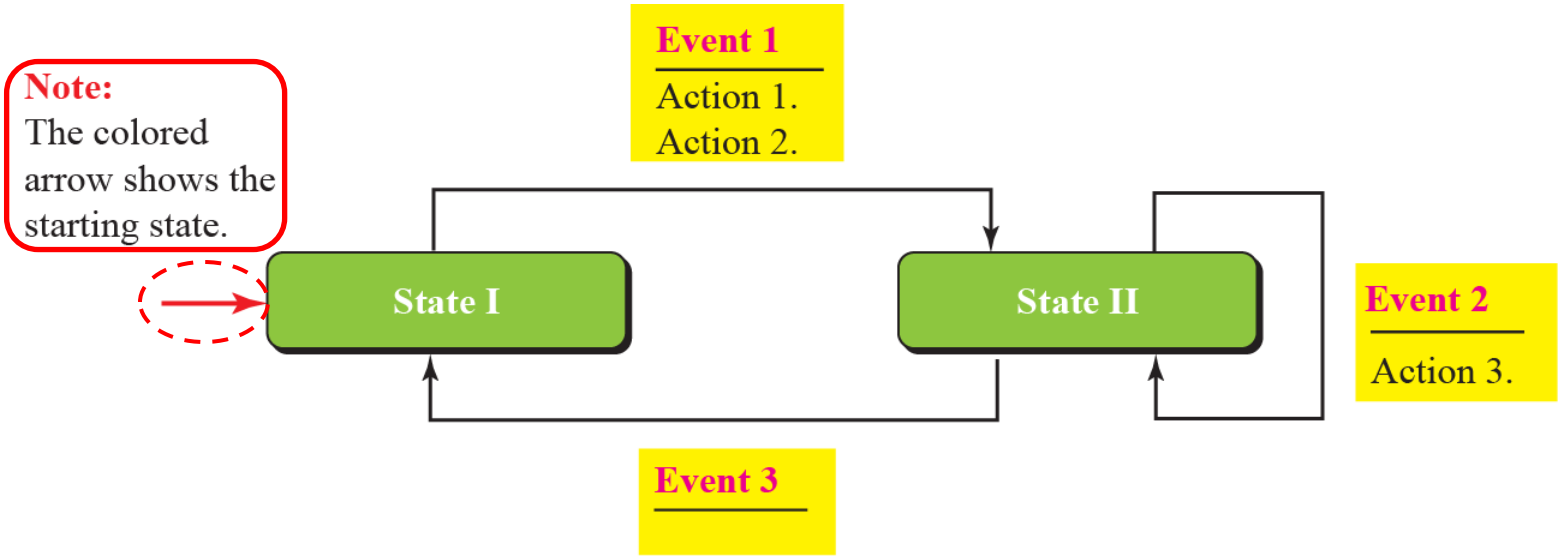
In a connectionless protocol, frames are sent from one node to the next without any relationship between the frames, i.e., each frame is independent. The frames are not numbered and there is no notion of numbering.

In a connection-oriented protocol, a logical connection should first be established between the two nodes, all the frames, related in some way, are transmitted and then the logical connection is terminated. The frames are usually numbered and sent in order.

11-2 DATA-LINK LAYER PROTOCOLS

The behavior of a data-link layer protocol can be shown as a finite state machine (FSM):

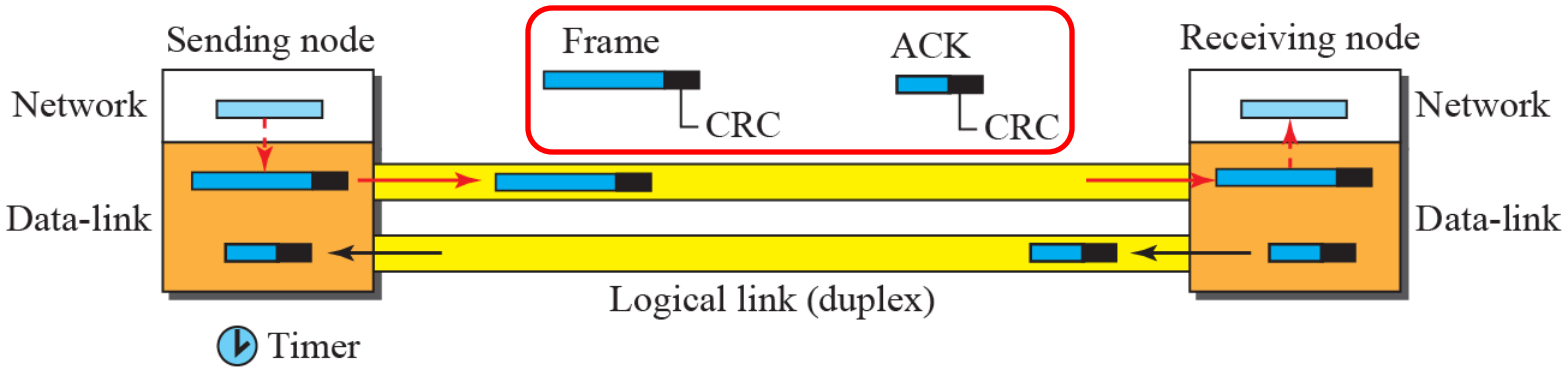
A FSM has a finite number of states and the machine is always in one of the states until an event occurs. One of the states must be defined as the initial state. Each event is associated with two reactions: (i) defining the list of actions to be performed and (ii) defining the next state.



