Chapter 3 Marks 12

Errors Detection, Correction and Wireless Communication

Introduction: --

We know that, it is virtually impossible to send any signal, analog or digital, over a distance without any distortion even in the most perfect condition. This is basically due to various impairments that can occur during the process of transmission as a result of an imperfect medium and/or environment. These errors can be classified into three main categories as given below:

- Delay Distortion
- Attenuation
- Noise

In the chapter we will study various errors that can take place during data transmission, and see how they can be trapped and corrected

3.1 Error Classification:--

3.1.1 Delay Distortion

Delay distortion is caused because the signal of varying frequencies travels at different speeds along the medium. We know that any complex signal can be decomposed into different sinusoidal signals of different frequencies resulting in a frequency bandwidth for every signal. One property of signal propagation is that the speed of travel of frequency is the highest at the center of this bandwidth, and lower at the both ends. Therefore at the receiving end, signals with different frequencies in a given bandwidth will arrive at different times. If the signal received is measured at a specific time, they will not measure up to the original signal resulting in its misinterpretation

3.1.2 Attenuation

Attenuation is another form of distortion.

In this case as a signal travels through any medium,

Its strength decreases, as shown in fig 3.1

Just the way our voice becomes weak over a

distance and loses its contents beyond a certain distance.

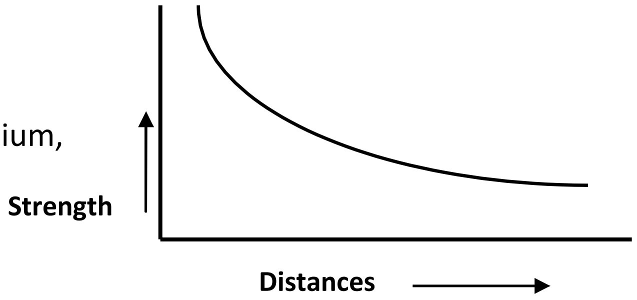


Fig 3.1 Attenuation

Attenuation is very small at short distance. The original signal therefore can be recognized as such without too much of distortion. Attenuation however increases with distance. This is because some of the signal energy is absorbed by the medium. Attenuation is also higher at higher frequencies.Techniques are available to equalize the attenuation for a band of frequencies over a medium. For telephone lines, this is achieved by using loading coils that change the electrical properties of the wire to smooth out the effects of attenuation. However it cannot be done away with. One can use amplifiers to boost the signal, but the amplifier boosts not just the signal, but also the accompanying noise.

3.1.3 Noise

Noise is yet another component that poses a problem in receiving the signal accurately. We know that a signal travels as an electromagnetic signal through any medium. Electromagnetic energy that gets inserted somewhere during transmission is called noise. Apart from distortion and attenuation, noise is one of the major limiting factors in the performance of any communication system.

In conclusion a signal can get changed over a distance leading to the misinterpretation of the signal (0 to 1 and vice versa). This can be very dangerous in many situations.

3.2 Types of Error: ----

If the signal is carrying binary data there can be two types of errors: Single-Bit errors and burst errors. This is shown in fig 3.2.In a single-bit error a bit value of 0 changes to 1 or vice versa. In a burst error, multiple bits of a binary value are changed

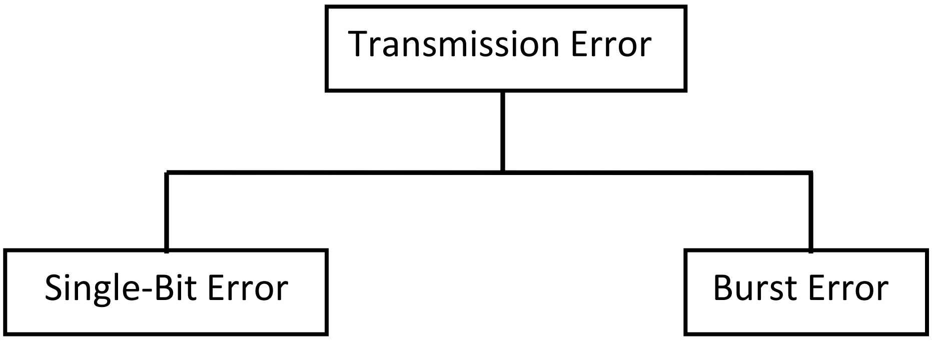


Fig 3.2 Types of transmission Error

As we mentioned in a single-bit errors, a single bit of the data unit changes. Thus effectively, either a 0 bit changes to 1, or a 1 bit changes to 0. Single-bit errors are more likely in the case of parallel transmission because it is possible that one of the eight wires carrying the bits has become noisy resultinginto corruption of a single bit for each byte. This can happen in the case of a parallel transmission between the CPU and memory inside a computer.

