CS-204: COMPUTER NETWORKS

Lecture 2

Chapter: Multiple Access

Instructor: Dr. Vandana Kushwaha

1. INTRODUCTION

Data link layer is divided into two sub layers:

- **Logical link control (LLC) layer**: The upper sub layer is responsible for data link control i.e. for flow and error control.
- **Media access control (MAC) layer**: The lower sub layer is responsible for resolving access to the shared media.

If the channel is dedicated, we do not need the lower sub layer. Figure 12.1 shows these two sub layers in the data link layer.

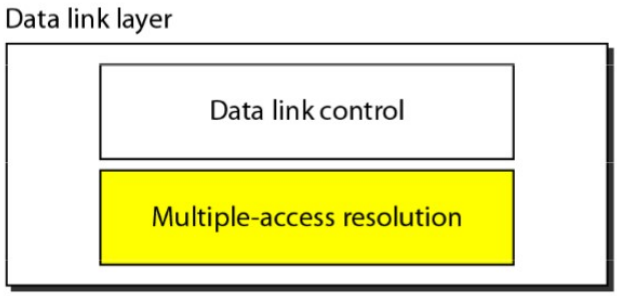

Figure 12.1 Data link layer divided into two functionality-oriented sub layers

When nodes or stations are connected and use a common link, called a multipoint or broadcast link, we need a multiple-access protocol to coordinate access to the link. The problem of controlling the access to the medium is similar to the rules of speaking in an assembly. The procedures guarantee that the right to speak is upheld and ensure that two people do not speak at the same time, do not interrupt each other, do not monopolize the discussion, and so on. The situation is similar for multipoint networks. Many formal protocols have been devised to handle access to a shared link. We categorize them into three groups. Protocols belonging to each group are shown in Figure 12.2.

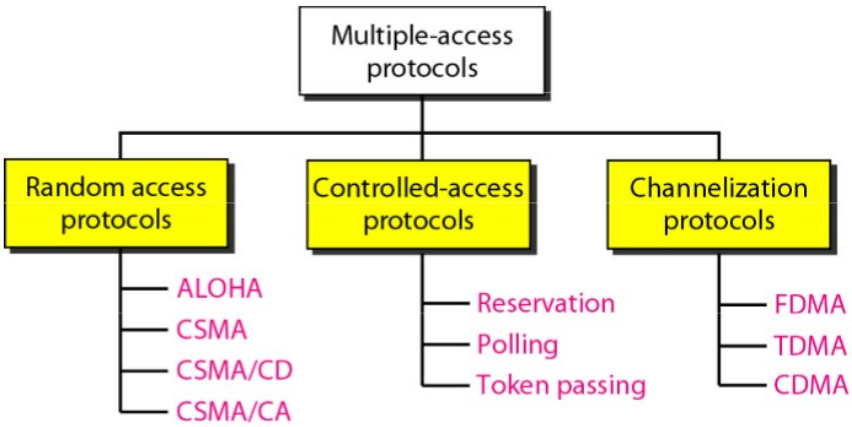

Figure 12.2 Taxonomy of multiple-access protocols discussed in this chapter

2. RANDOM ACCESS PROTOCOLS

In random access or contention methods, no station is superior to another station and none is assigned the control over another. No station permits, or does not permit, another station to send. At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send. This decision depends on the state

of the medium (idle or busy). In other words, each station can transmit when it desires on the condition that it follows the predefined procedure, including the testing of the state of the medium. Two features give this method its name.

- First, there is no scheduled time for a station to transmit. Transmission is random among the stations. That is why these methods are called random access.
- Second, no rules specify which station should send next. Stations compete with one another to access the medium. That is why these methods are also called contention methods.

In a random access method, each station has the right to the medium without being controlled by any other station. However, if more than one station tries to send, there is an access conflict-collision-and the frames will be either destroyed or modified. To avoid access conflict or to resolve it when it happens, each station follows a procedure that answers the following questions:

- *When* can the station access the medium?
- *What* can the station do if the medium is busy?
- *How* can the station determine the success or failure of the transmission?
- What can the station do if there is an access conflict?

Following section explains each protocol under random access category:

2.1. ALOHA

ALOHA, the earliest random access method was developed at the University of Hawaii in early 1970. It was designed for a radio (wireless) LAN, but it can be used on any shared medium. It is obvious that there are potential collisions in this arrangement. The medium is shared between the stations. When a station sends data, another station may attempt to do so at the same time. The data from the two stations collide and become garbled. There are two variations of ALOHA protocol:

2.1.1. Pure ALOHA

The original ALOHA protocol is called *pure ALOHA*. This is a simple, but elegant protocol. The idea is that each station sends a frame whenever it has a frame to send. However, since there is only one channel to share, there is the possibility of collision between frames from different stations. Figure 12.3 shows an example of frame collisions in pure ALOHA.

There are four stations (unrealistic assumption) that contend with one another for access to the shared channel. The figure shows that each station sends two frames; there are a total of eight frames on the shared medium. Some of these frames collide because multiple frames are in contention for the shared channel. Figure 12.3 shows that only two frames survive: frame 1.1 from station 1 and frame 3.2 from station 3. We need to mention that even if one bit of a frame coexists on the channel with one bit from another frame, there is a collision and both will be destroyed. It is obvious that we need to resend the frames that have been destroyed during transmission.

