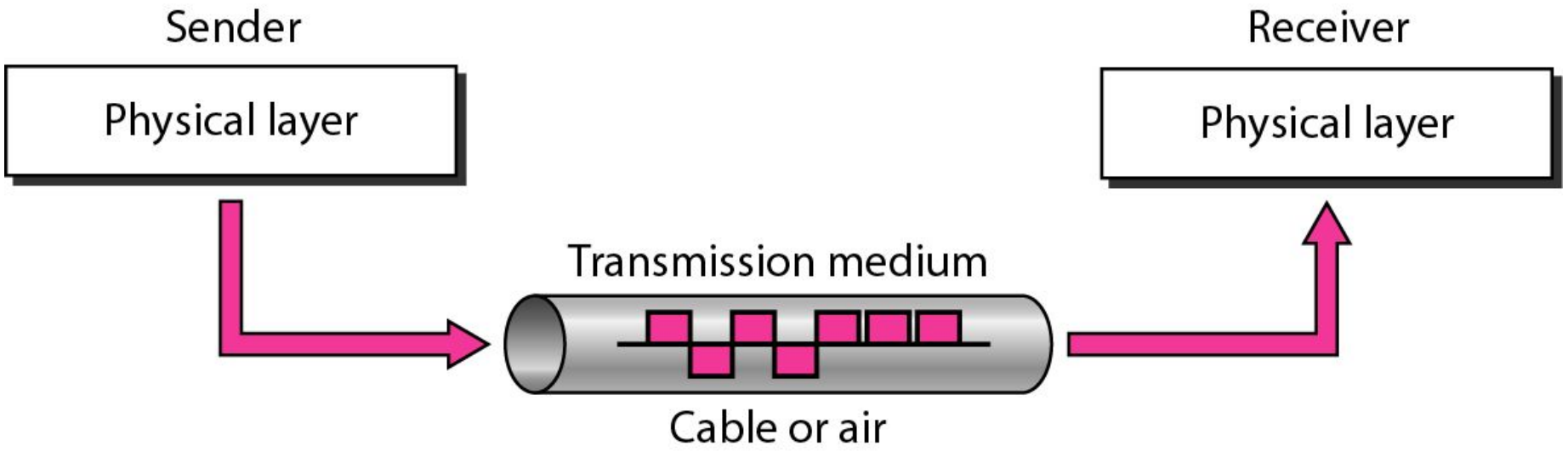
Module 2: Data Links and Transmission

Medium selection for installing cables, Reliable data delivery in industrial environment.

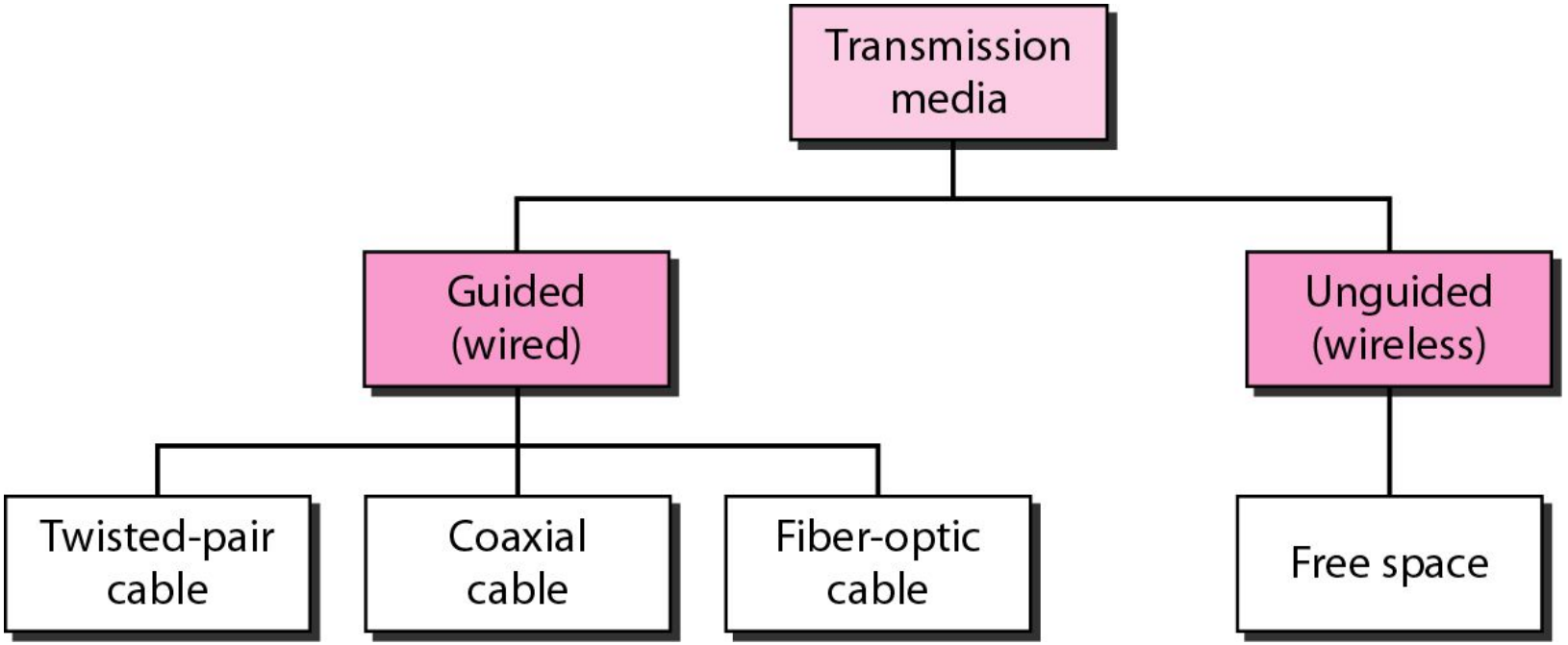
Transmission media, Error detection and correction, block codes, linear block codes, cyclic codes, checksum, Data link control-Framing, Flow and error control, Channels, HDLC and PPP.

Transmission medium and physical layer



- A **transmission medium** can be broadly defined as anything that can carry information from a source to a destination.
- For example, the transmission medium for two people having a dinner conversation is the air. The air can also be used to convey the message in a smoke signal or semaphore.
- For a written message, the transmission medium might be a mail carrier, a truck, or an airplane.
- In **data communications**, the **transmission medium** is usually **free space, metallic cable, or fiber-optic cable**. The information is usually a signal that is the result of a conversion of data from another form.
- In telecommunications, **transmission media can be divided into two broad categories**: guided and unguided.
- **Guided media** include twisted-pair cable, coaxial cable, and fiber-optic cable.
- **Unguided medium** is free space.

Classes of transmission media



GUIDED MEDIA

Guided media, which are those that provide a conduit from one device to another, include twisted-pair cable, coaxial cable, and fiber-optic cable.

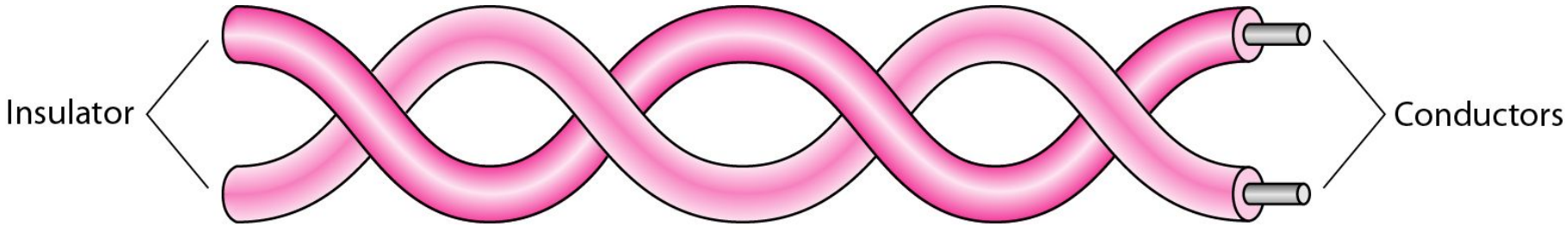
A signal traveling along any of these media is directed and contained by the physical limits of the medium.

Twisted-pair and coaxial cable use metallic (copper) conductors that **accept and transport signals** in the form of **electric current**.

Optical fiber is a cable that accepts and transports signals in the form of **light**.

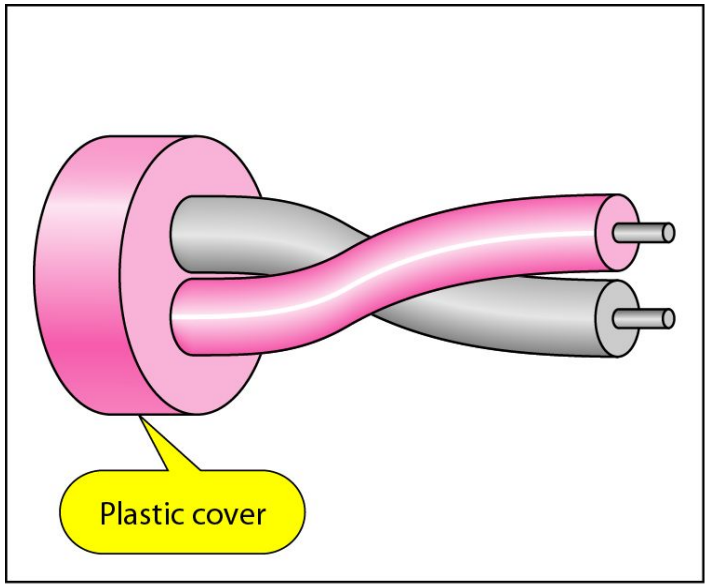
- A **twisted pair** consists of **two conductors** (normally copper), each with its own plastic insulation, twisted together.
- One of the wires is used to **carry signals to the receiver**, and the other is used only as a **ground reference**. The receiver uses the difference between the two.
- In addition to the signal sent by the sender on one of the wires, interference (noise) and crosstalk may affect both wires and create unwanted signals.
- If the two wires are parallel, the effect of these unwanted signals is not the same in both wires because they are at different locations relative to the noise or crosstalk sources (e,g., one is closer and the other is farther). This results in a difference at the receiver.
- By twisting the pairs, a balance is maintained.
- Twisting makes it probable that both wires are equally affected by external influences (noise or crosstalk). This means that the receiver, which calculates the difference between the two, receives no unwanted signals.
- The unwanted signals are mostly canceled out. It is clear that the number of twists per unit of length (e.g., inch) has some effect on the quality of the cable.

Twisted-pair cable

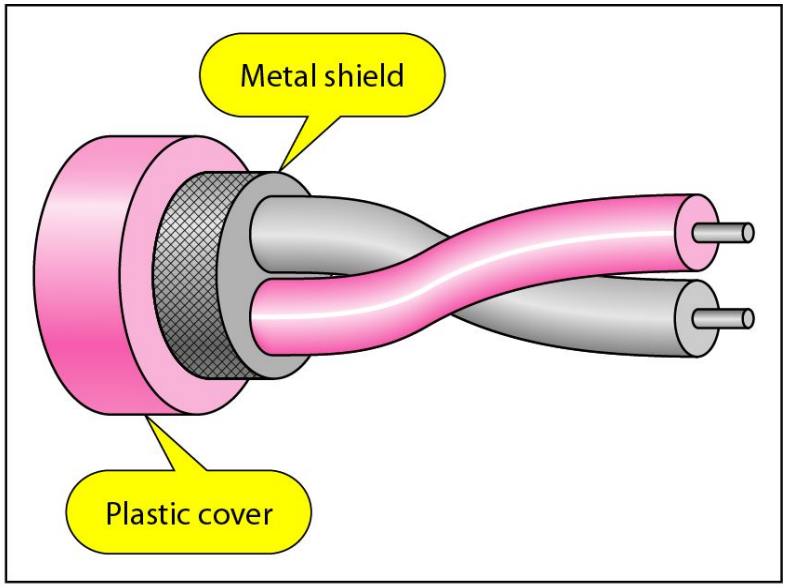


Unshielded Versus Shielded Twisted-Pair Cable

- The most common twisted-pair cable used in communications is referred to as **unshielded twisted-pair (UTP).**
- IBM has also produced a version of twisted-pair cable for its use called **shielded twisted-pair (STP).**
- STP cable has a **metal foil or braided mesh** covering that encases each pair of insulated conductors. Although metal casing improves the quality of cable by preventing the penetration of noise or crosstalk, it is bulkier and more expensive.



a. UTP



b. STP

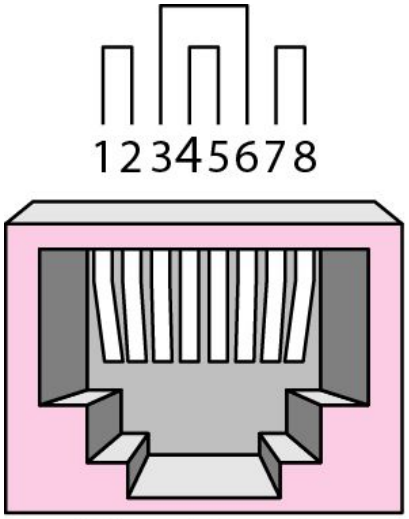
Categories

- The **Electronic Industries Association** (EIA) has developed standards to classify unshielded twisted-pair cable into seven categories.
- Categories are determined by cable quality, with 1 as the lowest and 7 as the highest.
- Each EIA category is suitable for specific uses.

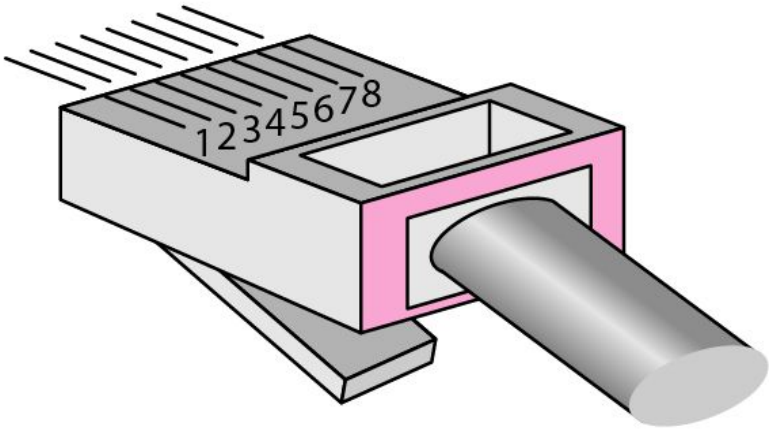
Category	*Specification*	*Data Rate (Mbps)*	*Use*
1	Unshielded twisted-pair used in telephone	< 0.1	Telephone
2	Unshielded twisted-pair originally used in T-lines	2	T-1 lines
3	Improved CAT 2 used in LANs	10	LANs
4	Improved CAT 3 used in Token Ring networks	20	LANs
5	Cable wire is normally 24 AWG with a jacket and outside sheath	100	LANs
5E	An extension to category 5 that includes extra features to minimize the crosstalk and electromagnetic interference	125	LANs
6	A new category with matched components coming from the same manufacturer. The cable must be tested at a 200-Mbps data rate.	200	LANs
7	Sometimes called SSTP (shielded screen twisted-pair). Each pair is individually wrapped in a helical metallic foil followed by a metallic foil shield in addition to the outside sheath. The shield decreases the effect of crosstalk and increases the data rate.	600	LANs

Connectors

- The most common UTP connector is RJ45 (RJ stands for registered jack),
- The RJ45 is a keyed connector, meaning the connector can be inserted in only one way.



RJ-45 Female

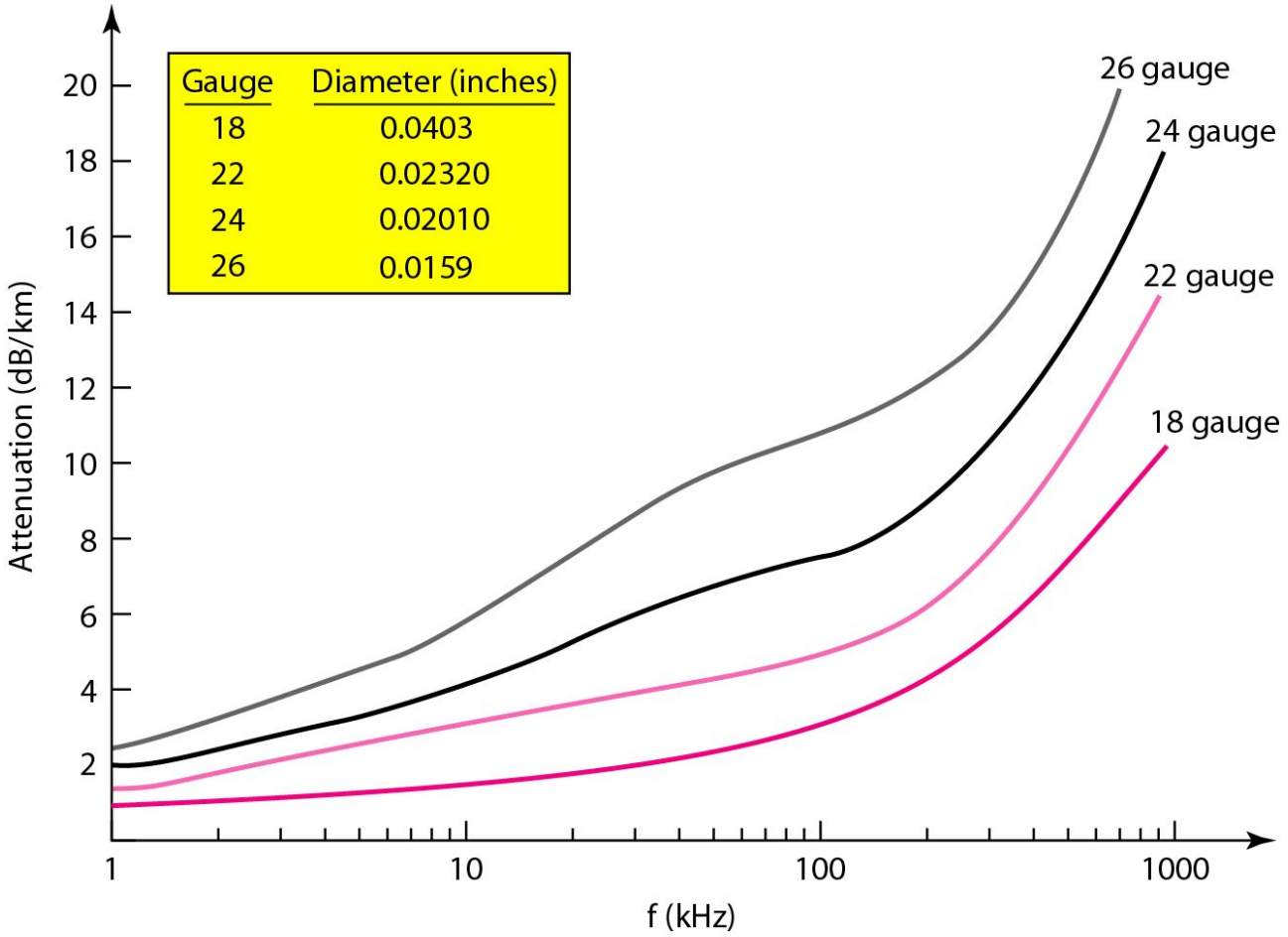


RJ-45 Male

Applications

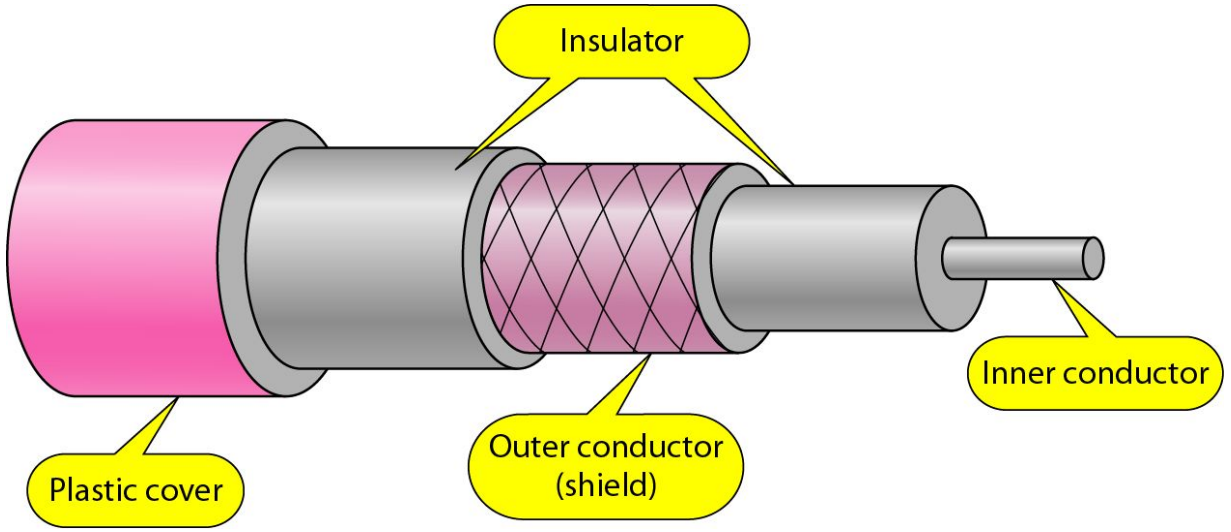
- Twisted-pair cables are used in telephone lines to provide voice and data channels.
- The local loop-the line that connects **subscribers to the central telephone office**---commonly consists of **unshielded twisted-pair cables**.
- The DSL lines that are used by the telephone companies to provide **high-data-rate** connections also use the **high-bandwidth** capability of unshielded twisted-pair cables.
- **Local-area networks**, such as lOBase-T and lOOBase-T, also use twisted-pair cables.

Figure 7.6 *UTP performance*



Coaxial Cable

- Coaxial cable (or *coax)* carries signals of higher frequency ranges than those in twisted pair cable.
- Coax has a central core conductor of solid or stranded wire (usually copper) enclosed in an insulating sheath, which is, in turn, encased in an outer conductor of metal foil, braid, or a combination of the two.
- The outer metallic wrapping serves both as a shield against noise and as the second conductor, which completes the circuit.
- This outer conductor is also enclosed in an insulating sheath, and the whole cable is protected by a plastic cover.



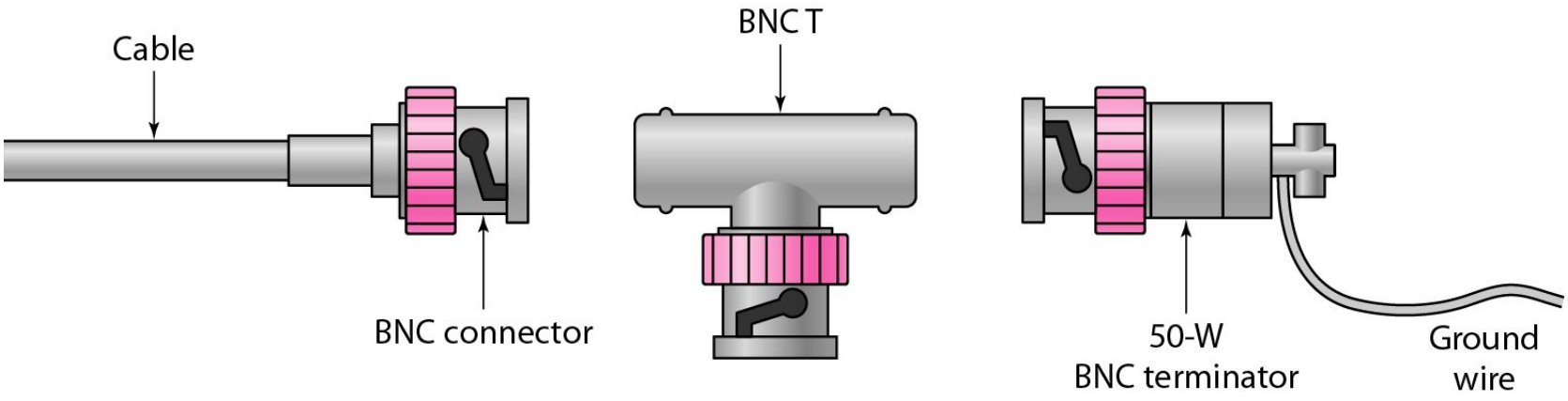
Coaxial Cable Standards

- Coaxial cables are categorized by their **radio government (RG) ratings**.
- Each RG number denotes **a unique set of physical specifications**, including the **wire gauge of the inner conductor**, the **thickness and type of the inner insulator**, the **construction of the shield**, and the **size and type of the outer casing.**

Category	*Impedance*	*Use*
RG-59	75 Ω	Cable TV
RG-58	50 Ω	Thin Ethernet
RG-11	50 Ω	Thick Ethernet

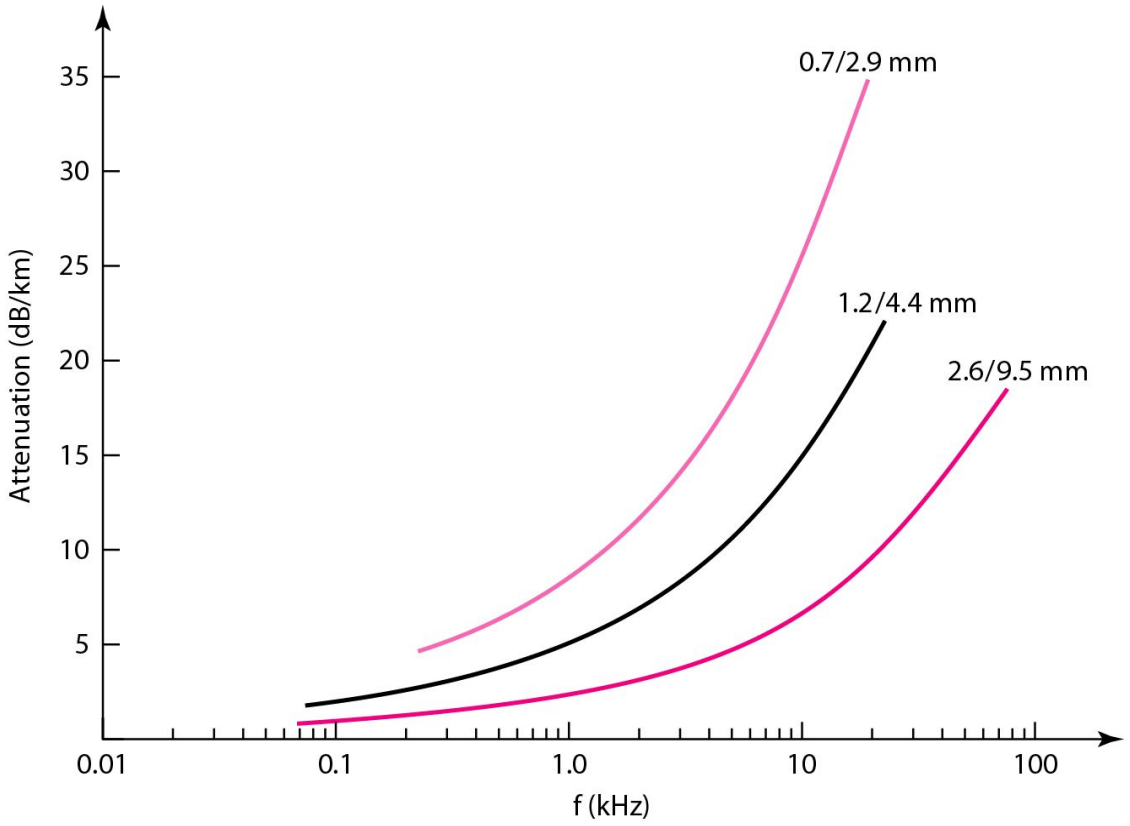
Coaxial Cable Connectors

- To connect coaxial cable to devices, we need coaxial connectors.
- The most **common type of connector** used today is the **Bayone-Neill-Concelman** (BNe), connector.
- Three popular types of these connectors: the BNC connector, the BNC T connector, and the BNC terminator.
- The **BNC connector** is used **to connect the end of the cable to a device**, such as a TV set.
- The **BNC T connector** is used in **Ethernet networks** to branch out to a connection to a computer or other device.
- The **BNC terminator** is used at the **end of the cable** to **prevent the reflection of the signal**.



Performance

- The **attenuation** is much **higher** in coaxial cables than in twisted-pair cable.
- **Coaxial cable** has a **higher bandwidth**, the signal weakens rapidly and requires the frequent use of repeaters.

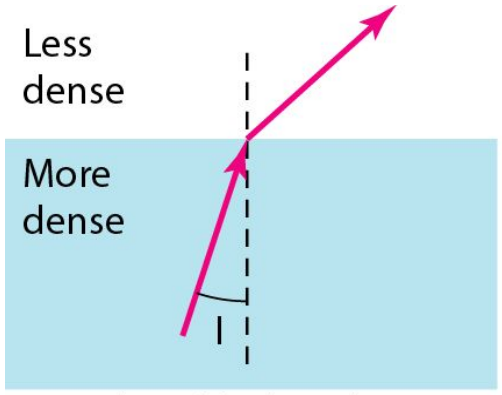


Applications

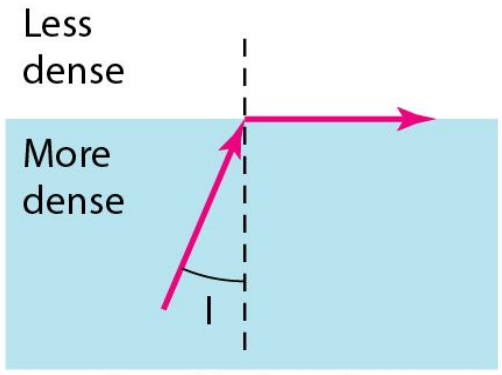
- Coaxial cable was widely used in **analog telephone networks** where a single coaxial network could carry 10,000 voice signals.
- Later it was used in **digital telephone networks** where a single coaxial cable could carry digital data up to 600 Mbps.
- Coaxial cable in telephone networks has largely been replaced today with fiber-optic cable.
- **Cable TV networks** also use coaxial cables.
- In the traditional cable TV network, the entire network used coaxial cable. Later, however, cable TV providers replaced most of the media with fiber-optic cable; hybrid networks use coaxial cable only at the network boundaries, near the consumer premises.
- Cable TV uses RG-59 coaxial cable.
- Another common application of coaxial cable is in **traditional Ethernet LANs**.
- Because of its high bandwidth, and consequently high data rate, coaxial cable was chosen for digital transmission in early Ethernet LANs.
- The 10Base-2, or Thin Ethernet, uses RG-58 coaxial cable with BNe connectors to transmit data at 10 Mbps with a range of 185 m.
- The 10Base5, or Thick Ethernet, uses RG-11 (thick coaxial cable) to transmit 10 Mbps with a range of 5000 m.
- Thick Ethernet has specialized connectors.