In case of serial transmission the duration of noise is usually longer than that of a single bit. Hence the chances of corrupting only a single bit are less. In contrast a burst changes at least two bits during data transmission because of errors. Note that burst errors can change any two or more bits in a transmission. These bits need not necessarily be adjacent bits. Burst errors are more likely in serial transmission, because the duration of noise is longer, which causes multiple bits to be corrupted.

3.3 Error Detection: ----

There are a number of techniques used for transmission error detection and correction. We shall examine the most common techniques used for this purpose.

3.3.1 Vertical Redundancy Check (VRC) or Parity Check:---

The Vertical Redundancy Check (VRC), also known as parity check, is quite simple. It is the least expensive technique as well as in this method, the sender appends a single additional bit, called the parity bit, to the message before transmission it. There are two schemes in this odd parity and even parity. In the odd parity scheme, given some bits, an additional parity bit is added in such a way that the number of 1s in the bit inclusive of the parity bit is odd. In the even parity scheme, the parity bit is added such that the number of 1s inclusive of the parity bit is even.

For example consider a message string 1100011 that needs to be transmitted. Let us assume the even parity scheme. The following will now happen.

1) The sender examines this message string and notes that the number of bits containing a value 1 in this message string is 4. Therefore it adds an extra 0 to the end of this message. This extra bit is called parity bit. This is done by the hardware itself, which is why it is very fast.
2) The sender sends the original bits 1100011 and the additional parity bit 0 together to the receiver.
3) The receiver separates the parity bit from the original bits and it also examines the original bits. Its sees the original bits as 1100011, and notes that the number of 1s in the message is four i.e. even.
4) The receiver now computes the parity bit again and compares this computed parity bit with the 0 parity bit received from the sender, it notes that are equal and accepts the bit string as correct. This is also done by the hardware itself. The process shown in the fig 3.3

In contrast, if the original message was 1010100 the number of 1s in the message would have been three (old), and therefore, the parity bit would contain a 1.
There is one problem with this scheme. This scheme can only catch a single bit error. If two bits reverse, this scheme will fail. For instance, if the first two bits in the bit stream shown in fig 3.3 change, we will get a stream 0000011, yielding a parity bit of 0 again, fooling us!