11.2.2 Stop-and-Wait Protocol

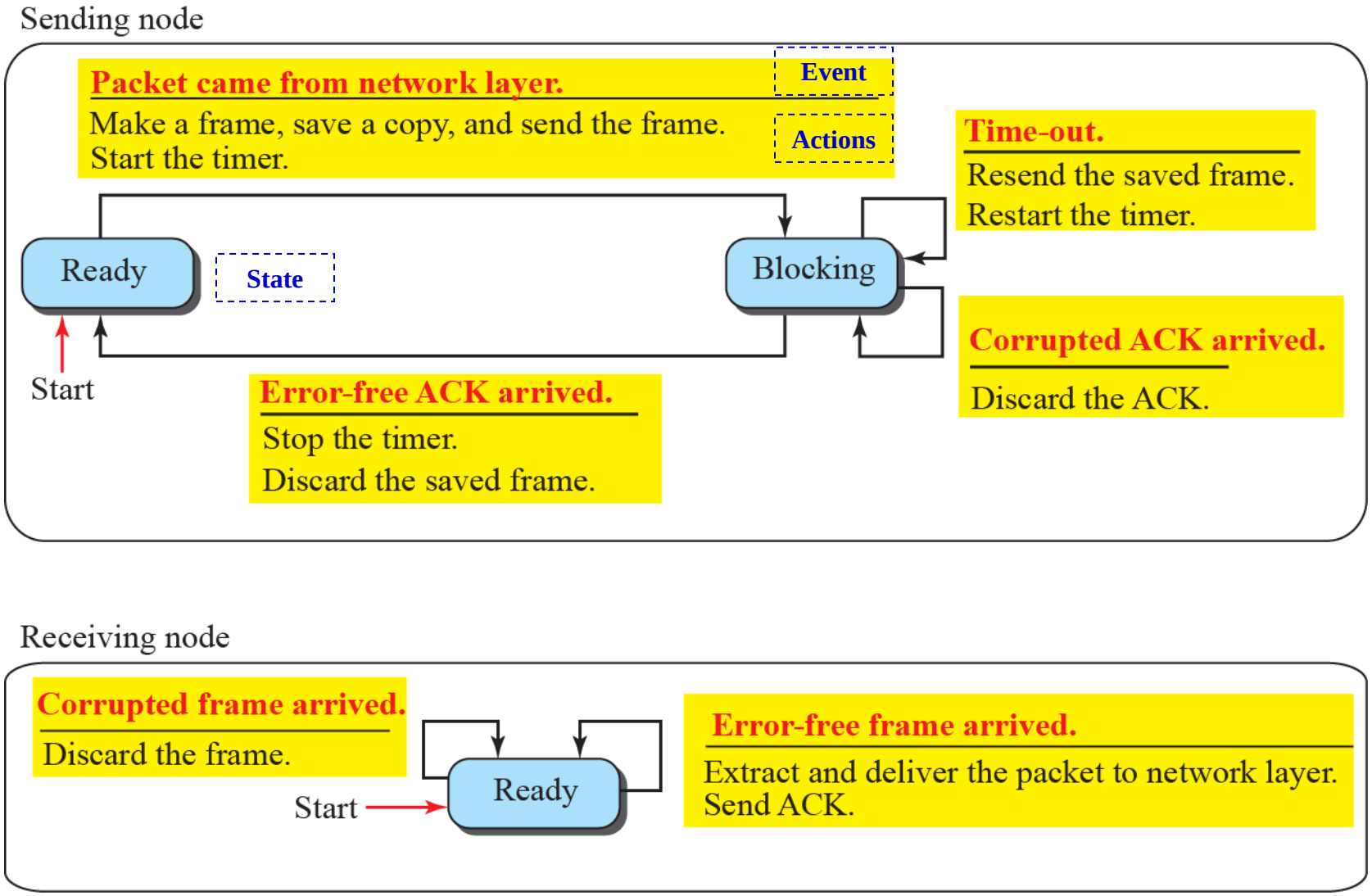
A DLC protocol that uses both flow and error control is called the Stop-and-Wait protocol.

In this protocol, the sender sends one frame at a time and waits for an acknowledgment before sending the next one. To detect corrupted frames, a CRC is added to each data frame. Every time the sender sends a frame, it starts a timer: if an acknowledgment arrives before the timer expires, it sends the next frame if it has one to send; if the timer expires, the sender resends the previous frame assuming that the frame was either lost or corrupted.

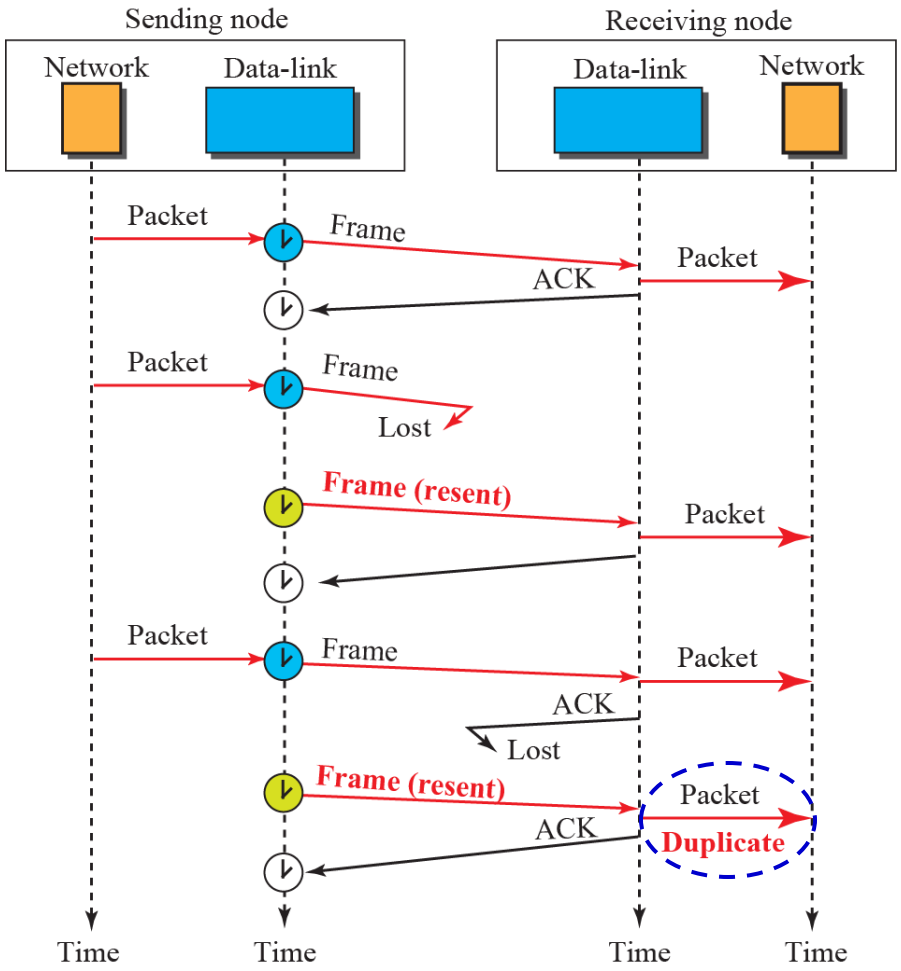


Note that only one frame and one acknowledgment can be in the channels at any time.

Figure 11.11: FSM for the Stop-and-Wait protocol



Example



Legend

- Start the timer.
- Stop the timer.
- Restart a time-out timer.

Notes:

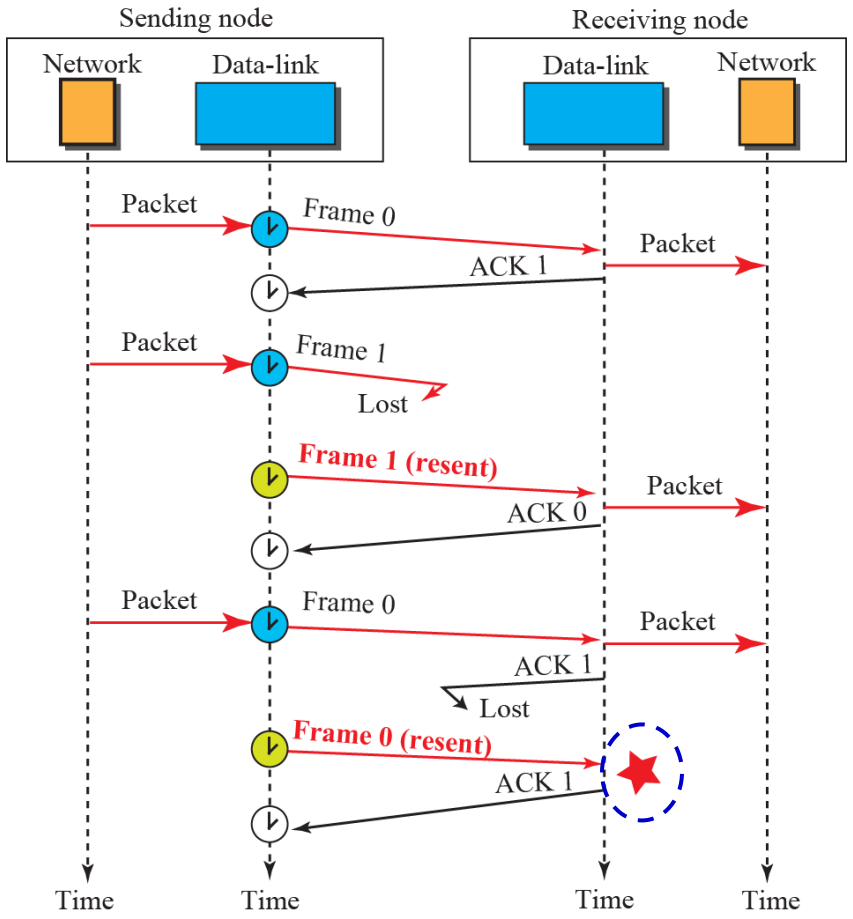
A lost frame means either lost or corrupted.