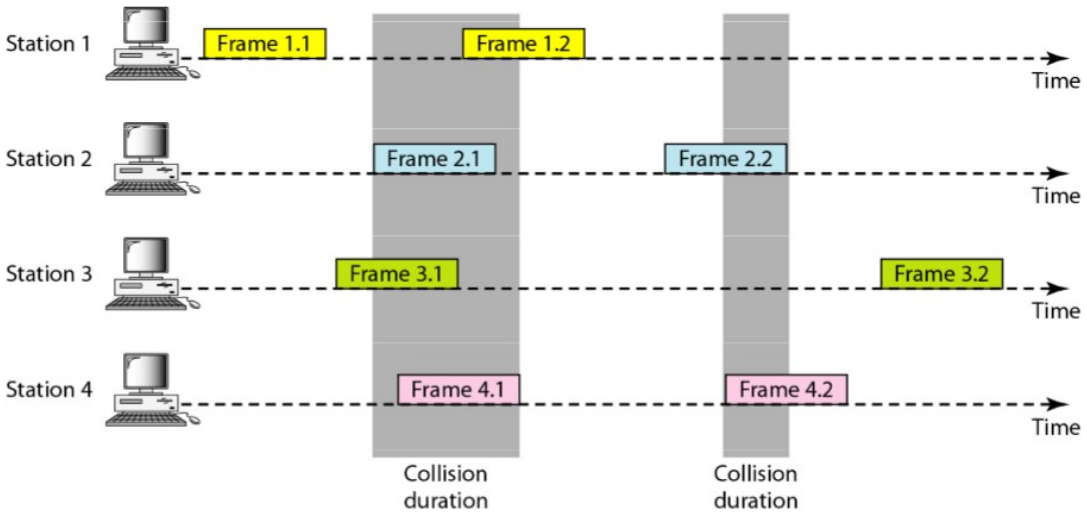

Figure 12.3 Frames in a pure ALOHA network

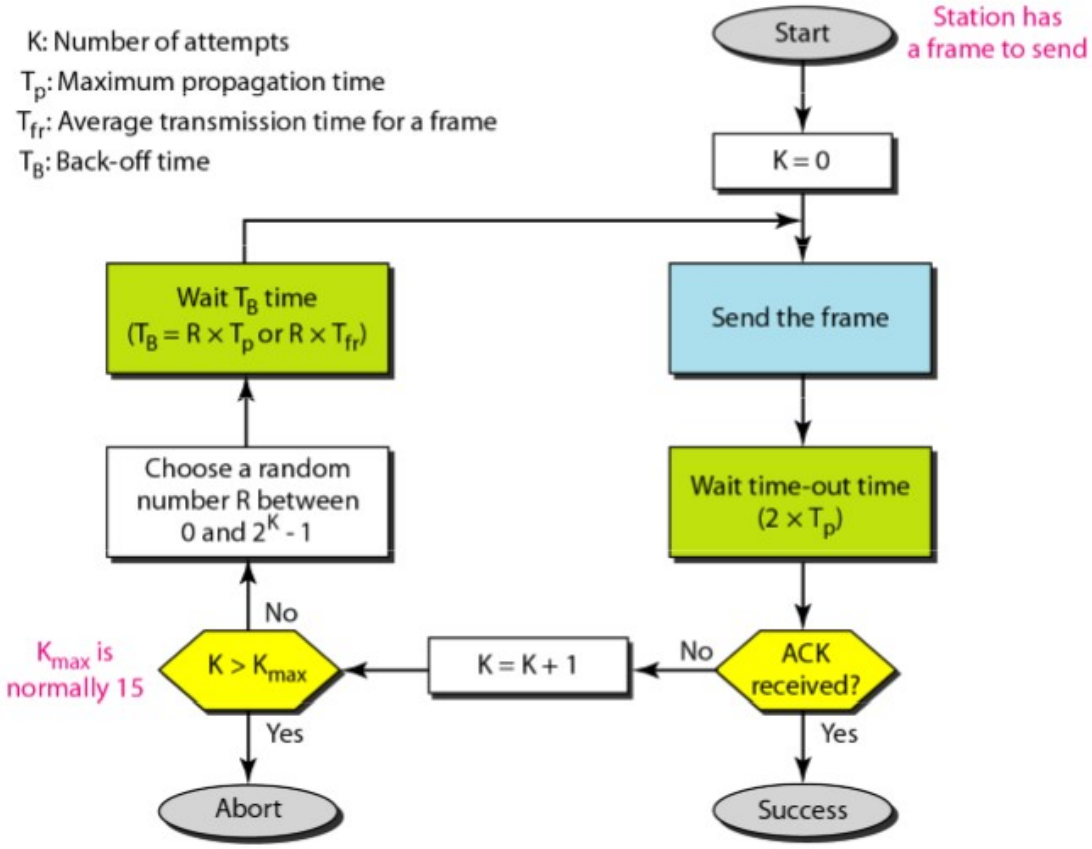

Figure 12.4 Procedure for pure ALOHA protocol

The pure ALOHA protocol relies on acknowledgments from the receiver. When a station sends a frame, it expects the receiver to send an acknowledgment. If the acknowledgment does not arrive after a time-out period, the station assumes that the frame (or the acknowledgment) has been destroyed and resends the frame. A collision involves two or more stations. If all these stations try to resend their frames after the time-out, the frames will collide again. Pure ALOHA dictates that when the time-out period passes, each station waits a random amount of time before resending its frame. The randomness will help avoid

more collisions. We call this time the back-off time T_B. Pure ALOHA has a second method to prevent congesting the channel with retransmitted frames. After a maximum number of retransmission attempts K_{max} a station must give up and try later. Figure 12.4 shows the procedure for pure ALOHA based on the above strategy.

The time-out period is equal to the maximum possible round-trip propagation delay, which is twice the amount of time required to send a frame between the two most widely separated stations (2 x Tp) The back-off time T_B is a random value that normally depends on K (the number of attempted unsuccessful transmissions). One common formula for calculating T_B is the *binary exponential back-off*. In this method, for each retransmission, a multiplier in the range 0 to 2^K - 1 is randomly chosen and multiplied by Tp (maximum propagation time) to find T_B. Note that in this procedure, the range of the random numbers increases after each collision. The value of K_{max} is usually chosen as 15.

Example 12.1

The stations on a wireless ALOHA networks are a maximum of 600 km apart. If we assume that signals propagate at 3 x 10^8 ms, we find Tp = (600 x 10^5) / (3 x 10^8) = 2 ms. Now we can find the value of T_B for different values of K.

a. For K = 1, the range is {0,1}. The station needs to generate a random number with a value of 0 or 1. This means that T_B is either 0 ms (0 x 2) or 2 ms (1 x 2), based on the outcome of the random variable.

b. For K =2, the range is {0, 1, 2, 3}. This means that T_B can be 0, 2, 4, or 6 ms, based on the outcome of the random variable.

c. For K =3, the range is {0,1,2,3,4,5,6,7}. This means that T_B can be 0,2,4, ... , 14 ms, based on the outcome of the random variable.

Vulnerable time

Vulnerable time is the Length of time, in which there is a possibility of collision. We assume that the stations send fixed-length frames with each frame taking T_{fr} s to send.

$$\textit{Pure ALOHA vulnerable time} = 2 \times T_{fr}$$

2.1.2. Slotted ALOHA

Slotted ALOHA was invented to improve the efficiency of pure ALOHA. In slotted ALOHA we divide the time into slots of T_{fr} s(time to send each fixed size frame) and force the station to send only at the beginning of the time slot. Figure 12.5 shows an example of frame collisions in slotted ALOHA.

Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame. Of course, there is still the possibility of collision if two stations try to send at the beginning of the same time slot.

Vulnerable time

Vulnerable time for slotted ALOHA is one-half that of pure ALOHA i.e.

$$\textit{Slotted ALOHA vulnerable time} = T_{fr}$$

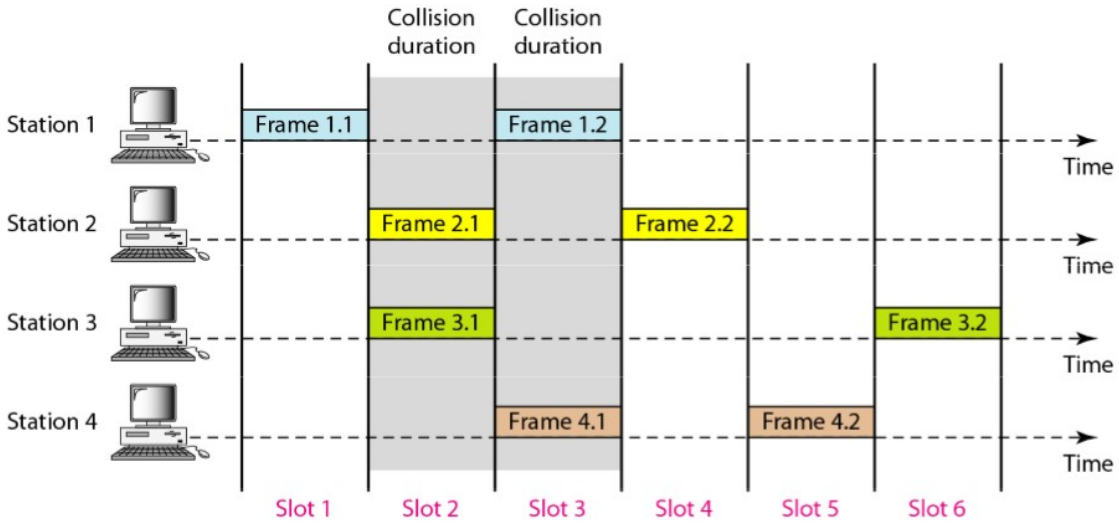

Figure 12.5 Frames in a slotted ALOHA network

2.2. Carrier Sense Multiple Access (CSMA)

To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed. The chance of collision can be reduced if a station senses the medium before trying to use it. Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending. In other words, CSMA is based on the principle "sense before transmit" or "listen before talk." CSMA can reduce the possibility of collision, but it cannot eliminate it.