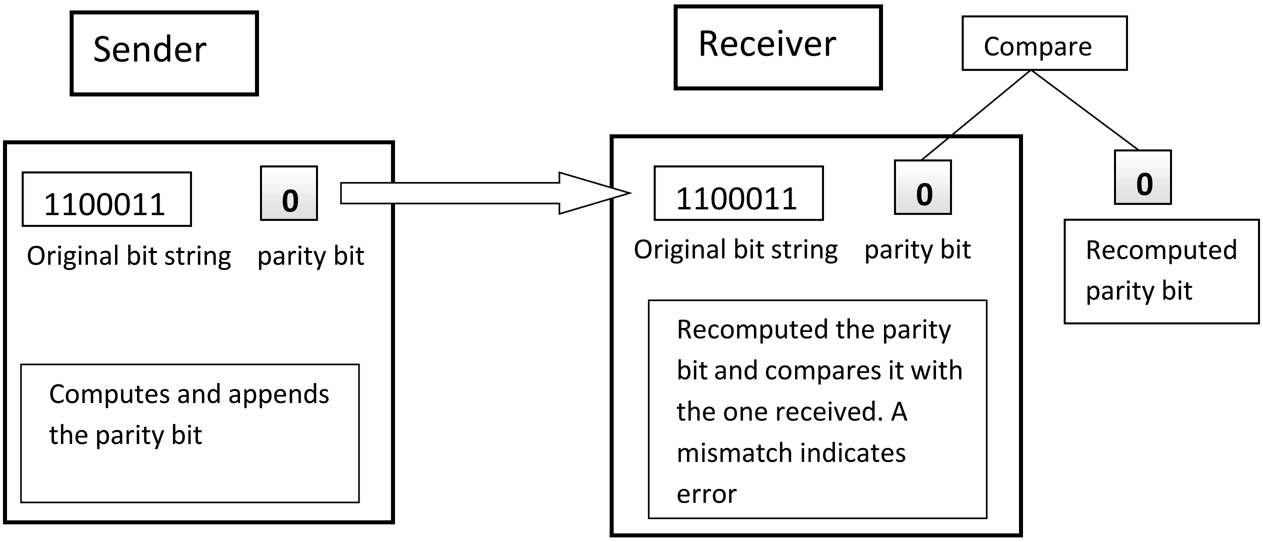

Sender
1100011
0
Original bit string
parity bit
Computes and appends the parity bit
Receiver
Compare
1100011
0
Original bit string
parity bit
0
Recomputed parity bit
Recomputed the parity bit and compares it with the one received. A mismatch indicates error

Fig 3.3 Even Parity VRC Check

Clearly, parity checking can detect single bit errors. However if multiple bits of a message are changed due to an error (burst), parity checking would not work.Better schemes are required to trap burst errors.

3.3.2 Longitudinal Redundancy Check (LRC):---

A block of bits is organized in the form of a list (as rows) in the Longitudinal Redundancy Check (LRC). Here, for instance, if we want to send 32 bits, we arrange them into a list of four rows. Then the parity bit for each column is calculated and a new row of eight bits is created. These become the parity bits for the whole block. An example of LRC is shown in fug 3.4.