Fiber-Optic Cable

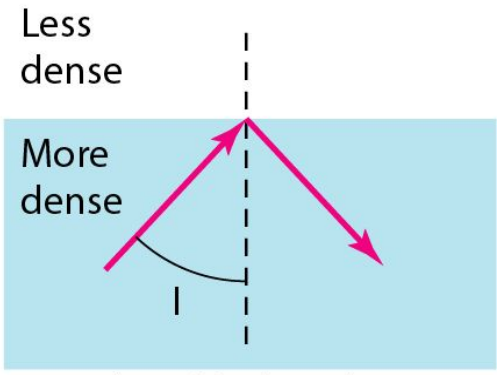
- A fiber-optic cable is made of glass or plastic and transmits signals in the form of light.
- Light travels in a straight line as long as it is moving through a single uniform substance.
- If a ray of light traveling through one substance suddenly enters another substance (of a different density), the ray changes direction.
- If the **angle of incidence I** (the angle the ray makes with the line perpendicular to the interface between the two substances **<** **the critical angle**, the **ray refracts and moves closer to the surface.**
- If the **angle of incidence = critical angle**, the **light bends along the interface**.
- If the **angle of incidence > critical angle**, the **ray reflects** (makes a turn) and **travels again in the denser substance.**
- The **critical angle** is a property of the substance, and its value differs from one substance to another.



I < critical angle, refraction



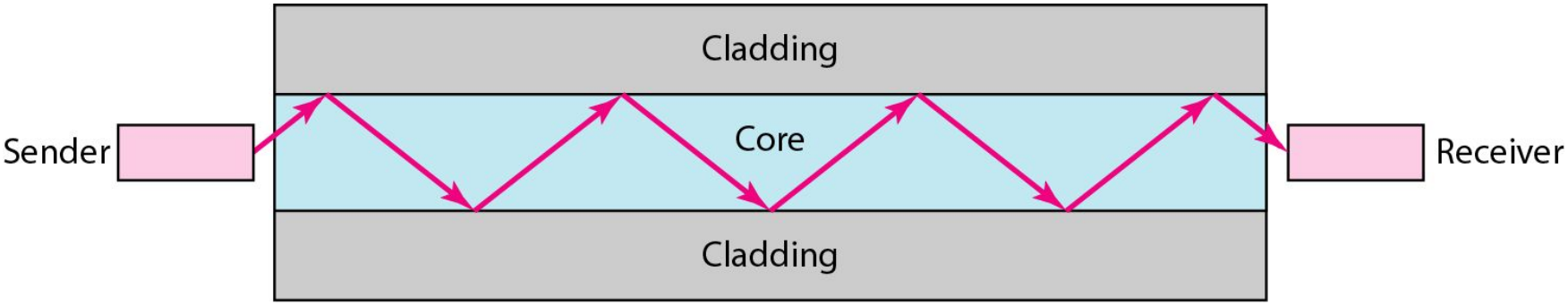
I = critical angle, refraction



I > critical angle, reflection

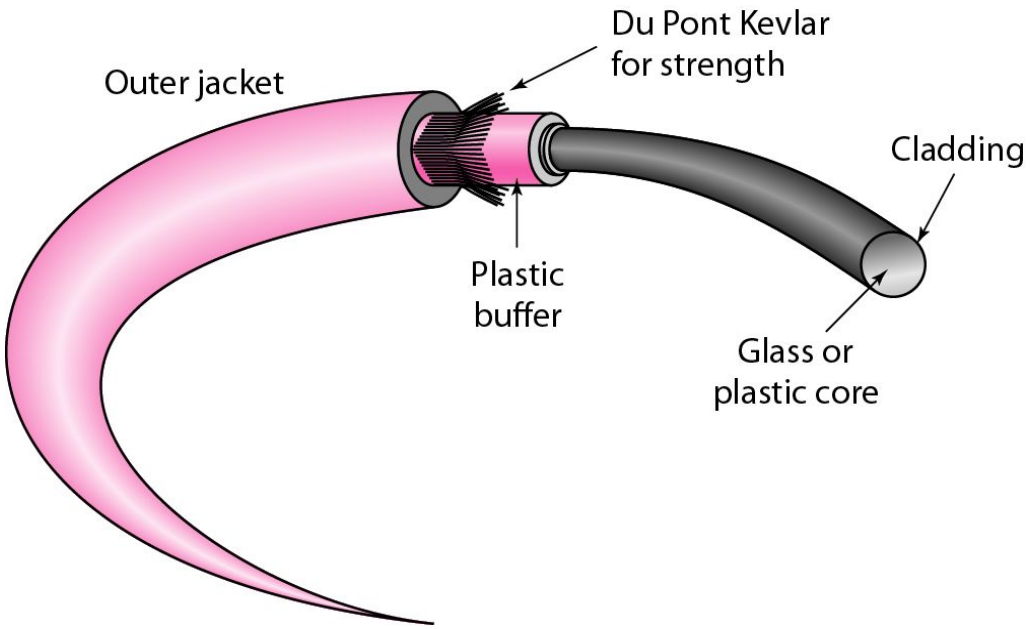
Optical Fibers

- Optical fibers use reflection to guide light through a channel.
- A glass or plastic core is surrounded by a cladding of less dense glass or plastic.
- The difference in density of the two materials must be such that a beam of light moving through the core is reflected off the cladding instead of being refracted into it.



Cable composition

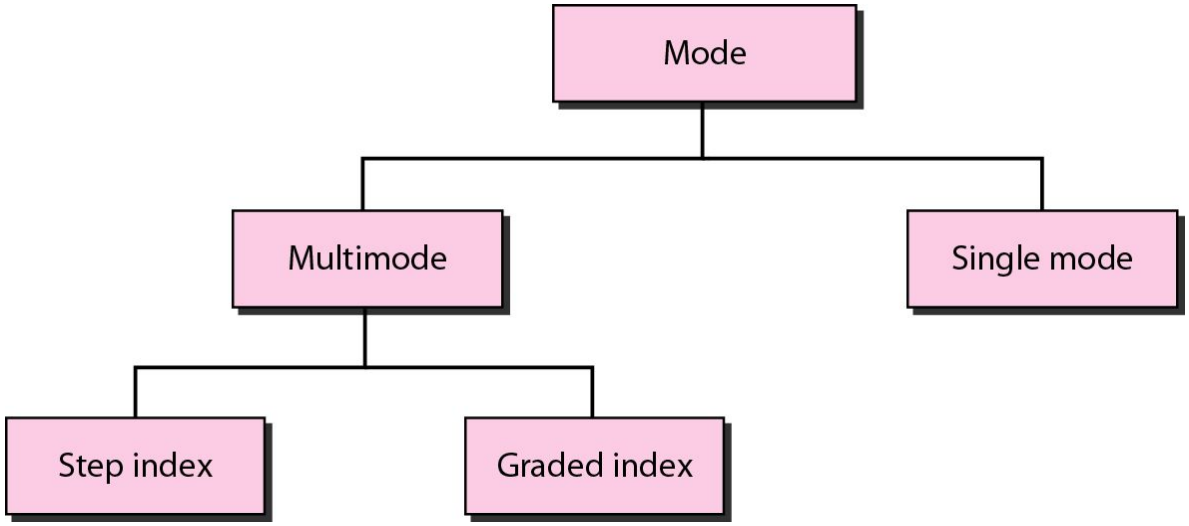
- The **outer jacket** is made of either **PVC or Teflon**.
- **Inside the jacket** are **Kevlar strands to strengthen the cable**. Kevlar is a strong material used in the **fabrication of bulletproof vests**.
- Below the Kevlar is another plastic coating to cushion the fiber. The fiber is at the center of the cable, and it consists of cladding and core.



Propagation Modes

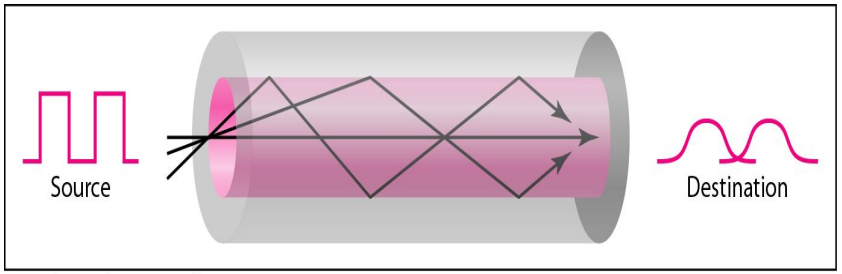
- Current technology supports two modes (**multimode and single mode**) for propagating light along optical channels, each requiring fiber with different physical characteristics.

  Multimode can be implemented in two forms: **step-index or graded-index**

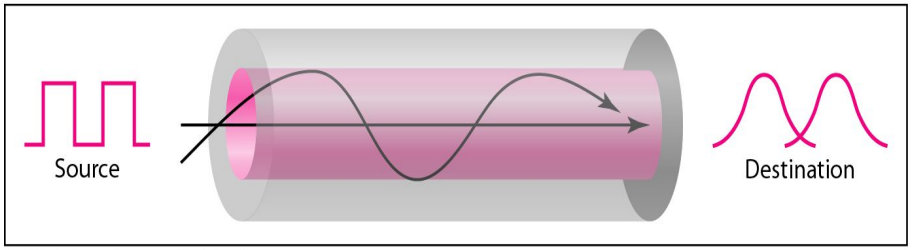


Multimode

- Multimode is so named because multiple beams from a light source move through the core in different paths. How these beams move within the cable depends on the structure of the core.
- In **multimode step-index fiber**, the **density of the core remains constant from the center to the edges.**
- A beam of light moves through this constant density in a straight line until it reaches the interface of the core and the cladding. At the interface, there is an abrupt change due to a lower density; this alters the angle of the beam's motion.
- The term ***step index*** refers to the **suddenness of this change**, which contributes to the **distortion of the signal as it passes through the fiber.**
- A second type of fiber, called **multimode graded-index fiber**, **decreases this distortion** of the signal through the cable.
- The word *index* here refers to the index of refraction. The index of refraction is related to density.
- A graded-index fiber is one with varying densities. **Density is highest at the center** of the core and decreases gradually to its lowest at the edge.



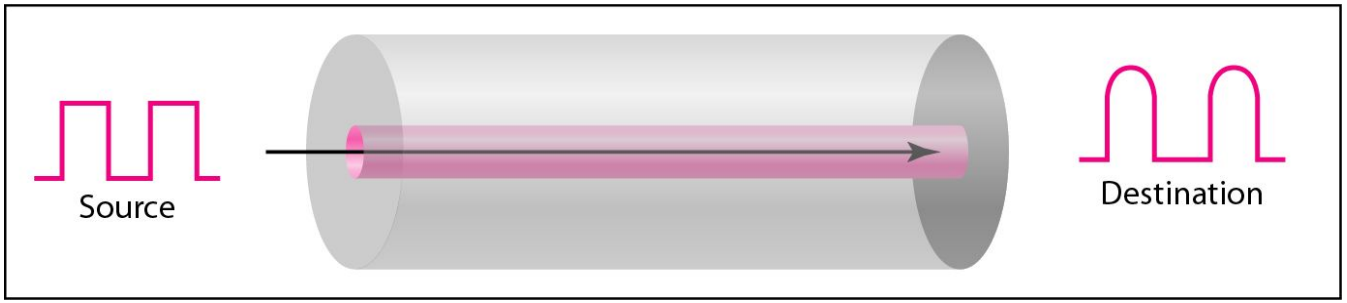
a. Multimode, step index



b. Multimode, graded index

Single-Mode

- Single-mode uses **step-index fiber** and a highly focused source of light that limits beams to a small range of angles, all close to the horizontal.
- The single mode fiber itself is manufactured with a much smaller diameter than that of multimode fiber, and with substantially lower density (index of refraction).
- The decrease in density results in a critical angle that is close enough to 90° to make the propagation of beams almost horizontal.
- In this case, propagation of different beams is almost identical, and delays are negligible.
- All the beams arrive at the destination "together" and can be recombined with little distortion to the signal.



c. Single mode

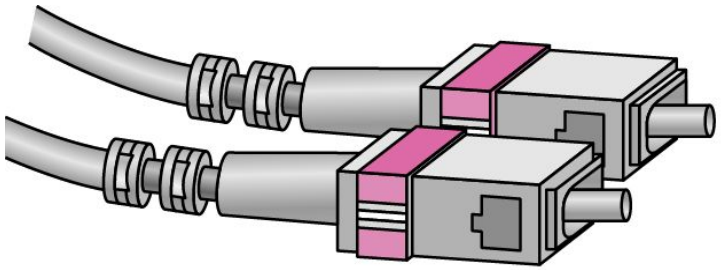
Fiber sizes

- **Optical fibers** are defined by the **ratio of the diameter of their core** to the **diameter of their cladding**, are expressed in micrometers.
- The last size listed is for single-mode only.

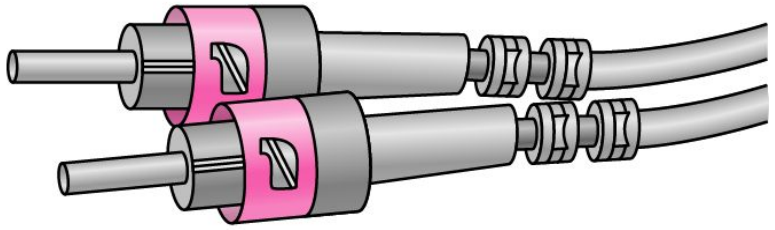
Type	*Core (μm)*	*Cladding (μm)*	*Mode*
50/125	50.0	125	Multimode, graded index
62.5/125	62.5	125	Multimode, graded index
100/125	100.0	125	Multimode, graded index
7/125	7.0	125	Single mode

Fiber-Optic Cable Connectors

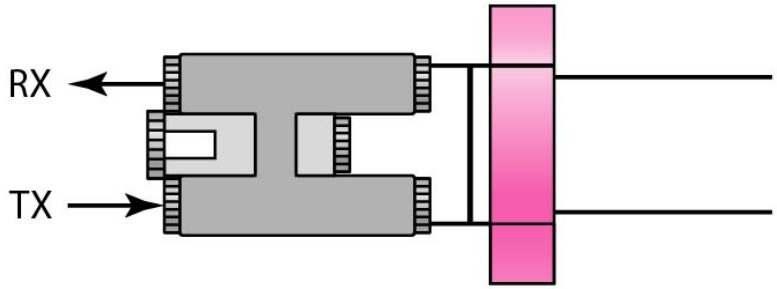
- There are **three types of connectors** for fiber-optic cables:
- The **subscriber channel** (SC) **connector** is used for cable TV. It uses a push/pull locking system.
- The **straight-tip** (ST) **connector** is used for connecting cable to networking devices. It uses a bayonet locking system and is more reliable than SC.
- **MT-RJ** is a connector that is the same size as RJ45.



SC connector



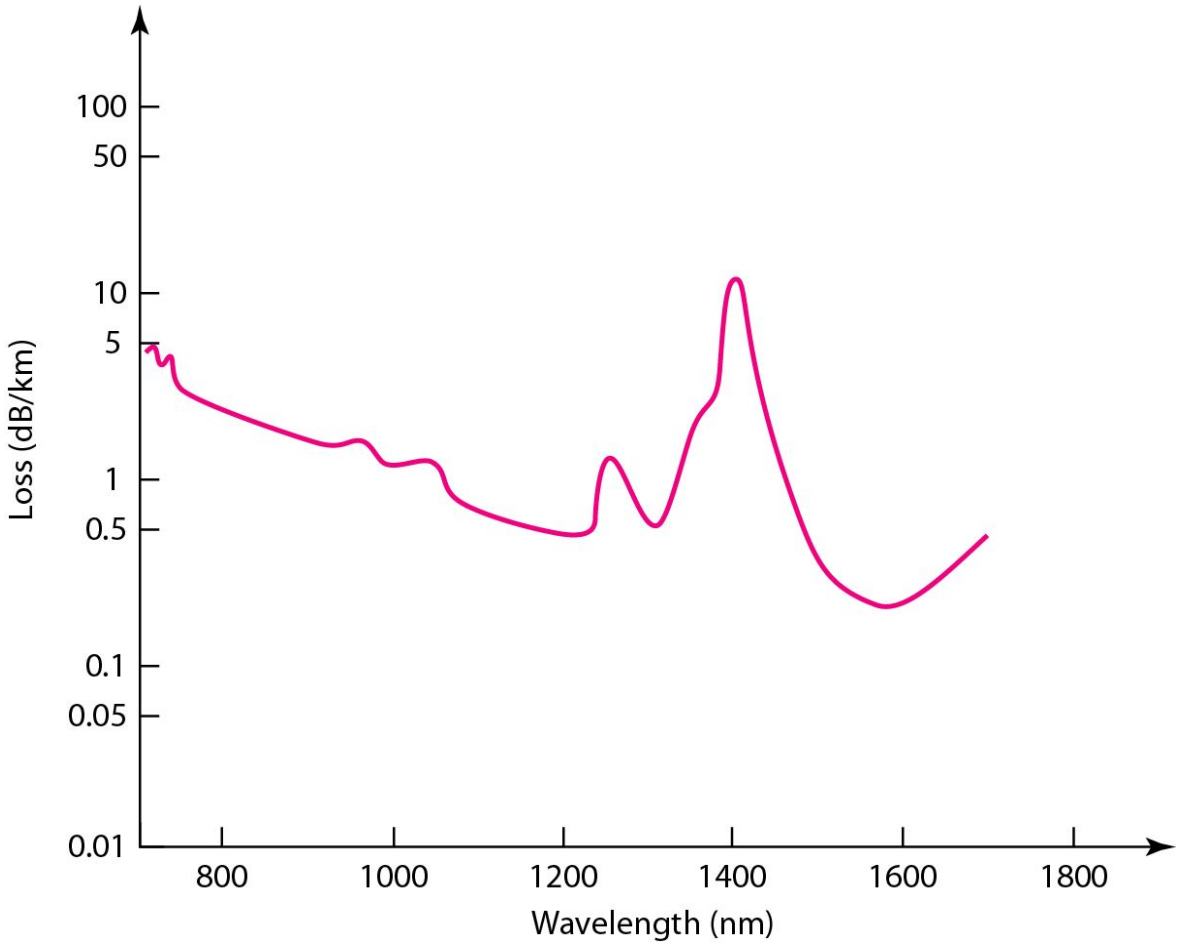
ST connector



MT-RJ connector

Performance

- Attenuation is flatter than in the case of twisted-pair cable and coaxial cable.
- The performance is such that fewer (actually 10 times less) repeaters are needed when fiber-optic cables are used.



Applications

- Fiber-optic cable is often found in backbone networks because its wide bandwidth is cost-effective. Today, with wavelength-division multiplexing (WDM), we can transfer data at a rate of 1600 Gbps.
- Some cable TV companies use a combination of optical fiber and coaxial cable, thus creating a hybrid network.
- Optical fiber provides the backbone structure while coaxial cable provides the connection to the user premises. This is a cost-effective configuration since the narrow bandwidth requirement at the user end does not justify the use of optical fiber.
- Local-area networks such as 100Base-FX network (Fast Ethernet) and 1000Base-X also use fiber-optic cable.

Advantages

Fiber-optic cable has several advantages over metallic cable (twisted pair or coaxial).

- **Higher bandwidth**. Fiber-optic cable can support dramatically higher bandwidths (and hence data rates) than either twisted-pair or coaxial cable. Currently, data rate and bandwidth utilization over fiber-optic cable are limited not by the medium but by the signal generation and reception technology available.
- **Less signal attenuation**. Fiber-optic transmission distance is significantly greater than that of other guided media. A signal can run for 50 km without requiring regeneration. We need repeaters every 5 km for coaxial or twisted-pair cable.
- **Immunity to electromagnetic interference**. Electromagnetic noise cannot affect fiber-optic cables.
- **Resistance to corrosive materials**. Glass is more resistant to corrosive materials than copper.
- **Light weight**. Fiber-optic cables are much lighter than copper cables.
- **Greater immunity to tapping**. Fiber-optic cables are more immune to tapping than copper cables. Copper cables create antenna effects that can easily be tapped.

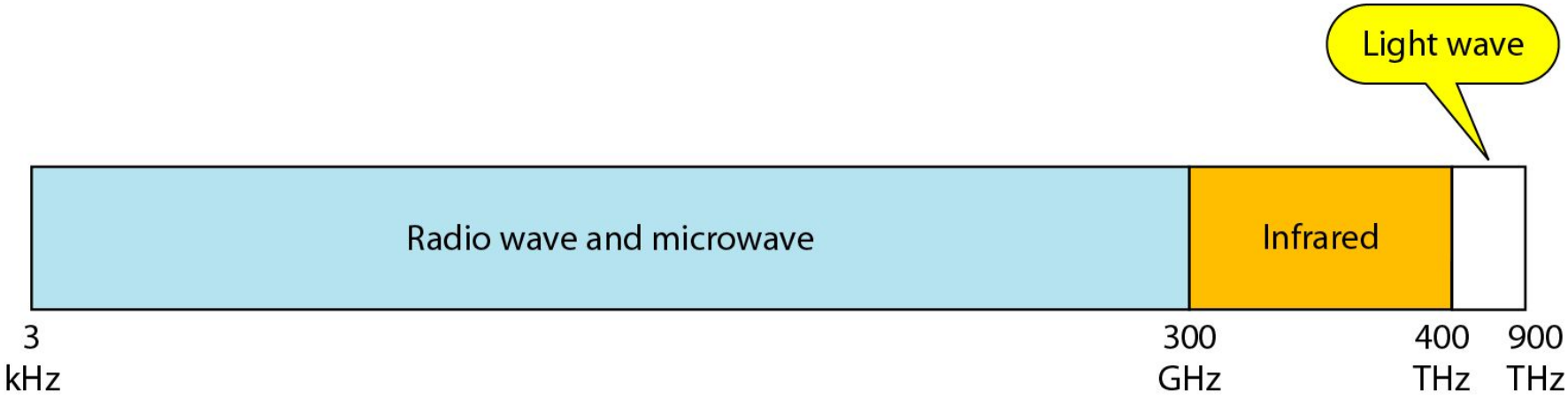
Disadvantages

- **Installation and maintenance**. Fiber-optic cable is a relatively new technology. Its installation and maintenance require expertise that is not yet available everywhere.
- **Unidirectional light propagation**. Propagation of light is unidirectional. If bidirectional communication is needed, two fibers are needed.
- **Cost**. The cable and the interfaces are relatively more expensive than those of other guided media. If the demand for bandwidth is not high, often the use of optical fiber cannot be justified.

UNGUIDED MEDIA: WIRELESS

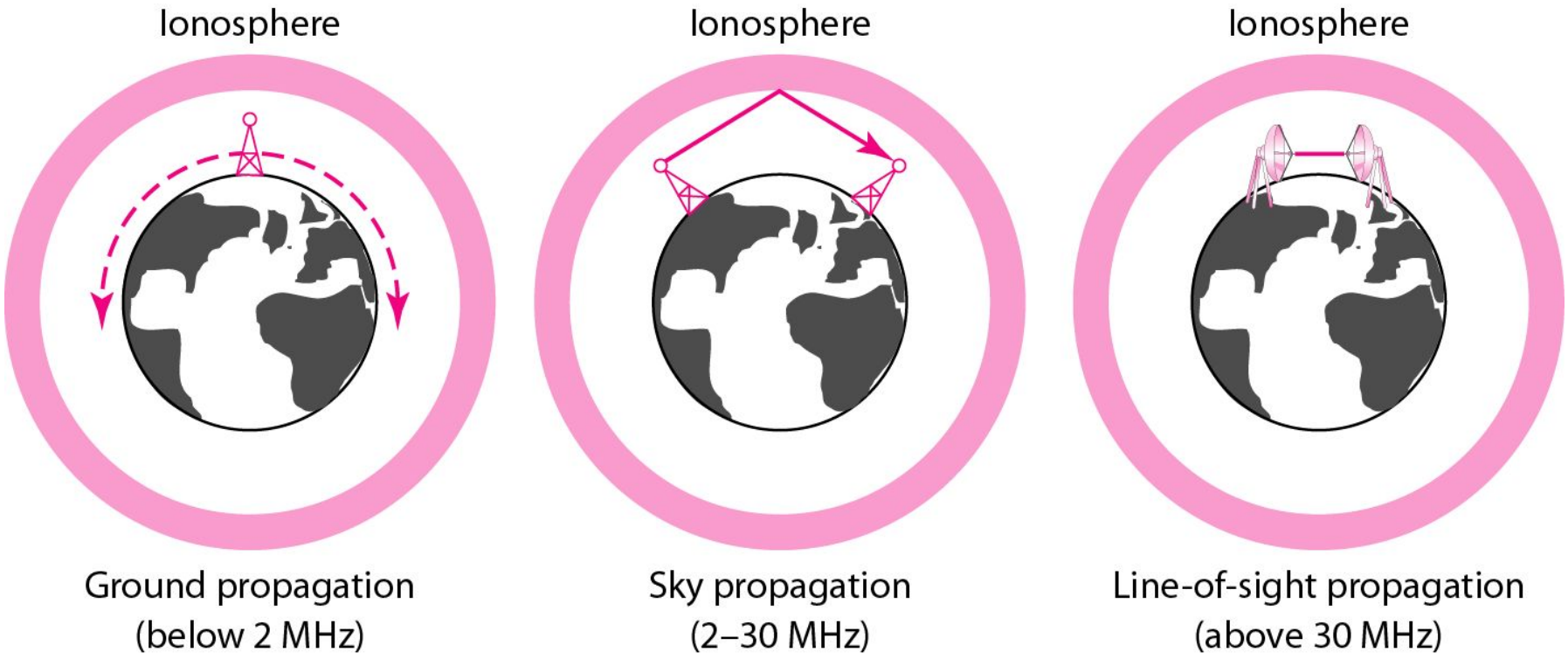
Unguided media transport electromagnetic waves without using a physical conductor.

This type of communication is often referred to as wireless communication.



- Unguided signals can travel from the source to destination in several ways:
  ground propagation, sky propagation, and line-of-sight propagation
- In **ground propagation**, **radio waves** travel through the **lowest portion of the atmosphere**, hugging the earth.
- These **low-frequency signals emanate in all directions** from the **transmitting antenna** and **follow the curvature of the planet**.
- **Distance depends on the amount of power in the signal**: The **greater the power**, the **greater the distance**.
- In **sky propagation**, **higher-frequency radio waves radiate upward into the ionosphere** (the layer of atmosphere where particles exist as ions) where they are reflected back to earth.
- This type of transmission allows for **greater distances with lower output power.**
- In **line-of-sight propagation**, **very high-frequency signals** are transmitted **in straight lines directly from antenna to antenna.**
- Antennas must be **directional**, facing each other and either tall enough or close enough together not to be affected by the curvature of the earth.
- Line-of-sight propagation is tricky because radio transmissions cannot be completely focused.