Vulnerable Time

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

Following sub sections explains different variations of CSMA protocols:

2.2.1. Persistence Methods

What should a station do if the channel is busy? What should a station do if the channel is idle? Three methods have been devised to answer these questions:

- 1-persistent method,
- Non-persistent method, and
- P-persistent method.

Figure 12.6 shows the behavior of three persistence methods when a station finds a channel busy.

2.2.1.1. 1-Persistent Method

The 1-persistent method is simple and straightforward. In this method, after the station finds the line idle, it sends its frame immediately (with probability 1). This method has the

highest chance of collision because two or more stations may find the line idle and send their frames immediately. Ethernet, a LAN standard uses this method.

2.2.1.2. Non-persistent Method

In the non-persistent method, a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again. The non-persistent approach reduces the chance of collision because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously. However, this method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.

2.2.1.3. p-Persistent Method

The p-persistent method is used if the channel has time slots with slot duration equal to or greater than the maximum propagation time. The p-persistent approach combines the advantages of the other two strategies. It reduces the chance of collision and improves efficiency. In this method, after the station finds the line idle it follows these steps:

(i). with probability p, the station sends its frame.
(ii). with probability q = 1 - p, the station waits for the beginning of the next time slot and checks the line again.

a. If the line is idle, it goes to step **(i)**.
b. If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.

Figure 12.6 shows the flow diagrams for these methods:

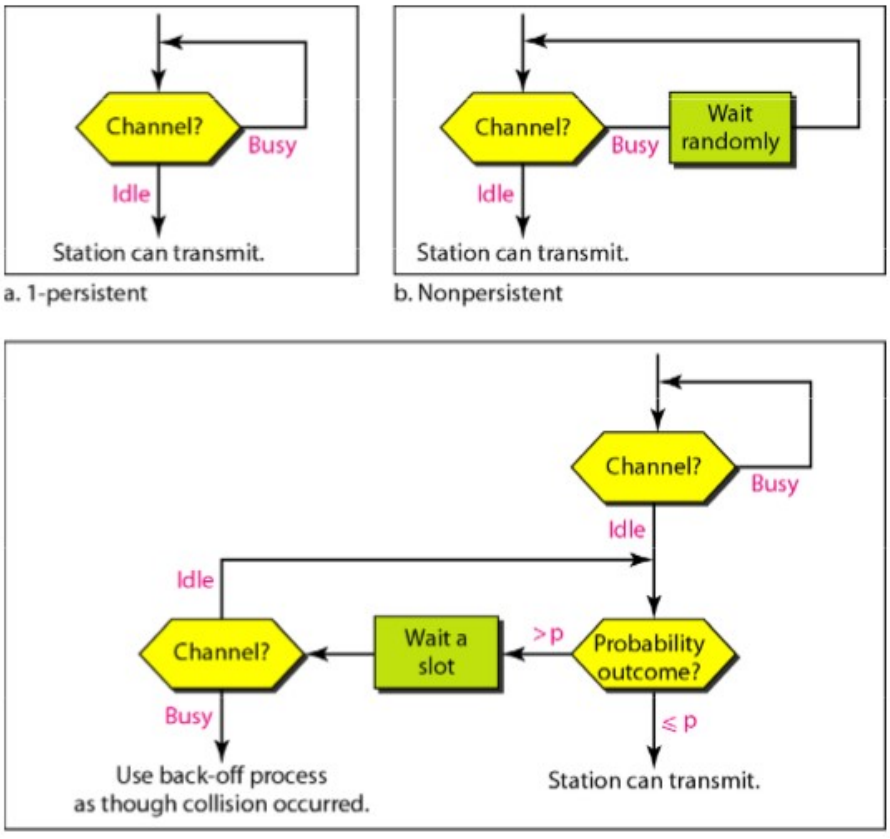

Figure 12.6 Flow diagram for three persistence methods

2.2.2. Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

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

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

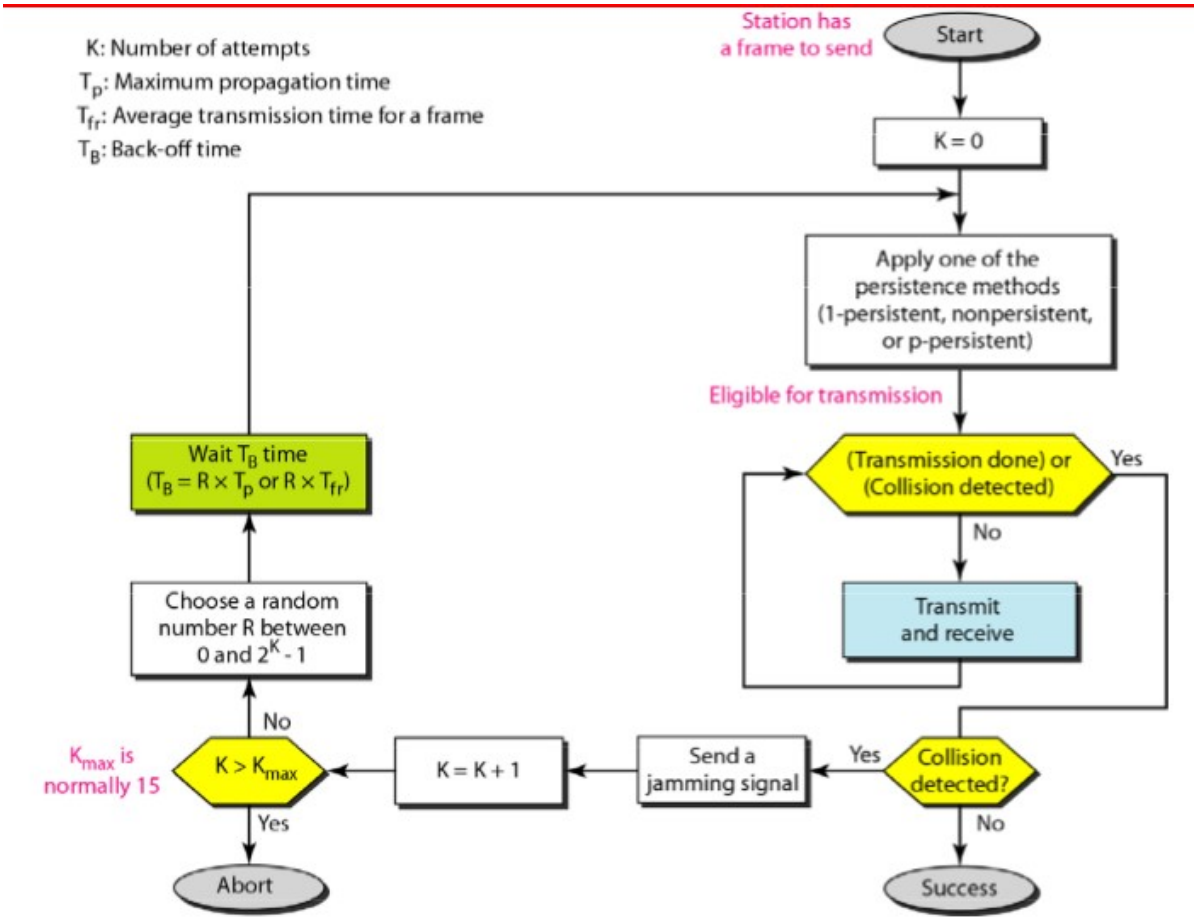

Figure 12.7 Flow diagram for the CSMA/CD

Flow diagram for CSMA/CD is depicted in Figure 12.7. It is similar to the one for the ALOHA protocol, but there are differences:

i. The first difference is the addition of the persistence process. We need to sense the channel before we start sending the frame by using one of the persistence processes we discussed previously (non-persistent, 1-persistent, or p-persistent). The corresponding box can be replaced by one of the persistence processes shown in Figure 12.6.

ii. The second difference is the frame transmission. In ALOHA, we first transmit the entire frame and then wait for an acknowledgment. In CSMA/CD, transmission and collision detection is a continuous process. We do not send the entire frame and then look for a collision. The station transmits and receives continuously and simultaneously (using two different ports). We constantly monitor in order to detect

one of two conditions: either transmission is finished or a collision is detected. Either event stops transmission. When we come out of the loop, if a collision has not been detected, it means that transmission is complete; the entire frame is transmitted. Otherwise, a collision has occurred.

iii. The third difference is the sending of a short jamming signal that enforces the collision in case other stations have not yet sensed the collision.

Energy Level

Energy level of a channel is used to monitor the current status of a channel. The level of energy in a channel can have three values: zero, normal, and abnormal.

- At the zero level, the channel is idle.
- At the normal level, a station has successfully captured the channel and is sending its frame.
- At the abnormal level, there is a collision and the level of the energy is twice the normal level.

A station that has a frame to send or is sending a frame needs to monitor the energy level to determine if the channel is idle, busy, or in collision mode. Figure 12.8 shows the situation.

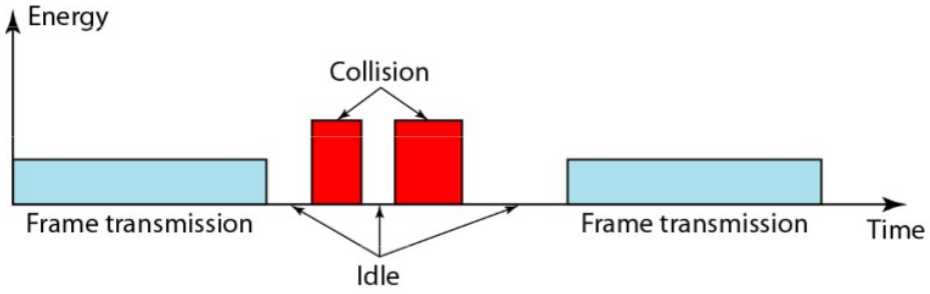

Figure 12.8 Energy level during transmission, idleness, or collision

Throughput

Let us call G the average number of frames generated by the system during one frame transmission time.

- The throughput for **pure ALOHA** is S =G x e^{-2G}. The maximum throughput **Smax =0.184** when G=(1/2).
- The throughput for **slotted ALOHA** is S =: G x e^{-G}. The maximum throughput **Smax = 0.368** when G=1.
- The throughput of **CSMA/CD** is greater than that of pure or slotted ALOHA. The maximum throughput occurs at a different value of G and is based on the persistence method and the value of p in the p-persistent approach.
    - For **1-persistent** method the maximum throughput is around **50** percent when G =1.
    - For **non-persistent** method, the maximum throughput can go up to **90** percent when G is between 3 and 8.

3. CONTROLLED ACCESS METHODS

In controlled access, the stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations. There are three popular controlled-access methods:

i. Reservation
ii. Polling
iii. Token passing

3.1. Reservation

In the reservation method, a station needs to make a reservation before sending data. Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval.

If there are N stations in the system, there are exactly N reservation mini-slots in the reservation frame. Each mini-slot belongs to a station. When a station needs to send a data frame, it makes a reservation in its own mini-slot. The stations that have made reservations can send their data frames after the reservation frame. Figure 12.9 shows a situation with five stations and a five mini-slot reservation frame. In the first interval, only stations 1, 3, and 4 have made reservations. In the second interval, only station 1 has made a reservation.

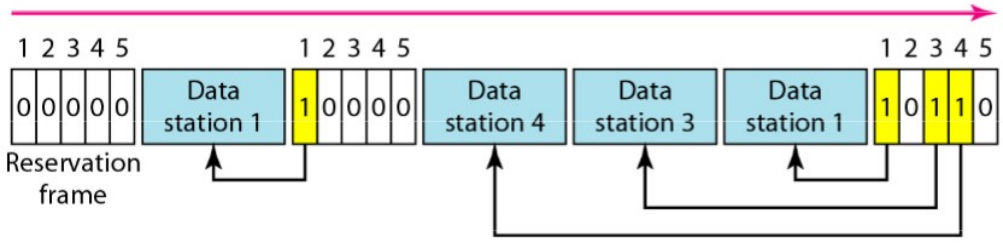

Figure 12.9 Reservation access method

3.2. Polling

Polling works with topologies in which one device is designated as a primary station and the other devices are secondary stations. All data exchanges must be made through the primary device even when the ultimate destination is a secondary device. The primary device controls the link; the secondary devices follow its instructions. It is up to the primary device to determine which device is allowed to use the channel at a given time. The primary device, therefore, is always the initiator of a session (see Figure 12.10).

If the primary wants to receive data, it asks the secondaries if they have anything to send; this is called **poll function**. If the primary wants to send data, it tells the secondary to get ready to receive; this is called **select function**.