A lost ACK means either lost or corrupted.

The first frame is sent and acknowledged. The second frame is sent, but lost. After time-out, it is resent. The third frame is sent and acknowledged, but the acknowledgment is lost. The frame is resent. However, there is a problem with this scheme: the network layer at the receiving node received two copies of the third packet (a problem that needs to be corrected).

Example

Duplicate packets, as much as corrupted packets, need to be avoided. Adding sequence numbers and acknowledgment numbers can prevent duplicates.



Legend

- Start the timer.
- Stop the timer.
- Restart a time-out timer.

Notes:

A lost frame means either lost or corrupted.

A lost ACK means either lost or corrupted.

Frame 0 is discarded because the receiver expects frame 1.

In this scheme, the sequence numbers start with 0 and the acknowledgment start with 1 (and alternate thereafter). The first frame is sent and acknowledged. The second frame is sent, but lost. After a time-out, it is resent. The third frame is sent and acknowledged, but the acknowledgment is lost. The frame is resent and acknowledged.

11.2.3 Piggybacking

Stop-and-Wait protocol is designed for unidirectional communication, i.e., data is flowing only in one direction although the acknowledgment may travel in the other direction. Protocols have been designed in the past to allow data to flow in both directions.

To make the communication more efficient, the data in one direction can be piggybacked with the acknowledgment in the other direction. In other words, when node A is sending data to node B, node A can also acknowledge the data received from node B. Note however, piggybacking makes communication at the data-link layer more complicated.

Chapter 12: Media Access Control

Outline

12.1 RANDOM ACCESS

12-1 RANDOM ACCESS

In random-access or contention methods, no station is superior to another station and none is assigned control over another. At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send. This decision may depend on whether the state of the medium is idle or busy.

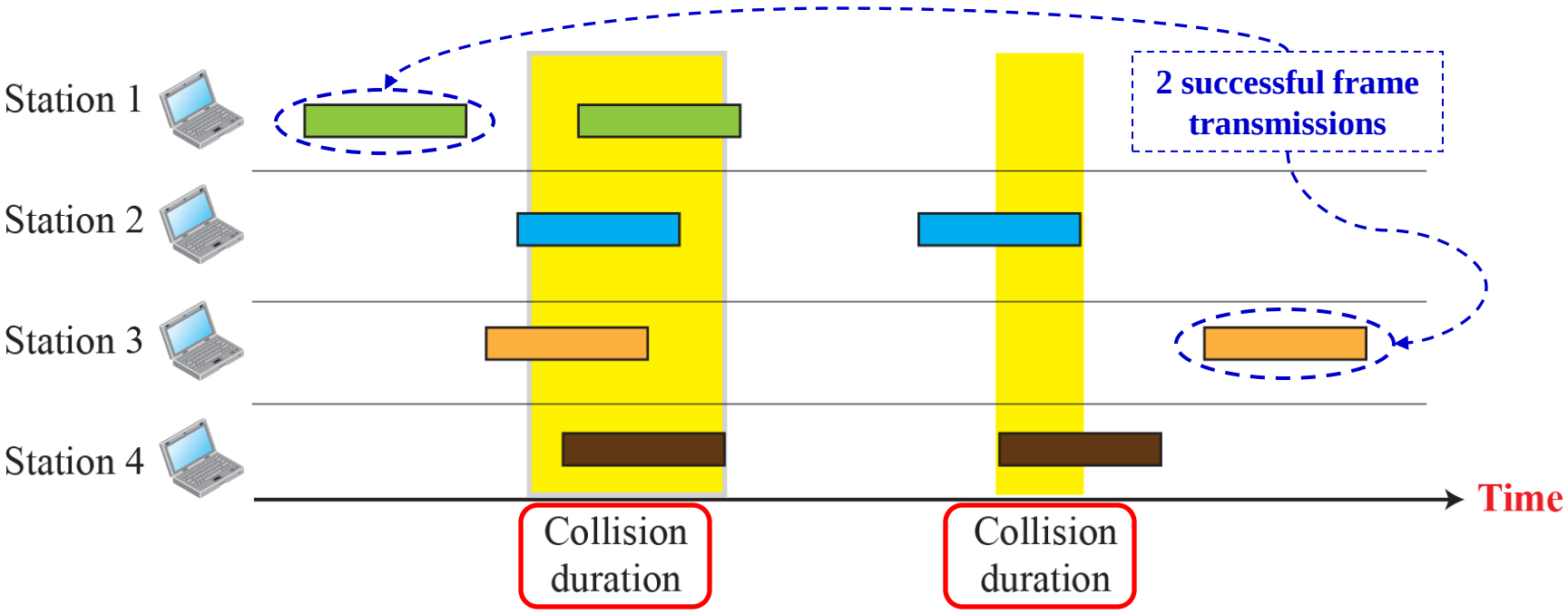
Two features give this method its name: (i) transmission is random among the stations and (ii) stations compete with one another to access the medium.

12.1.1 Pure ALOHA

ALOHA, the earliest random access method, was developed at the University of Hawaii in early 1970. It was designed for a wireless radio LAN, but it can be used on any shared medium.

In the original ALOHA protocol, also known as pure ALOHA, the idea is that each station sends a frame whenever it has a frame to send. However, since there is only one channel to share, there is a possibility of collision between frames from different stations.

Figure 12.2: Frames in a pure ALOHA network



As the medium is shared between the stations, it is obvious that there are potential collisions in a pure ALOHA network. When a station sends data, another station may attempt to do so at the same time: the data from the two stations collide and become garbled.

Figure 12.3: Procedure for pure ALOHA protocol

The pure ALOHA protocol relies on acknowledgments from the receiver. When a station sends a frame, it expects the receiver to send an acknowledgment. If the acknowledgment does not arrive after a time-out period, the station assumes that either the frame (or the acknowledgment) is corrupted/lost and resends the frame.