Propagation methods



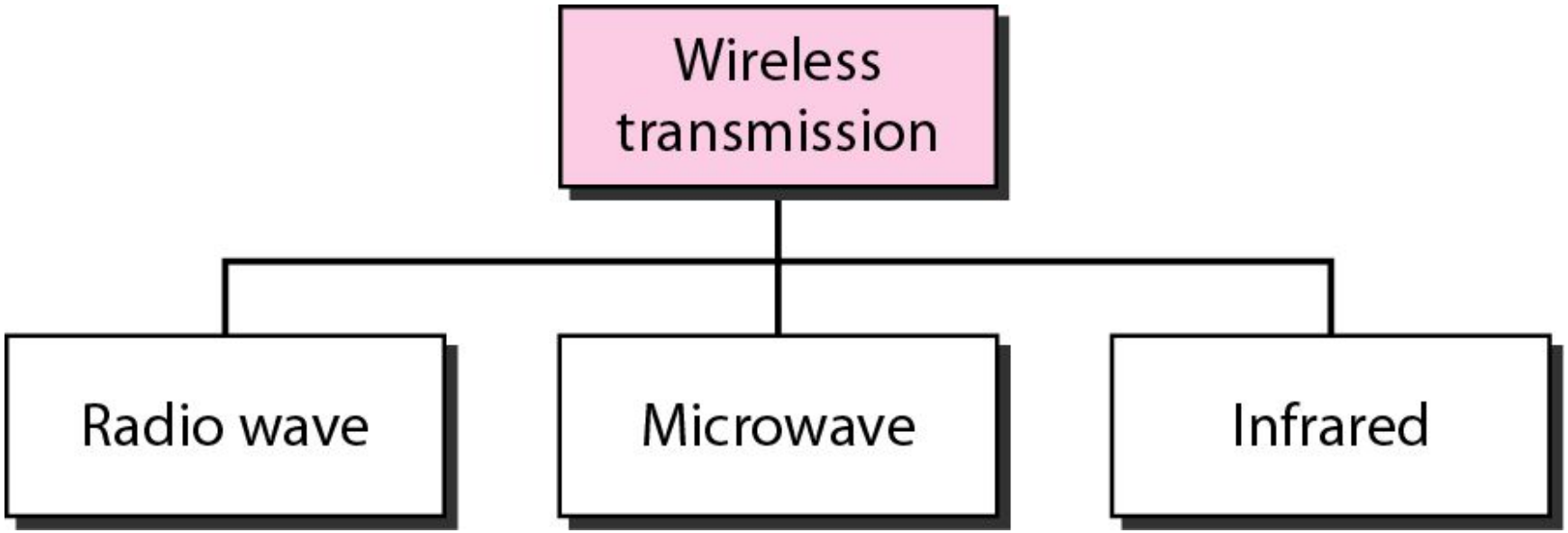
Bands

- The section of the electromagnetic spectrum defined as radio waves and microwaves is divided into eight ranges, called *bands,* regulated by government authorities.
- These bands are rated from **very low frequency (VLF) to extremely high frequency (EHF).**

Band	*Range*	*Propagation*	*Application*
VLF (very low frequency)	3–30 kHz	Ground	Long-range radio navigation
LF (low frequency)	30–300 kHz	Ground	Radio beacons and navigational locators
MF (middle frequency)	300 kHz–3 MHz	Sky	AM radio
HF (high frequency)	3–30 MHz	Sky	Citizens band (CB), ship/aircraft communication
VHF (very high frequency)	30–300 MHz	Sky and line-of-sight	VHF TV, FM radio
UHF (ultrahigh frequency)	300 MHz–3 GHz	Line-of-sight	UHF TV, cellular phones, paging, satellite
SHF (superhigh frequency)	3–30 GHz	Line-of-sight	Satellite communication
EHF (extremely high frequency)	30–300 GHz	Line-of-sight	Radar, satellite

Wireless Transmission

- Wireless transmission are divided into three broad groups:
  - radio waves, microwaves, and infrared waves

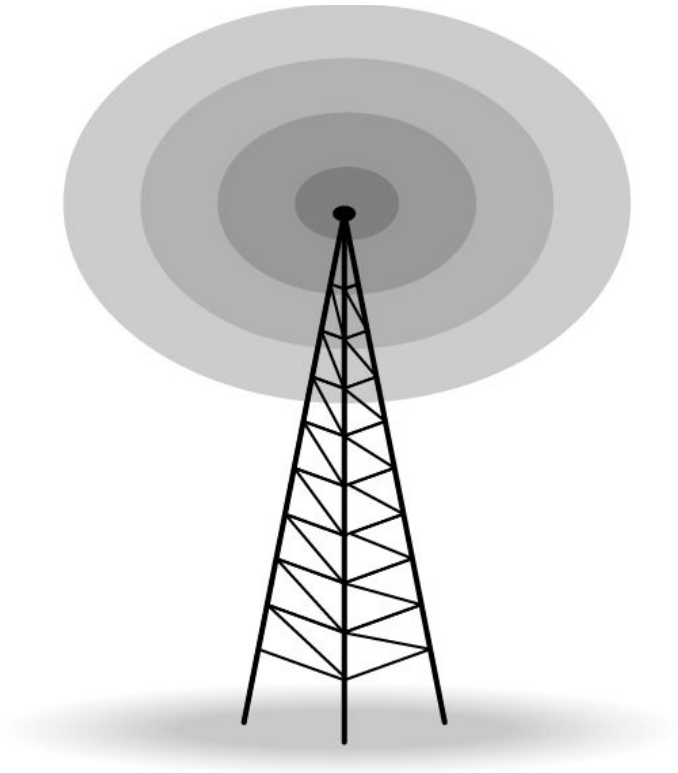


Radio Waves

- **Electromagnetic waves** ranging in frequencies between **3 kHz and 1 GHz** are normally called **radio waves.**
- **Waves** ranging in frequencies between **1 and 300 GHz** are called **microwaves**.
- Radio waves are **omnidirectional**. When an **antenna transmits radio waves**, they are **propagated in all directions**. This means that the **sending and receiving antennas do not have to be aligned**. A sending antenna sends waves that can be received by any receiving antenna.
- The omnidirectional property has a disadvantage - The radio waves transmitted by one antenna are susceptible to interference by another antenna that may send signals using the same frequency or band.
- Radio waves, particularly those waves that propagate in the **sky mode**, can travel long distances. This makes radio waves a good candidate for long-distance broadcasting such as AM radio.
- Radio waves of **low and medium frequencies**, can penetrate walls. This characteristic can be both an advantage and a disadvantage. It is an advantage because, for example, an AM radio can receive signals inside a building. It is a disadvantage because we cannot isolate a communication to just inside or outside a building.
- The **radio wave band is relatively narrow, just under 1 GHz**, compared to the microwave band. When this band is divided into subbands, the subbands are also narrow, leading to a low data rate for digital communications.

Omnidirectional Antenna

- Radio waves use **omnidirectional antennas** that send out signals in all directions.
- Based on the wavelength, strength, and the purpose of transmission, we can have several types of antennas.



Applications

- The omnidirectional characteristics of radio waves make them useful for **multicasting**, in which there is **one sender but many receivers**.
- **AM and FM radio, television, maritime radio, cordless phones, and paging** are examples of multicasting.

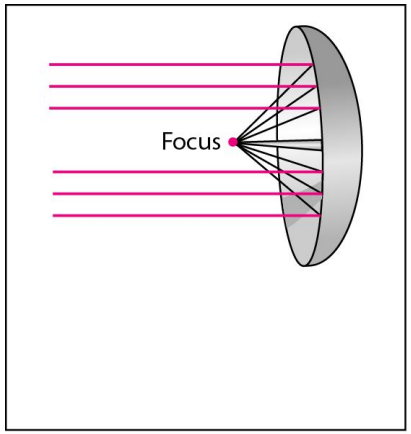
Radio waves are used for multicast communications, such as radio and television, and paging systems.
They can penetrate through walls.
Highly regulated.
Use omni directional antennas.

Microwaves

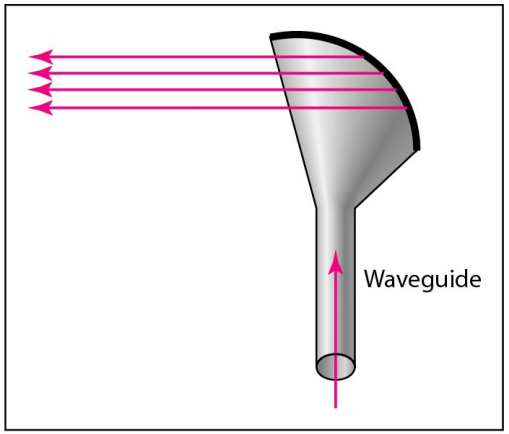
- **Electromagnetic waves** having frequencies between **1 and 300 GHz** are called **microwaves**.
- Microwaves are **unidirectional**. The unidirectional property has an advantage - **A pair of antennas can be aligned** without interfering with another pair of aligned antennas.
- When an antenna transmits microwave waves, they can be narrowly focused. This means that the sending and receiving antennas need to be aligned.
- The following describes some characteristics of microwave propagation:
  - Microwave propagation is **line-of-sight.** Since the towers with the mounted antennas need to be in direct sight of each other, towers that are far apart need to be very tall.
  - The curvature of the earth as well as other blocking obstacles do not allow two short towers to communicate by using microwaves.
  - **Repeaters** are often needed for long distance communication.
  - **Very high-frequency microwaves cannot penetrate walls**. This characteristic can be a disadvantage if receivers are inside buildings.
  - The microwave band is relatively wide, almost 299 GHz. Therefore wider subbands can be assigned, and a high data rate is possible.
  - Use of certain portions of the band requires permission from authorities.

Unidirectional Antenna

- Microwaves need unidirectional antennas that send out signals in one direction.
- Two types of antennas are used for microwave communications:
- the parabolic dish and the horn.
- A parabolic dish antenna is based on the **geometry of a parabola**: Every **line parallel** to the **line of symmetry** (line of sight) reflects off the curve at angles such that all the lines intersect in a common point called the **focus**. The parabolic dish works as a funnel, catching a wide range of waves and directing them to a common point. In this way, more of the signal is recovered than would be possible with a single-point receiver.
- Outgoing transmissions are broadcast through a horn aimed at the dish. The microwaves hit the dish and are deflected outward in a reversal of the receipt path.
- A horn antenna looks like a gigantic scoop. Outgoing transmissions are broadcast up a stem (resembling a handle) and deflected outward in a series of narrow parallel beams by the curved head. Received transmissions are collected by the scooped shape of the horn, in a manner similar to the parabolic dish, and are deflected down into the stem.



a. Dish antenna



b. Horn antenna

Note

Microwaves are used for unicast communication such as cellular telephones, satellite networks, and wireless LANs.
Higher frequency ranges cannot penetrate walls.
Use directional antennas - point to point line of sight communications.

Infrared

- **Infrared waves**, with frequencies from **300 GHz to 400 THz** (wavelengths from 1 mm to 770 nm), can be used for short-range communication.
- Infrared waves, **having high frequencies, cannot penetrate walls**. This advantageous characteristic **prevents interference** between one system and another; a short-range communication system in one room cannot be affected by another system in the next room.
- When we use our infrared remote control, we do not interfere with the use of the remote by our neighbors.
- However, this characteristic makes infrared signals useless for long-range communication.
- In addition, we cannot use infrared waves outside a building because the sun's rays contain infrared waves that can interfere with the communication.

Applications

- The infrared band, almost 400 THz, has an excellent potential for data transmission. Such a wide bandwidth can be used to transmit digital data with a very high data rate.
- The ***Infrared Data Association*** **(IrDA),** an association for sponsoring the use of infrared waves, has established standards for using these signals for communication between devices such as keyboards, mice, PCs, and printers. For example, some manufacturers provide a special port called the IrDA port that allows a wireless keyboard to communicate with a PC.
- The standard originally **defined a data rate of 75 kbps for a distance up to 8 m**. The recent standard defines a data rate of **4 Mbps**.
- Infrared signals defined by IrDA transmit through line of sight; the IrDA port on the keyboard needs to point to the PC for transmission to occur.

Note

Infrared signals can be used for short-range communication in a closed area using line-of-sight propagation.

Wireless Channels

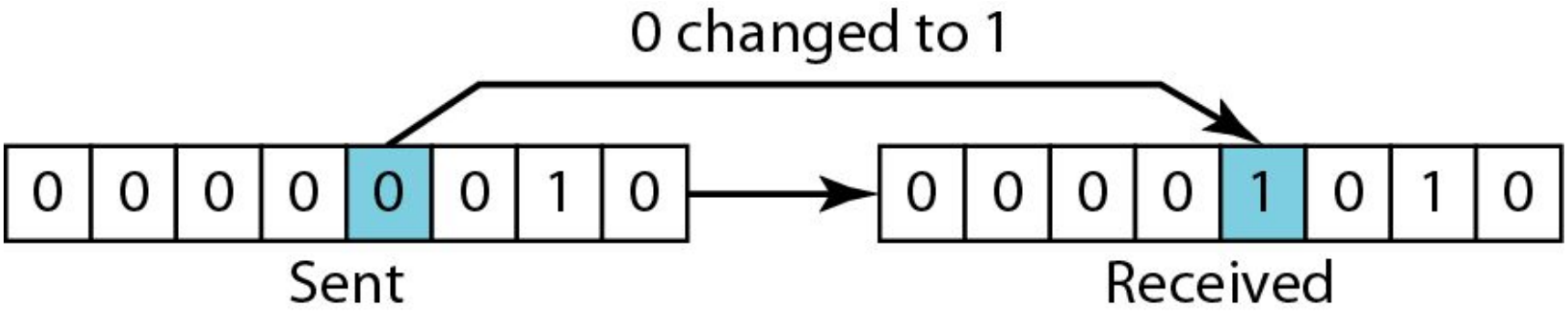
- Are subject to a lot more errors than guided media channels.
- Interference is one cause for errors, can be circumvented with high SNR.
- The higher the SNR the less capacity is available for transmission due to the broadcast nature of the channel.
- Channel also subject to fading and no coverage holes.

Error Detection and Correction

- Data can be corrupted during transmission.
- Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference. This interference can change the shape of the signal.

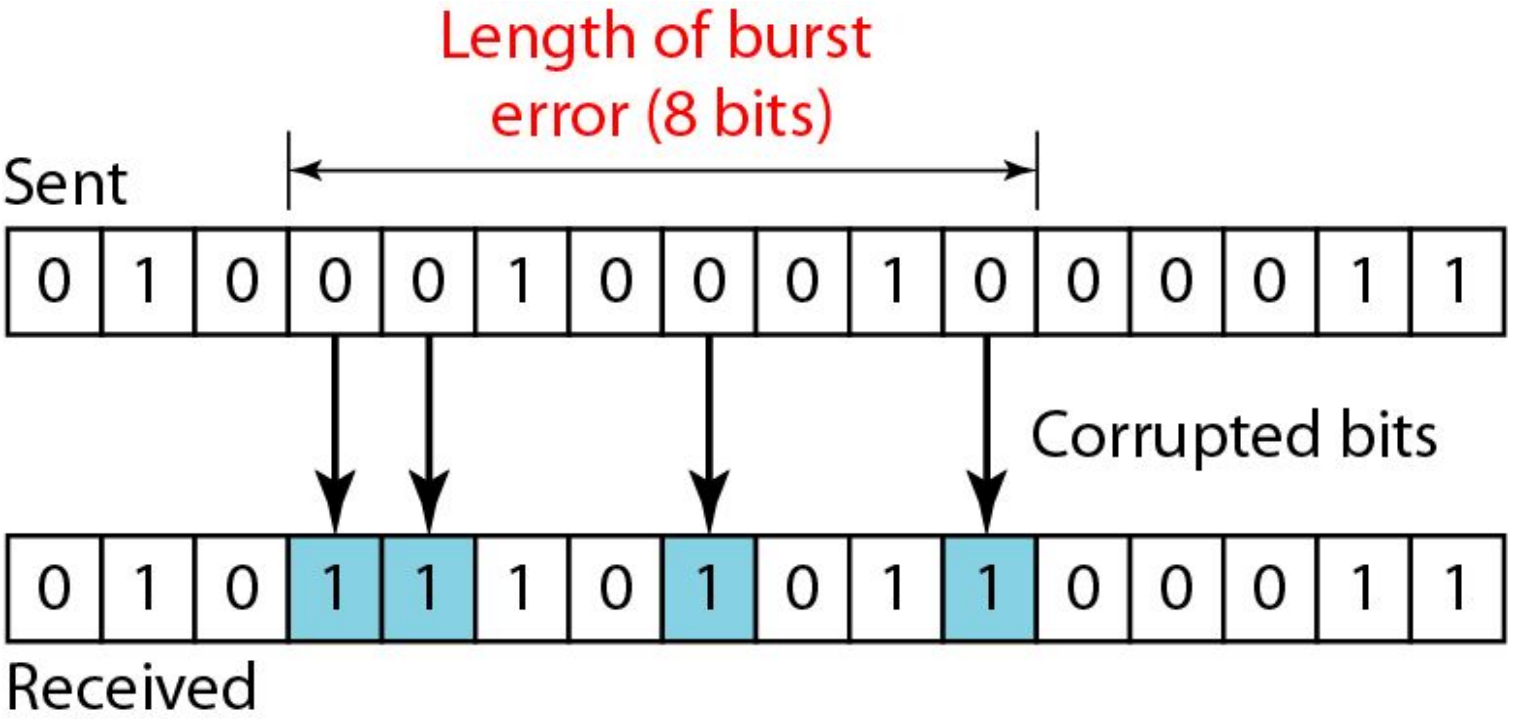
Single-Bit Error

- The term ***single-bit error*** means that only 1 bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or from 0 to 1.
- Single-bit errors are the least likely type of error in serial data transmission.
- For a single-bit error to occur, the noise must have a duration of only 1μs, which is very rare; noise normally lasts much longer than this



Burst error

- The term **burst error** means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1.
- A burst error is more likely to occur than a single-bit error.
- The duration of noise is normally longer than the duration of 1 bit, which means that when noise affects data, it affects a set of bits.



Redundancy

- The central concept in **detecting or correcting errors** is **redundancy**.
- To be able to **detect or correct errors**, we need to **send some extra bits with the data**. These **redundant bits are added by the sender** and **removed by the receiver**. Their presence allows the receiver to detect or correct corrupted bits.

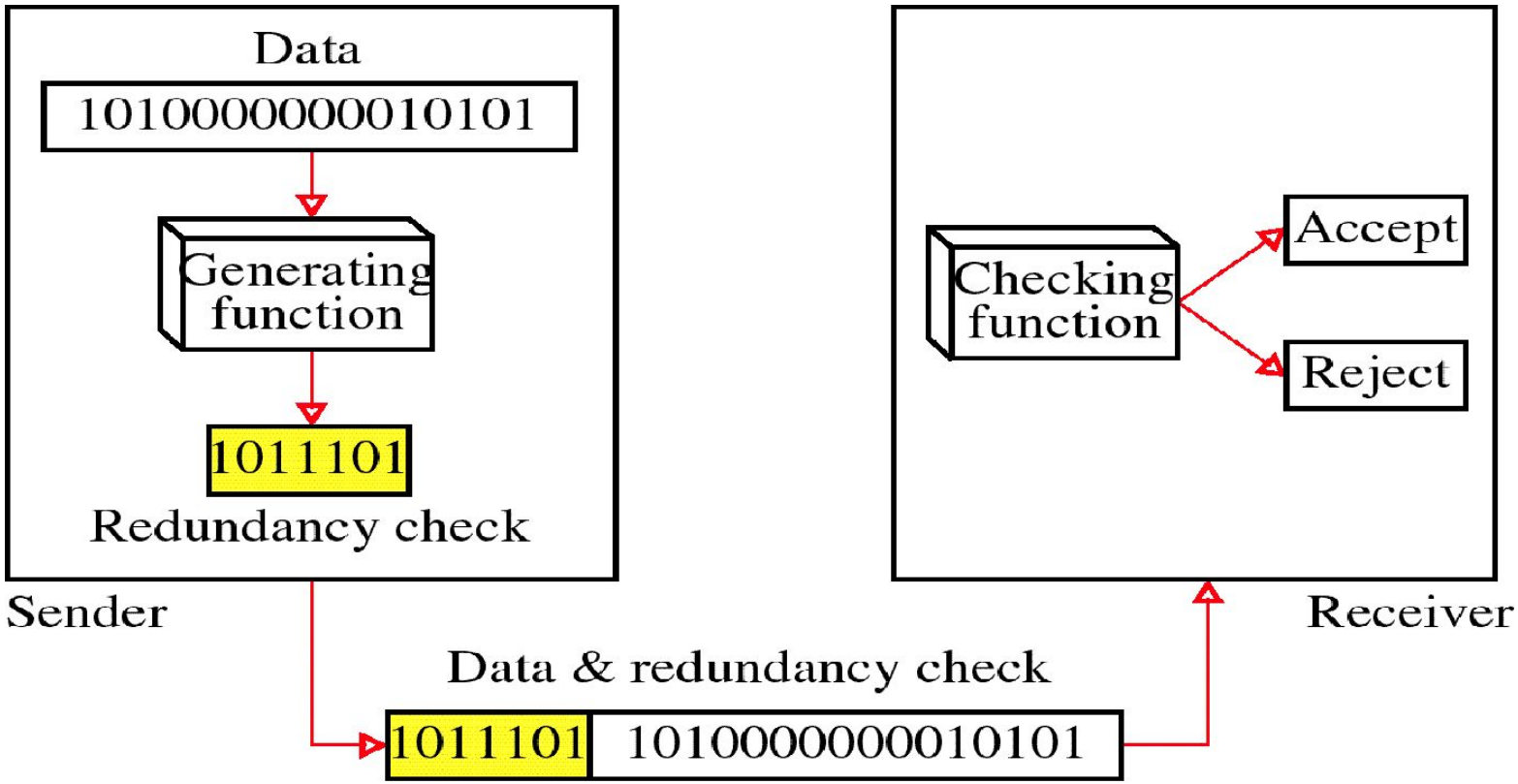
Detection Versus Correction

- The correction of errors is more difficult than the detection.
- In error detection, we are looking only to see if any error has occurred. The answer is a simple yes or no. We are not even interested in the number of errors.
- A single-bit error is the same as a burst error.
- In error correction, the exact number of bits that are corrupted and more importantly, their location in the message are needed to be known. The number of the errors and the size of the message are important factors.
- Inorder to correct one single error in an 8-bit data unit, we need to consider eight possible error locations; if we need to correct two errors in a data unit of the same size, we need to consider 28 possibilities.

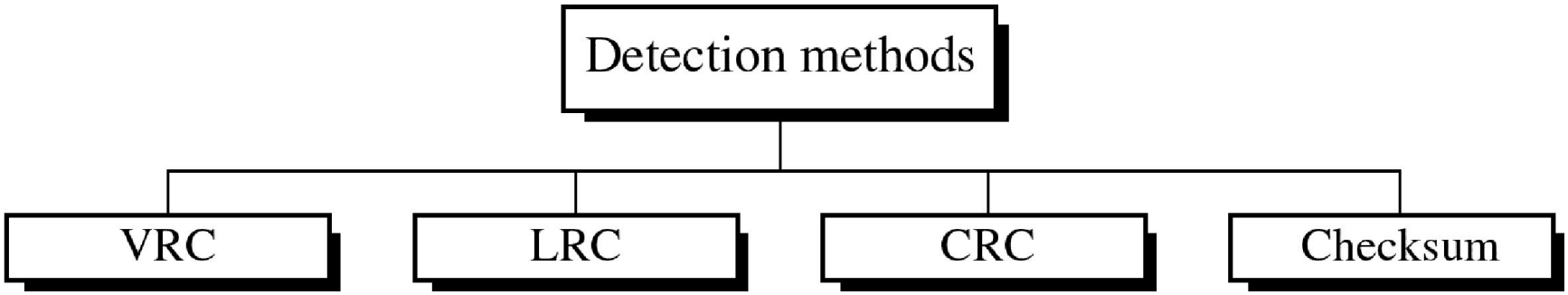
Error detection

- Error detection means to decide whether the received data is correct or not without having a copy of the original message.

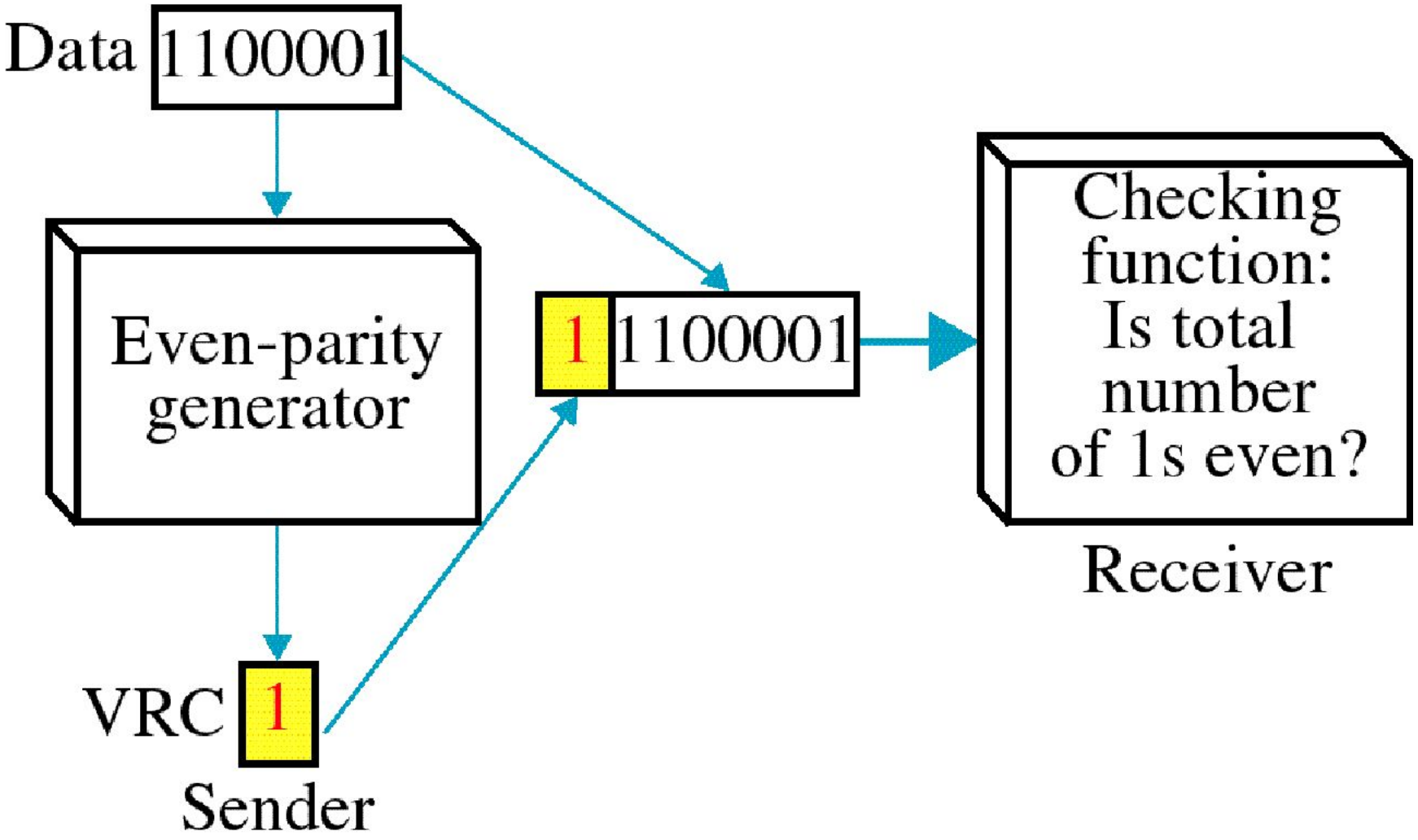
- Error detection **uses the concept of redundancy**, **which means** adding extra bits for detecting errors at the destination.



Four types of redundancy checks are used in data communications



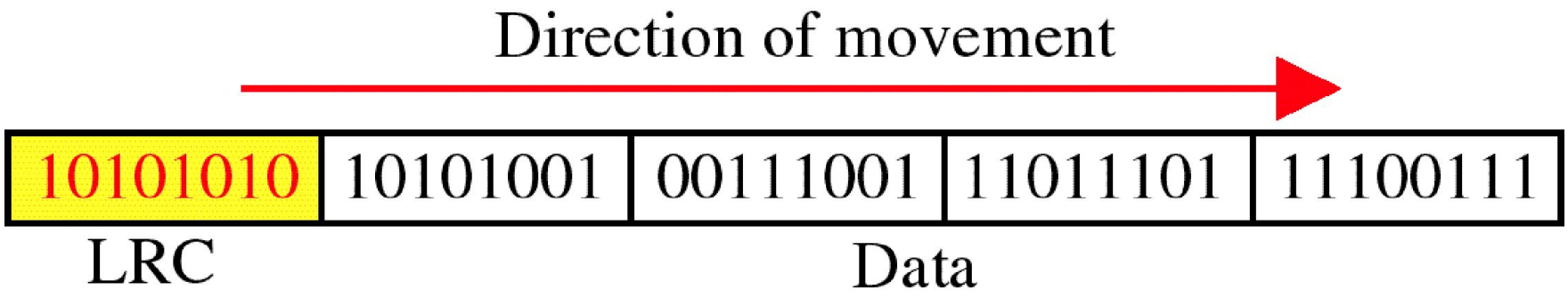
Vertical Redundancy Check
VRC (Parity Check)



Performance

- It can detect single bit error
- It can detect burst errors only if the total number of errors is odd.