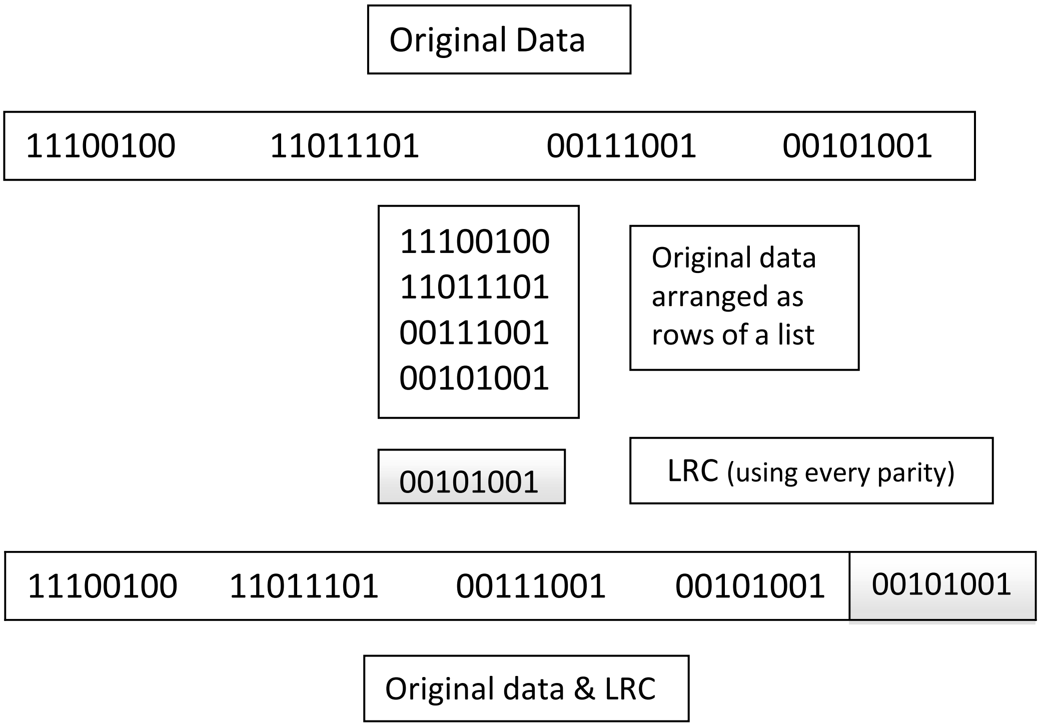
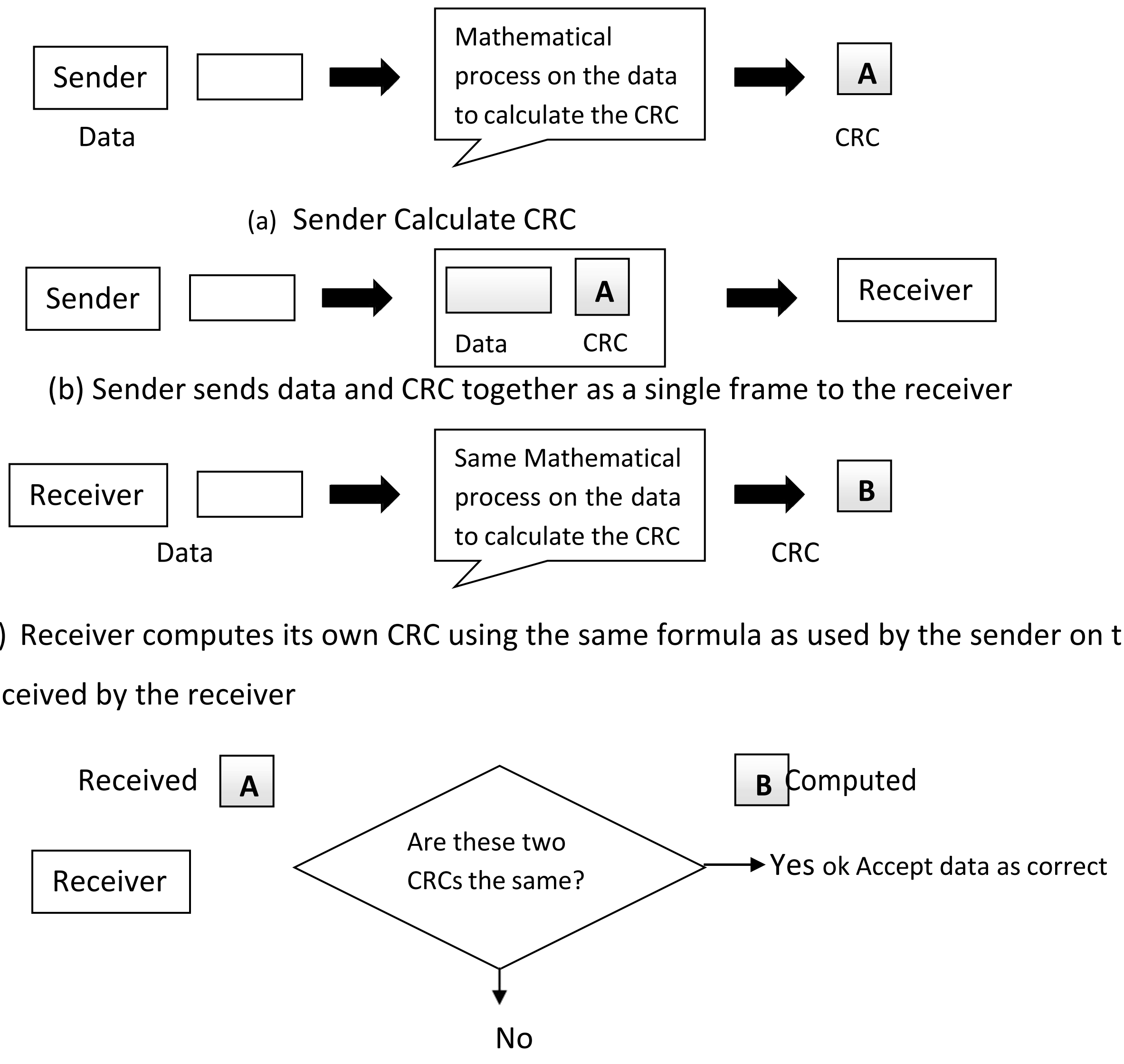


Fig 3.4 Longitudinal Redundancy Check (LRC)