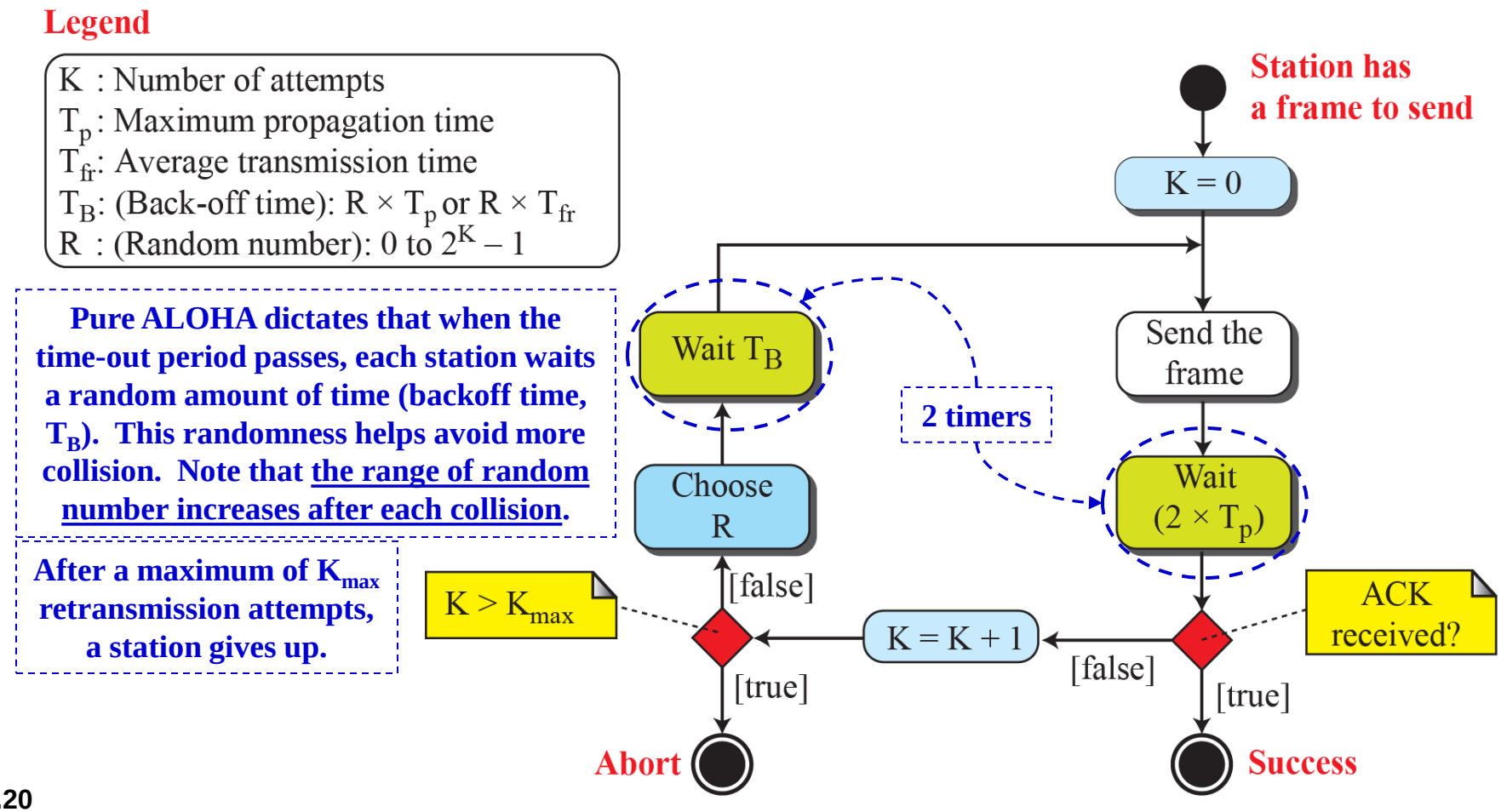
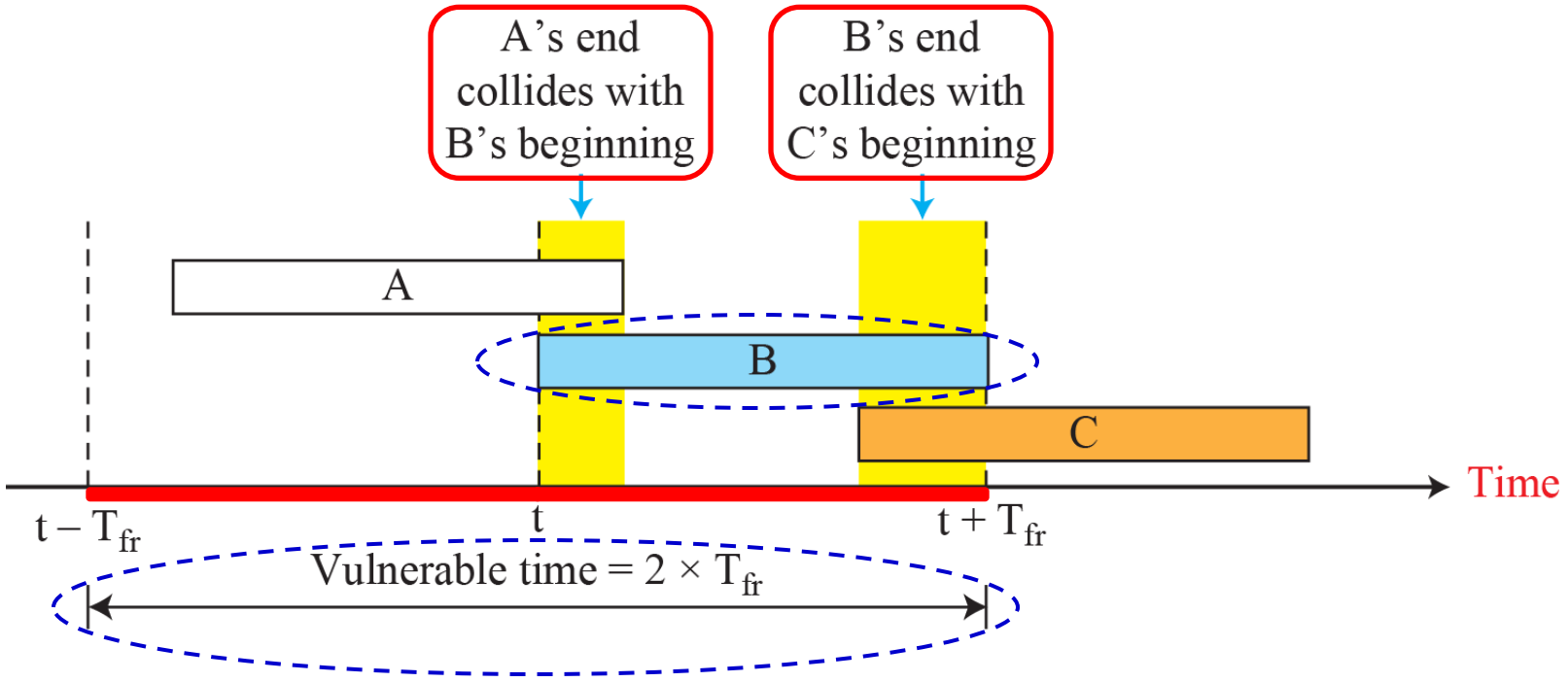


Figure 12.4: Vulnerable time for pure ALOHA protocol

Let's define the vulnerable time, i.e., the length of time in which there is a possibility of collision. Assuming that the stations send fixed-length frames, with each frame taking T_{fr} sec to transmit, the vulnerable time for station B (transmitting at time t) can be shown as follows:



Problem

A station in a pure ALOHA network intends to transmit 200 bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

Solution

The frame transmission time, T_{fr}, is 200 bits/200 kbps = 1 ms. The vulnerable time is 2×1 ms = 2 ms.