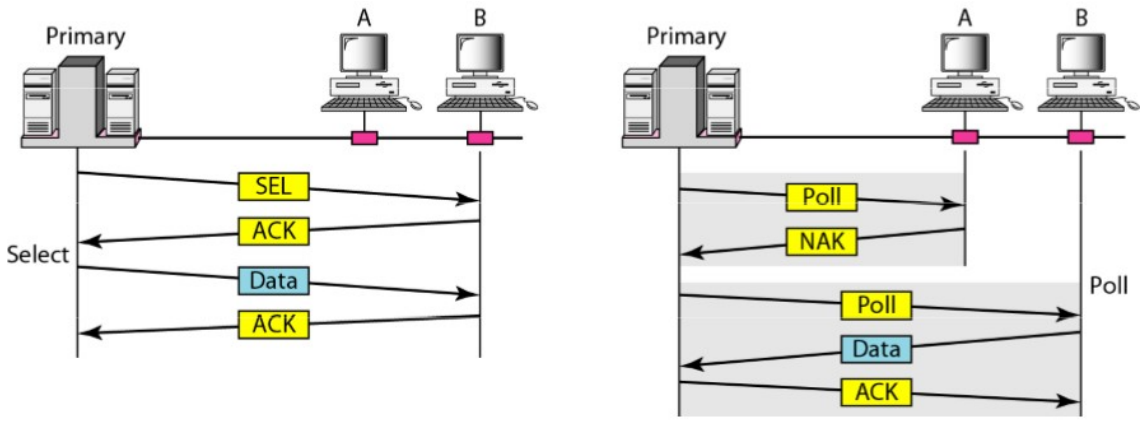

Figure 12.10 Select and poll functions in polling access method

Select

The select function is used whenever the primary device has something to send. Remember that the primary controls the link. If the primary is neither sending nor receiving data, it knows the link is available. If it has something to send, the primary device sends it. What it does not know, however, is whether the target device is prepared to receive. So the primary must alert the secondary to the upcoming transmission and wait for an acknowledgment of the secondary's ready status. Before sending data, the primary creates and transmits a select (SEL) frame, one field of which includes the address of the intended secondary.

Poll

The poll function is used by the primary device to solicit transmissions from the secondary devices. When the primary is ready to receive data, it must ask (poll) each device in turn if it has anything to send. When the first secondary is approached, it responds either with a NAK frame if it has nothing to send or with data (in the form of a data frame) if it does. If the response is negative (a NAK frame), then the primary polls the next secondary in the same manner until it finds one with data to send. When the response is positive (a data frame), the primary reads the frame and returns an acknowledgment (ACK frame), verifying its receipt.

3.3. Token Passing

In the token-passing method, the stations in a network are organized in a *logical ring*. In other words, for each station, there is a predecessor and a successor.

- The predecessor is the station which is logically before the station in the ring;
- The successor is the station which is after the station in the ring.
- The current station is the one that is accessing the channel now.

The right to this access has been passed from the predecessor to the current station. The right will be passed to the successor when the current station has no more data to send. But how is the right to access the channel passed from one station to another?

In this method, a special packet called a **token** circulates through the ring. The **possession** of the token gives the station the right to access the channel and send its data. When a station

has some data to send, it waits until it receives the token from its predecessor. It then holds the token and sends its data.

When the station has no more data to send, it releases the token, passing it to the next logical station in the ring. The station cannot send data until it receives the token again in the next round. In this process, when a station receives the token and has no data to send, it just passes the data to the next station.

Token management is needed for this access method. Stations must be limited in the time they can have possession of the token. The token must be monitored to ensure it has not been lost or destroyed. For example, if a station that is holding the token fails, the token will disappear from the network. Another function of token management is to assign priorities to the stations and to the types of data being transmitted. And finally, token management is needed to make low-priority stations release the token to high priority stations.

Logical Ring

In a token-passing network, stations do not have to be physically connected in a ring; the ring can be a logical one. Figure 12.11 show four different physical topologies that can create a logical ring.

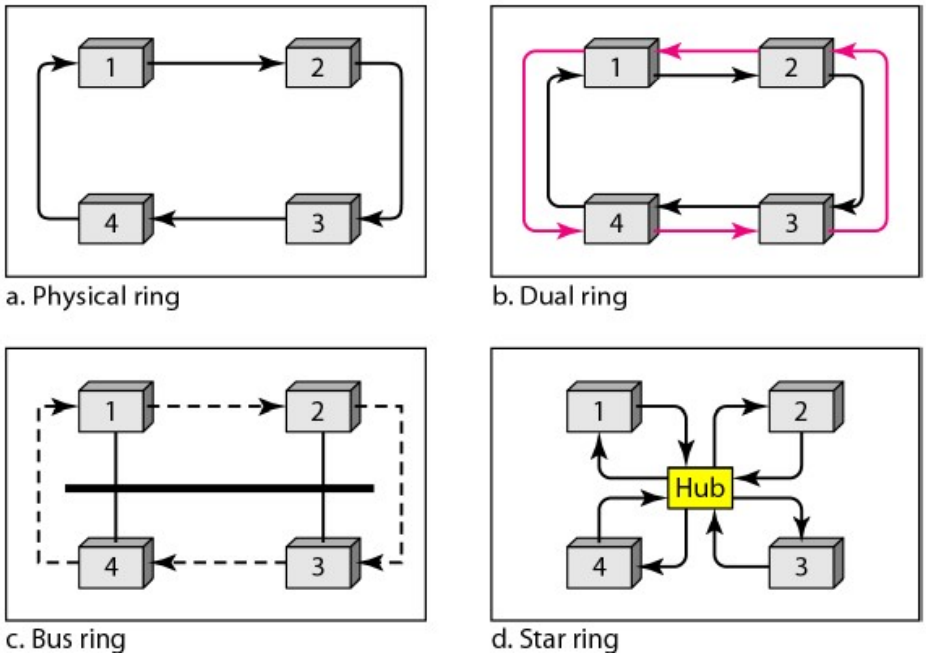

Figure 12.11 Logical ring and physical topology in token-passing access method

In the **Physical Ring Topology**, when a station sends the token to its successor, the token cannot be seen by other stations; the successor is the next one in line. This means that the token does not have to have the address of the next successor. The problem with this topology is that if one of the links-the medium between two adjacent stations fails, the whole system fails. The dual ring topology uses a second (auxiliary) ring which operates in the reverse direction compared with the main ring. The second ring is for emergencies only (such as a spare tire for a car). If one of the links in the main ring fails, the system automatically combines the two rings to form a temporary ring. After the failed link is restored, the auxiliary ring becomes idle again. Note that for this topology to work, each

station needs to have two transmitter ports and two receiver ports. The high-speed Token Ring networks called FDDI (Fiber Distributed Data Interface) and CDDI (Copper Distributed Data Interface) use this topology.

In the **Bus Ring Topology**, also called a token bus, the stations are connected to a single cable called a bus. They, however, make a logical ring, because each station knows the address of its successor (and also predecessor for token management purposes). When a station has finished sending its data, it releases the token and inserts the address of its successor in the token. Only the station with the address matching the destination address of the token gets the token to access the shared media. The Token Bus LAN, standardized by IEEE, uses this topology.

In a **Star Ring Topology**, the physical topology is a star. There is a hub, however, that acts as the connector. The wiring inside the hub makes the ring; the stations are connected to this ring through the two wire connections. This topology makes the network less prone to failure because if a link goes down, it will be bypassed by the hub and the rest of the stations can operate. Also adding and removing stations from the ring is easier. This topology is still used in the Token Ring LAN designed by IBM.

4. CHANNELIZATION

Channelization is a multiple-access method in which the available bandwidth of a link is shared *in time, frequency, or through code*, between different stations. In this section, we discuss three channelization protocols: FDMA, TDMA, and CDMA.

4.1. Frequency-Division Multiple Access (FDMA)

In frequency-division multiple access (FDMA), the available bandwidth is divided into frequency bands. Each station is allocated a frequency band to send its data. In other words, each band is reserved for a specific station, and it belongs to the station all the time. To prevent station interferences, the allocated bands are separated from one another by small *guard bands.* Figure 12.12 shows the idea of FDMA.