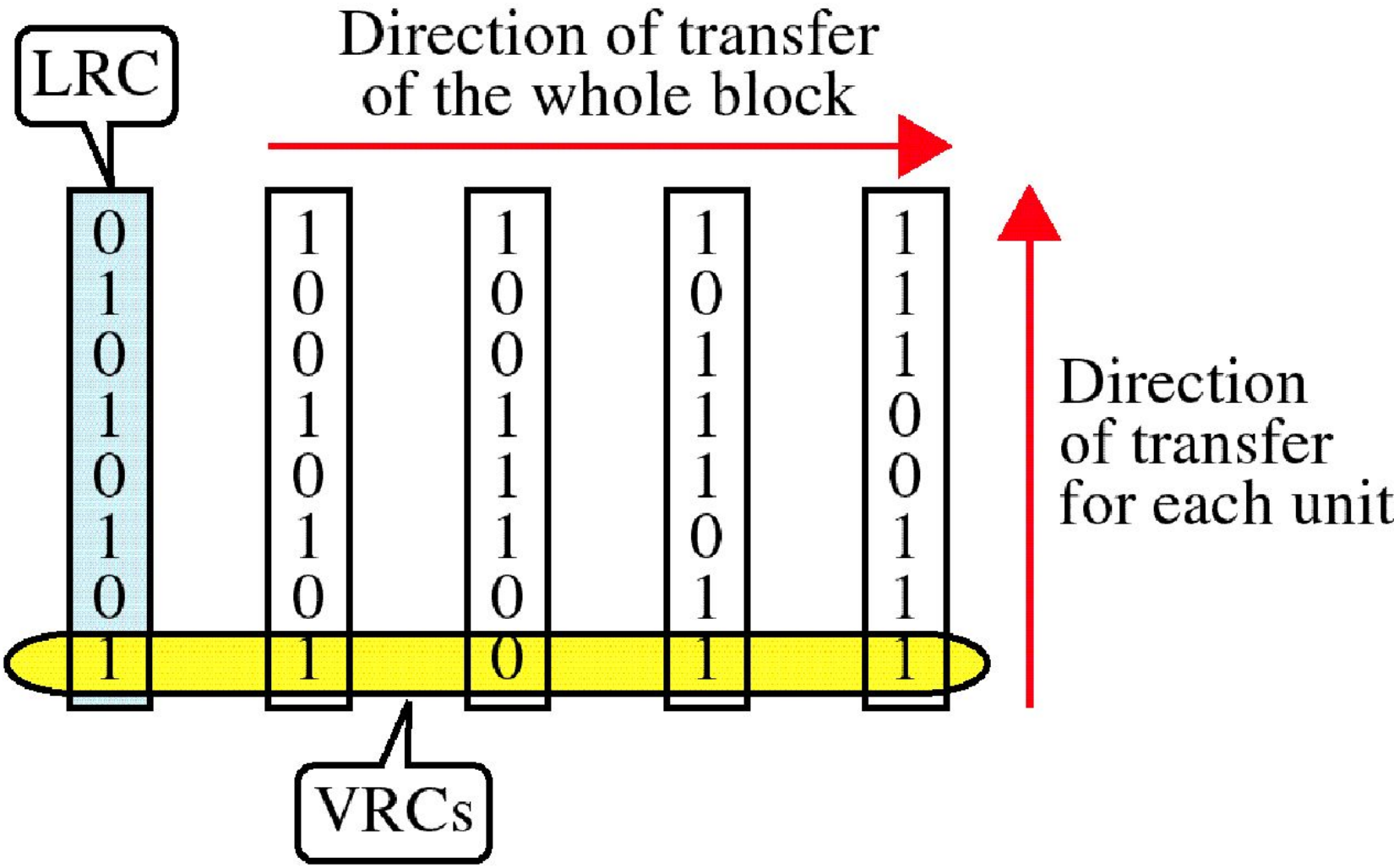
Longitudinal Redundancy Check (LRC) (2-D Parity Check)



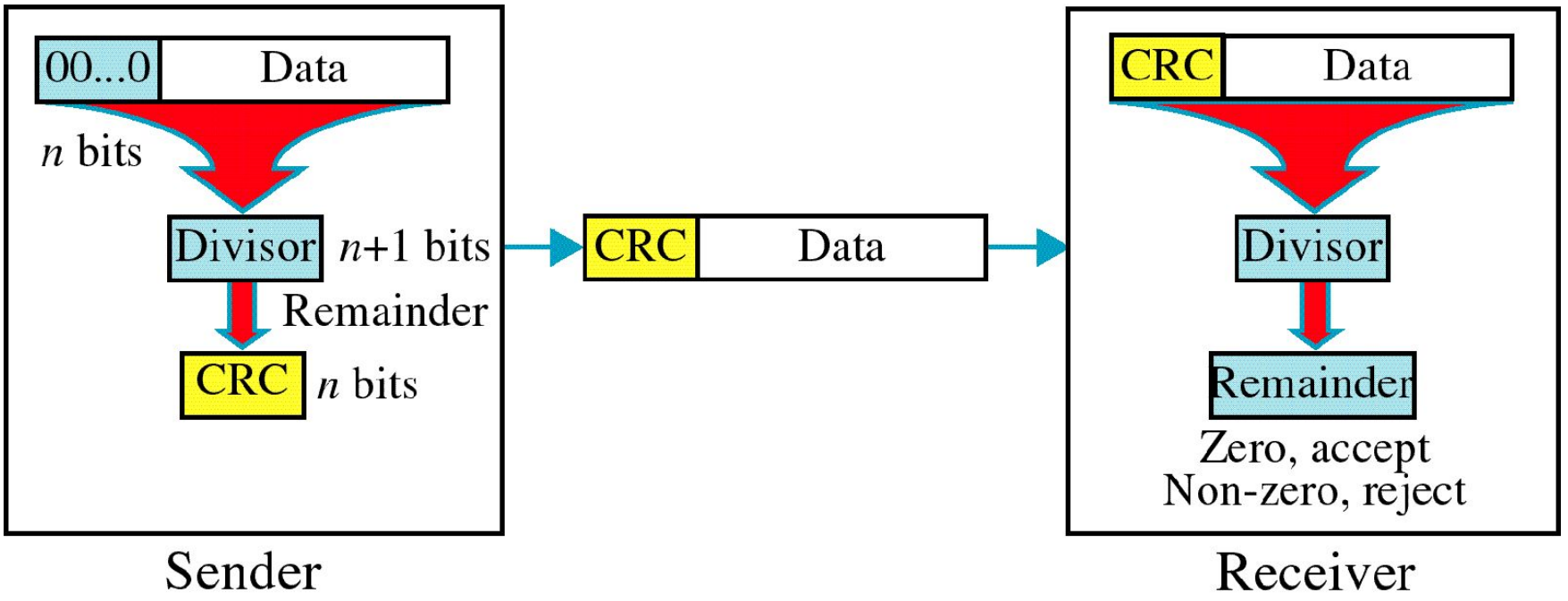
Performance

- **LCR increases the likelihood of detecting burst errors.**
- **If two bits in one data units are damaged and two bits in exactly the same positions in another data unit are also damaged, the LRC checker will not detect an error.**

VRC and LRC


LRC
Direction of transfer
of the whole block
0 1 0 1 0 1 0 1
1 0 0 1 0 1 0 1
1 0 0 1 1 1 1 0
1 0 1 1 1 0 1 1
1 1 1 0 0 1 1 1
Direction
of transfer
for each unit
VRCs

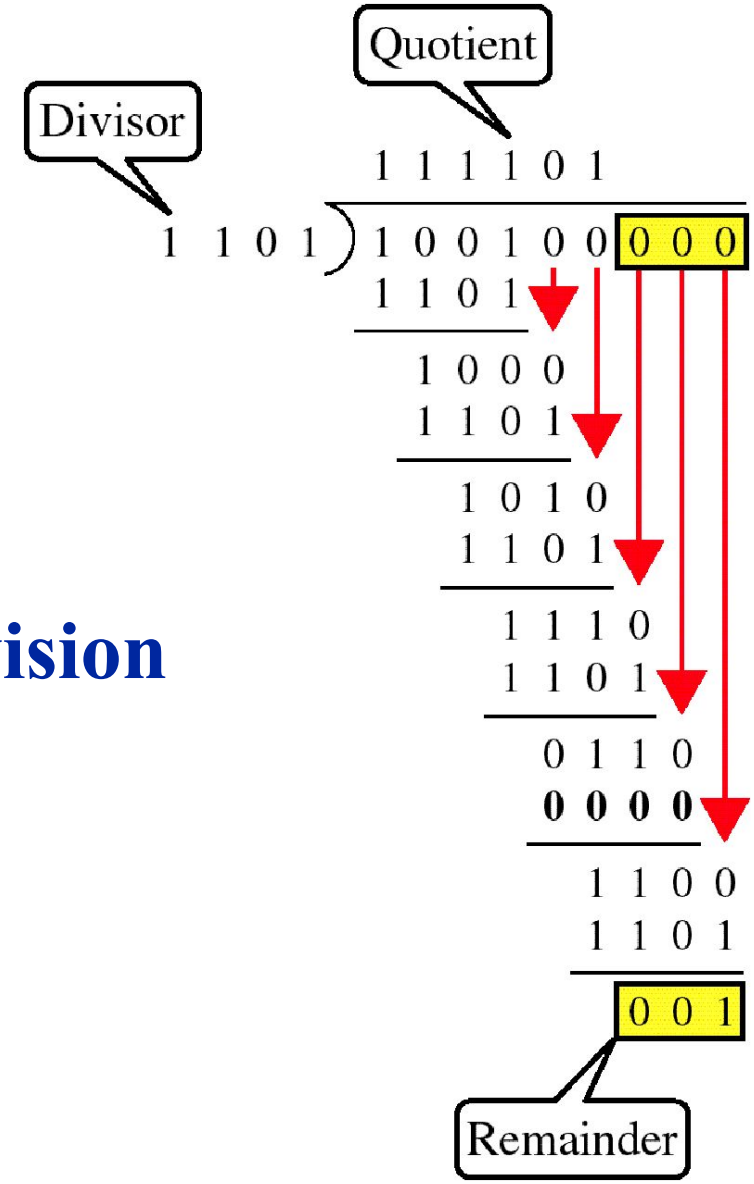
Cyclic Redundancy Check
CRC



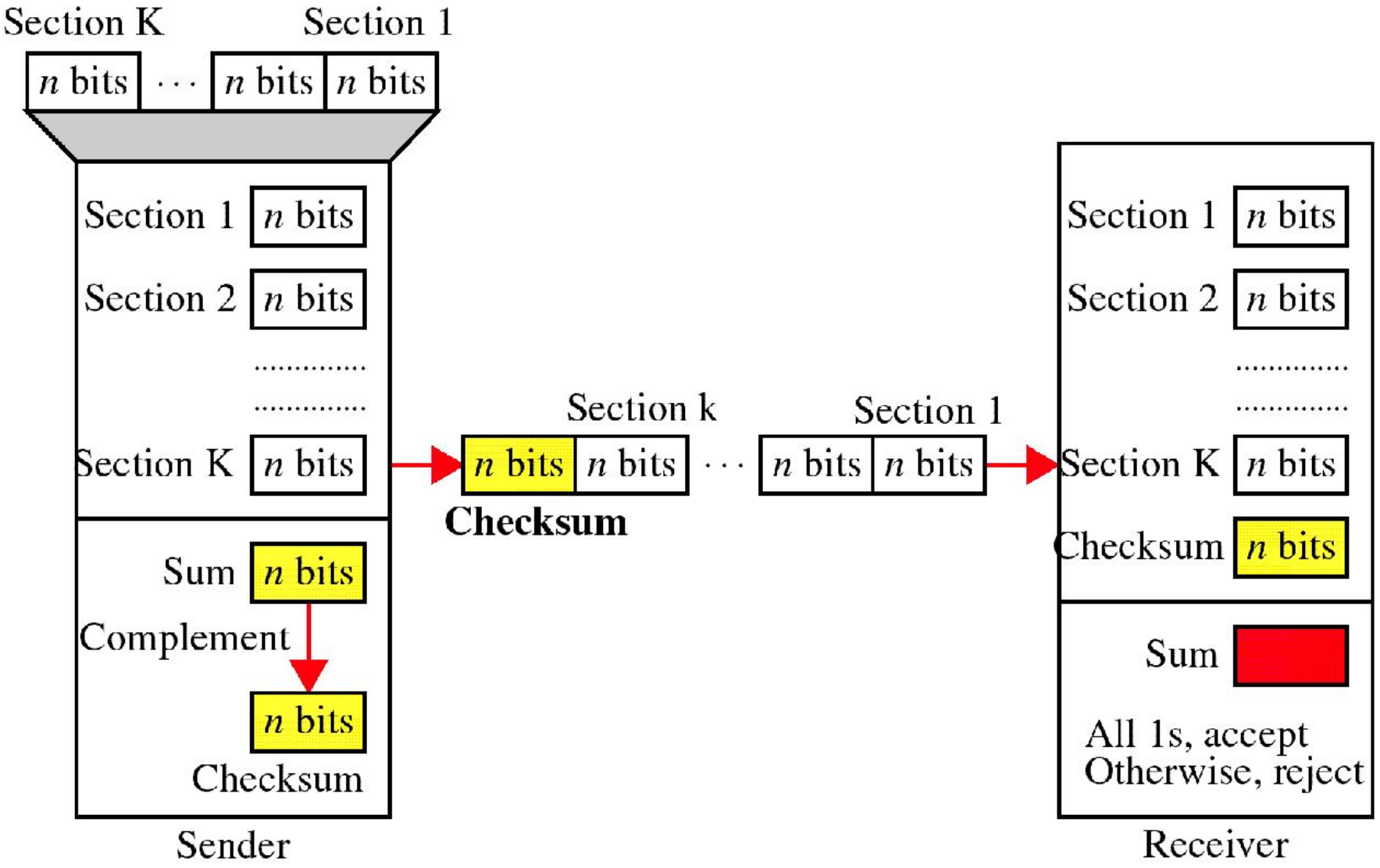
Cyclic Redundancy Check

- Given a k-bit frame or message, the transmitter generates an n-bit sequence, known as a *frame check sequence (FCS)*, so that the resulting frame, consisting of (k+n) bits, is exactly divisible by some predetermined number.
- The receiver then divides the incoming frame by the same number and, if there is no remainder, assumes that there was no error.

Binary Division



Checksum


Section K
Section 1
n bits
n bits
n bits
Section 1
n bits
Section 2
n bits
Section K
n bits
Sum
n bits
Complement
n bits
Checksum
Sender
Section k
Section 1
n bits
n bits
n bits
n bits
Checksum
Section 1
n bits
Section 2
n bits
Section K
n bits
Checksum
n bits
Sum
All 1s, accept
Otherwise, reject
Receiver

At the sender

- The unit is divided into *k* sections, each of *n* bits.
- All sections are added together using one's complement to get the sum.
- The sum is complemented and becomes the checksum.
- The checksum is sent with the data

At the receiver

- The unit is divided into *k* sections, each of *n* bits.
- All sections are added together using one's complement to get the sum.
- The sum is complemented.
- If the result is zero, the data are accepted: otherwise, they are rejected.

Performance

- The checksum detects all errors involving an odd number of bits.
- It detects most errors involving an even number of bits.
- If one or more bits of a segment are damaged and the corresponding bit or bits of opposite value in a second segment are also damaged, the sums of those columns will not change and the receiver will not detect a problem.

Methods of error correction

- There are two main methods of error correction.
- **Forward error correction** is the process in which the receiver tries to guess the message by using redundant bits. This is possible, if the number of errors is small.
- **Correction by retransmission** is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message.
  - Resending is repeated until a message arrives that the receiver believes is error-free (usually, not all errors can be detected).

Coding

- Redundancy is achieved through various coding schemes. The sender adds redundant bits through a process that creates a relationship between the redundant bits and the actual data bits.
- The receiver checks the relationships between the two sets of bits to detect or correct the errors. The ratio of redundant bits to the data bits and the robustness of the process are important factors in any coding scheme.

Hamming Code

- Hamming code is a set of error-correction codes that can be used to **detect and correct the errors** that can occur when the data is moved or stored from the sender to the receiver.
- It is **technique developed by R.W. Hamming for error correction**.

Redundant bits:

Redundant bits are extra binary bits that are generated and added to the information carrying bits of data transfer to ensure that no bits were lost during the data transfer. The number of redundant bits can be calculated using the following formula:

$$\mathbf{2^r \geq m + r + 1}$$ where, r = redundant bit, m = data bit

Suppose the number of data bits is 7, then the number of redundant bits can be calculated using:

$= 2^4 \geq 7 + 4 + 1$

Thus, the number of redundant bits= 4

- Relationship between data and redundancy bits:

No. of data bits m	No. of redundancy bits r	Total bits m+r
1	2	3
2	3	5
3	3	6
4	3	7
5	4	9
6	4	10
7	4	11

- The hamming code can be applied to data units of any length and uses the relationship between data and redundancy bits.
- A 7-bit ASCII code requires 4 redundancy bits that can be added to the end of the data or interspersed with the original data bits.

11	10	9	8	7	6	5	4	3	2	1
d	d	d	r_8	d	d	d	r_4	d	r_2	r_1

- In the hamming code, each r bit is the parity bit for one combination of data bits.
- r1 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the least significant position.

  r1: bits 1, 3, 5, 7, 9, 11
- r2 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the second position from the least significant bit.

  r2: bits 2,3,6,7,10,11
- r4 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the third position from the least significant bit.

  r4: bits 4, 5, 6, 7
- r8 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the fourth position from the least significant bit.

  r8: bit 8,9,10,11

Calculating the r values

- Place each bit of the original character in its appropriate position in the 11-bit unit.
- Calculate the even parities for the various bit combinations.
- The parity value for each combination is the value of the corresponding **r** bit.
- **Data : 1001101**

- **Adding r_1: bits 1, 3, 5, 7, 9, 11**

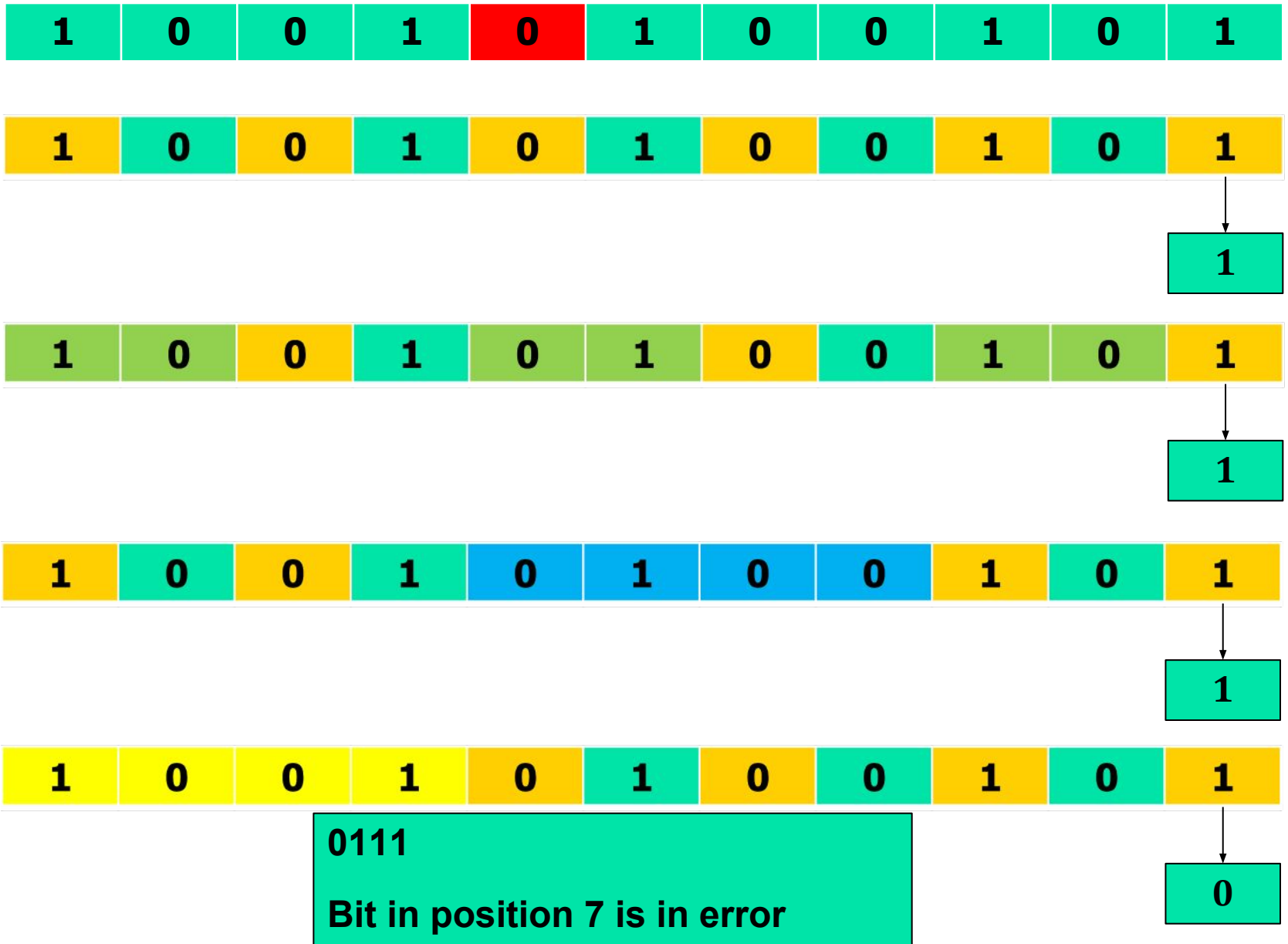
- **Adding r_2: bits 2, 3, 6, 7, 10, 11**

- **Adding r_4: bits 4, 5, 6, 7**

- **Adding r_8: bits 8, 9, 10, 11**

- **Code: 10100111001**

Error detection and correction



Data link control - Framing

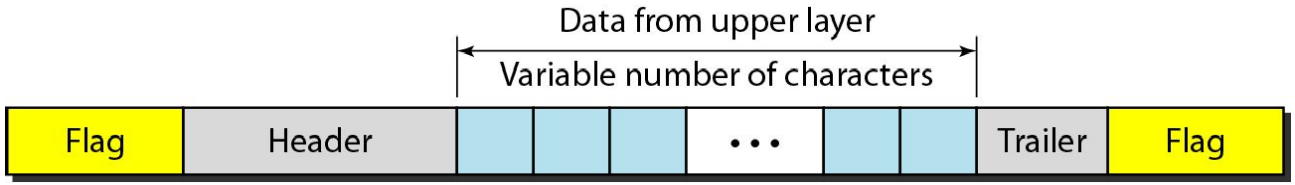
- The data link layer needs to pack bits into frames, so that each frame is distinguishable from another.
- Our postal system practices a type of framing. The simple act of inserting a letter into an envelope separates one piece of information from another; the envelope serves as the delimiter.
- **Framing** in the data link layer separates a message from one source to a destination, or from other messages to other destinations, by adding a sender address and a destination address.
- The **destination address** defines where the packet is to go; the **sender address** helps the recipient acknowledge the receipt.

Fixed-size & Variable-size framing

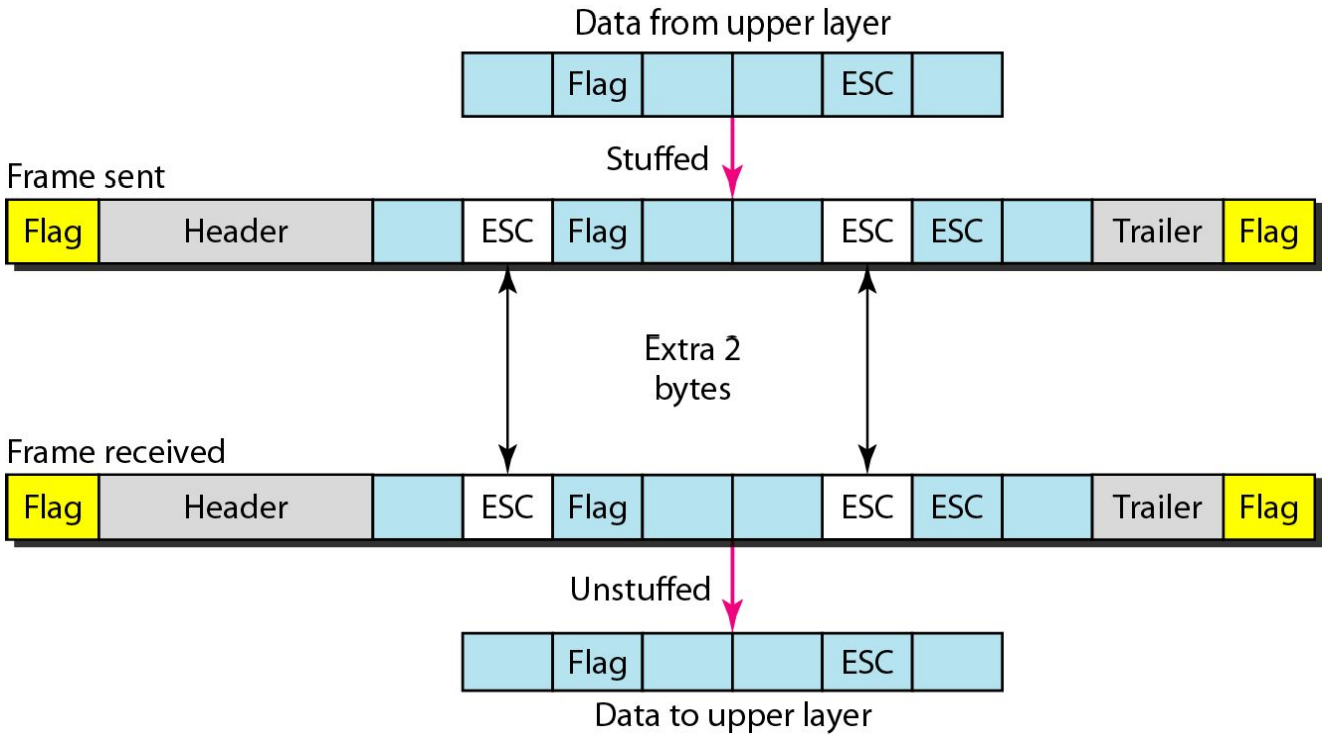
- Frames can be of fixed or variable size.
- In **fixed-size framing**, there is **no need for defining the boundaries of the frames**; the size itself can be used as a delimiter.
- An example of this type of framing is the **ATM wide-area network**, which uses **frames of fixed size** called **cells**.
- In **variable-size framing**, a way is needed to define the end of the frame and the beginning of the next frame.
- **Two approaches**: character-oriented approach and a bit-oriented approach.

Character-Oriented Protocols

- In a **character-oriented protocol**, data to be carried are 8-bit characters from a coding system such as ASCII. The **header,** which normally carries the **source and destination addresses and other control information**, and the **trailer**, which carries **error detection or error correction redundant bits**, are also multiples of 8 bits.
- To **separate one frame from the next, an 8-bit (1-byte) flag is added at the beginning and the end of a frame**. The flag, composed of protocol-dependent special characters, signals the start or end of a frame.
- A **byte-stuffing strategy** is added to character-oriented framing.
- In **byte stuffing** (or character stuffing), a **special byte** is added to the **data section** of the frame when there is **a character with the same pattern as the flag**. The data section is stuffed with an extra byte. This byte is usually called the **escape character (ESC),** which has a predefined bit pattern.
- Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag.

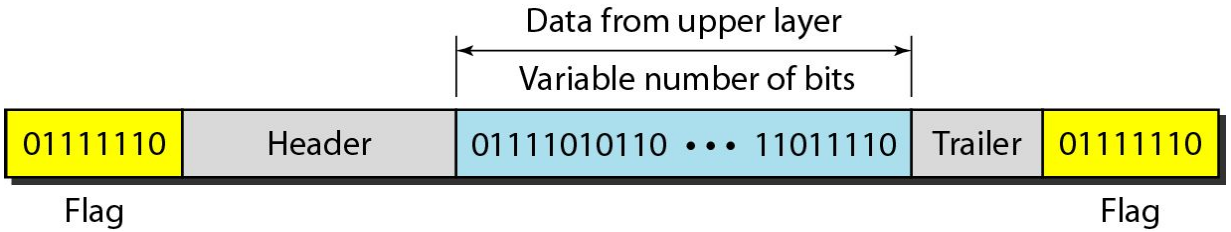


Byte stuffing and unstuffing

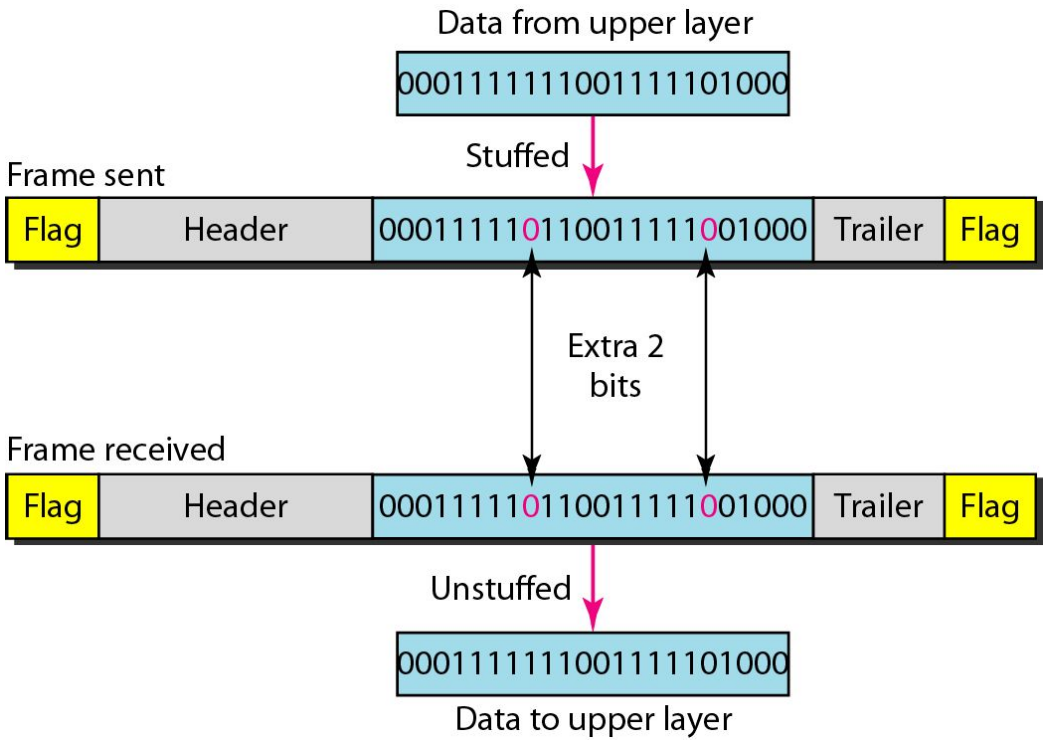


Bit-Oriented Protocols

- In a bit-oriented protocol, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, and so on.
- However, in addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other.
- Most protocols use a special 8-bit pattern **flag 01111110** as the **delimiter** to define the beginning and the end of the frame.
- By stuffing 1 single bit (instead of I byte) to prevent the pattern from looking like a flag. The strategy is called bit stuffing.
- **In bit stuffing, if a 0 and five consecutive I bits are encountered, an extra 0 is added.** This extra stuffed bit is eventually removed from the data by the receiver. Note that the extra bit is added after one 0 followed by five 1s regardless of the value of the next bit.



uffing and unstuffing



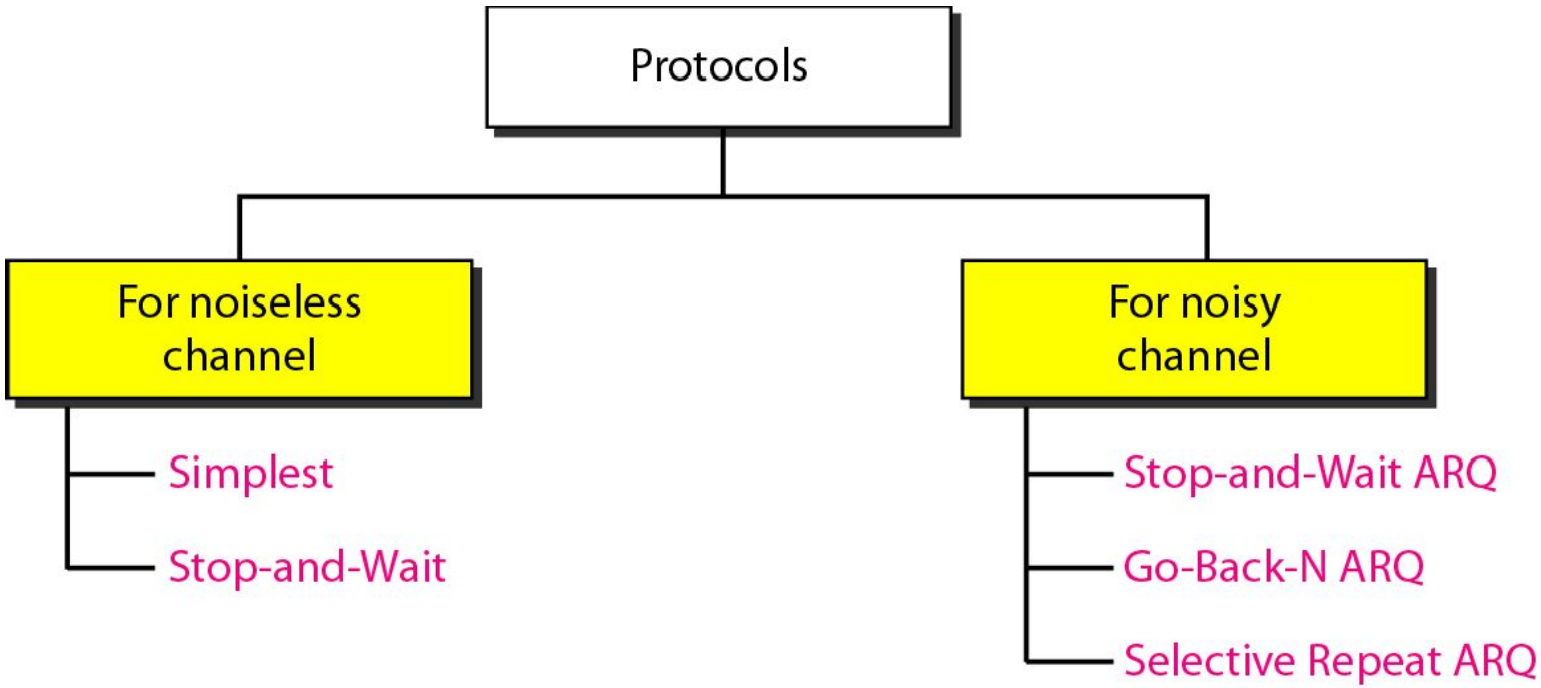
FLOW AND ERROR CONTROL

- The most important responsibilities of the data link layer are flow control and error control. Collectively, these functions are known as data link control.
- **Flow control** is a set of procedures that tells the sender

  (i) how much data it can transmit before it must wait for an acknowledgment from the receiver. The flow of data must not be allowed to overwhelm the receiver.
- Each receiving device has a block of memory, called a ***buffer***, reserved for storing incoming data until they are processed. If the buffer begins to fill up, the receiver must be able to tell the sender to halt transmission until it is once again able to receive.
- In the data link layer, the term **error control** refers primarily to methods of error detection and retransmission. Error control in the data link layer is often implemented simply: Any time an error is detected in an exchange, specified frames are retransmitted. This process is called **automatic repeat request (ARQ)**.