3.3.2 Cyclic Redundancy Check (CRC):---

A mathematical algorithm is used on the data block to be sent to arrive at the Cyclic Redundancy Check (CRC) a small block of bits which are append to the data block and sent by the sender. At the destination, the receiver separate the data block, recomputes the CRC using the same algorithm and matches the received CRC with the computed CRC. A mismatch indicates an error. It is possible to change a few specific bits in specific position in the data block to get the same CRC ,thus fooling us ,but the probability of that is very rare thus making this method the sturdiest and the most widely used. At a broad level, the process is shown in fig 3.5


Sender
Data
Mathematical process on the data to calculate the CRC
A
CRC
Sender
A
Data
CRC
Receiver
Receiver
Data
Same Mathematical process on the data to calculate the CRC
B
CRC
Received
A
B
Computed
Receiver
Are these two CRCs the same?
Yes ok Accept data as correct
No

(a) Sender Calculate CRC

(b) Sender sends data and CRC together as a single frame to the receiver

(c) Receiver computes its own CRC using the same formula as used by the sender on the data received by the receiver

The transmission has some errors. Retransmission is required
(d) if the two CRCs match the receiver accepts the transmission, else rejects it

Fig 3.5 The Logical Process of CRC Computation

The main feature of CRC is as follows:

1) CRC is a very sturdy and better error detection method compared to others. Why? As in the case of VRC, if two adjacent bits change will the CRC be the same, fooling us? The answer is no. The algorithm to compute the CRC is so chosen that given the length of the data block in bits there are only a few and finite number of permutations and combinations for which the CRC is the same. The possibilities are also normally such that you have to inverse a number of specific and distant bits(e.g. 1, 43, and 91) to get the same CRC (which can fool us). As we know, normally, errors occurs in a burst, causing many consecutive bits and rarely will exactly the bits required to get the same CRC change.

2) CRC is normally implemented in hardware rather than in software. This makes this operation very fast, though a little more expansive. Depending on the method of CRC used,the corresponding type of modem has to be used. For computing the CRC, two simple hardware components are used an XOR gate and a shift register using a combination of these two, we can calculate the CRC for any data.

3) The data to be transmitted is divided into a number of blocks consisting of several bits each. After this, a block is treated as a mammoth string of 1s and 0s in a binary number. It is then divided by a prime number and the remainder is treated as CRC. This is the normal method.

3.3 IEEE Standards:

A set of network standards developed by the IEEE they include:

- IEEE 802.1: Standards related to network management.
- IEEE 802.2: General standard for the data link layer in the OSI Reference Model. The IEEE divides this layer into two sublayers -- the logical link control (LLC) layer and the media access control (MAC) layer. The MAC layer varies for different network types and is defined by standards IEEE 802.3 through IEEE 802.5.
- IEEE 802.3: Defines the MAC layer for bus networks that use CSMA/CD. This is the basis of the Ethernet standard.
- EEE 802.4: Defines the MAC layer for bus networks that use a token passing mechanism (token bus networks).
- IEEE 802.5: Defines the MAC layer for token-ring networks.
- IEEE 802.6: Standard for Metropolitan Area Networks (MANs).
- IEEE 802.11 Wireless Network Standards: 802.11 is the collection of standards setup for wireless networking.

3.4 Wireless LAN

Wireless communication is one of the fastest growing technologies. The demand for connecting devices without cable is increasing everywhere. Wireless LANs are found on college campuses, office buildings, and public areas. At home, a **wireless LAN** can connect roaming devices to the Internet.

IEEE 802.11

IEEE has defined the specification for a wireless LAN, called **IEEE 802.11,** which covers the physical and data link layers. But before discussing these layers, we describe the architecture of the protocol in general.

Architecture

The standard defines two kinds of services: the basic service set (BSS) and the extended service set (ESS).

Basic Service Set

IEEE 802.11 defines the **basic service set (BSS)** as the building block of a wireless LAN. A basic service set is made of stationary or mobile wireless stations and a possible central base station, known as the **access point (AP).** Figure 3.6 shows two sets in this standard.

The BSS without an AP is a stand-alone network and cannot send data to other BSSs. It is what is called an *ad hoc architecture.* In this architecture, stations can form a network without the need of an AP; they can locate each other and agree to be part of a BSS. A BSS with an AP is sometimes referred to as an infrastructure network.

