Multiple Access

Propagation Time

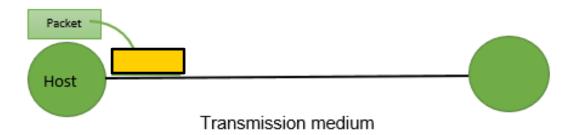
• Propagation time measures the time required for a bit to travel from the source to the destination. The propagation time is calculated by dividing the distance by the propagation speed.



Transmission Time

In data communications we don't send just 1 bit, we send a message. Time taken to put a packet onto link. In other words, it is simply time required to put data bits on the wire/communication medium. It depends on length of packet and bandwidth of network.

Transmission time = Message sizeBandwidth



Queuing Time

The third component in latency is the queuing time, the time needed for each intermediate or end device to hold the message before it can be processed. Latency

• The latency or delay defines how long it takes for an entire message to completely arrive at the destination from the time the first bit is sent out from the source.

Latency = propagation time +transmission time +queuing time + processing delay

Figure 12.1 Data link layer divided into two functionality-oriented sublayers

The upper sublayer is responsible for data link control (a mechanism which provides a link with reliable communication), and the lower sublayer is responsible for resolving access to the shared media.

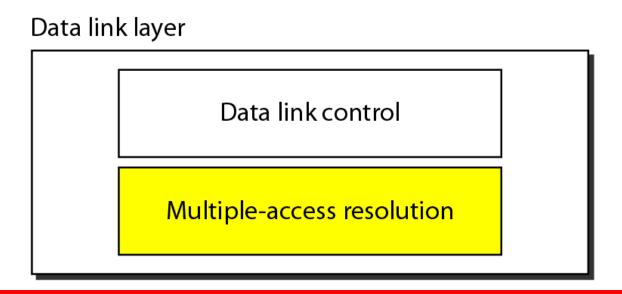
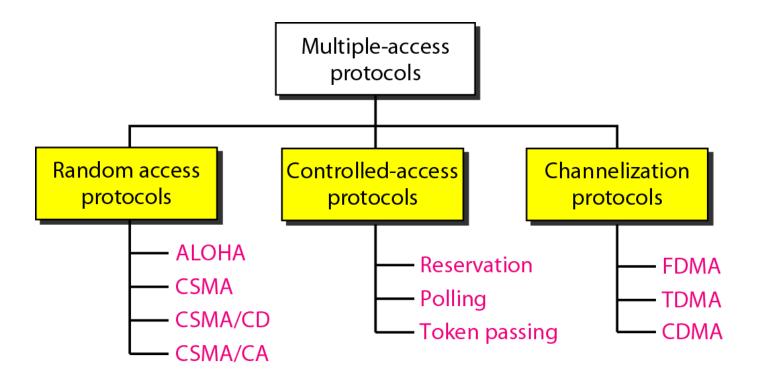


Figure 12.2 Taxonomy of multiple-access protocols discussed in this chapter



12-1 RANDOM ACCESS

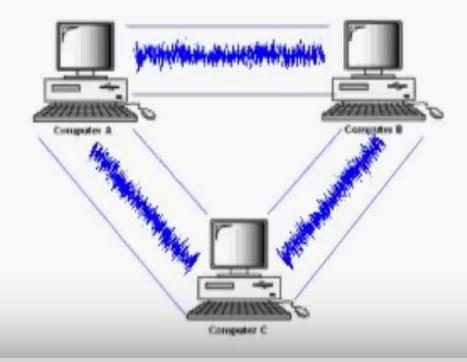
In random access or contention methods, no station is superior to another station and none is assigned the control over another. No station permits, or does not permit, another station to send.

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?

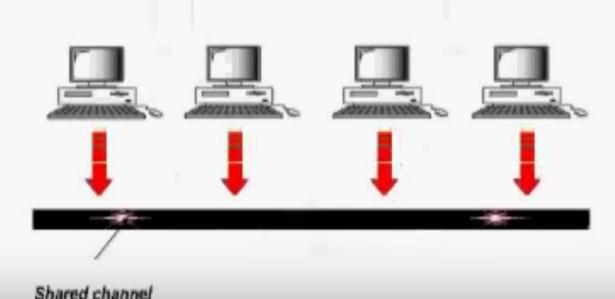
Point-to-point networks

Point-to-point networks are those in which when a message is sent from one computer to another, it usually has to be sent via other computers in the network. A point-to-point network consists of many connections between individual pairs of computers.