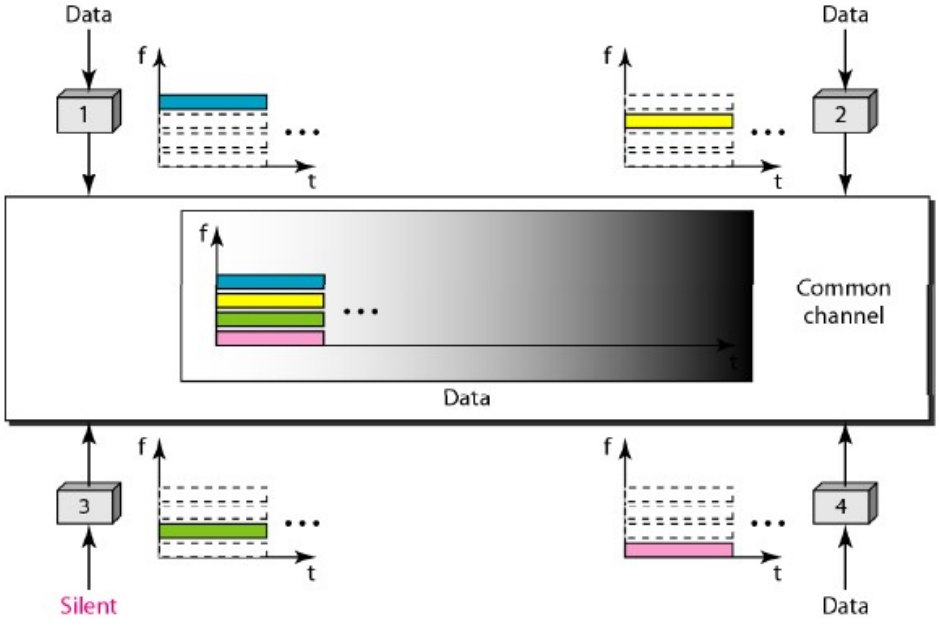

Figure 12.12 Frequency-division multiple access (FDMA)

4.2. Time-Division Multiple Access (TDMA)

In time-division multiple access (TDMA), the stations share the bandwidth of the channel in time. Each station is allocated a time slot during which it can send data. Each station transmits its data in is assigned time slot. Figure 12.13 shows the idea behind TDMA.

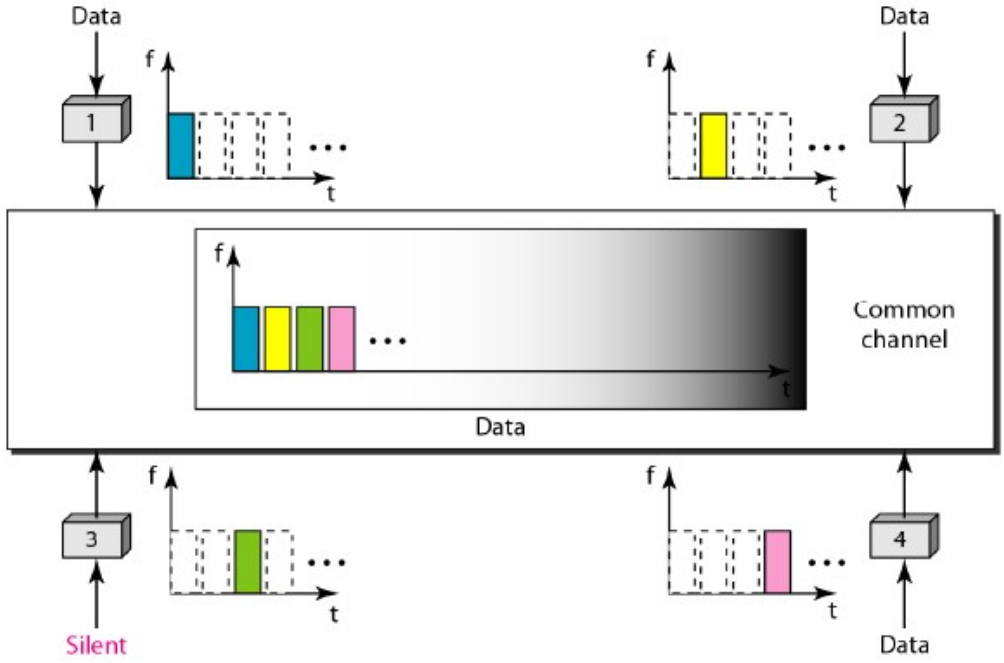

Figure 12.13 Time-division multiple access (TDMA)

The main problem with TDMA lies in achieving synchronization between the different stations. Each station needs to know the beginning of its slot and the location of its slot. This may be difficult because of propagation delays introduced in the system if the stations are spread over a large area. To compensate for the delays, we can insert *guard times.* Synchronization is normally accomplished by having some synchronization bits (normally referred to as preamble bits) at the beginning of each slot.

4.3. Code-Division Multiple Access (CDMA)

Code-division multiple access (CDMA) was conceived several decades ago. Recent advances in electronic technology have finally made its implementation possible. CDMA differs from FDMA because only one channel occupies the entire bandwidth of the link. It differs from TDMA because all stations can send data simultaneously; there is no timesharing.

IDEA

Let us assume we have four stations 1, 2, 3, and 4 connected to the same channel. The data from station 1 are *d1 ,* from station 2 are *d2,* and so on. The code assigned to the first station is *c1*, to the second is *c2,* and so on. We assume that the assigned codes have **two properties**:

i. cixcj=0 for all i≠j .i.e. If we multiply each code by another, we get 0.

ii. cixci=n(number of stations) i.e. If we multiply each code by itself, we get 4 (the number of stations).

With these two properties in mind, let us see how the above four stations can send data using the same common channel, as shown in Figure 12.14.

- Station 1 multiplies (a special kind of multiplication, as we will see) its data by its code to get *d1 . c1*. Station 2 multiplies its data by its code to get *d2 . c2 a*nd so on. The data that go on the channel are the sum of all these terms, as shown in the box.
- Any station that wants to receive data from one of the other three multiplies the data on the channel by the code of the sender. For example, suppose stations 1 and 2 are talking to each other. Station 2 wants to hear what station 1 is saying. It multiplies the data on the channel by c1, the code of station 1. Because (*c1 . c1*) is 4, but *(*c2 . c1), *(*c3 . c1), and *(c4 . c1*) are all 0s, station 2 divides the result by 4 to get the data from station 1.