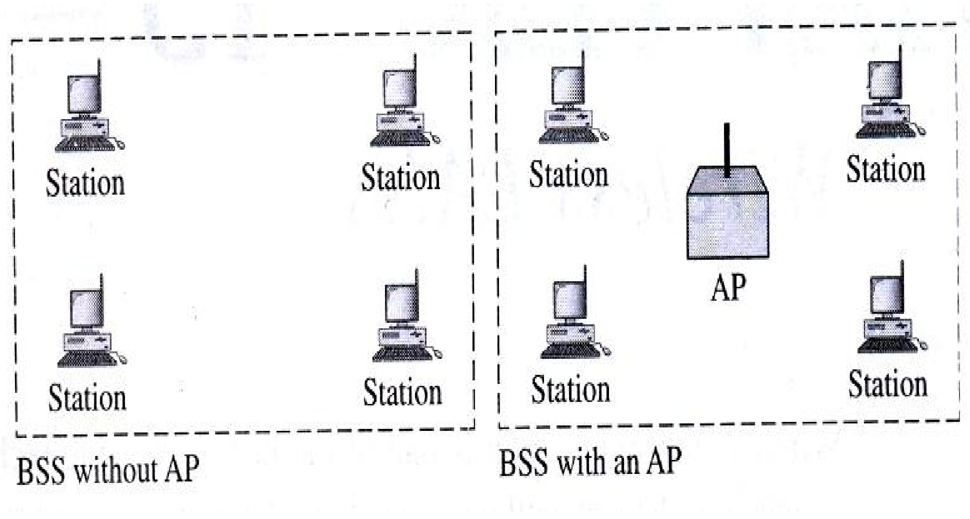


Figure 3.6 BSSs

Extended Service Set

An **extended service set (ESS)** is made up of two or more BSSs with *APs.* In this case, the BSSs are connected through a *distribution system,* which is usually a wired LAN. The distribution system connects the APs in the BSSs. IEEE 802.11 does not restrict the distribution system; it can be any IEEE LAN such as an Ethernet. Note that the extended service set uses two types of stations: mobile and stationary. The mobile stations are normal stations inside a BSS. The stationary stations are AP stations that are part of a wired LAN. Figure 3.7 shows an ESS.

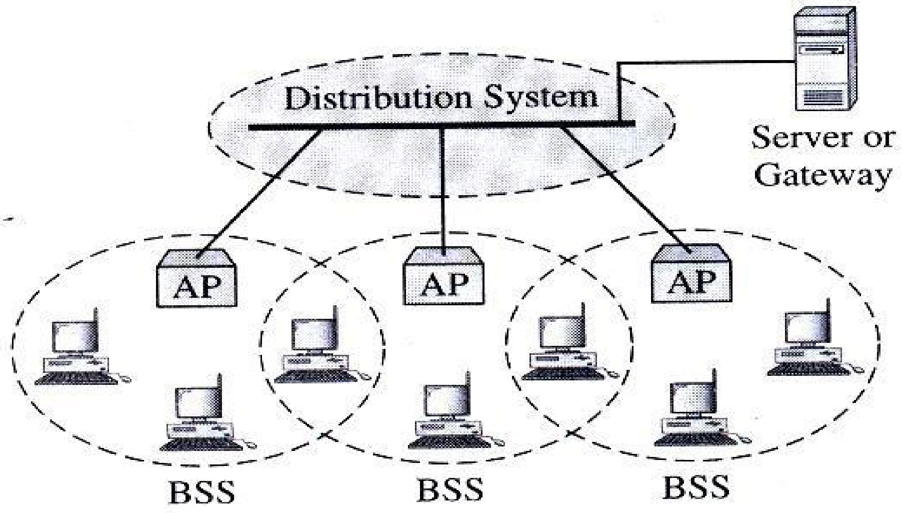


Figure 3.7 ESS

When BSSs are connected, we have what is called an *infrastructure network.* In this network, the stations within reach of one another can communicate without the use of an AP. However, communication between two stations in two different BSSs usually occurs via two APs. The idea is similar to communication in a cellular network if we consider each BSS to be a cell and each AP to be a base station. Note that a mobile station can belong to more than one BSS at the same time.