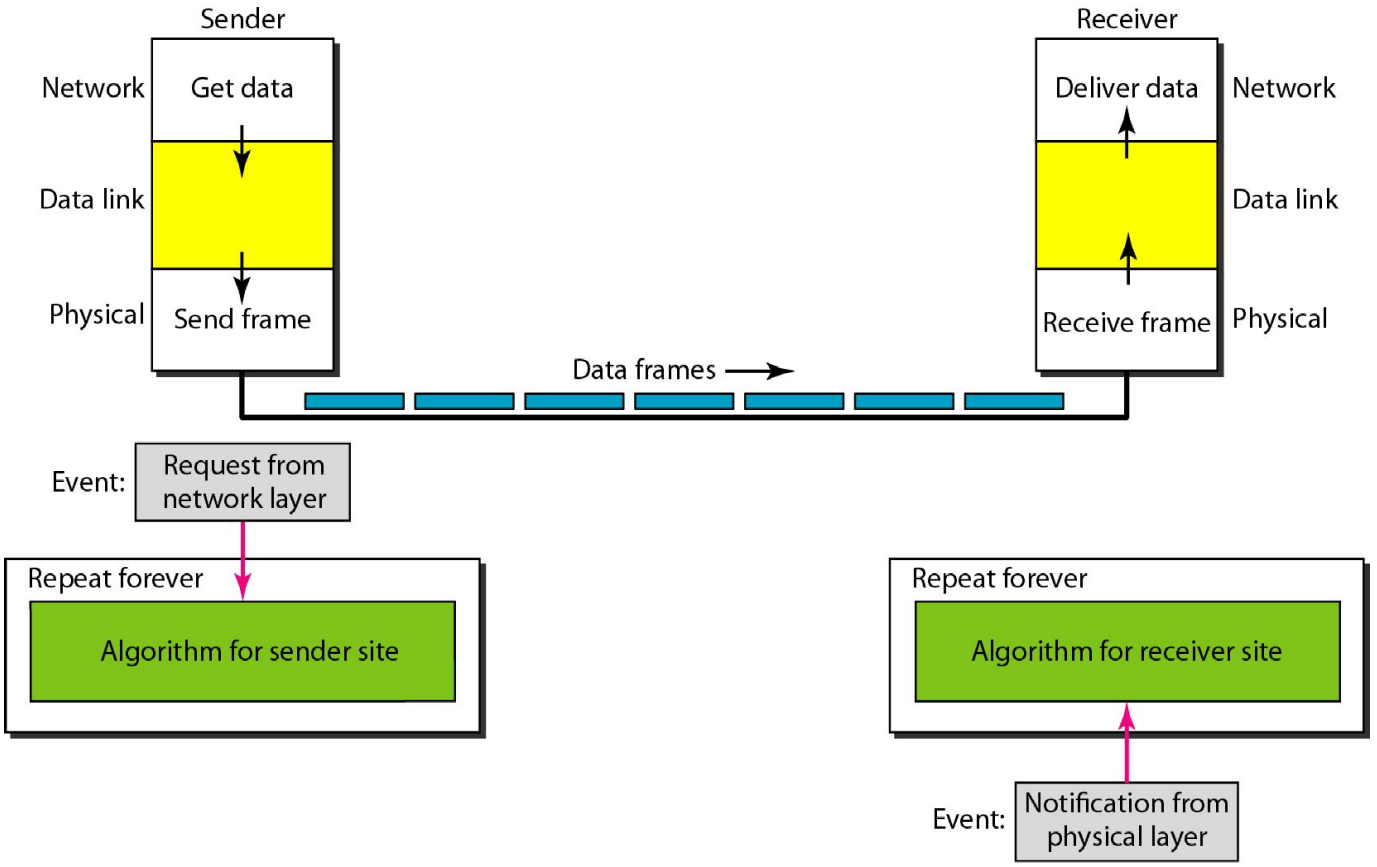
PROTOCOLS

Protocols into those that can be used for noiseless (error-free) channels and those that can be used for noisy (error-creating) channels.



Simplest Protocol

- **Simplest Protocol** is one that has no flow or error control. It is a unidirectional protocol in which data frames are traveling in only one direction-from the sender to receiver.



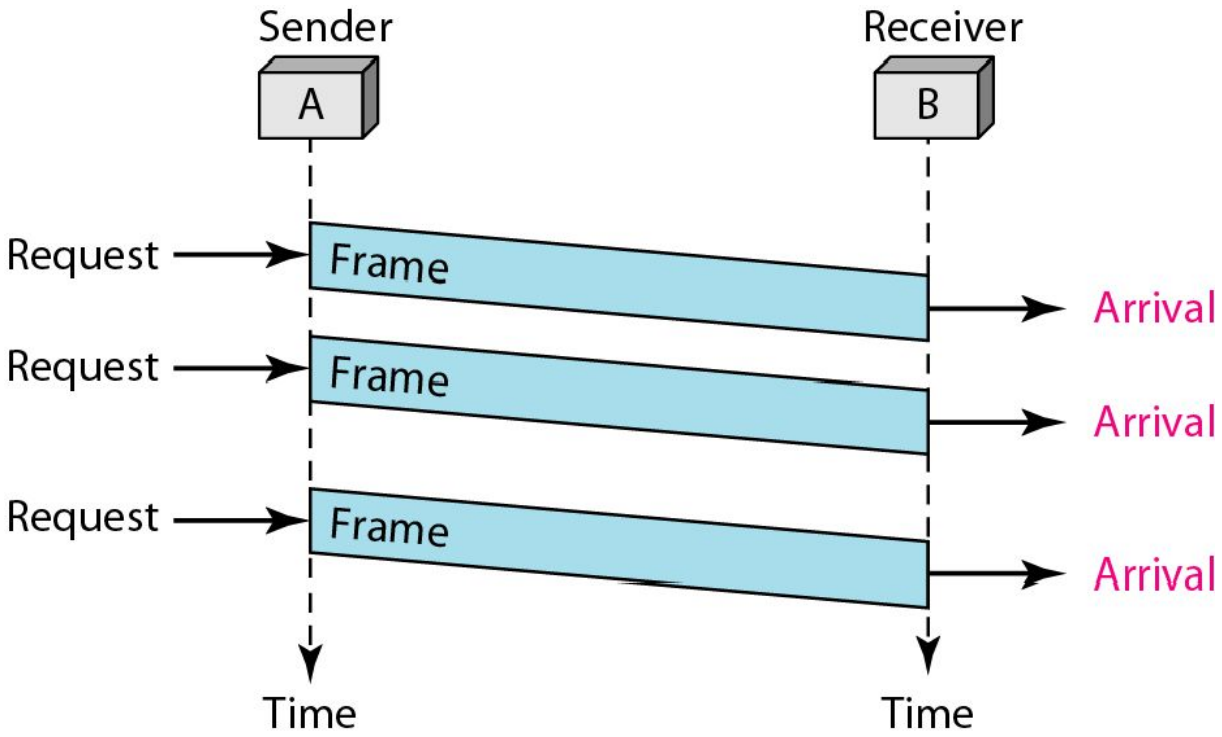
Algorithm 11.1 *Sender-site algorithm for the simplest protocol*

```
1  while(true)                          // Repeat forever
2  {
3    WaitForEvent();                    // Sleep until an event occurs
4    if(Event(RequestToSend))           //There is a packet to send
5    {
6       GetData();
7       MakeFrame();
8       SendFrame();                    //Send the frame
9    }
10 }
```

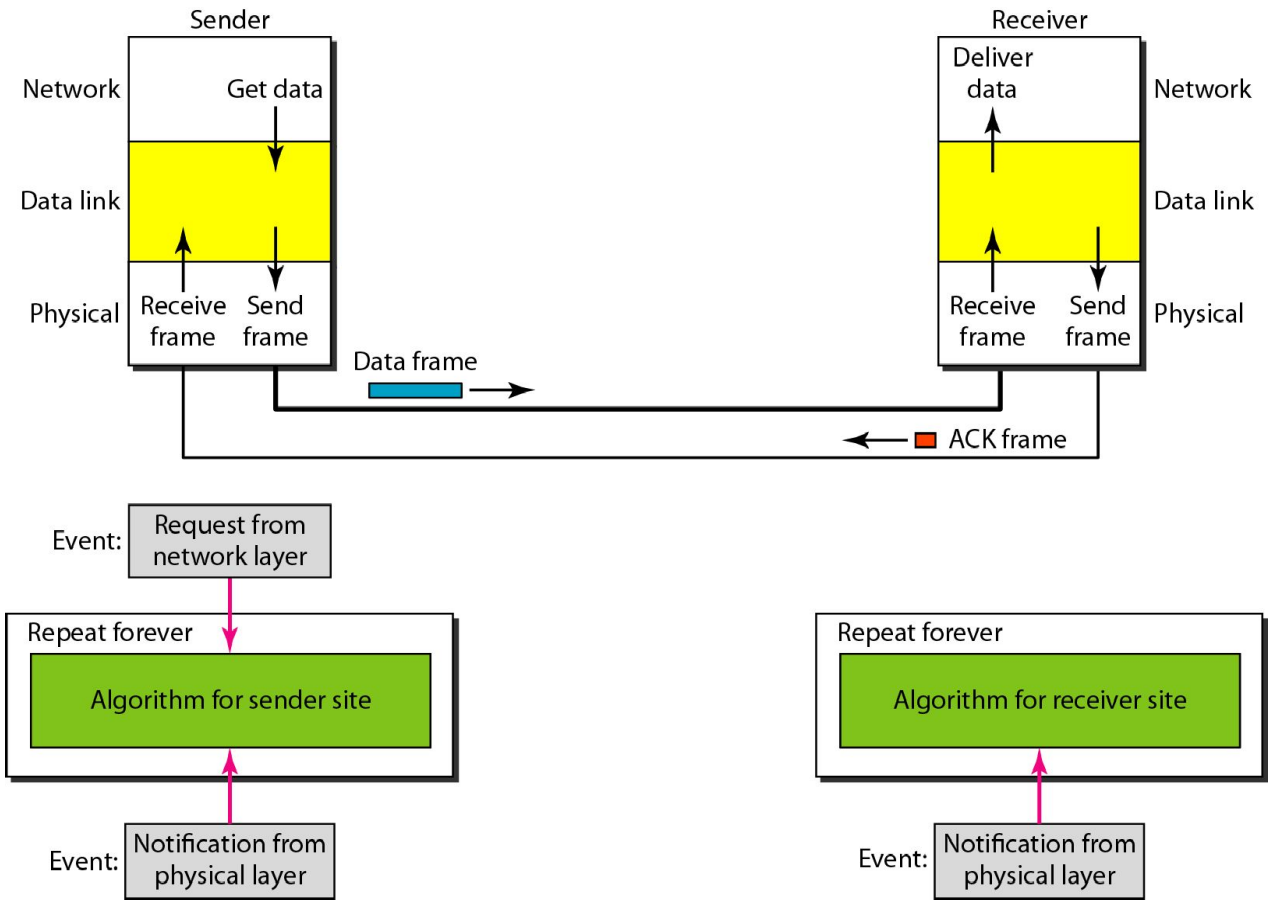
Algorithm 11.2 *Receiver-site algorithm for the simplest protocol*

```
1  while(true)                          // Repeat forever
2  {
3    WaitForEvent();                    // Sleep until an event occurs
4    if(Event(ArrivalNotification))     //Data frame arrived
5    {
6       ReceiveFrame();
7       ExtractData();
8       DeliverData();                  //Deliver data to network layer
9    }
10 }
```

Simplest Protocol



Stop-and-Wait Protocol



Algorithm 11.3 *Sender-site algorithm for Stop-and-Wait Protocol*

```
1  while(true)                          //Repeat forever
2  canSend = true                       //Allow the first frame to go
3  {
4    WaitForEvent();                    // Sleep until an event occurs
5    if(Event(RequestToSend) AND canSend)
6    {
7       GetData();
8       MakeFrame();
9       SendFrame();                    //Send the data frame
10      canSend = false;                //Cannot send until ACK arrives
11   }
12   WaitForEvent();                    // Sleep until an event occurs
13   if(Event(ArrivalNotification) // An ACK has arrived
14    {
15      ReceiveFrame();                 //Receive the ACK frame
16      canSend = true;
17    }
18 }
```

Algorithm 11.4 *Receiver-site algorithm for Stop-and-Wait Protocol*

```
1  while(true)                          //Repeat forever
2  {
3    WaitForEvent();                    // Sleep until an event occurs
4    if(Event(ArrivalNotification))     //Data frame arrives
5    {
6       ReceiveFrame();
7       ExtractData();
8       Deliver(data);                  //Deliver data to network layer
9       SendFrame();                    //Send an ACK frame
10   }
11 }
```

Note

Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting of the frame when the timer expires.

Note

In Stop-and-Wait ARQ, we use sequence numbers to number the frames. The sequence numbers are based on modulo-2 arithmetic.

Note

In Stop-and-Wait ARQ, the acknowledgment number always announces in modulo-2 arithmetic the sequence number of the next frame expected.

Stop-and-wait ARQ

11.2 STOP-AND-WAIT ARQ

Stop-and-Wait ARQ is the simplest flow and error control mechanism. It has the following features:

- The sending device keeps a copy of the last frame transmitted until it receives an acknowledgment for that frame. Keeping a copy allows the sender to retransmit lost or damaged frames until they are received correctly.
- For identification purposes, both data frames and **acknowledgment (ACK)** frames are numbered alternately 0 and 1. A data 0 frame is acknowledged by an ACK 1 frame, indicating that the receiver has received data frame 0 and is now expecting data frame 1. This numbering allows for identification of data frames in case of duplicate transmission (important in the case of lost acknowledgment or delayed acknowledgment, as we will see shortly).
- A damaged or lost frame is treated in the same manner by the receiver. If the receiver detects an error in the received frame, it simply discards the frame and sends no acknowledgment. If the receiver receives a frame that is out of order (0 instead of 1 or 1 instead of 0), it knows that a frame is lost. It discards the out-of-order received frame.
- The sender has a control variable, which we call *S*, that holds the number of the recently sent frame (0 or 1). The receiver has a control variable, which we call *R*, that holds the number of the next frame expected (0 or 1).

- The sender starts a timer when it sends a frame. If an acknowledgment is not received within an allotted time period, the sender assumes that the frame was lost or damaged and resends it.
- The receiver sends only positive acknowledgment for frames received safe and sound; it is silent about the frames damaged or lost. The acknowledgment number always defines the number of the next expected frame. If frame 0 is received, ACK 1 is sent; if frame 1 is received, ACK 0 is sent.

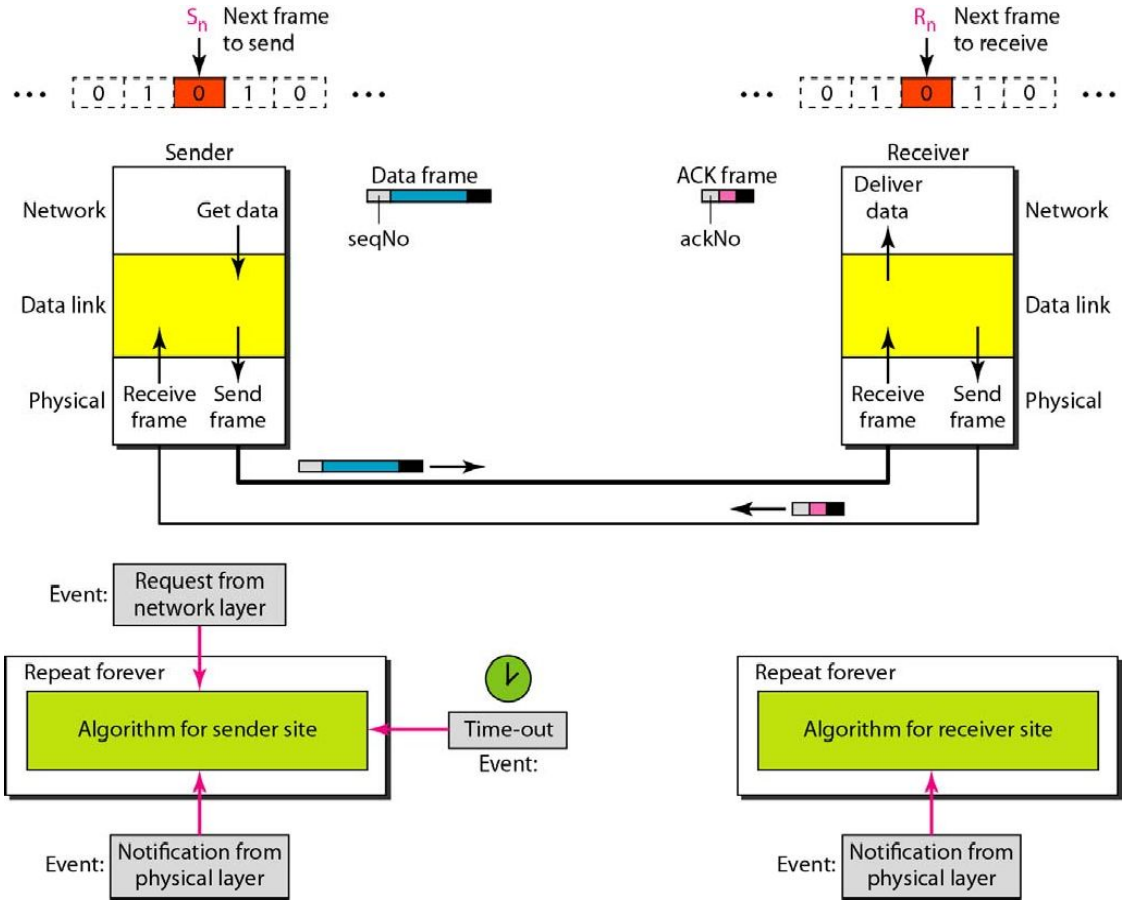
Operation

In the transmission of a frame, we can have four situations: normal operation, the frame is lost, the acknowledgment is lost, or the acknowledgment is delayed.

Normal Operation

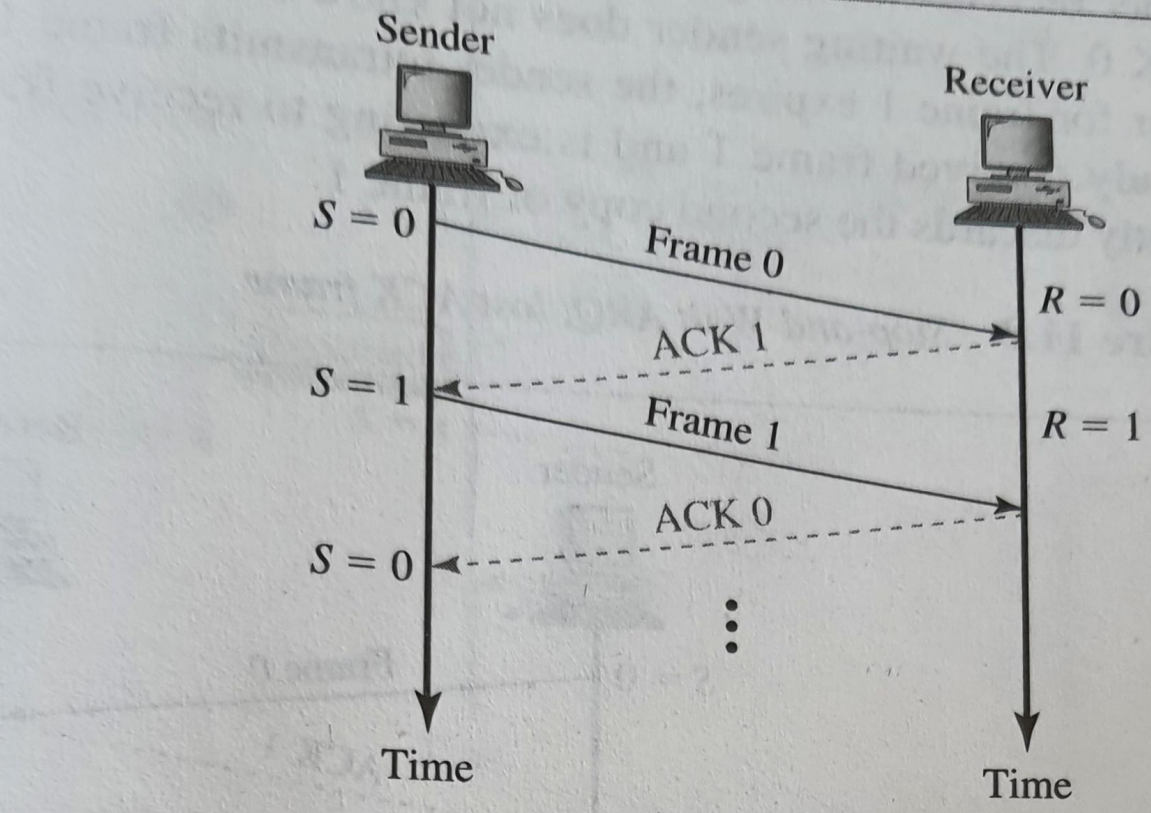
In a normal transmission, the sender sends frame 0 and waits to receive ACK 1. When ACK 1 is received, it sends frame 1 and then waits to receive ACK 0, and so on. The ACK must be received before the timer set for each frame expires. Figure 11.1 shows successful frame transmissions.