(This means no station should send later than 1 ms before this station starts transmission of a frame and no station should start sending during the 1 ms period that this station is sending.)

Throughput for pure ALOHA

Let's define G as the average number of frames generated by the system during one frame transmission time, T_{fr}. It can be shown that the average number of successfully transmitted frames, S, for pure ALOHA is

$$S = G \times e^{-2G}$$

The maximum throughput, S_{max}, is 0.184 or 18.4% when G = ½ by solving $\frac{dS}{dG} = 0$.

(Note that we expect that G = ½, i.e., one frame during two frame transmission times, to obtain the maximum throughput as the vulnerable time is $2T_{fr}$).

Problem

A pure ALOHA network transmits 200 bit frames on a shared channel of 200 kbps. What is the throughput if the system, i.e., all stations together, produces

a. 1000 frames per second? **b.** 500 frames per second? **c.** 250 frames per second?

Solution

The frame transmission time, T_{fr}, is 200 bits/200 kbps = 1 ms or 0.001 s.

a. If the system creates 1000 frames per second, then $G = 1$ (1 frame per T_{fr} of 1 ms). In this case, $S = G \times e^{-2G} = 0.135$ (13.5%). The throughput is $1000 \times 0.135 = 135$ frames.

b. If the system creates 500 frames per second, then $G = ½$ (½ frame per T_{fr}). In this case $S = G \times e^{-2G} = 0.184$ (18.4%). The throughput is $500 \times 0.184 = 92$ frames. This is the maximum throughput case for pure ALOHA.

c. If the system creates 250 frames per second, then $G = ¼$ (¼ frame per T_{fr}). In this case $S = G \times e^{-2G} = 0.152$ (15.2%). The throughput is $250 \times 0.152 = 38$ frames.

12.1.1 Slotted ALOHA

Pure ALOHA has a vulnerable time of $2T_{fr}$ as there are no rules to define when a station can transmit. Slotted ALOHA was created to improve the efficiency of pure ALOHA. In slotted ALOHA, the time is divided into slots of T_{fr} seconds and stations can only transmit at the beginning of the time slots.

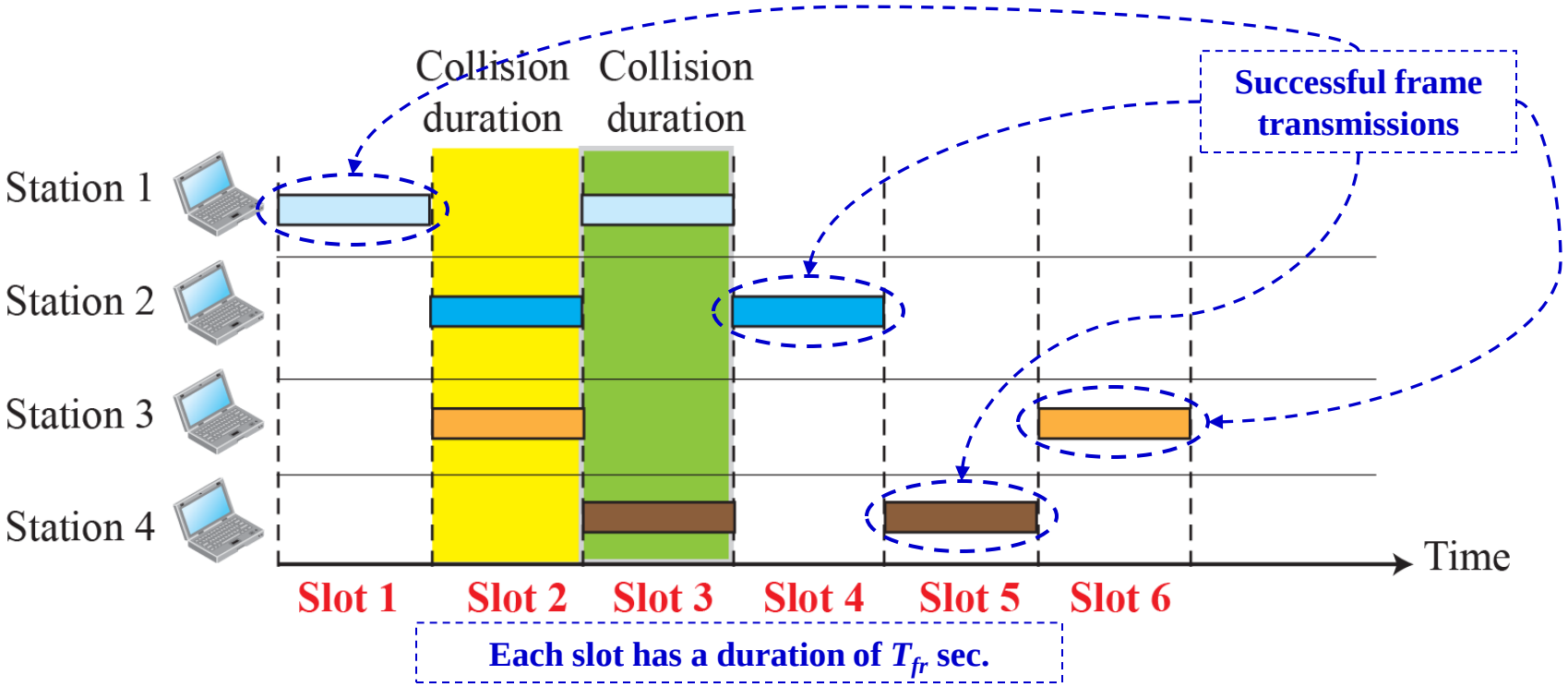
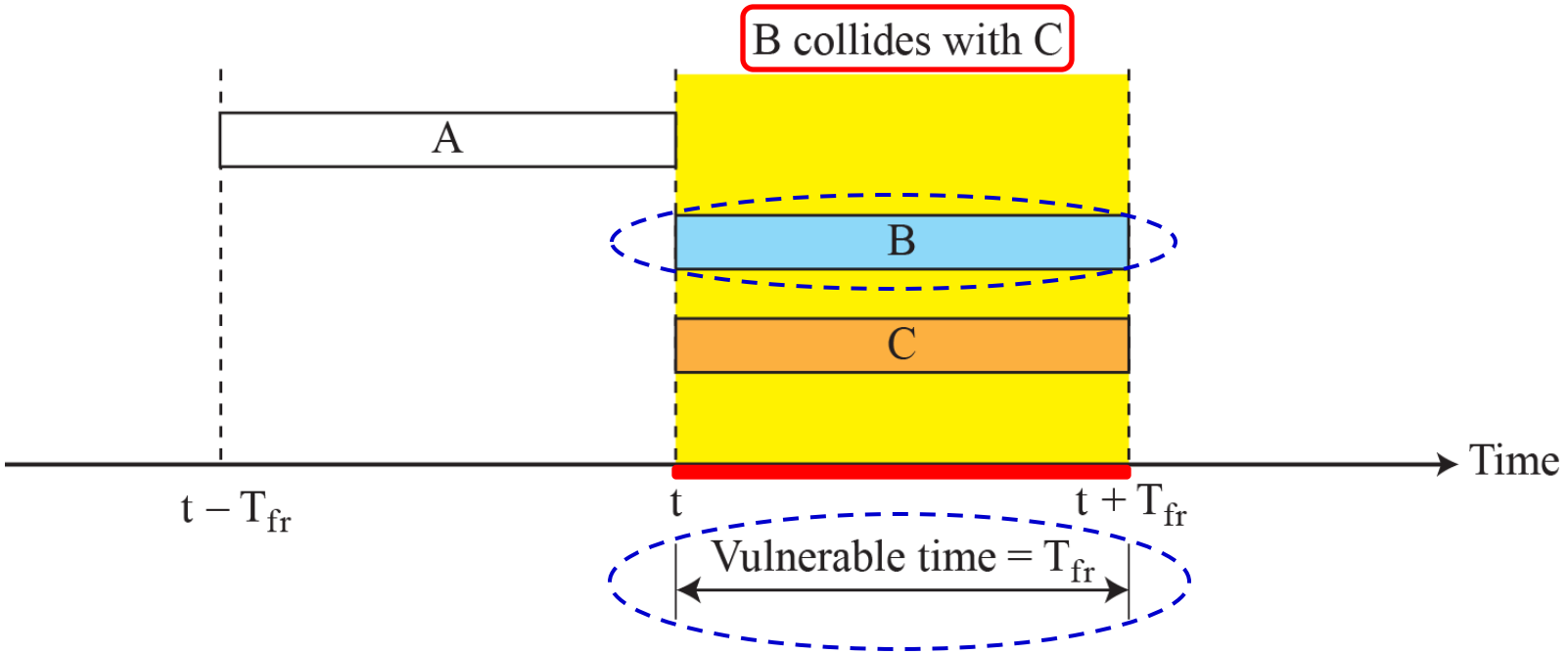