Station Types

IEEE 802.11 defines three types of stations based on their mobility in a wireless LAN: **no-transition, BSS-transition, and ESS-transition.**

No-Transition Mobility A station with no-transition mobility is either stationary (not moving) or moving only inside a BSS.

BSS-Transition Mobility A station with BSS-transition mobility can move from one BSS to another, but the movement is confined inside one ESS.

ESS-Transition Mobility A station with ESS-transition mobility can move from one ESS to another. However, IEEE *802.11* does not guarantee that communication is continuous during the move.

3.5 BLUETOOTH

Bluetooth is a wireless LAN technology designed to connect devices of different functions such as telephones, notebooks, computers (desktop and laptop), cameras, printers, *coffee makers,* and so on, A Bluetooth LAN is an ad hoc network, which means that the network is formed spontaneously; the devices, sometimes called gadgets, find *each* other and make a network called piconet. A Bluetooth LAN can even be connected to the Internet if one of the gadgets has this capability. A Bluetooth LAN, by nature, cannot be large. If there are many gadgets that try to connect, there is chaos.

Bluetooth technology has several applications. Peripheral devices of a computer can communicate with the computer through this technology (wireless mouse or keyboard).Monitoring devices can communicate with sensor devices in a small health care center. Home Security devices can use this technology to connect different sensors to the main security controller. Conference attendees can synchronize their palmtop computers at a conference.

Bluetooth was originally started as a project by the Ericsson Company. It is named for Harald Blaatand, the king of Denmark (940-981) who united Denmark and Norway. *Blaatand* translates to *Bluetooth* in English.

Today, Bluetooth technology is the implementation of a protocol defined by the IEEE 802.15 standard. The standard defines a wireless personal- area network (PAN) operable in an area the size of a room or a hall.

Architecture

Bluetooth defines two types of networks: piconets and scatternet.

Piconets

A Bluetooth network is called a **piconet,** or a small net. A piconet can have up to eight stations, one of which is called the **master;** the rest are called **slaves.** All the slave stations synchronize their clocks and hopping sequence with the master slave. Note that a piconet can have only one master station. The communication between the master and the slaves can be one-to-one or one-to-many. Figure 3.8 shows a piconet.

As indicated earlier, a Bluetooth device can be either a master or a slave and any of the devices within a piconet can be the master. However, the device that establishes the piconet automatically becomes the master and all the other devices then become slaves. There can be up to seven **active slave(AS)** devices active at a time within a single piconet. In addition, as shown in Figure 3.8 (a), a device may be a **standby slave (SS)** or a **parked slave (PS).** Devices in the standby mode cannot participate in the piconet. A parked slave, however, cannot actively participate in the piconet but is known by the master and can be reactivated by it. Typically, these are devices that the master has switched to a low power state to save the power in the batteries of the device.

All parked slaves have an 8-bit **parked member address (PMA)** and there can be up to 255 such devices. If there are already seven active slaves, then a parked slave must wait until one of the active slaves switches to the parked mode before it can become active. All active slaves have a 3-bit **active member address (AMA)** and this is used by the master both to send blocks, of data - referred to as packets in the standard - to a specific slave and to identify the slave that has sent a response packet. All communications are via the master and slave-to-slave communications are prohibited.

To utilize the 80 MHz of bandwidth in an efficient way, multiple piconets can be present in a room/office at the same time. In addition, multiple piconets can be interconnected together to provide a wider area of coverage. The resulting configuration is then known as a scatternet and a simple example is shown in Figure 3.8(b).

As we can see, in this example one of the active slave devices is located in the region of the two overlapping piconets. However, since each piconet has a different pseudo-random hopping sequence, both piconets can operate concurrently. Hence the slave can become a member of either piconet but not both at the same time. To change its membership, the slave first informs its current master that it will be unavailable for a specified time. It then proceeds to synchronize, with the master of the other piconet in the described way

In practice, creating more piconets in the same area leads to a degradation in the performance of each piconet. This is because the probability of the same carrier frequency is being used at the same time increases so resulting in the respective bit in the bit stream being corrupted.