Figure 11.10 *Design of the Stop-and-Wait ARQ Protocol*

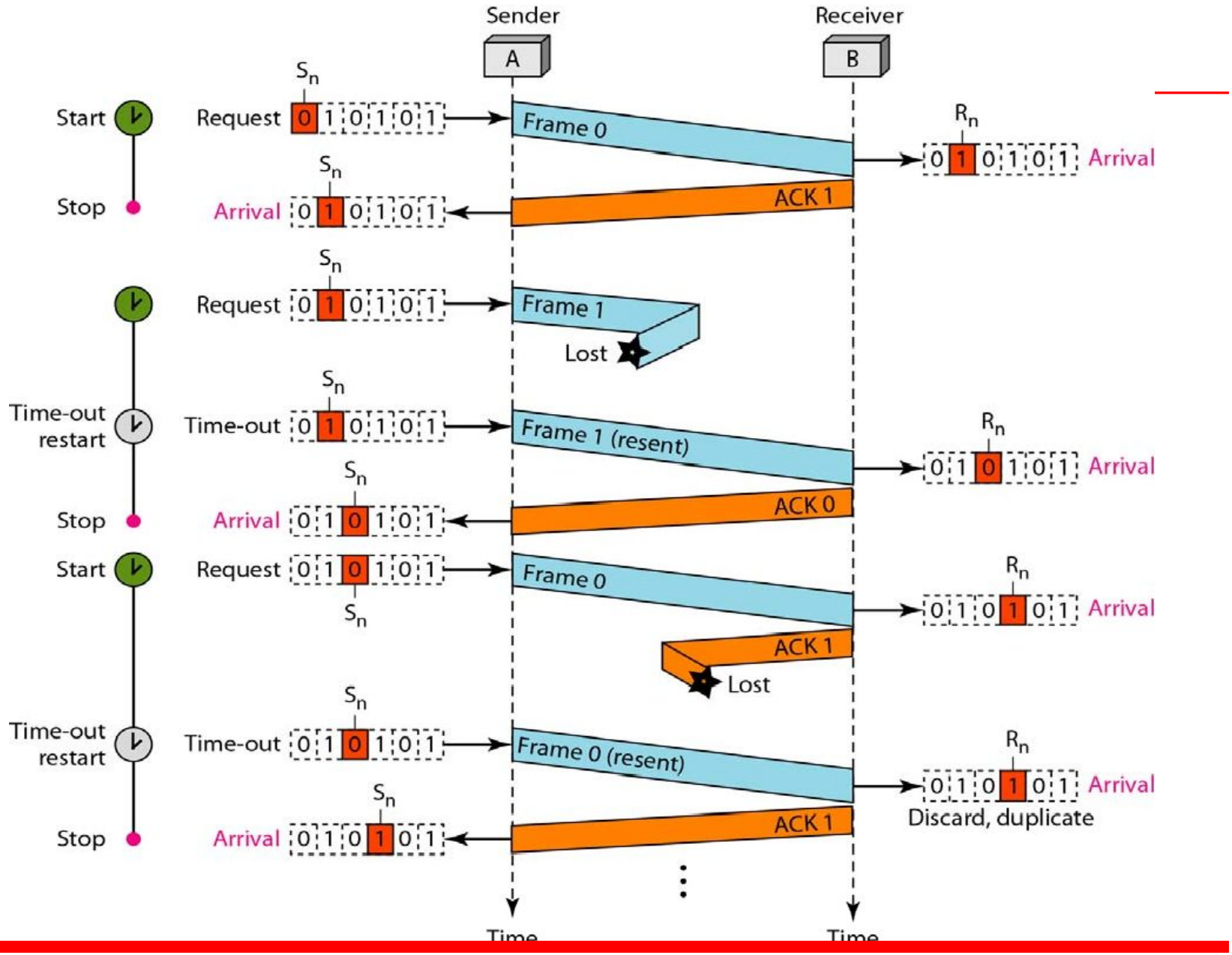


Normal operation

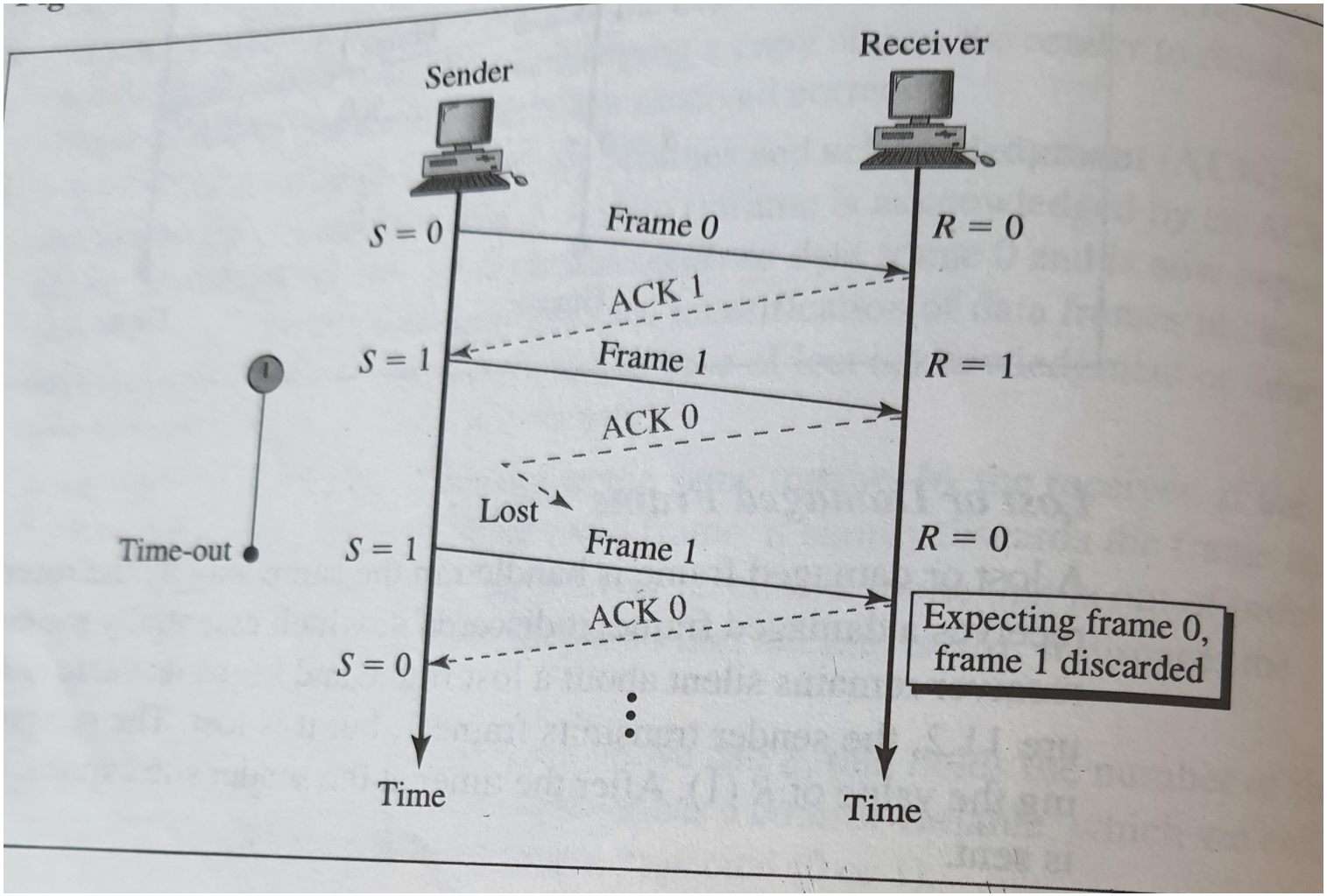
Figure 11.1 *Normal operation*



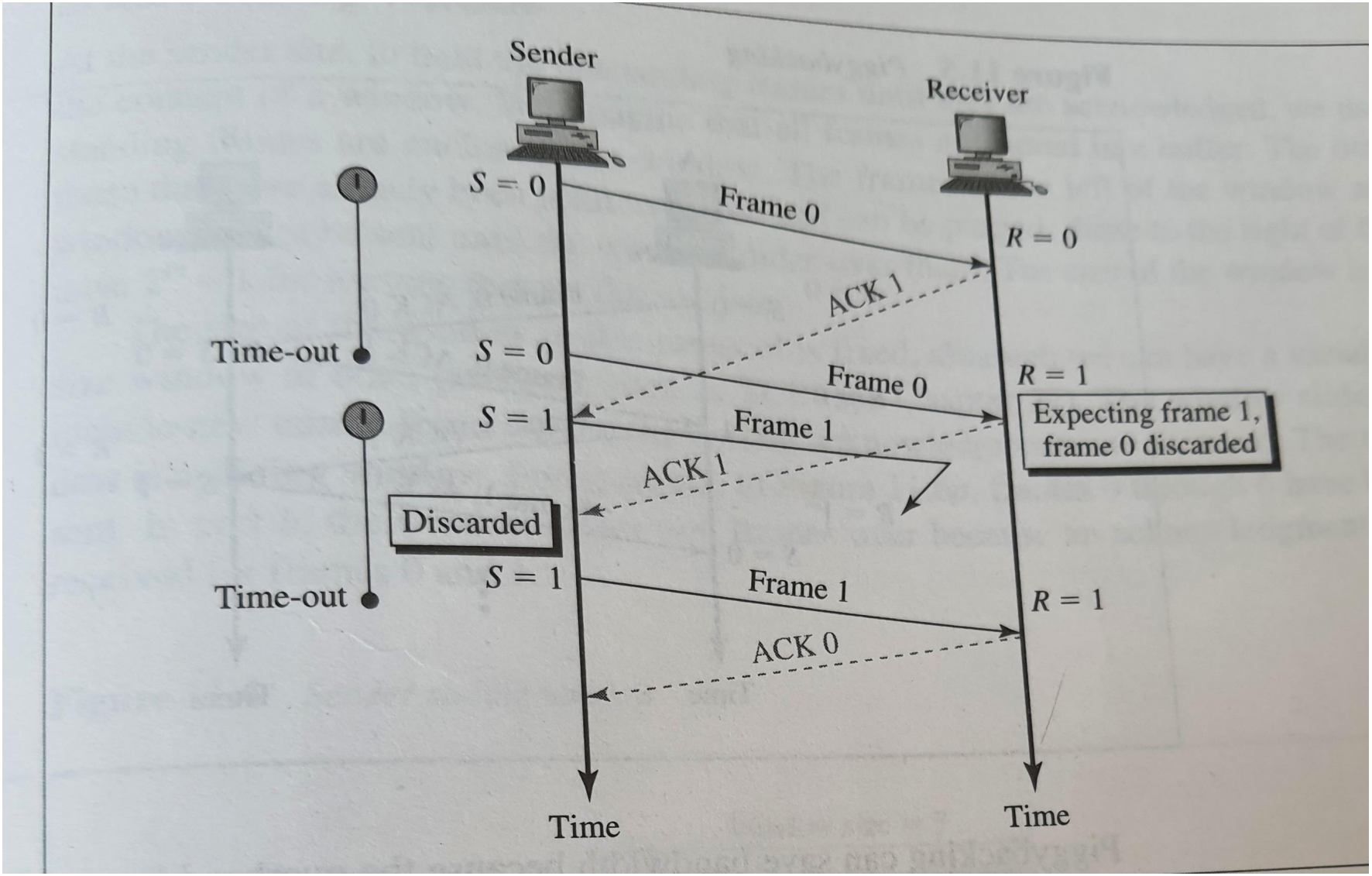
Stop-and-wait ARQ, lost frame



Stop-and-wait ARQ, lost ACK frame



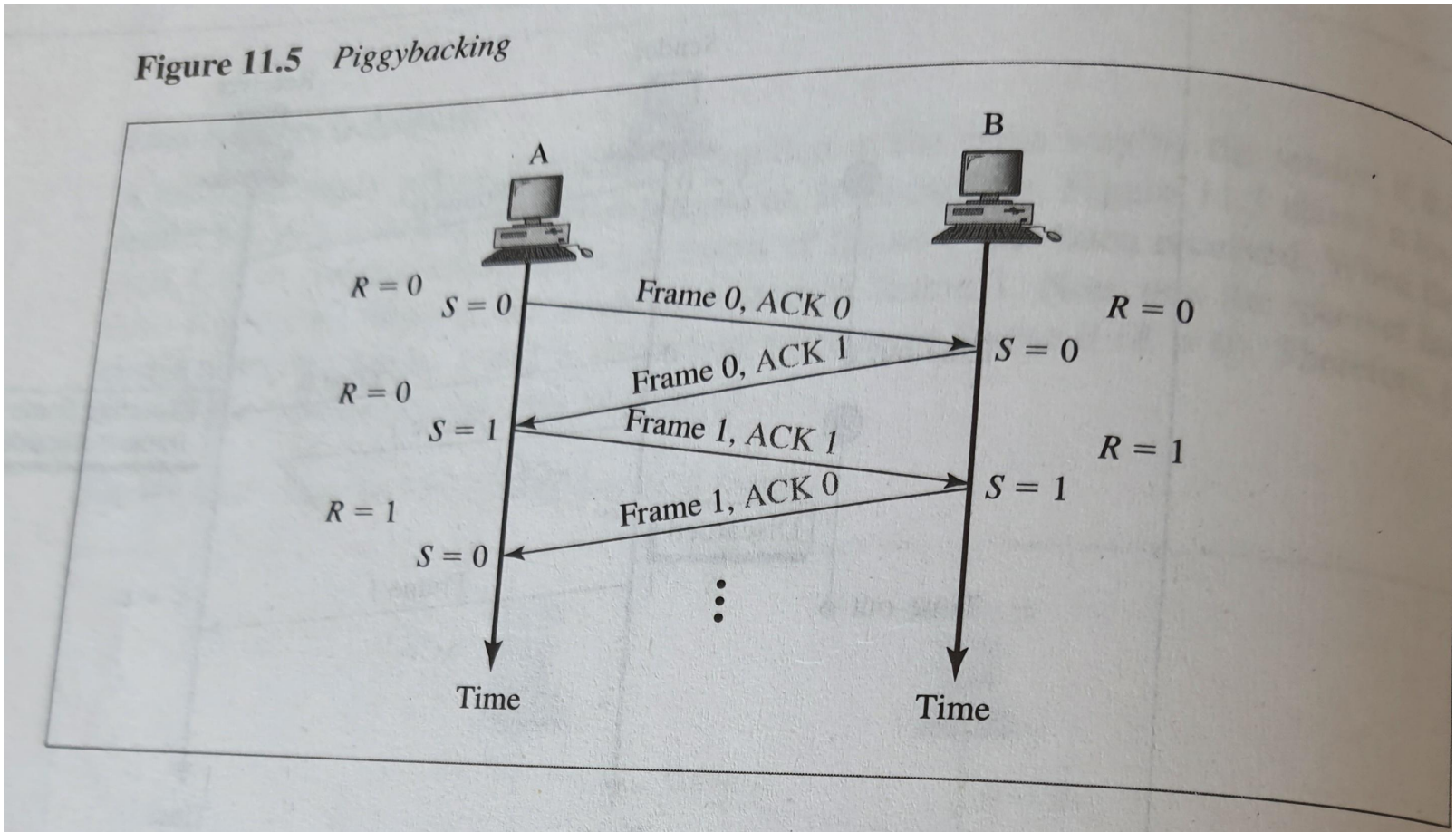
Stop-and-Wait ARQ, delayed ACK



Piggybacking

Piggybacking is a method to combine a data frame with an acknowledgement.

Piggybacking can save bandwidth .



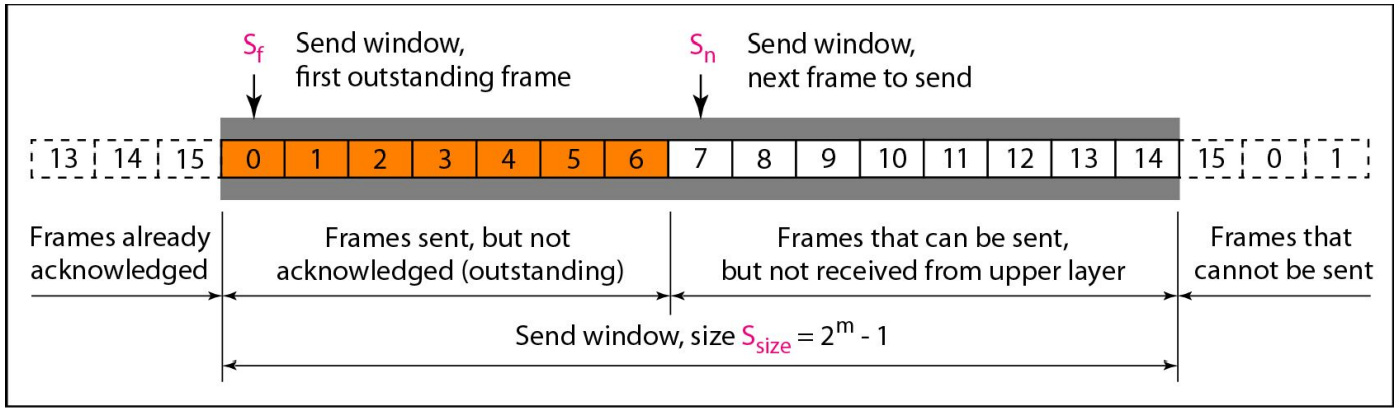
Go-Back-N ARQ

- In Go-Back-N Automatic Repeat Request, can send several frames before receiving acknowledgments; a copy of the frames are kept until the acknowledgments arrive.
- Frames from a sending station are numbered sequentially.
- If the header of the frame allows *m* bits for the sequence number, the sequence numbers range from 0 to 2^m - 1. For example, if *m* is 4, the only sequence numbers are 0 through 15 inclusive. However, we can repeat the sequence. So the sequence numbers are

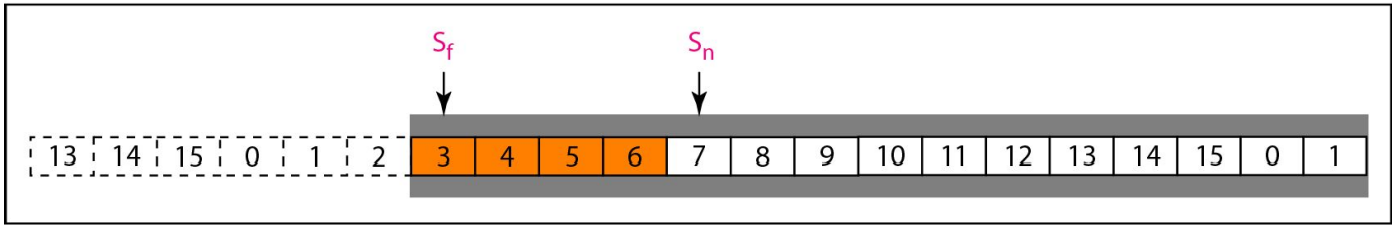
  0, 1,2,3,4,5,6, 7,8,9, 10, 11, 12, 13, 14, 15,0, 1,2,3,4,5,6,7,8,9,10, 11, ...
- In the Go-Back-N Protocol, the sequence numbers are modulo 2^m where *m* is the size of the sequence number field in bits.
- In this protocol, the sliding window is an abstract concept that defines the range of sequence numbers that is the concern of the sender and receiver.
- The range which is the concern of the sender is called the **send sliding window**; the range that is the concern of the receiver is called the **receive sliding window**.
- The **send window** is an imaginary box covering the sequence numbers of the data frames which can be in transit. In each window position, some of these sequence numbers define the frames that have been sent; others define those that can be sent. The maximum size of the window is $\boldsymbol{2^m - 1}$.

Send Window

- The window at any time divides the possible sequence numbers into four regions:
- The **first region**, from the far left to the left wall of the window, defines the sequence numbers belonging to **frames that are already acknowledged**. The sender does not worry about these frames and keeps no copies of them.
- The **second region** defines the range of sequence numbers belonging to **the frames that are**
- **sent and have an unknown status**. The sender needs to wait to find out if these frames have been received or were lost. We call these outstanding frames.
- The **third range** defines the range of sequence numbers for **frames that can be sent**; however, the corresponding data packets have not yet been received from the network layer.
- The **fourth region** defines sequence numbers that **cannot be used until the window slides**.



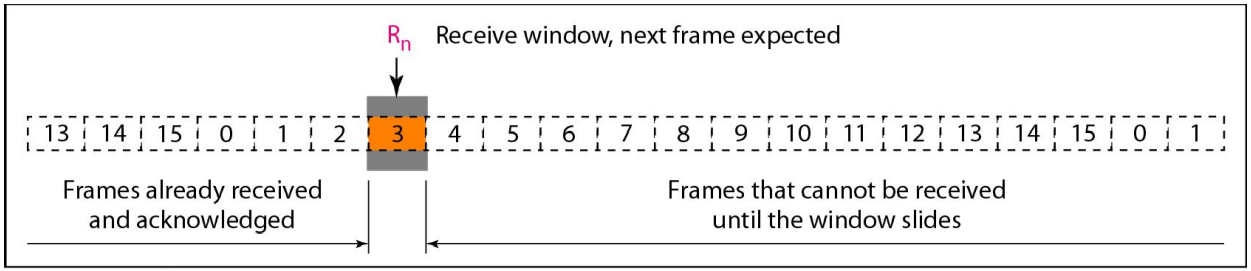
- The window itself is an abstraction; three variables define its size and location at any time. S_f (*send* window, the first outstanding frame), S_n (send window, the next frame to be sent), and S_{size} (send window, size).
- The variable S_f defines the **sequence number of the first (oldest) outstanding frame**.
- The variable S_n holds the **sequence number that will be assigned to the next frame to be sent.**
- The variable S_{size} **defines the size of the window**.
- In Figure, frames 0, 1, and 2 are acknowledged, so the window has slid to the right three slots. Note that the value of $S_f = 3$ because frame 3 is now the first outstanding frame.



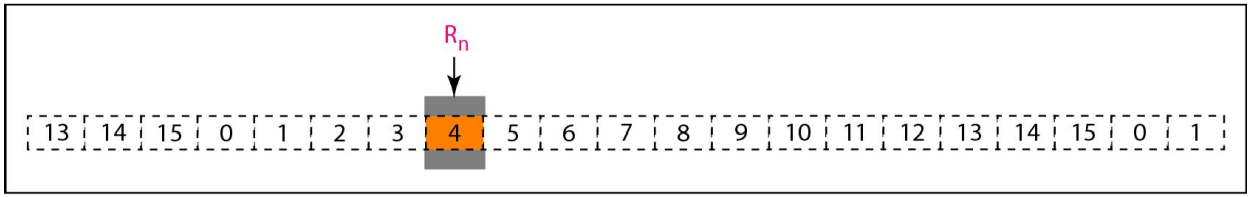
b. Send window after sliding

Receive Window

- The receive window makes sure that the correct data frames are received and that the correct acknowledgments are sent.
- The size of the receive window is always 1.
- The receiver looks for the arrival of a specific frame. Any frame arriving out of order is discarded and needs to be resent.
- Only one variable R_n (receive window, next frame expected) to define this abstraction. The sequence numbers to the left of the window belong to the frames already received and acknowledged; the sequence numbers to the right of this window define the frames that cannot be received.
- Any received frame with a sequence number in these two regions is discarded. Only a frame with a sequence number matching the value of R_n is accepted and acknowledged.
- The receive window also slides, but only one slot at a time. When a correct frame is received (and a frame is received only one at a time), the window slides.



a. Receive window

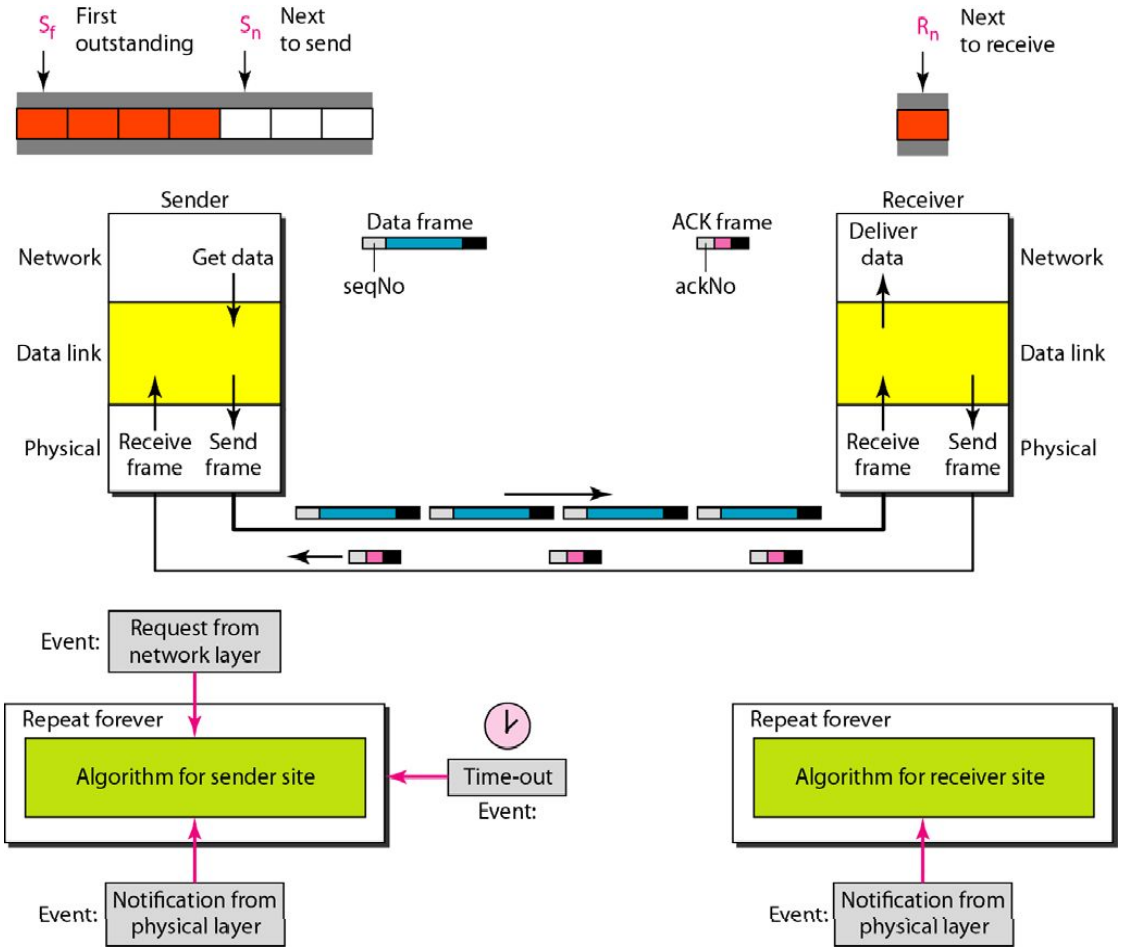


- The receiver sends a positive acknowledgment if a frame has arrived safe and sound and in order.
- If a frame is damaged or is received out of order, the receiver is silent and will discard all subsequent frames until it receives the one it is expecting. The silence of the receiver causes the timer of the unacknowledged frame at the sender site to expire.
- This, in turn, causes the sender to go back and resend all frames, beginning with the one with the expired timer. The receiver does not have to acknowledge each frame received.
- It can send one cumulative acknowledgment for several frames.
- When the timer expires, the sender resends all outstanding frames. For example, suppose the sender has already sent frame 6, but the timer for frame 3 expires. This means that frame 3 has not been acknowledged; the sender goes back and sends frames 3, 4,5, and 6 again. That is why the protocol is called ***Go-Back-N*** **ARQ**.

Note

The receive window is an abstract concept defining an imaginary box of size 1 with one single variable R_n. The window slides when a correct frame has arrived; sliding occurs one slot at a time.

Figure 11.14 *Design of Go-Back-N ARQ*



Note

In Go-Back-N ARQ, the size of the send window must be less than 2^m; the size of the receiver window is always 1.

Algorithm 11.7 *Go-Back-N sender algorithm*

```
1  Sw = 2^m - 1;
2  Sf = 0;
3  Sn = 0;
4
5  while (true)                          //Repeat forever
6  {
7   WaitForEvent();
8     if(Event(RequestToSend))           //A packet to send
9     {
10       if(Sn-Sf >= Sw)                //If window is full
11             Sleep();
12       GetData();
13       MakeFrame(Sn);
14       StoreFrame(Sn);
15       SendFrame(Sn);
16       Sn = Sn + 1;
17       if(timer not running)
18            StartTimer();
19    }
20
```

(continued)

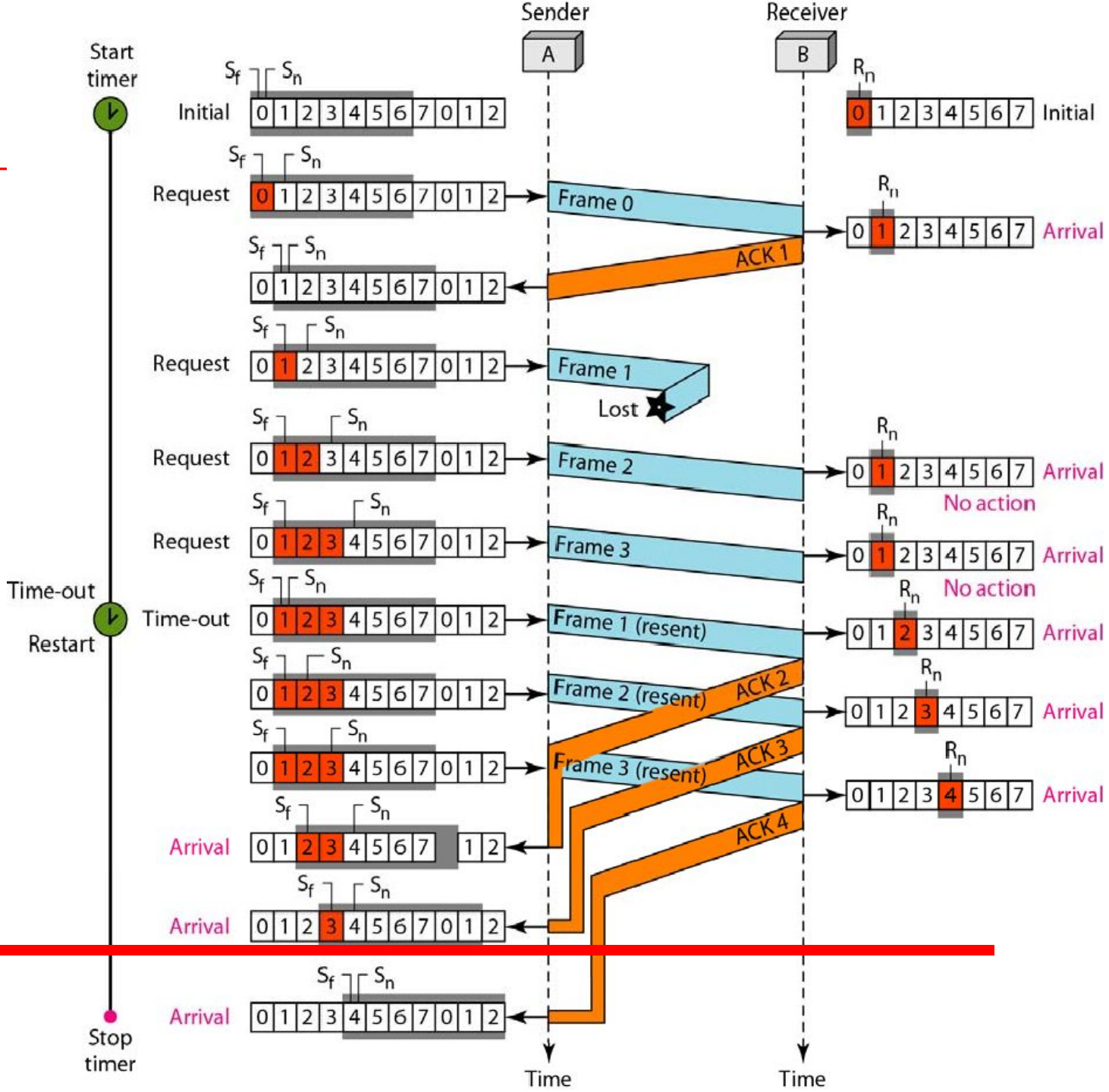
Algorithm 11.7 *Go-Back-N sender algorithm* **(continued)**

```
21   if(Event(ArrivalNotification)) //ACK arrives
22   {
23      Receive(ACK);
24      if(corrupted(ACK))
25           Sleep();
26      if((ackNo>Sf)&&(ackNo<=Sn))  //If a valid ACK
27      While(Sf <= ackNo)
28       {
29        PurgeFrame(Sf);
30        Sf = Sf + 1;
31       }
32       StopTimer();
33   }
34
35   if(Event(TimeOut))                  //The timer expires
36   {
37    StartTimer();
38    Temp = Sf;
39    while(Temp < Sn);
40      {
41       SendFrame(Sf);
42       Sf = Sf + 1;
43      }
44   }
45 }
```

Algorithm 11.8 *Go-Back-N receiver algorithm*

```
1  Rn = 0;
2
3  while (true)                          //Repeat forever
4  {
5    WaitForEvent();
6
7    if(Event(ArrivalNotification)) /Data frame arrives
8    {
9       Receive(Frame);
10      if(corrupted(Frame))
11           Sleep();
12      if(seqNo == Rn)                  //If expected frame
13      {
14        DeliverData();                 //Deliver data
15        Rn = Rn + 1;                   //Slide window
16        SendACK(Rn);
17      }
18   }
19 }
```

Go-back N ARQ


Sender
Receiver
A
B
Start timer
Initial
Request
Frame 0
ACK 1
Request
Frame 1
Lost
Request
Frame 2
Arrival
No action
Request
Frame 3
Arrival
No action
Time-out
Restart
Time-out
Frame 1 (resent)
Arrival
Frame 2 (resent)
ACK 2
Arrival
Frame 3 (resent)
ACK 3
Arrival
ACK 4
Arrival
Arrival
Stop timer
Time
Time

Note

Stop-and-Wait ARQ is a special case of Go-Back-N ARQ in which the size of the send window is 1.

Figure 11.18 *Send window for Selective Repeat ARQ*

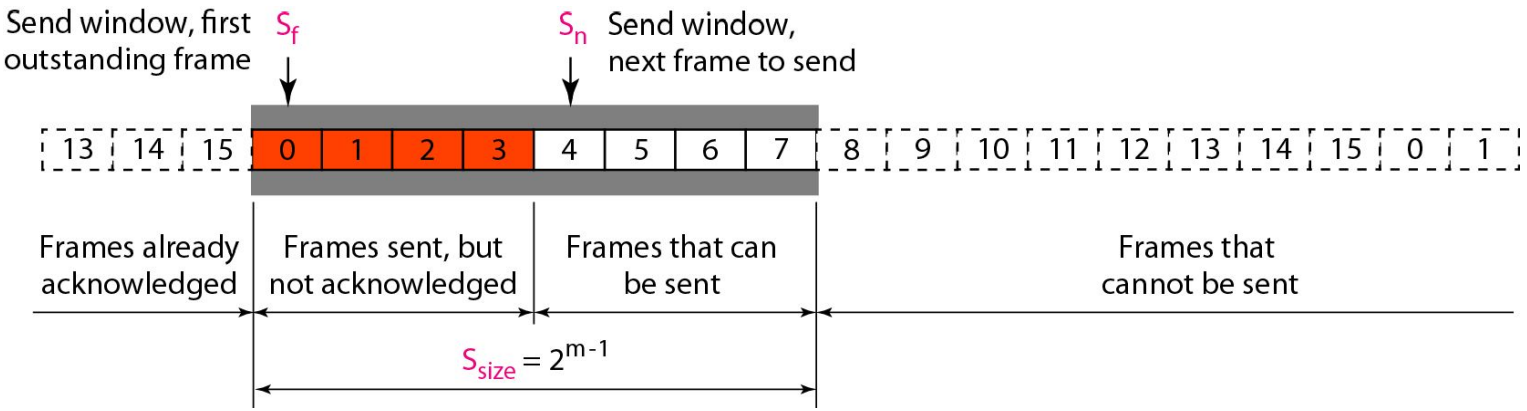


Figure 11.19 *Receive window for Selective Repeat ARQ*

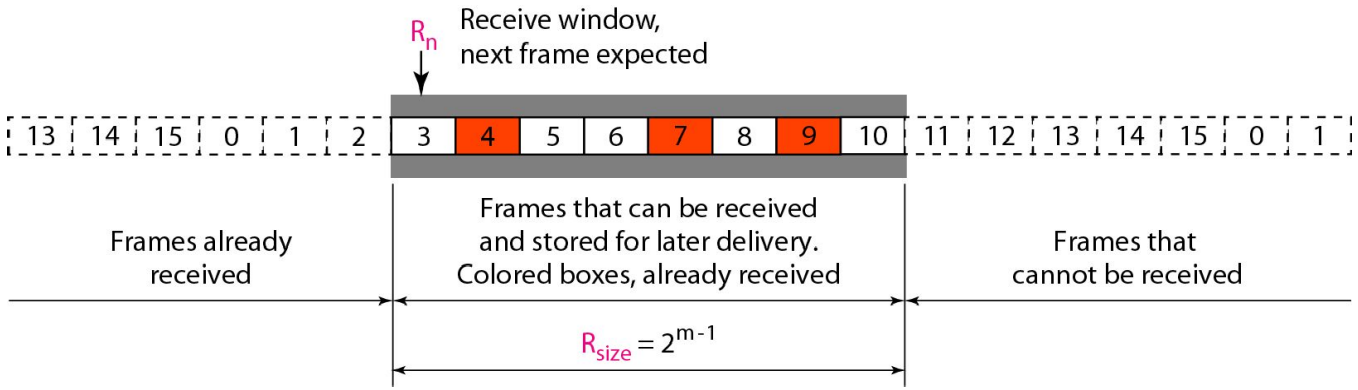
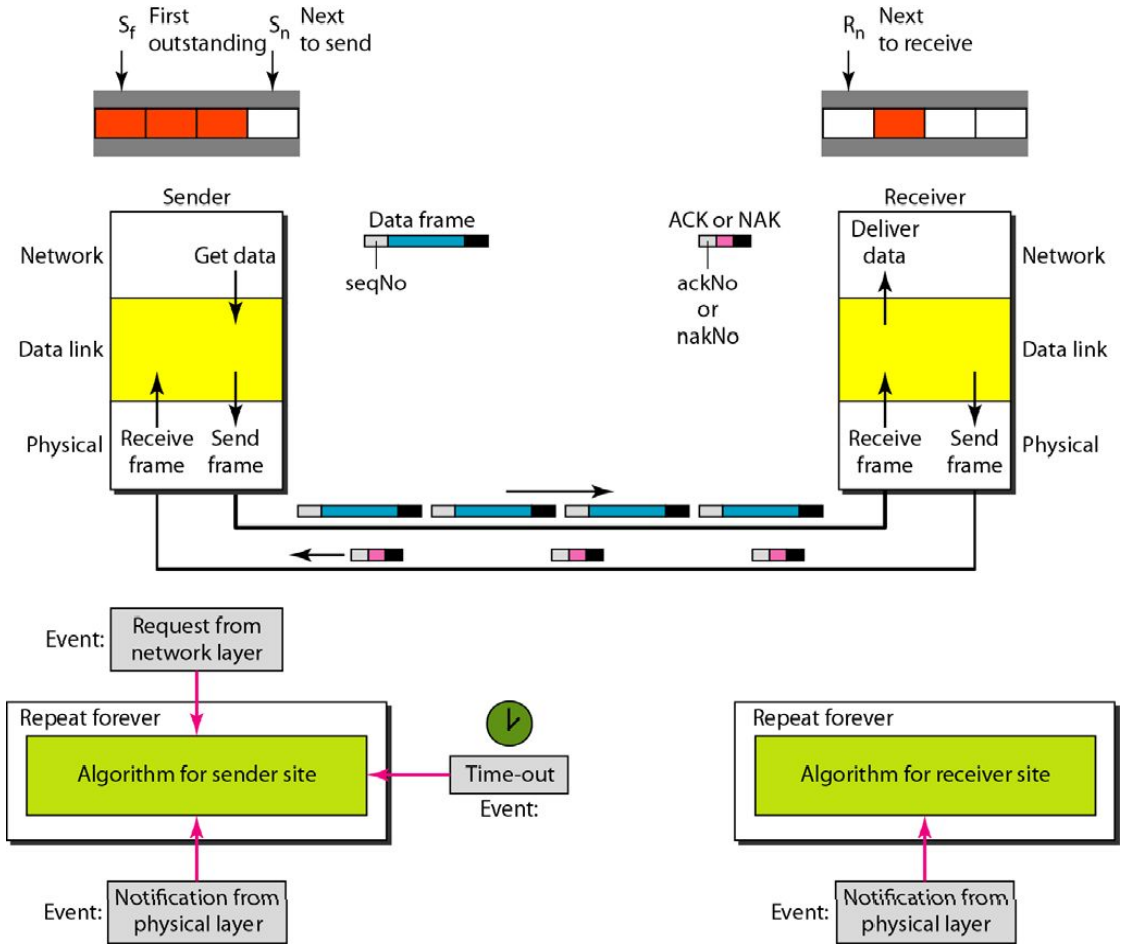


Figure 11.20 *Design of Selective Repeat ARQ*



Note

In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of 2^m.