Figure 12.6: Vulnerable time for slotted ALOHA protocol

Since a station is only allowed to transmit at the beginning of a time slot, if a station misses the current time slot, it must wait until the beginning of the next time slot. This means that if a station started transmission at the beginning of the current time slot, the transmission has already finished by the next time slot. Note there is still a possibility of collision if two stations attempt to transmit at the beginning of the same time slot. However, the vulnerable time is reduced to T_{fr} .



Throughput for slotted ALOHA

It can be shown that the average number of successfully transmitted frames, S, for slotted ALOHA is

$$S = G \times e^{-G}$$

where G is the average number of frames generated by the system during one frame transmission time. The maximum throughput, S_{max}, is 0.368 or 36.8% when G = 1 by solving $\frac{dS}{dG} = 0$.

Problem

A slotted ALOHA network transmits 200 bit frames using a shared channel with a 200 kbps bandwidth. Find the throughput if the system, i.e., all stations together, produces

a. 1000 frames per second. **b.** 500 frames per second. **c.** 250 frames per second.

Solution

The frame transmission time, T_{fr}, is 200 bits/200 kbps = 1 ms or 0.001 s.

a. If the system creates 1000 frames per second, then $G = 1$ (1 frame per T_{fr} of 1 ms). In this case, $S = G \times e^{-G} = 0.368$ (36.8%). The throughput is $1000 \times 0.368 = 368$ frames. Note that this is the maximum throughput case for slotted ALOHA.

b. If the system creates 500 frames per second, then $G = ½$ (½ frame per T_{fr}). In this case $S = G \times e^{-G} = 0.303$ (30.3%). The throughput is $500 \times 0.303 = 151$ frames.

c. If the system creates 250 frames per second, then $G = ¼$ (¼ frame per T_{fr}). In this case $S = G \times e^{-G} = 0.195$ (19.5%). The throughput is $250 \times 0.195 = 49$ frames.

12.1.2 CSMA

To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed. The chance of collision can be reduced if a station senses the medium before trying to use it. Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending.

In other words, CSMA is based on the principle "sense before transmit" or "listen before talk."

Figure 12.7: Space/time model of a collision in CSMA

CSMA can reduce the possibility of collision, but it cannot eliminate it. The possibility of collision still exists because of propagation delay: when a station sends a frame, it still takes time for the first bit to reach every station and for every station to sense it.

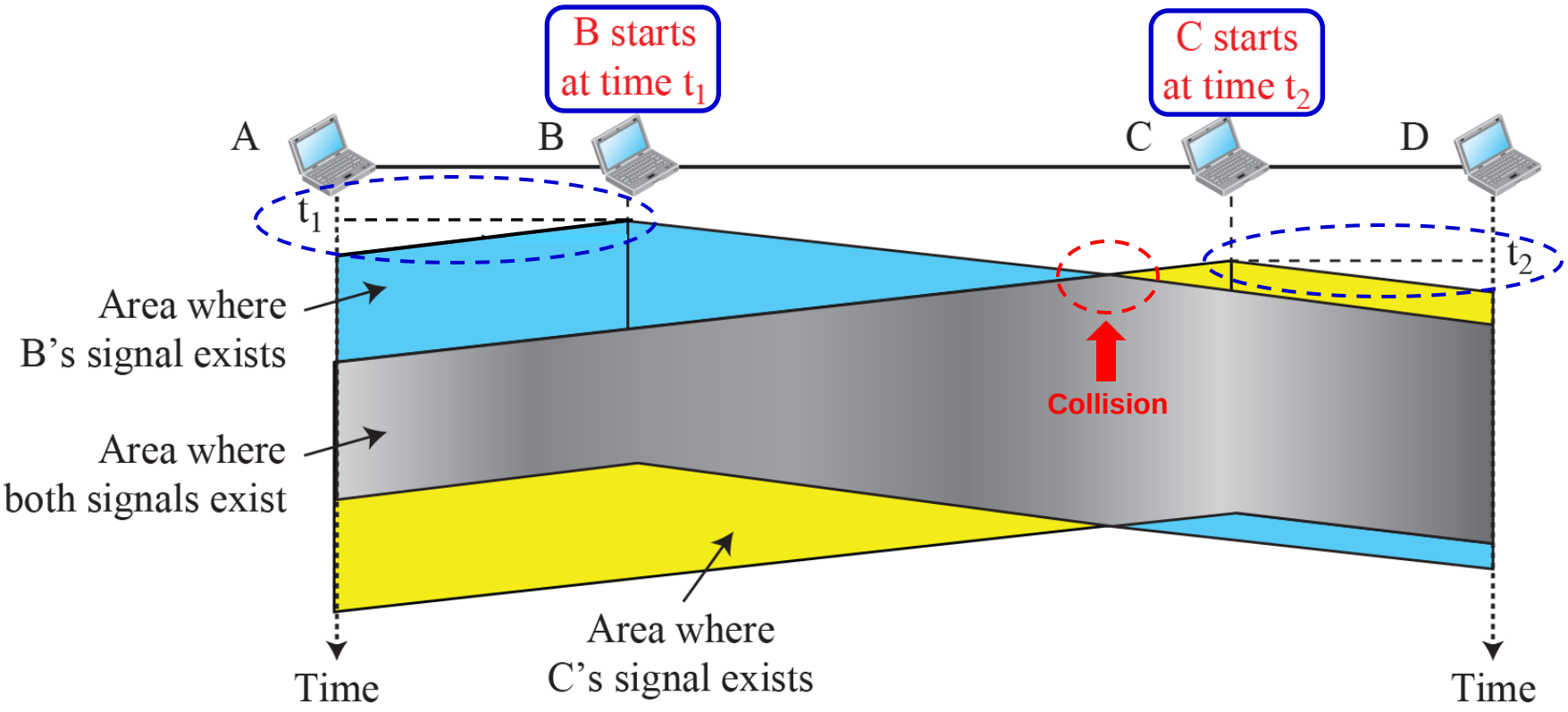
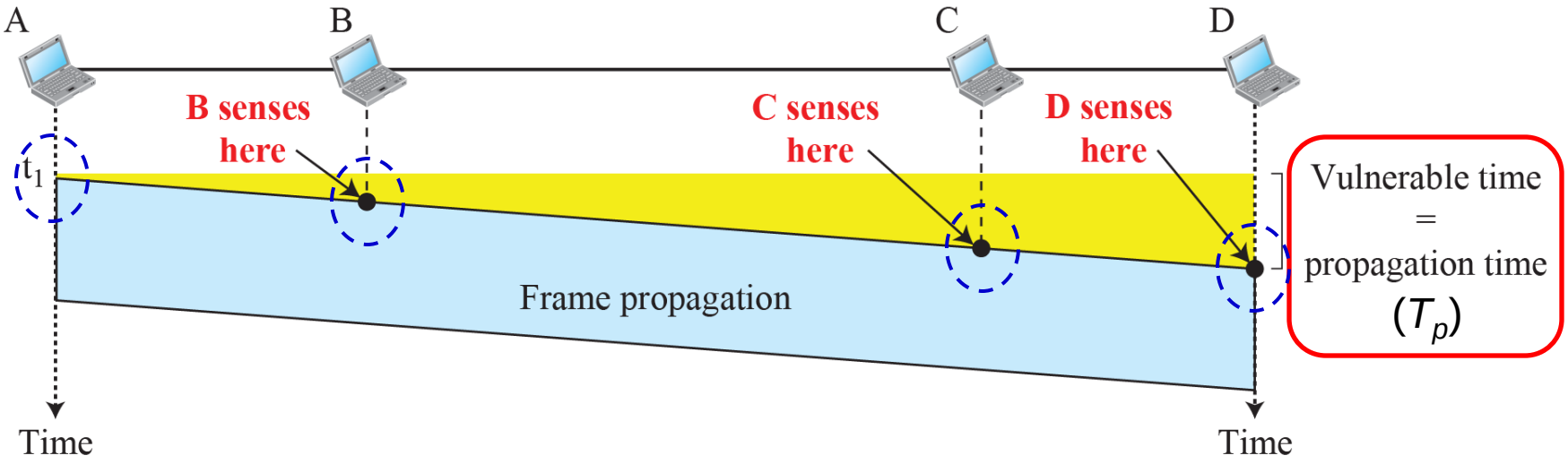