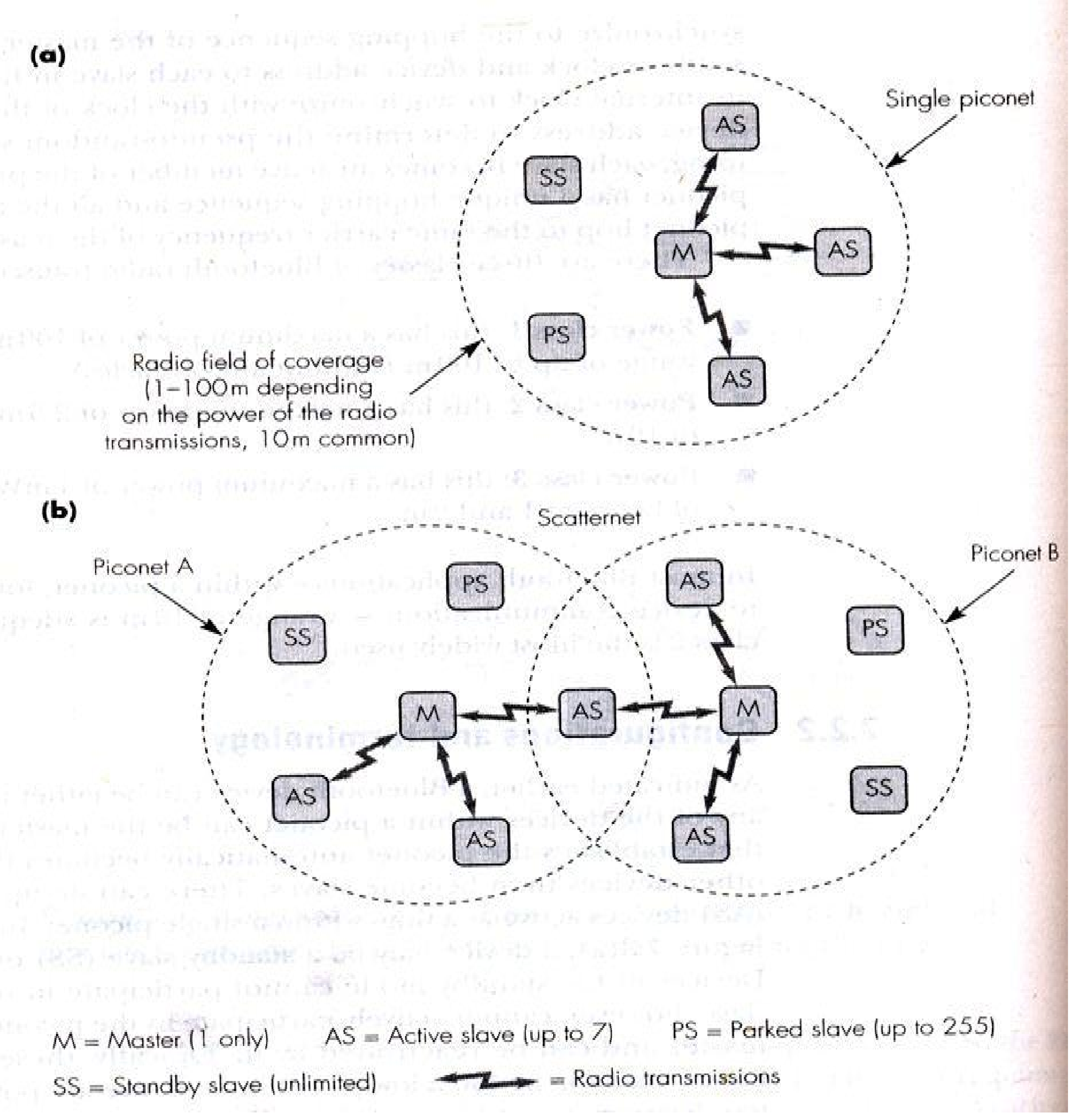

(a)
Single piconet
AS
SS
M
AS
PS
AS
Radio field of coverage (1–100m depending on the power of the radio transmissions, 10m common)
(b)
Scatternet
Piconet A
Piconet B
PS
AS
SS
PS
M
AS
M
AS
AS
SS
AS
M = Master (1 only)
AS = Active slave (up to 7)
PS = Parked slave (up to 255)
SS = Standby slave (unlimited)
= Radio transmissions

Figure 3.8 Bluetooth configurations and terminology:

(a) Single piconet; (b) scatternet.