Algorithm 11.9 *Sender-site Selective Repeat algorithm*

```
1  Sw = 2^(m-1) ;
2  Sf = 0;
3  Sn = 0;
4
5  while (true)                          //Repeat forever
6  {
7    WaitForEvent();
8    if(Event(RequestToSend))            //There is a packet to send
9    {
10     if(Sn-Sf >= Sw)                   //If window is full
11           Sleep();
12     GetData();
13     MakeFrame(Sn);
14     StoreFrame(Sn);
15     SendFrame(Sn);
16     Sn = Sn + 1;
17     StartTimer(Sn);
18   }
19
```

(continued)

Algorithm 11.9 *Sender-site Selective Repeat algorithm* (continued)

```
20  if(Event(ArrivalNotification)) //ACK arrives
21  {
22     Receive(frame);              //Receive ACK or NAK
23     if(corrupted(frame))
24          Sleep();
25     if (FrameType == NAK)
26        if (nakNo between Sf and Sn)
27        {
28         resend(nakNo);
29         StartTimer(nakNo);
30        }
31     if (FrameType == ACK)
32        if (ackNo between Sf and Sn)
33        {
34          while(sf < ackNo)
35          {
36           Purge(sf);
37           StopTimer(sf);
38           Sf = Sf + 1;
39          }
40        }
41  }
```

(continued)

Algorithm 11.9 *Sender-site Selective Repeat algorithm* **(continued)**

```
42
43    if(Event(TimeOut(t)))              //The timer expires
44    {
45     StartTimer(t);
46     SendFrame(t);
47    }
48  }
```

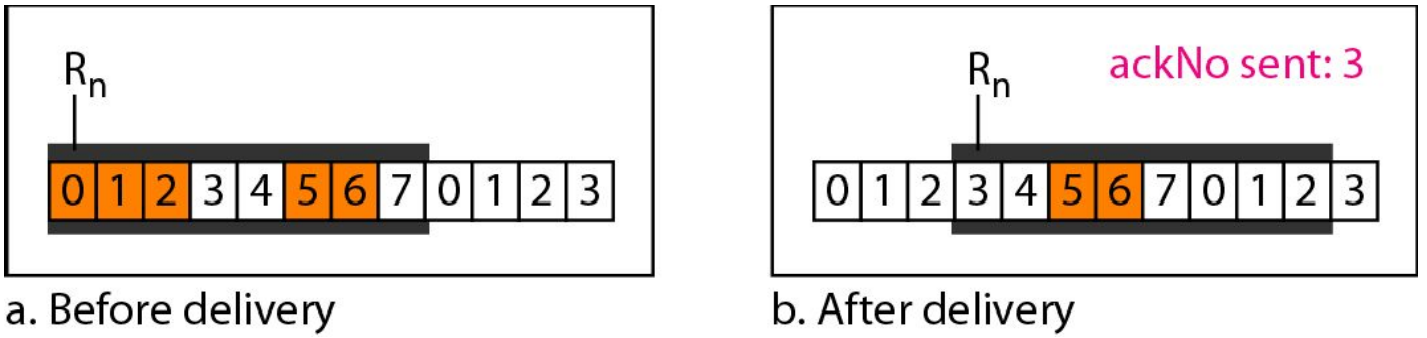
Algorithm 11.10 *Receiver-site Selective Repeat algorithm*

```
 1  Rn = 0;
 2  NakSent = false;
 3  AckNeeded = false;
 4  Repeat(for all slots)
 5      Marked(slot) = false;
 6
 7  while (true)                                  //Repeat forever
 8  {
 9    WaitForEvent();
10
11    if(Event(ArrivalNotification))              /Data frame arrives
12    {
13       Receive(Frame);
14       if(corrupted(Frame))&& (NOT NakSent)
15       {
16        SendNAK(Rn);
17        NakSent = true;
18        Sleep();
19       }
20       if(seqNo <> Rn)&& (NOT NakSent)
21       {
22        SendNAK(Rn);
```

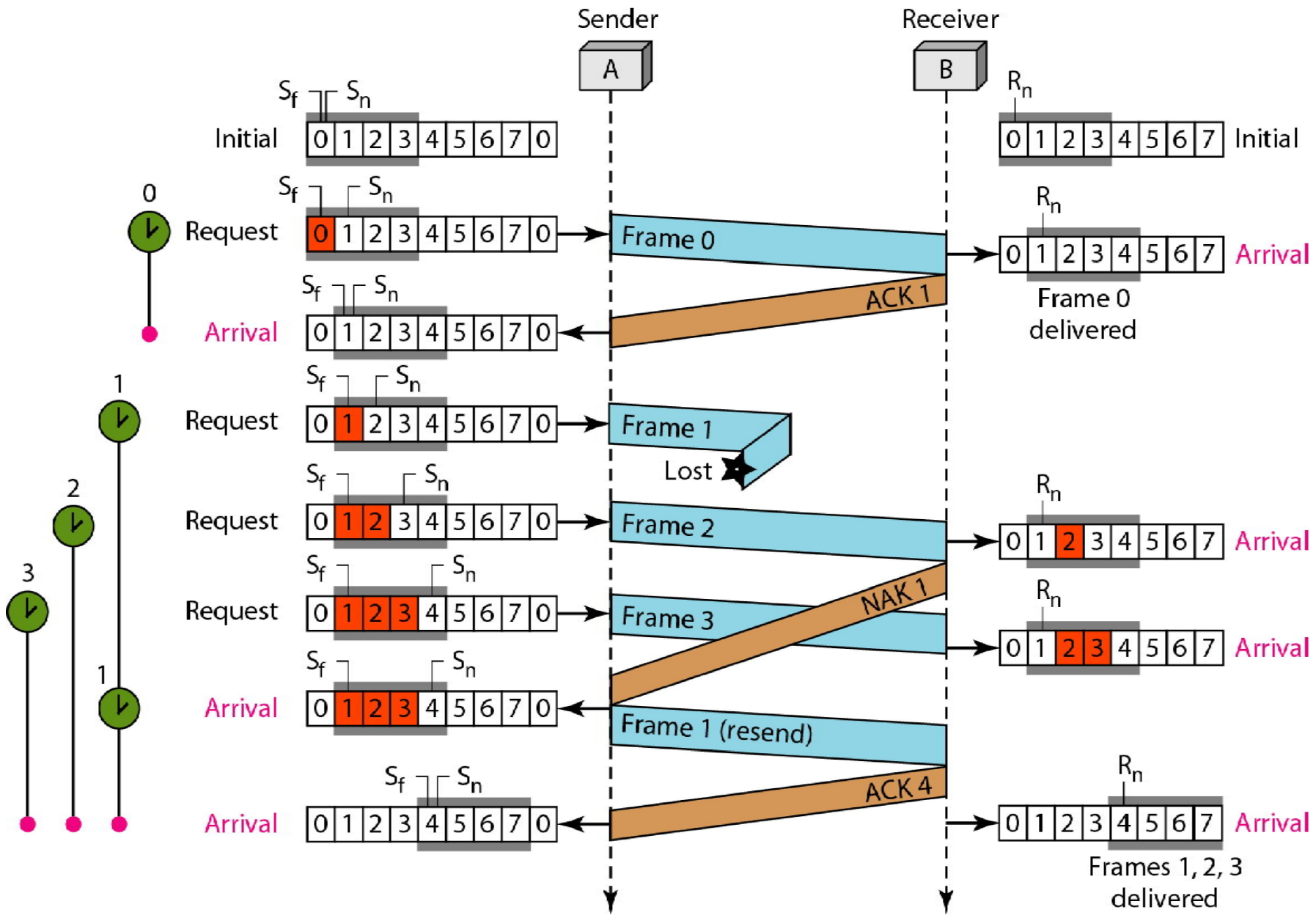
Algorithm 11.10 *Receiver-site Selective Repeat algorithm*

```
23          NakSent = true;
24          if ((seqNo in window)&&(!Marked(seqNo))
25          {
26           StoreFrame(seqNo)
27           Marked(seqNo)= true;
28           while(Marked(Rn))
29           {
30            DeliverData(Rn);
31            Purge(Rn);
32            Rn = Rn + 1;
33            AckNeeded = true;
34           }
35            if(AckNeeded);
36            {
37            SendAck(Rn);
38            AckNeeded = false;
39            NakSent = false;
40            }
41          }
42        }
43    }
44  }
```

Figure 11.22 *Delivery of data in Selective Repeat ARQ*



Selective Repeat ARQ



HDLC

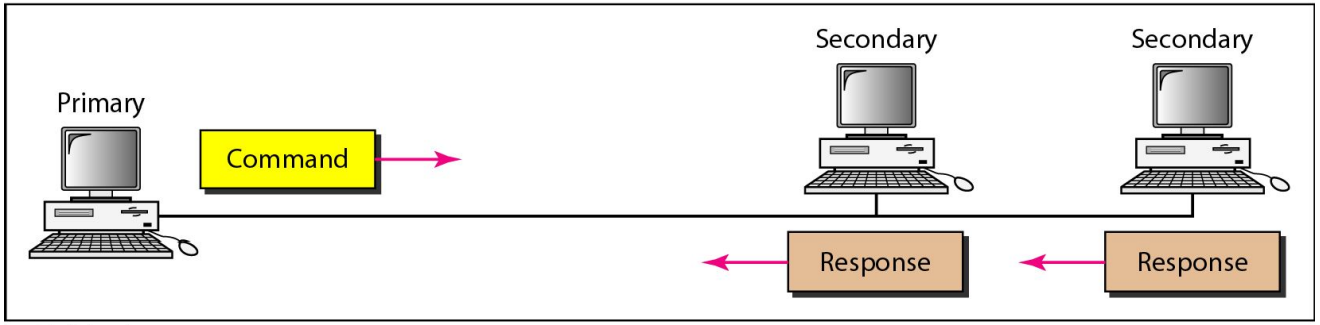
High-level Data Link Control (HDLC) is a bit-oriented protocol

- designed to support half-duplex & full duplex communication
- over point-to-point and multipoint links.
- It implements the ARQ mechanisms.

Configurations & Transfer Modes

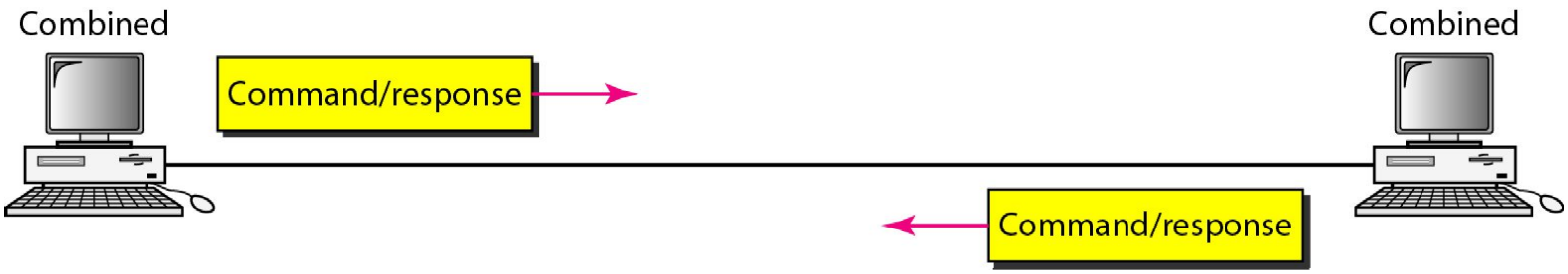
- HDLC provides 2 modes of transmission:
  - Normal Response Mode (NRM)
  - Asynchronous Balanced Mode (ABM)
- **NRM**
  - Station configuration is **unbalanced** – one **primary station** and **multiple secondary stations**.
  - A **primary station** can **send commands**; a **secondary station** can only **respond**.
  - The NRM is used for both point-to-point and multiple-point links.

a. Point-to-point



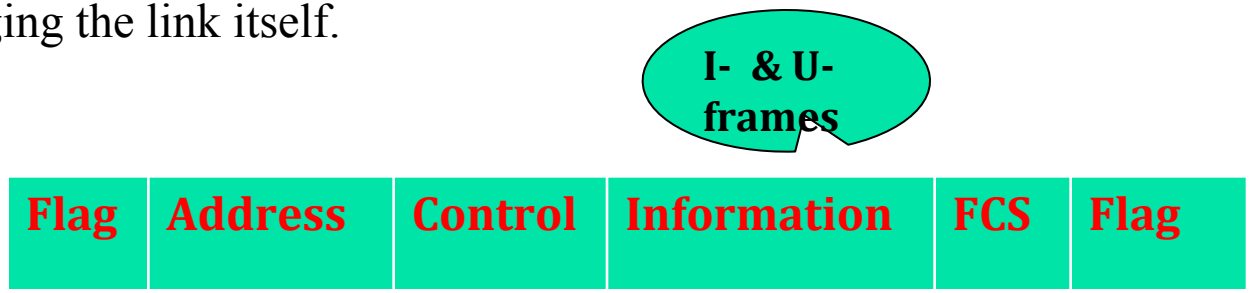
b. Multipoint

- ABM
  - configuration is balanced.
  - The link is point-to-point, and each station can function as a primary and a secondary station.



Frames

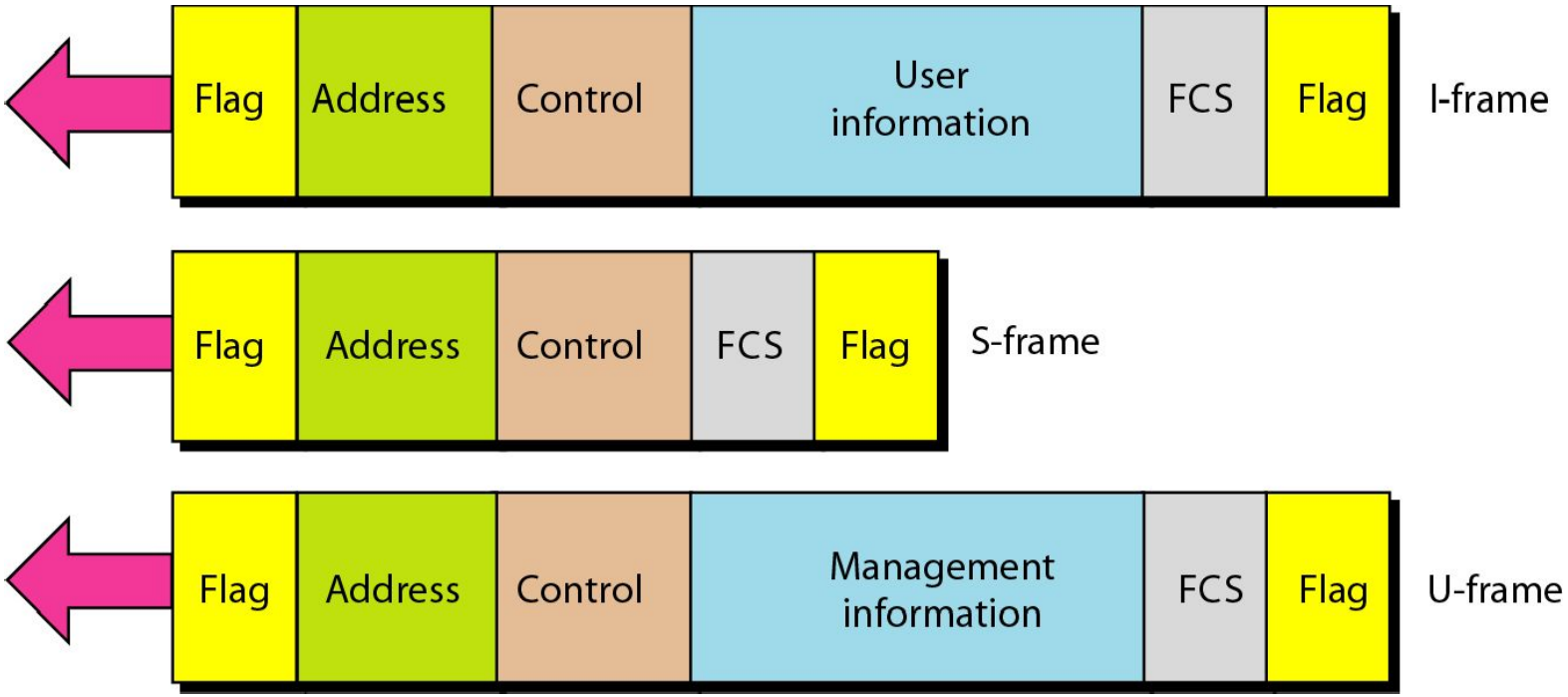
- HDLC defines **three types of frames**: information frames (I-frames), supervisory frames (S-frames), and unnumbered frames (U-frames).
- **I-frames** are used to **transport user data and control information** relating to user data (piggybacking).
- **S-frames** are used only to **transport control information**.
- **U-frames** are reserved for **system management**. Information carried by U-frames is intended for managing the link itself.



- Each frame in HDLC may contain up to six fields: a beginning flag field, an address field, a control field, an information field, a frame check sequence (FCS) field, and an ending flag field.
- In multiple-frame transmissions, the ending flag of one frame can serve as the beginning flag of the next frame.

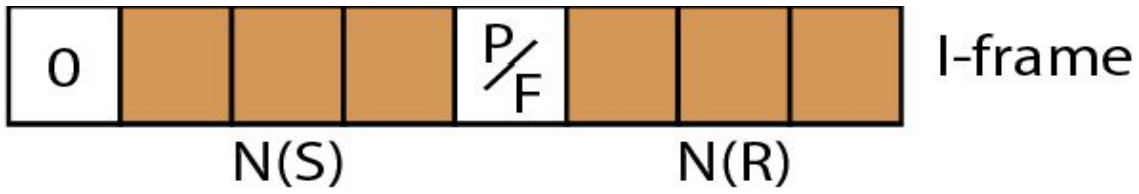
- **Flag field**. The flag field of an HDLC frame is an **8-bit sequence** with the bit pattern **01111110** that identifies both the **beginning and the end of a frame** and serves as a **synchronization pattern for the receiver**.
- **Address field**. The second field of an HDLC frame contains the **address of the secondary station.** If **a primary station created the frame**, it contains a ***to*** **address.** If a **secondary creates the frame**, it contains ***a from*** **address**. An address field can be **1 byte or several bytes long**, depending on the needs of the network. One byte can identify up to **128 stations**. If the **address is more than 1 byte**, all bytes but the last one will **end with 0**; only the last will end with 1. Ending each intermediate byte with 0 indicates to the receiver that there are more address bytes to come.
- **Control field**. The control field is a **1- or 2-byte segment** of the frame used for **flow and error control.** The interpretation of bits in this field depends on the frame type.
- **Information field**. The information field contains the **user's data from the network layer** or management information. Its length can vary from one network to another.
- **FCS field**. The frame check sequence (FCS) is the **HDLC error detection** field. It can contain either a **2- or 4-byte ITU-T CRC**.

Figure 11.27 *HDLC frames*



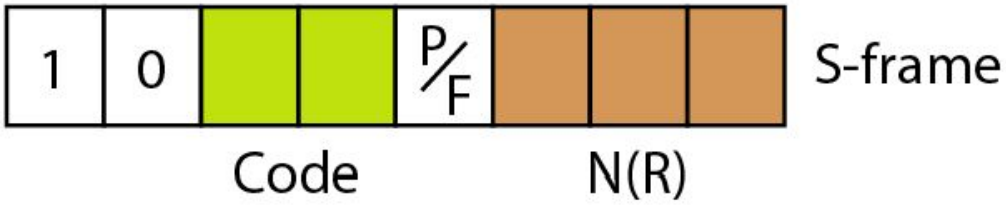
Control field - I frames

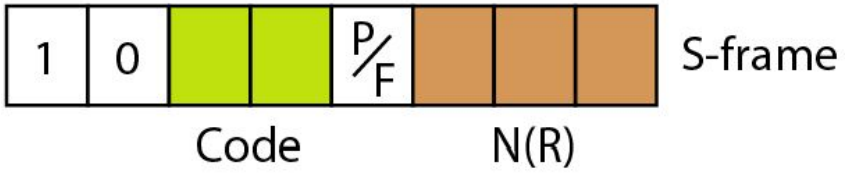
- **I-frames** are designed to carry **user data from the network layer**. In addition, they can include flow and error control information (piggybacking).
- The **first bit** defines the **type.** If the first bit of the control field is **0**, this means the frame is an **I-frame.**
- The **next 3 bits, called *N(S),*** define the **sequence number** of the frame. Note that with 3 bits, we can define a sequence number between **0 and 7**; but in the extension format, in which the control field is 2 bytes, this field is larger.
- The **last 3 bits, called *N(R),*** correspond to the **acknowledgment number** when piggybacking is used. The single bit between *N(S)* and *N(R)* is called the *P/F* bit.
- The *P/F* field is a single bit with a dual purpose. It has meaning only when it is set (bit = 1) and can mean poll or final. It means ***poll*** when the frame is sent by a **primary station to a secondary.**
- It means ***final*** when the frame is sent by a **secondary station to a primary station**.



Control field - S frames

- **Supervisory frames** are used for **flow and error control** whenever piggybacking is either impossible or not.
- S-frames do not have information fields. If the **first 2 bits** of the control field is **10**, this means the frame is an S-frame.
- The **last 3 bits, called *N(R)*,** corresponds to the **acknowledgment number** (ACK) or **negative acknowledgment number** (NAK) depending on the type of S-frame.
- The **2 bits called code** is used to define the **type of S-frame** itself.

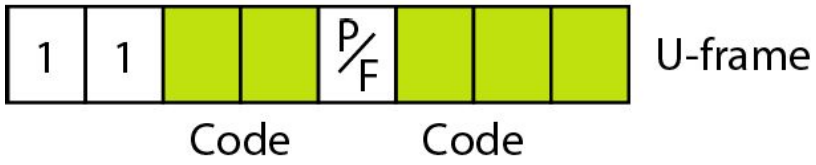




- With 2 bits, **four types of S-frames**:
  - **Receive ready (RR)** If the value of the code subfield is **00**, it is an RR S-frame. This kind of frame **acknowledges the receipt of a safe and sound frame or group of frames**. In this case, the value ***N(R)* field defines the acknowledgment number**.
  - **Receive not ready (RNR)** If the value of the code subfield is **10,** it is an RNR S-frame. This kind of frame is an RR frame with additional functions. It **acknowledges the receipt of a frame or group of frames, and it announces that the receiver is busy and cannot receive more frames**. It acts as a kind of **congestion control mechanism** by asking the sender to slow down. The value of ***N(R)*is the acknowledgment number**.
  - **Reject (REJ)** If the value of the code subfield is **01**, it is a REJ S-frame. This is a **NAK frame.** It is a NAK that can be used in ***Go-Back-N* ARQ** to improve the efficiency of the process by informing the sender, before the sender time expires, that the last frame is lost or damaged. The value of ***N(R)* is the negative acknowledgment number.**
  - **Selective reject (SREJ)** If the value of the code subfield is **11**, it is an SREJ S-frame. This is a **NAK frame used in Selective Repeat ARQ**. Note that the HDLC Protocol uses the term *selective reject* instead of *selective repeat.* The value of ***N(R)* is the negative acknowledgment number.**

Control field - U frames

- Unnumbered frames are used to exchange **session management and control information between connected devices.**
- U-frames contain an information field, but one used for system management information, not user data.
- U-frame codes are divided into two sections: a **2-bit prefix before the P/F bit** and a **3-bit suffix after the P/F bit**. Together, these two segments (5 bits) can be used to create up to **32 different types of U-frames.**



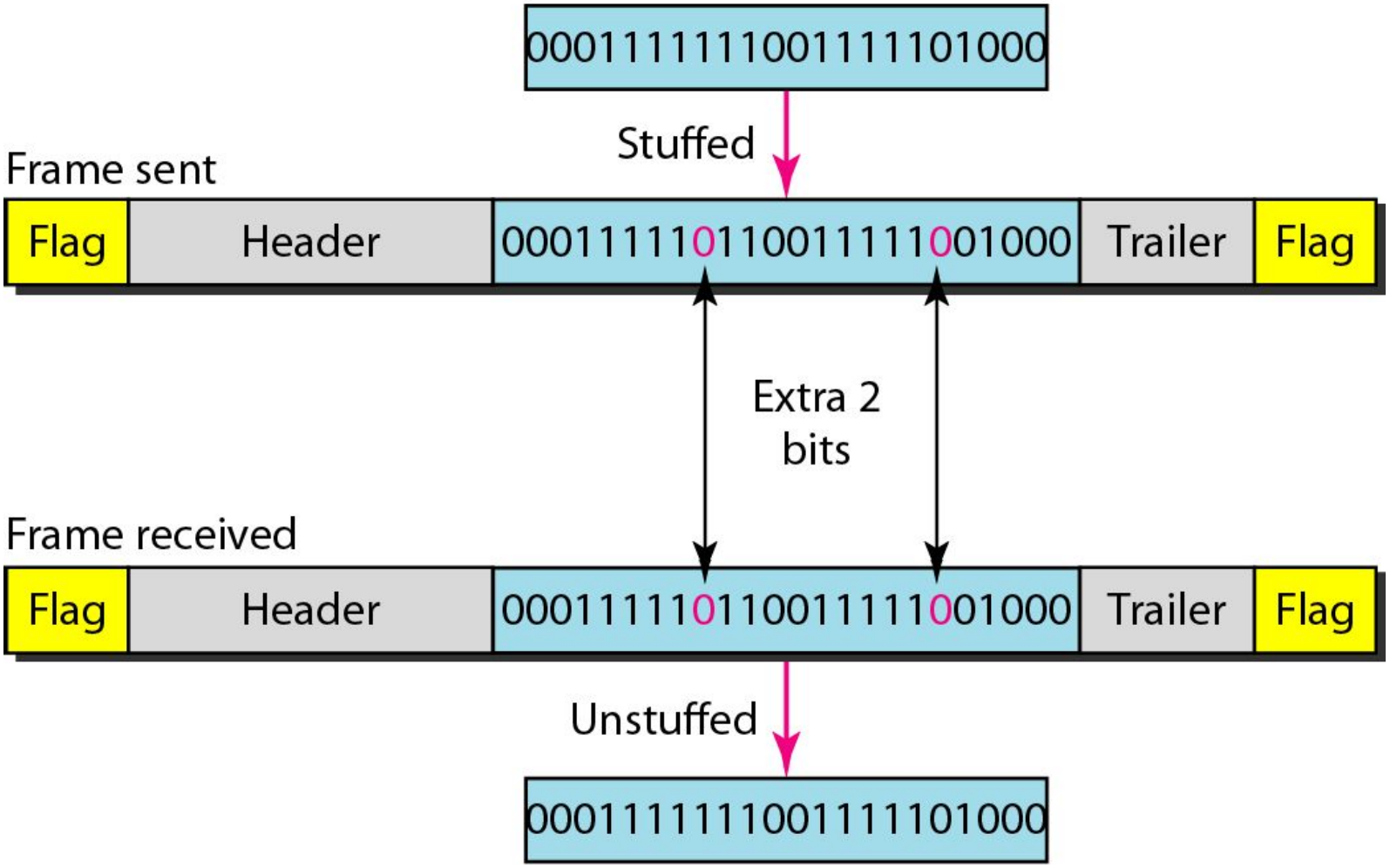
U-frame control command and response

Code	*Command*	*Response*	*Meaning*
00 001	SNRM		Set normal response mode
11 011	SNRME		Set normal response mode, extended
11 100	SABM	**DM**	Set asynchronous balanced mode or **disconnect mode**
11 110	SABME		Set asynchronous balanced mode, extended
00 000	UI	**UI**	Unnumbered information
00 110		**UA**	**Unnumbered acknowledgment**
00 010	DISC	**RD**	Disconnect or **request disconnect**
10 000	SIM	**RIM**	Set initialization mode or **request information mode**
00 100	UP		Unnumbered poll
11 001	RSET		Reset
11 101	XID	**XID**	Exchange ID
10 001	FRMR	**FRMR**	Frame reject

Data Transparency

- If the data field of an HDLC frame contains a pattern that is same as the flag 01111110, the receiver considers it as an ending flag. The rest of the bits are assumed to be the next frame. This phenomena is known as lack of **data transparency**.
- When the data is transparent, all data are recognized as data and all control information is recognized as control information.