Figure 12.8: Vulnerable time in CSMA

The vulnerable time for CSMA is the maximum propagation time, T_p, the time needed for a signal to propagate from one end of the medium to the other. When a station sends a frame and any other station tries to send a frame during this time, a collision will result. However, if the first bit of the frame reaches the end of the medium, every station will have heard the bit and will refrain from sending.



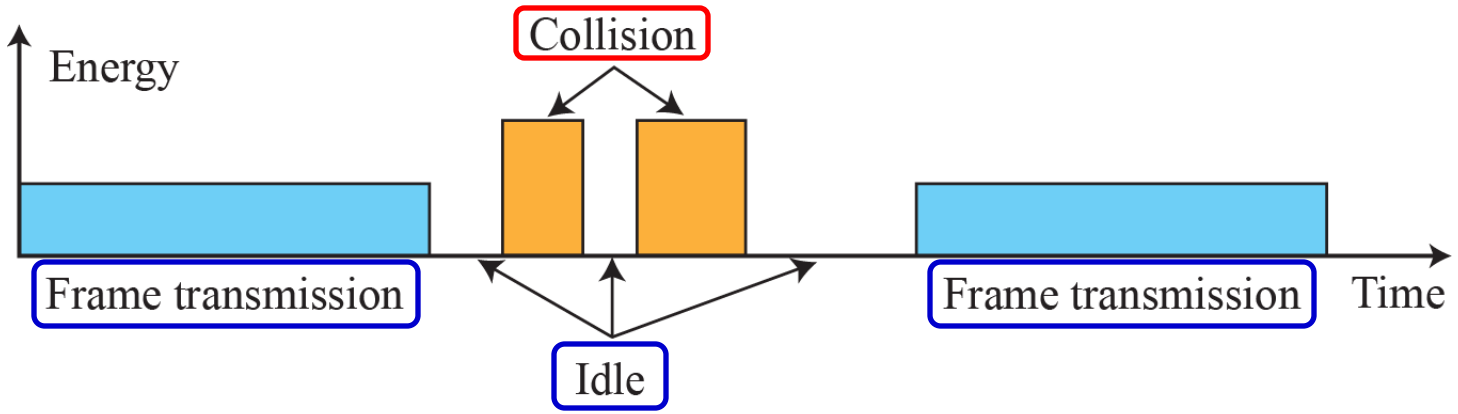
12.1.3 CSMA/CD

The CSMA method does not specify the procedure following a collision. Carrier sense multiple access with collision detection (CSMA/CD) augments the algorithm to handle the collision.

In this method, a station monitors the medium after it sends a frame to see if the transmission was successful. If it was successful, the station is done. If, however, there was a collision, the frame is sent again.

Figure 12.14: *Energy level during transmission, idleness, or collision*

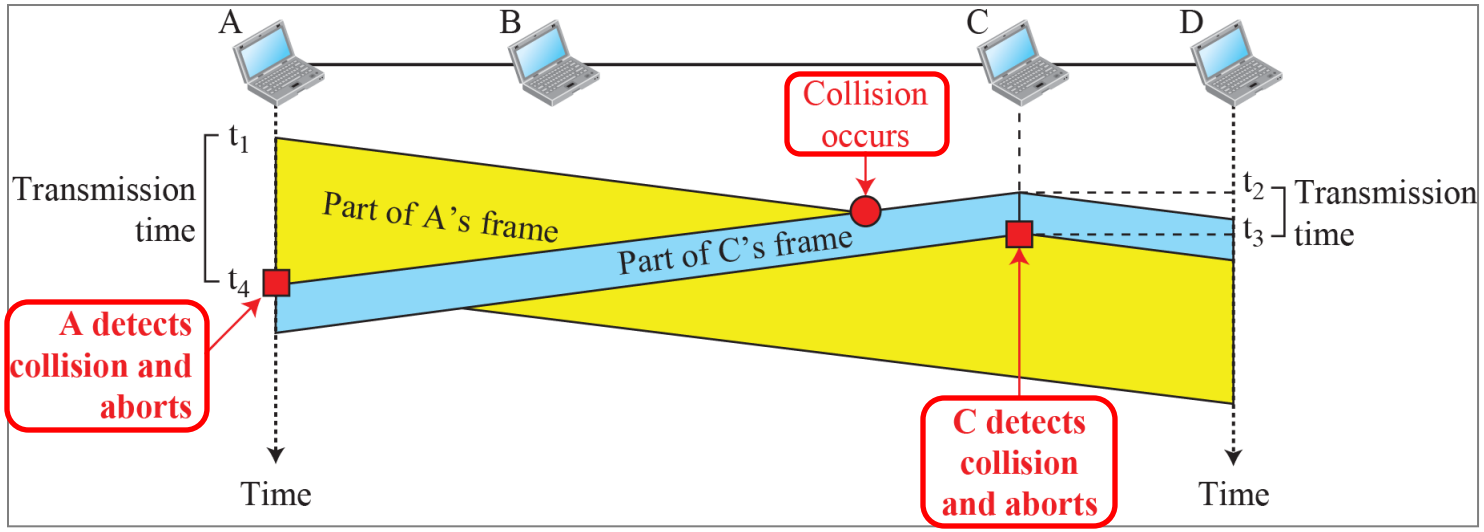
To understand CSMA/CD, let's see how a station monitors the energy level on the channel to determine if the channel is idle, busy or in collision mode. Depending on the energy level in the channel:



1) ***At zero level:*** ***channel is idle.***
2) ***At normal level:*** ***channel is busy. A station has successfully captured the channel and is transmitting a frame.***
3) ***At abnormal level:*** ***channel is in collision mode. There is a collision and the energy level is twice the normal level.***

Figure 12.11: Collision and Abortion in CSMA/CD

In CSMA/CD, a station monitors the medium after it sends a frame to see if the transmission was successful. Let's look at the first bits transmitted by the two stations involved in the collision:



@t_1: A starts sending the bits of its frame.

@t_2: C has not yet sensed the first bit sent by A and starts sending the bits of its frame. A collision occurs sometime after t_2.

@t_3: C detects a collision and immediately aborts transmission.

@t_4: A detects a collision and immediately aborts transmission.

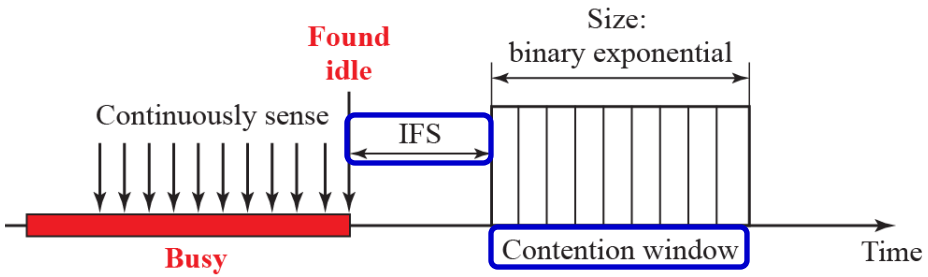
A transmits for duration ($t_4 - t_1$) and C transmits for duration ($t_3 - t_2$).

12.1.4 CSMA/CA

Carrier sense multiple access with collision avoidance (CSMA/CA) was invented for wireless networks. Collisions are avoided through the use of CSMA/CA's three strategies: the interframe space (IFS), the contention window and acknowledgments along with Ready to Send (RTS) and Clear to Send (CTS) frames:

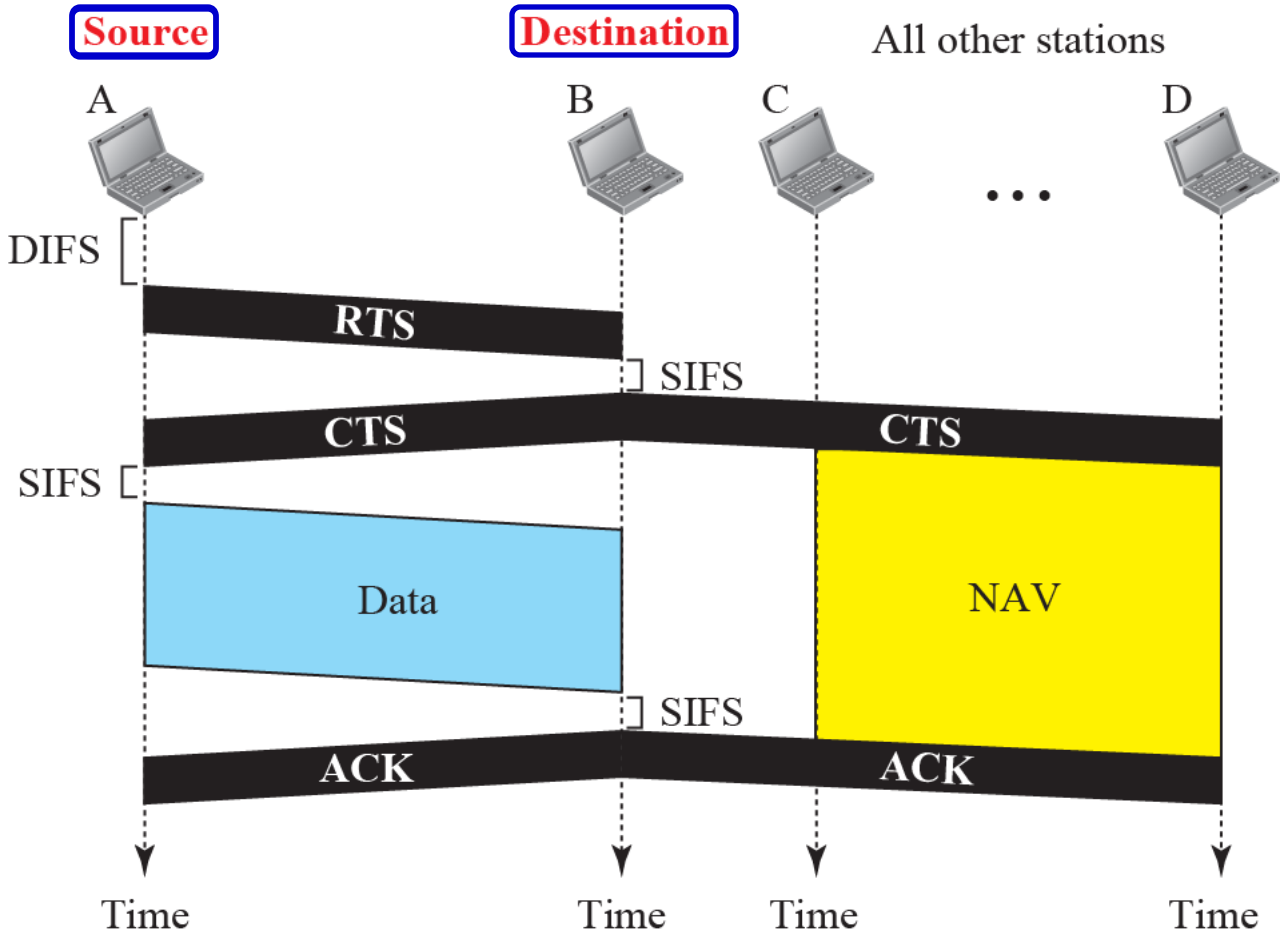
Interframe Space: Collisions are avoided by deferring transmission even if the channel is idle. It waits for a IFS period of time as the channel may appear idle even though a distant station may have started transmission.

Contention Window: A station that is ready to send chooses a random (binary exponential) number of slots as its wait time.



Acknowledgment: The use of positive acknowledgment and time-out timers helps guarantee that the receiver has received the frame.

Figure 12.17: CSMA/CA and NAV



DIFS: Distributed Coordination Function (DCF) Interframe Space

SIFS: Short Interframe Space

NAV: Network Allocation Vector - When a station sends a RTS frame, it includes the duration of the time it needs to occupy the channel. The stations that are affected by this transmission create a NAV timer that shows how much time must pass before these stations are allowed to check the channel for idleness.