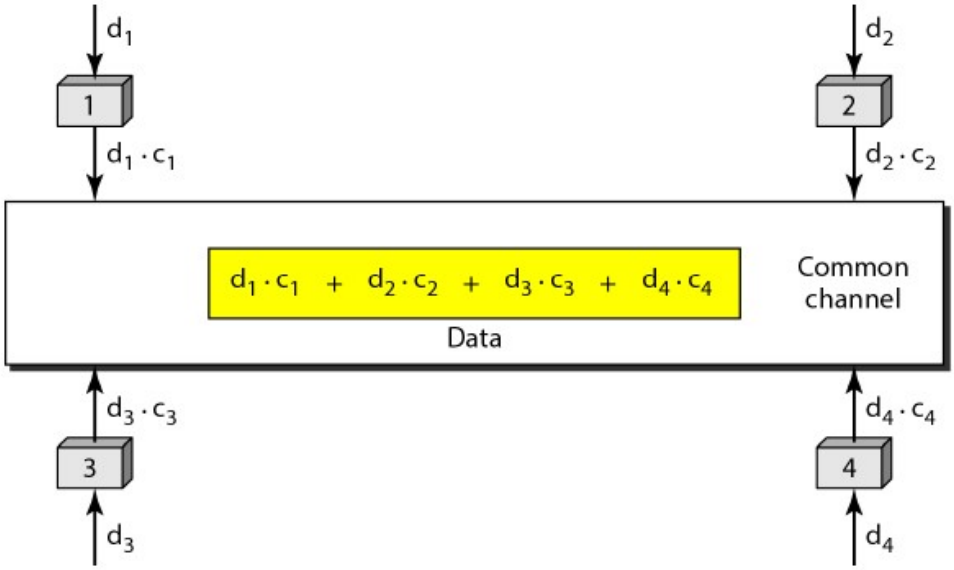

Figure 12.14 Simple idea of communication with code

Data = [*(d1 . c1 + d2 . c2 +d3 . c3 + d4 . c4) . c1]/4*

= [*d 1. c1* . c1 + d2. c2 . c1 + *d3 . c3* . c1 + *d4 . c4.* c1]/4

= [4 x *d1+0+0+0]/4*

= *4d1/4*

= *d1*

Chips

CDMA is based on coding theory. Each station is assigned a code, which is a sequence of numbers called chips, as shown in Figure 12.15. The codes are for the previous example.

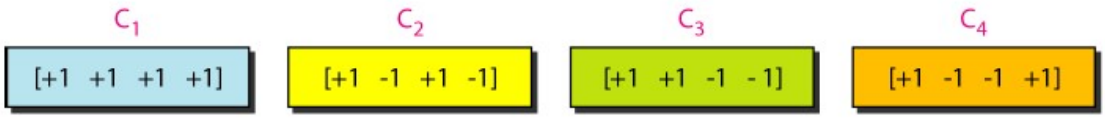

Figure 12.15 Chip sequences

We need to know that we did not choose the sequences randomly; they were carefully selected. They are called orthogonal sequences and have the following properties:

1. Each sequence is made of *N* elements, where *N* is the number of stations.
2. If we multiply a sequence by a number, every element in the sequence is multiplied by that element. This is called multiplication of a sequence by a scalar. For example,

   2.[+1 +1-1-1]=[+2+2-2-2]

3. If we multiply two equal sequences, element by element, and add the results, we get *N,* where *N* is the number of elements in the each sequence. This is called the inner product of two equal sequences. For example,

   [+1 +1-1 -1]· [+1 +1 -1 -1] = 1 + 1 + 1 + 1 = 4

4. If we multiply two different sequences, element by element, and add the results, we get 0. This is called inner product of two different sequences. For example,

   [+1 +1 -1 -1]. [+1 +1 +1 +1] = 1 + 1 - 1 - 1 = 0

5. Adding two sequences means adding the corresponding elements. The result is another sequence. For example,

   [+1+1-1-1]+[+1+1+1+1]=[+2 +2 0 0]

Data Representation

We follow these rules for encoding: If a station needs to send a 0 bit, it encodes it as -1; if it needs to send a 1 bit, it encodes it as +1. When a station is idle, it sends no signal, which is interpreted as a 0. These are shown in Figure 12.16.

Figure 12.16 Data representation in CDMA

Encoding and Decoding

As a simple example, we show how four stations share the link during a 1-bit interval. The procedure can easily be repeated for additional intervals. We assume that stations 1 and 2 are sending a 0 bit and channel 4 is sending a 1 bit. Station 3 is silent. The data at the sender site are translated to -1, -1, 0, and +1. Each station multiplies the corresponding number by its chip (its orthogonal sequence), which is unique for each station. The result is a new sequence which is sent to the channel. For simplicity, we assume that all stations send the resulting sequences at the same time. The sequence on the channel is the sum of all four sequences as defined before. Figure 12.17 shows the situation.

Now imagine station 3, which we said is silent, is listening to station 2. Station 3 multiplies the total data on the channel by the code for station 2, which is [+1 -1 +1-1], to get

[-1 -1 -3 +1]· [+1 -1 +1 -1] = -4/4 = -1 → bit 1

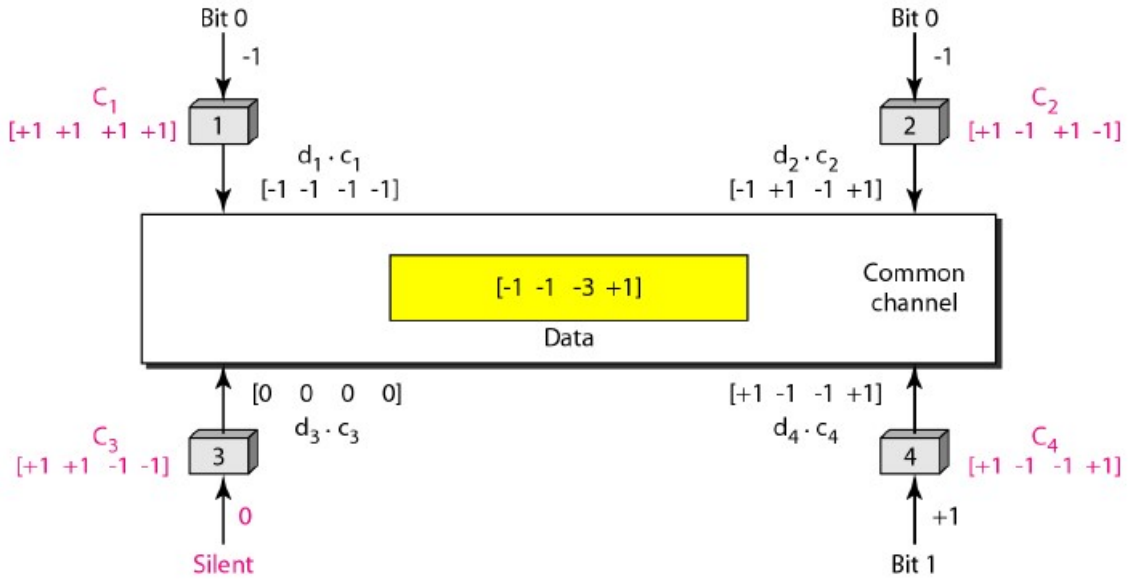

Figure 12.17 Sharing channel in CDMA

Signal Level

The process can be better understood if we show the digital signal produced by each station and the data recovered at the destination (see Figure 12.18). The figure shows the corresponding signals for each station (using NRZ-L for simplicity) and the signal that is on the common channel.