Bit Stuffing

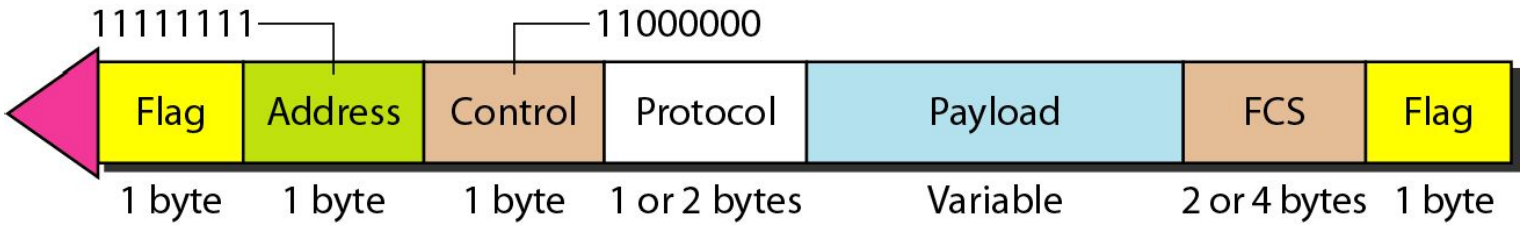


POINT-TO-POINT PROTOCOL

- One of the most common protocols for point-to-point access is the Point-to-Point Protocol (PPP).
- PPP is a byte-oriented protocol.
- PPP defines the **format of the frame to be exchanged** between devices.
- PPP defines how 2 **devices negotiate the establishment of the link** and **exchange of data**.
- PPP defines how **network layer data are encapsulated in the data link layer**.
- PPP defines how 2 devices can **authenticate** each other.
- PPP provides **multiple network layer services** supporting a variety of network layer protocols.
- PPP provides **connections over multiple links**.
- PPP provides **network address configuration**. This is particularly useful when a home user needs a temporary network address to connect to the Internet.

Frame Format

- **Flag** A PPP frame starts and ends with a 1-byte flag with the bit pattern **01111110**. PPP is a byte-oriented protocol; HDLC is a bit-oriented protocol. The flag is treated as a byte.
- **Address** The address field in this protocol is a constant value and set to 11111111, uses the broadcast address of HDLC.
- **Control** uses the format of U-frame in HDLC. The value is 11000000 to show that the frame does not contain any sequence numbers and there is no flow or error control.
- **Protocol** The protocol field defines what is being carried in the data field: either user data or other information. This field is by default 2 bytes long, but the two parties can agree to use only 1 byte.
- **Payload field** This field carries either the user data or other information. The data field is a sequence of bytes with the default of a maximum of 1500 bytes; but this can be changed during negotiation. The data field is byte stuffed if the flag byte pattern appears in this field. Because there is no field defining the size of the data field, padding is needed if the size is less than the maximum default value or the maximum negotiated value.
- **FCS** The frame check sequence (FCS) is simply a 2-byte or 4-byte standard CRC.



Byte stuffing

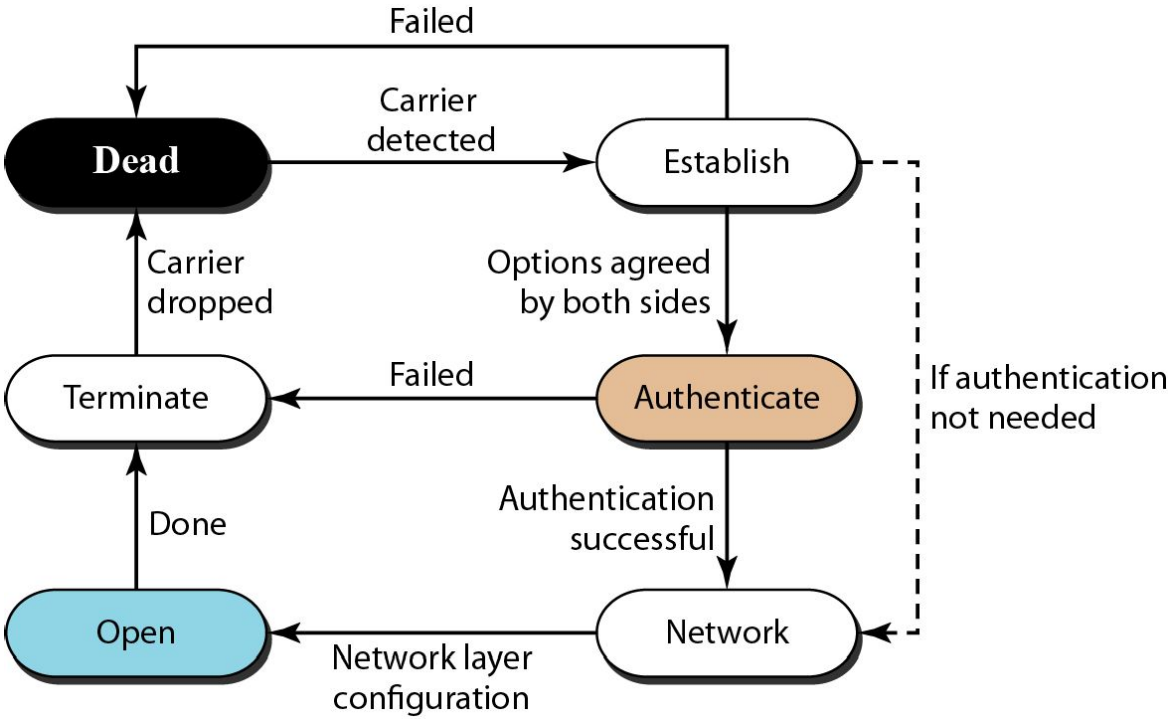
- As a byte-oriented protocol, the flag in PPP is a byte and needs to be escaped whenever it appears in the data section of the frame.
- The **escape byte** is **01111101**, which means that every time the flag-like pattern appears in the data, this extra byte is stuffed to tell the receiver that the next byte is not a flag.

PPP is a byte-oriented protocol using byte stuffing with the escape byte 01111101

Transition Phases

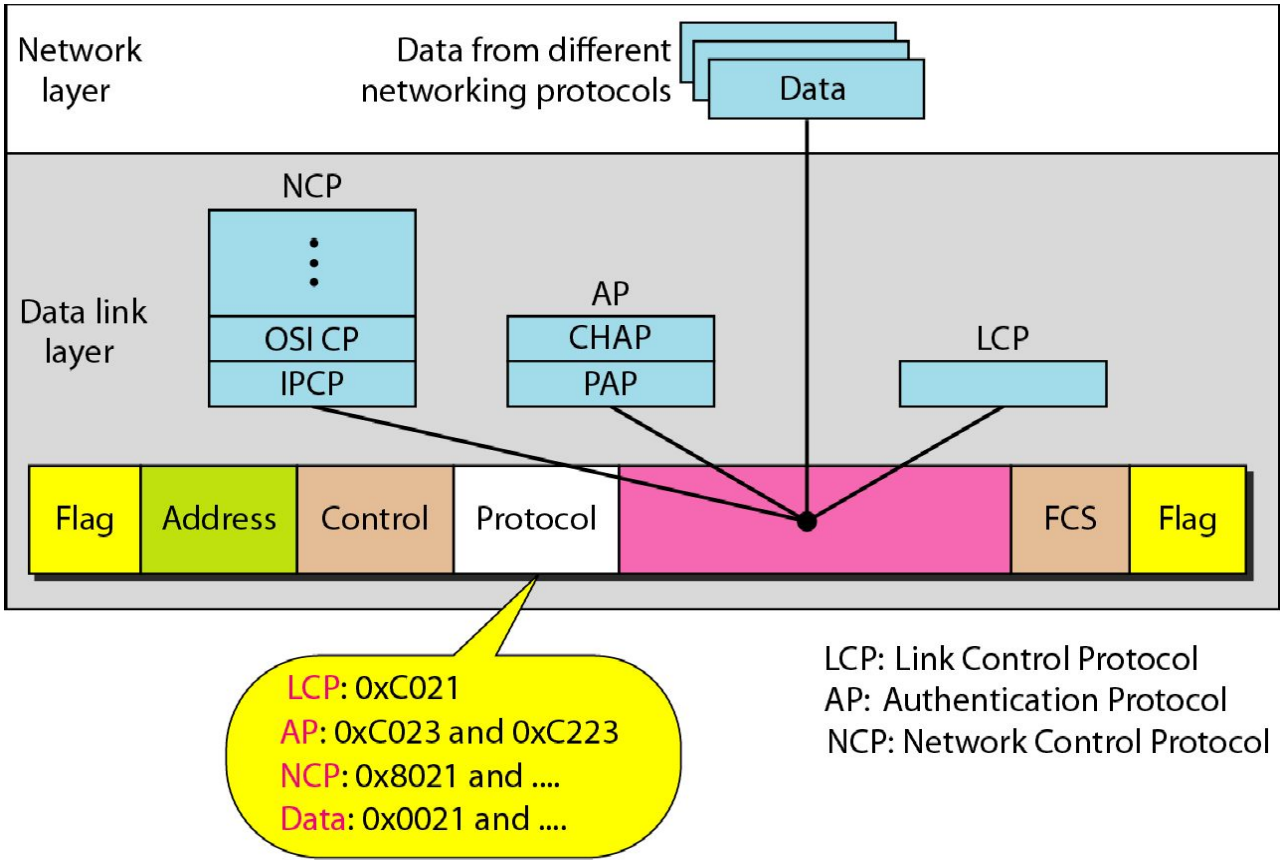
- **Dead** In the dead phase the link is not being used. There is no active carrier (at the physical layer) and the line is quiet.
- **Establish** When one of the nodes starts the communication, the connection goes into this phase. In this phase, options are negotiated between the two parties. If the negotiation is successful, the system goes to the authentication phase (if authentication is required) or directly to the networking phase.
- **Authenticate** The authentication phase is optional; the two nodes may decide, during the establishment phase, not to skip this phase. However, if they decide to proceed with authentication, they send several authentication packets. If the result is successful, the connection goes to the networking phase; otherwise, it goes to the termination phase.
- **Network** In the network phase, negotiation for the network layer protocols takes place. PPP specifies that two nodes establish a network layer agreement before data at the network layer can be exchanged.
- **Open** In the open phase, data transfer takes place. When a connection reaches this phase, the exchange of data packets can be started. The connection remains in this phase until one of the endpoints wants to terminate the connection.
- **Terminate** In the termination phase the connection is terminated. Several packets are exchanged between the two ends for closing the link.

Transition phases



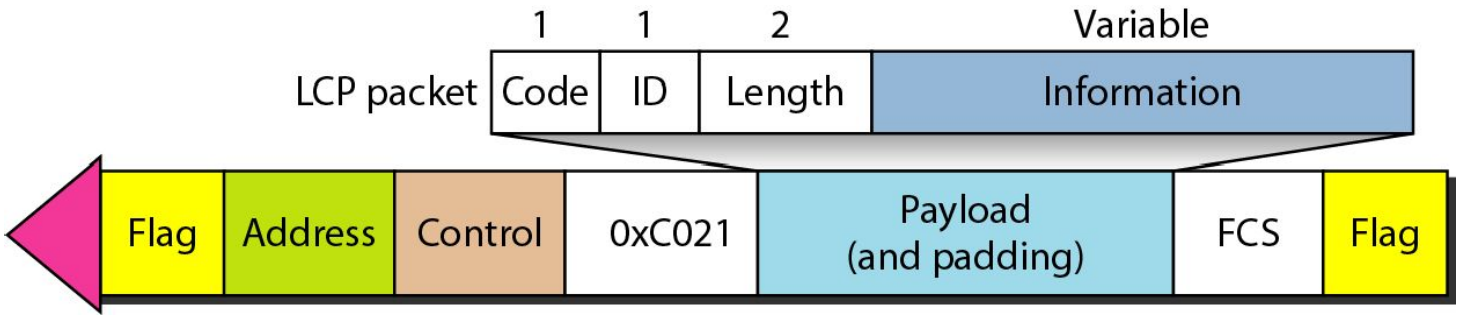
PPP stack

- PPP – data link layer protocol
- PPP uses a stack of other protocols to **establish the link**, to **authenticate** the parties involved and to **carry the network layer data**.
- 3 protocols defines PPP – **Link Control Protocol**, **authentication protocols**, **Network Control Protocol**.



Link Control Protocol (LCP)

- The **Link Control Protocol** (LCP) is responsible for **establishing, maintaining, configuring, and terminating links**.
- It also provides **negotiation mechanisms** to set options between the two endpoints. Both endpoints of the link must reach an agreement about the options before the link can be established.
- When **PPP is carrying an LCP packet**, it is either in the **establishing state or in the terminating state**. **No user data** are carried during these states.
- All **LCP packets** are carried in the **payload field** of the **PPP frame** with the **protocol field set to C021 in hexadecimal**.
- **Code**: defines the type of LCP packet. There are 11 types of packets.
- **ID**: holds a value to match a request with the reply. One endpoint inserts a value in this field which will be copied in the reply packet.
- **Length**: defines the length of the LCP packet.
- **Information**: contains extra information needed for the LCP packets.



LCP Packets

- **Configuration packets** – used for **link configuration during the establish phase.**
- **Link Termination packets** - used for **link termination during the termination phase**.
- **Link Monitoring and Debugging packets** - The last five packets are used for **link monitoring and debugging.**
- **Configuration Packets:**

Configuration Packets Configuration packets are used to negotiate the options between two ends. Four different packets are used for this purpose: configure-request, configure-ak, configure-nak, and configure-reject.

- **Configure-request.** The endpoint that wishes to start a connection sends a configure-request message with a list of zero or more options to the other endpoint. Note that all the options are negotiated in one packet.
- **Configure-ack.** If all the options listed in the configure-request packet are accepted by the receiving end, it sends a configure-ack packet, which repeats all the options requested.
- **Configure-nak.** If the receiver of the configure-request packet recognizes all the options but finds that some need to be omitted or revised (the values must be changed), it sends a configure-nak packet to the sender. The sender then omits or revises the options and sends a totally new configure-request packet.
- **Configure-reject.** If some of the options are not recognized by the receiving party, it responds with a configure-reject packet, marking those options that are not recognized. The sender of the request must revise the configure-request message and send a totally new one.

Link Termination packets

Link Termination Packets The link termination packets are used to disconnect the link between two endpoints.

- **Terminate-request.** Either party can terminate the link by sending a terminate-request packet.
- **Terminate-ack.** The party that receives the terminate-request packet must answer with a terminate-ack packet.

Link Monitoring and Debugging packets

Link Monitoring and Debugging Packets These packets are used for monitoring and debugging the link.

- **Code-reject.** If the endpoint receives a packet with an unrecognized code in the packet, it sends a code-reject packet.
- **Protocol-reject.** If the endpoint receives a packet with an unrecognized protocol in the frame, it sends a protocol-reject packet.
- **Echo-request.** This packet is sent to monitor the link. Its purpose is to see if the link is functioning. The sender expects to receive an echo-reply packet from the other side as proof.
- **Echo-reply.** This packet is sent in response to an echo-request. The information field in the echo-request packet is exactly duplicated and sent back to the sender of the echo-request packet.
- **Discard-request.** This is a kind of loopback test packet. It is used by the sender to check its internal condition. The receiver of the packet just discards it.

Code - LCP packets

Code	*Packet Type*	*Description*
0x01	Configure-request	Contains the list of proposed options and their values
0x02	Configure-ack	Accepts all options proposed
0x03	Configure-nak	Announces that some options are not acceptable
0x04	Configure-reject	Announces that some options are not recognized
0x05	Terminate-request	Request to shut down the line
0x06	Terminate-ack	Accept the shutdown request
0x07	Code-reject	Announces an unknown code
0x08	Protocol-reject	Announces an unknown protocol
0x09	Echo-request	A type of hello message to check if the other end is alive
0x0A	Echo-reply	The response to the echo-request message
0x0B	Discard-request	A request to discard the packet

Information field - Common options

- The **information field** contains information, such as options, needed for some LCP packets.
- There are many options that can be negotiated between the two endpoints. Options are inserted in the information field of the configuration packets.
- Information field is divided into three fields: **option type, option length, and option data.**

Option	*Default*
Maximum receive unit (payload field size)	1500
Authentication protocol	None
Protocol field compression	Off
Address and control field compression	Off

Authentication Protocols

- Authentication plays an important role in PPP – designed for use over dial-up links where verification of user identity is necessary.
- Authentication – validating the identity of a user who needs to access a set of resources.
- PPP has 2 protocols for authentication:
  - Password Authentication Protocol (PAP)
  - Challenge Handshake Authentication Protocol (CHAP)
- The protocols are used during the authentication state - No user data are exchanged only the corresponding packets to authenticate the user.

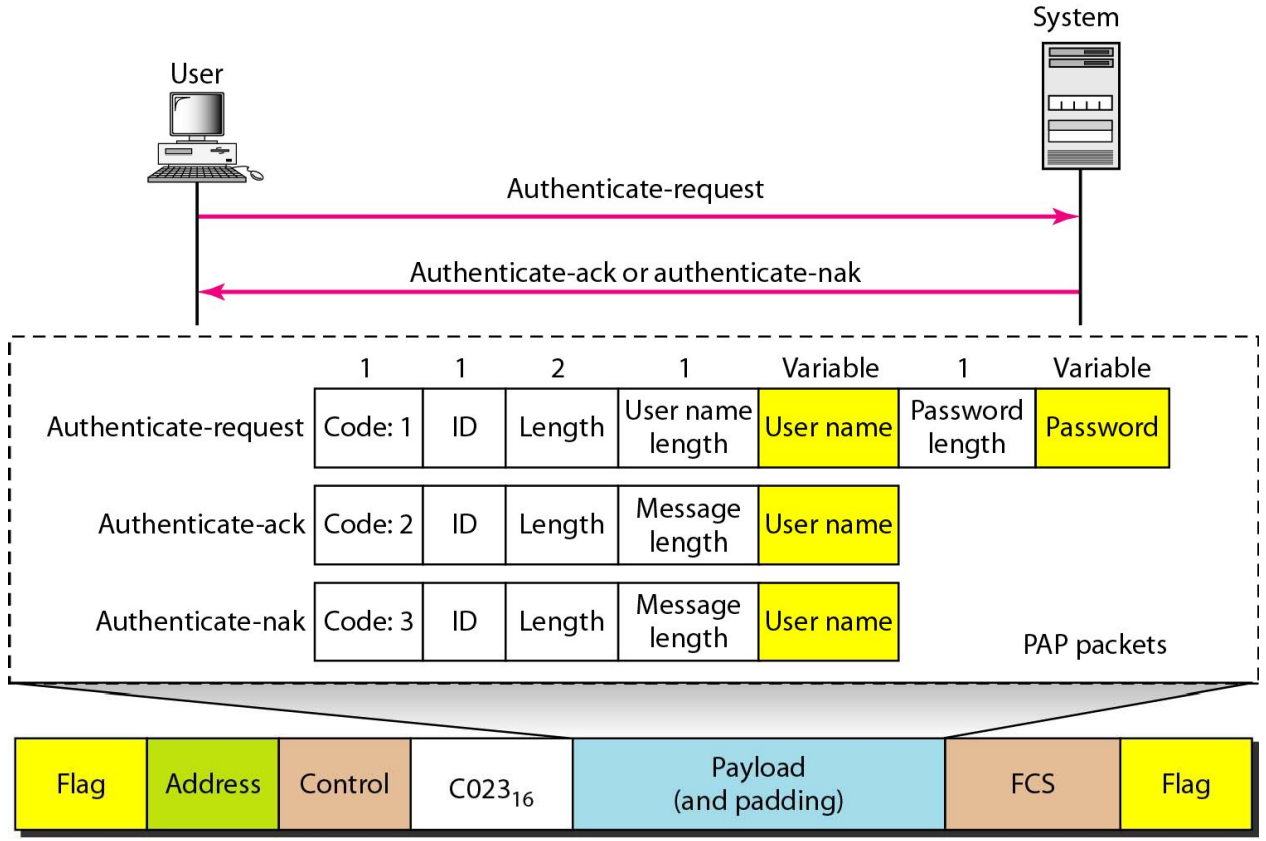
PAP

- The **Password Authentication Protocol (PAP)** is a simple authentication procedure with a two-step process:

  1. The user who wants to access a system sends an authentication identification **user name and a password**.

  2. The system checks the **validity of the identification and password** and either accepts or denies connection. (Accept/reject)

- The three types of packets used by PAP and how they are actually exchanged.
- When a PPP frame is carrying any PAP packets, the value of the protocol field is 0xC023. The three PAP packets are authenticate-request, authenticate-ack, and authenticate-nak.
- The **first packet** is used by the user to send the user name and password.
- The **second** is used by the system to allow access.
- The **third** is used by the system to deny access.

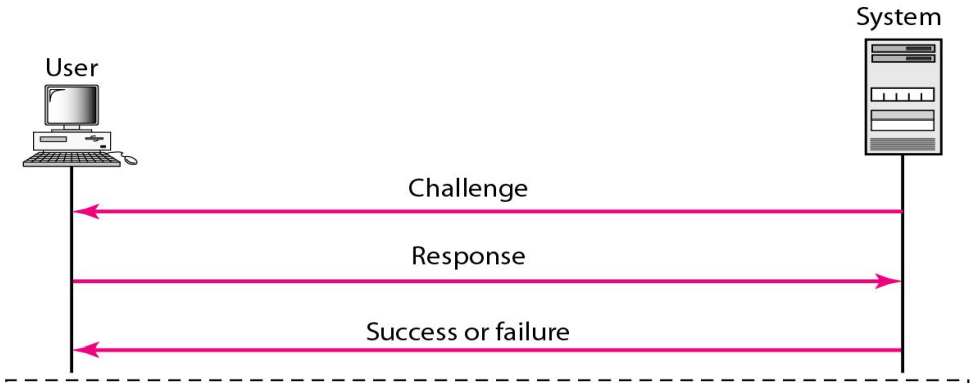


CHAP

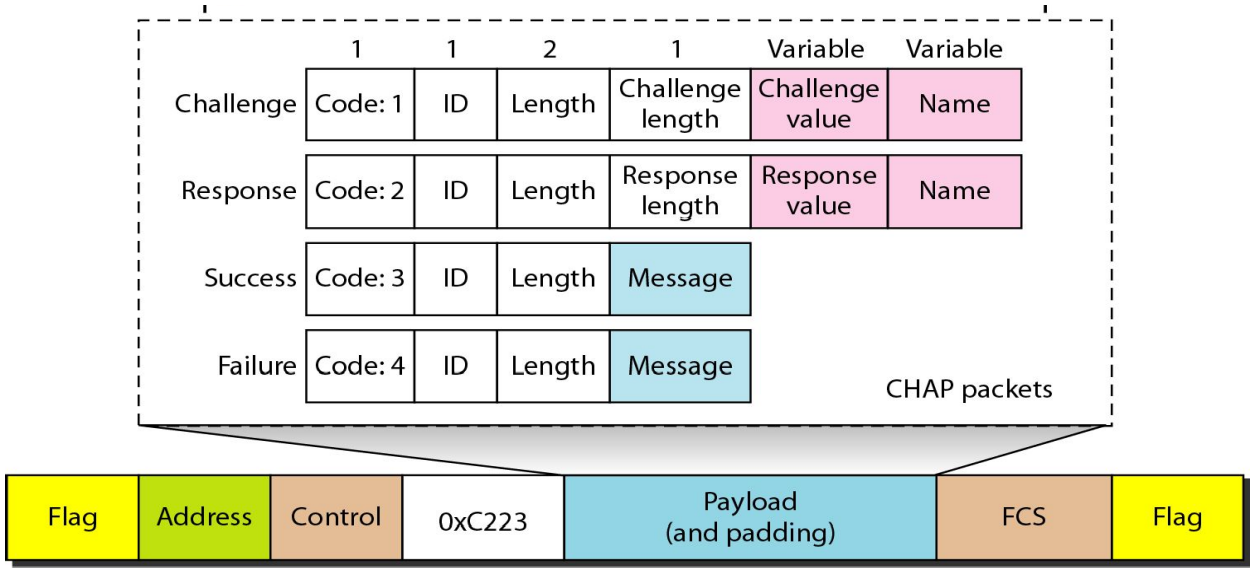
- A **three-way hand-shaking authentication protocol** that provides greater security than PAP.
- In this method, the **password is kept secret**; it is never sent online.
- The system sends the **user a challenge packet containing a challenge value**, usually a few bytes.
- The **user applies a predefined function that takes the challenge value and the user's own password and creates a result**. The user sends the result in the **response packet** to the system.
- The system does the same. It applies the same function to the password of the user (known to the system) and the challenge value to create a result. If the result created is the same as the result sent in the response packet, access is granted; otherwise, it is denied.
- CHAP is more secure than PAP, especially if the system continuously changes the challenge value.
- Even if the intruder learns the challenge value and the result, the password is still secret.

CHAP packets

- **CHAP packets** are **encapsulated** in the **PPP frame** with the **protocol value C223** in hexadecimal.
- There are four CHAP packets: challenge, response, success, and failure.
  - **Challenge** - used by the system to send the challenge value.
  - **Response** - used by the user to return the result of the calculation.
  - **Success** - used by the system to allow access to the system.
  - **Failure** - used by the system to deny access to the system.



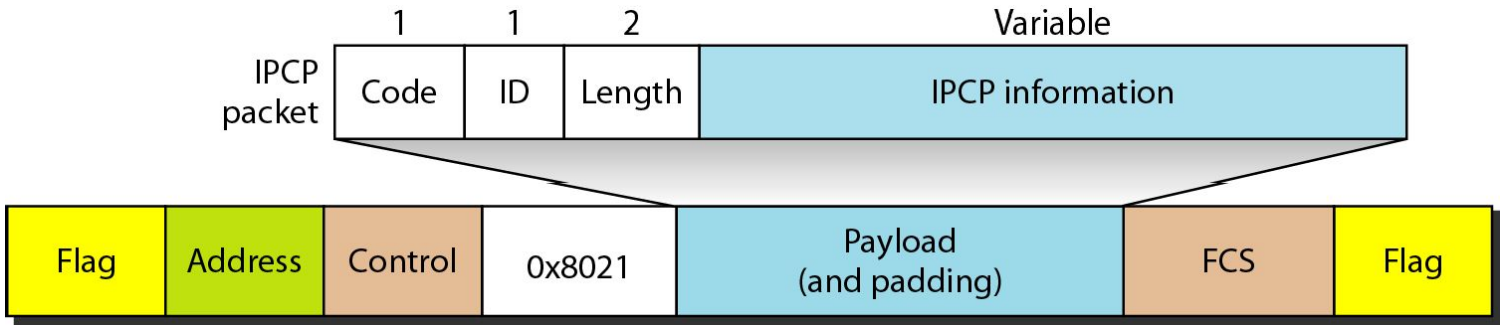
CHAP packets



Network Control Protocol (NCP)

- After the link is established and authentication is successful, the connection goes to the networking state.
- PPP uses the protocol – NCP – a set of control protocols which allows the encapsulation of data coming from the network layer protocols into the PPP frame.
- Internet Protocol Control Protocol (IPCP) – set of packets that establish and terminate a network layer connection for IP packets.

- **Format** :



- The value of the **protocol field** is 8021_{16}.

IPCP packets

- 7 packets are defined for IPCP

Code	*IPCP Packet*
0x01	Configure-request
0x02	Configure-ack
0x03	Configure-nak
0x04	Configure-reject
0x05	Terminate-request
0x06	Terminate-ack
0x07	Code-reject

- A party uses the configure-request packet to negotiate options with the other party to set the IP addresses.
- After configuration, the link is ready to carry IP data in the payload field of a PPP frame.
- The value of the protocol field is 0021_{16} to show that an IP data packet is being carried across the link and not an IPCP packet.
- After the IP has sent all its packets, the IPCP can take control and use the terminate-request packet and terminate-ack packets to end the network connection.

Figure 11.39 *IP datagram encapsulated in a PPP frame*

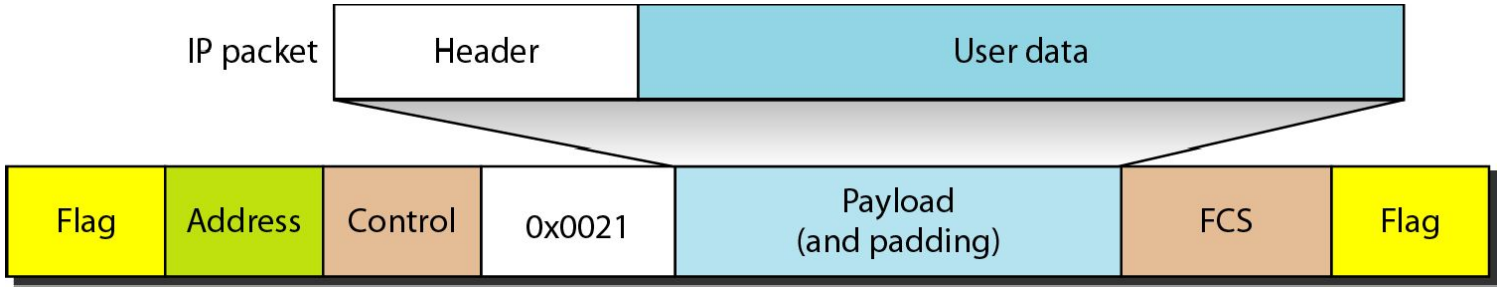


Figure 11.40 *Multilink PPP*