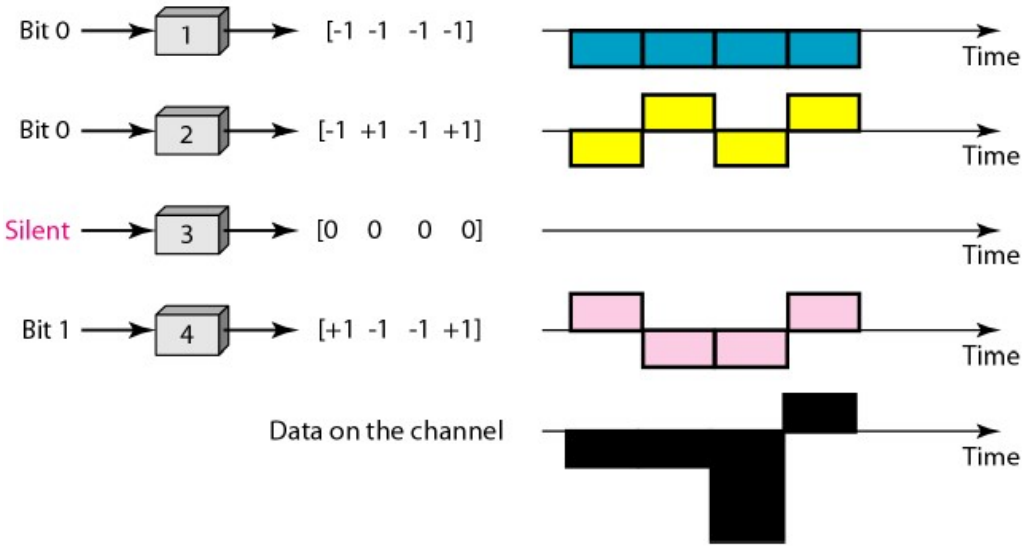

Figure 12.18 Digital signal created by four stations in CDMA

Figure 12.19 shows how station 3 can detect the data sent by station 2 by using the code for station 2. The total data on the channel are multiplied (inner product operation) by the signal representing station 2 chip code to get a new signal. The station then integrates and adds the area under the signal, to get the value -4, which is divided by 4 and interpreted as bit 0.

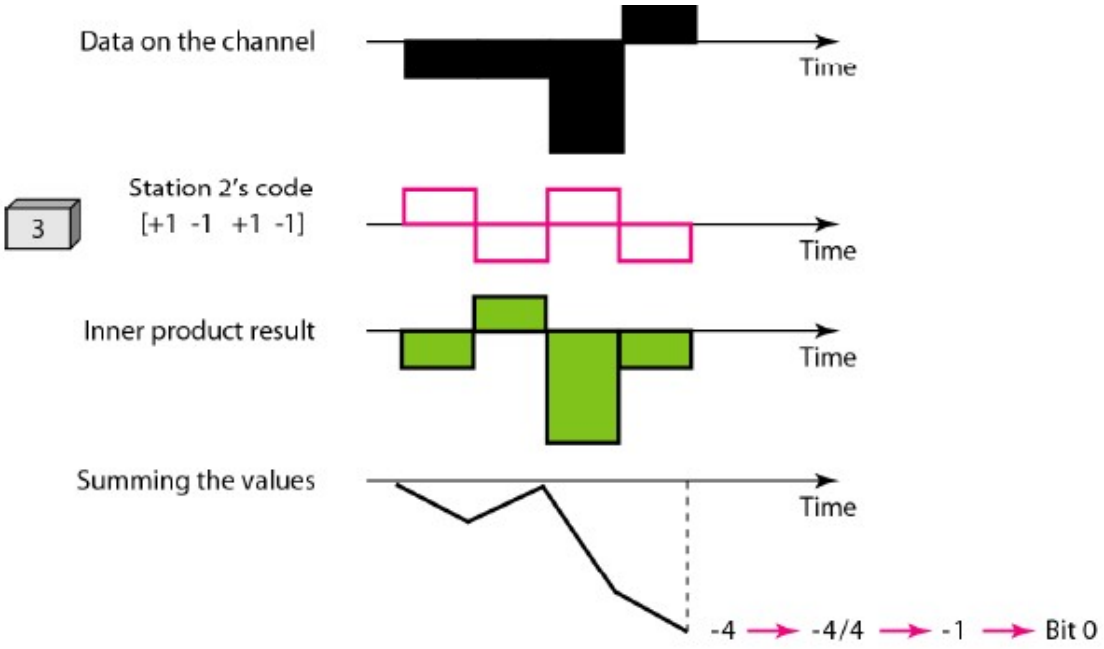

Figure 12.19 Decoding of the composite signal for one in CDMA

Sequence Generation

To generate chip sequences, we use a **Walsh table,** which is a two-dimensional table with an equal number of rows and columns, as shown in Figure 12.20.

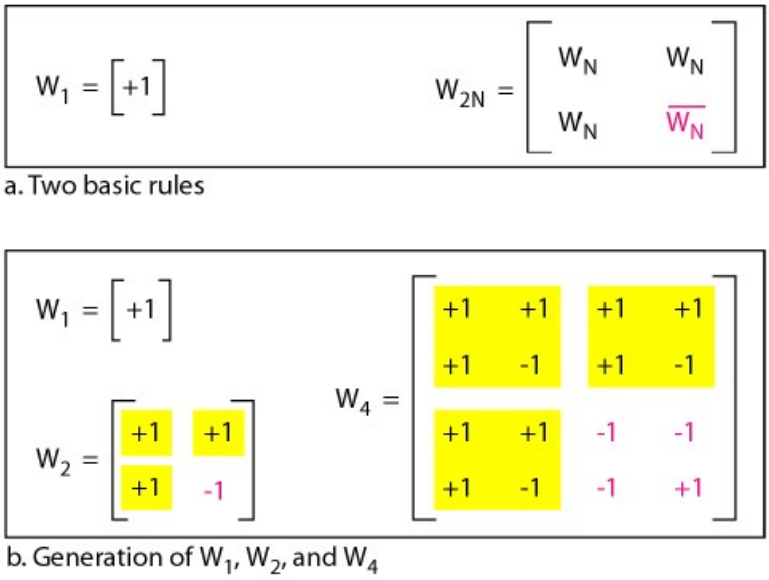

Figure 12.20 General rule and examples of creating Walsh tables

In the Walsh table, each row is a sequence of chips. W_1 for a one-chip sequence has one row and one column. We can choose -1 or +1 for the chip for this trivial table (we chose +1). According to Walsh, if we know the table for N sequences W_N we can create the table for $2N$ sequences W_{2N}, as shown in Figure 12.20. The W_N with the overbar ($\overline{\ }$) stands for the complement of W_N where each +1 is changed to -1 and vice versa. Figure 12.20 also shows how we can create W_2 and W_4 from W_1 .After we select W_1, W_2 can be made from four W_j

's, with the last one the complement of W_1 .After W_2 is generated, W_4 can be made of four W_2's, with the last one the complement of W_2. Of course, W_8 is composed of four W_4's, and so on. Note that after W_N is made, each station is assigned a chip corresponding to a row.

Something we need to emphasize is that the number of sequences N needs to be a power of 2. In other words, we need to have $N = 2^m$.

Example 12.6

Find the chips for a network with

a. Two stations

b. Four stations

Solution

We can use the rows of W_2 and W_4 in Figure 12.20:

a. For a two-station network, we have [+1 +1] and [+1 -1].

b. For a four-station network we have [+1 +1 +1 +1], [+1 -1 +1 -1], [+1 +1 -1 -1], and

[+1-1-1 +1].

Example 12.7

What is the number of sequences if we have 90 stations in our network?

Solution

The number of sequences needs to be 2^m. We need to choose $m = 7$ and $N = 2^7$ or 128. We can then use 90 of the sequences as the chips.

Reference:

1. B. A. Forouzan: Data Communications and Networking, Fourth edition, TMH .