Broadcast networks

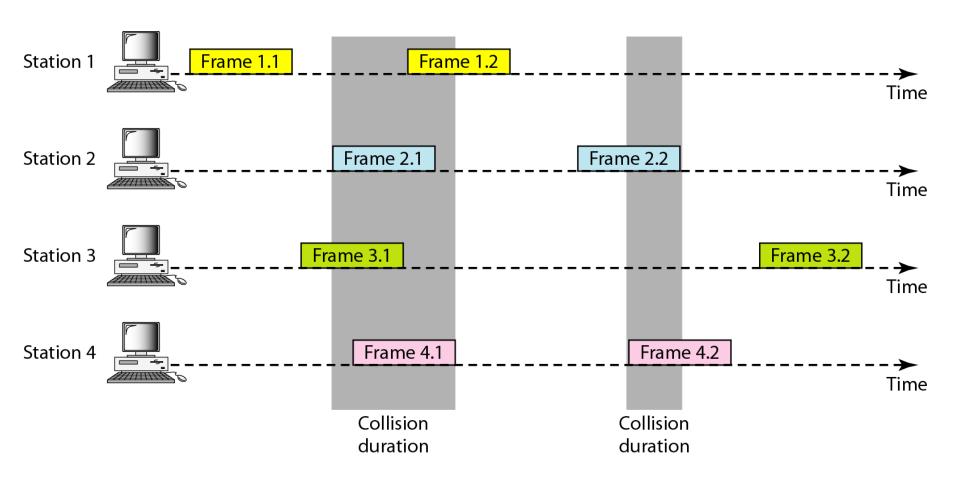
Broadcast networks have a single communication channel that is shared by all the machines on the network. A packet sent by one computer is received by all the other computers on the network. The packets that are sent contain the address of the receiving computer, each computer checks this field to see if it matches its own address. If it does not then it is usually ignored; if it does then it is read. Broadcast channels are sometimes known as multi-access channel.

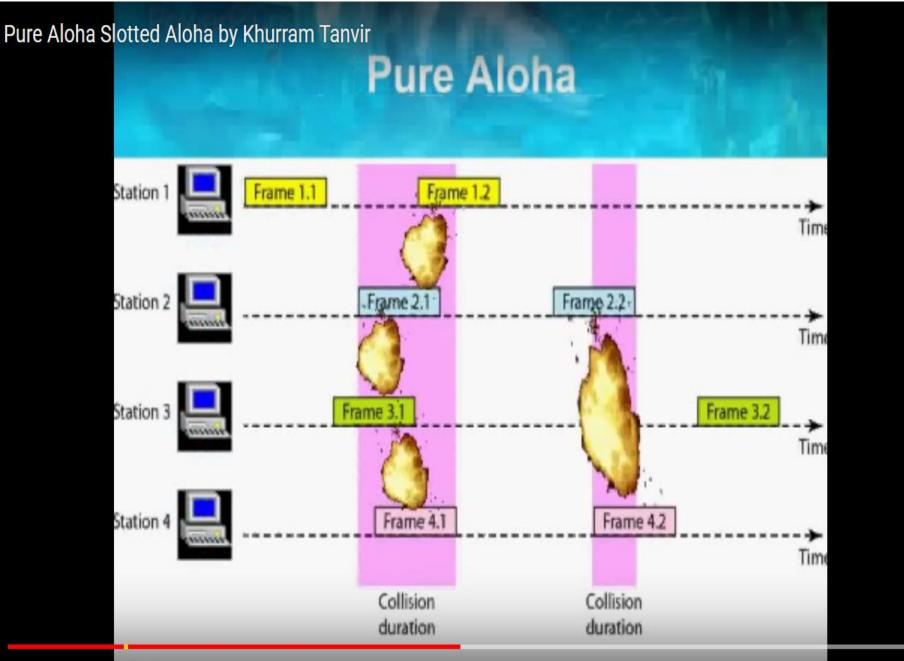


ALOHA

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.

Figure 12.3 Frames in a pure ALOHA network







Two frames survive: frame 1.1 from station 1 and frame 3.2 from station 3.

The pure ALOHA protocol relies on acknowledgments from the receiver.

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.

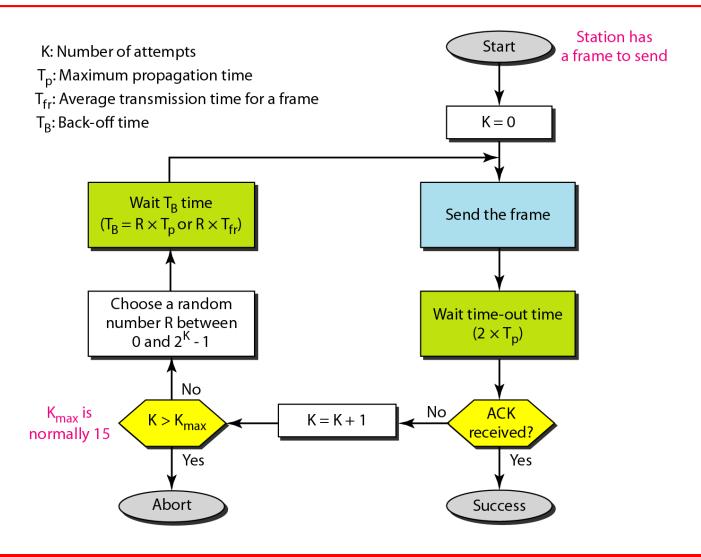
A collision involves two or more stations. If all these stations try to resend their frames after the time-out, the frames will collide again

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 TB.

Pure ALOHA has a second method to prevent congesting the channel with retransmitted frames

After a maximum number of retransmission attempts Kmax' a station must give up and try later

Figure 12.4 Procedure for pure ALOHA protocol



Example 12.1

The stations on a wireless ALOHA network are a maximum of 600 km apart. If we assume that signals propagate at 3×10^8 m/s, we find

$$T_p = (6 \times 10^5) / (3 \times 10^8) = 2 \text{ ms.}$$

Now we can find the value of T_B for different values of K.

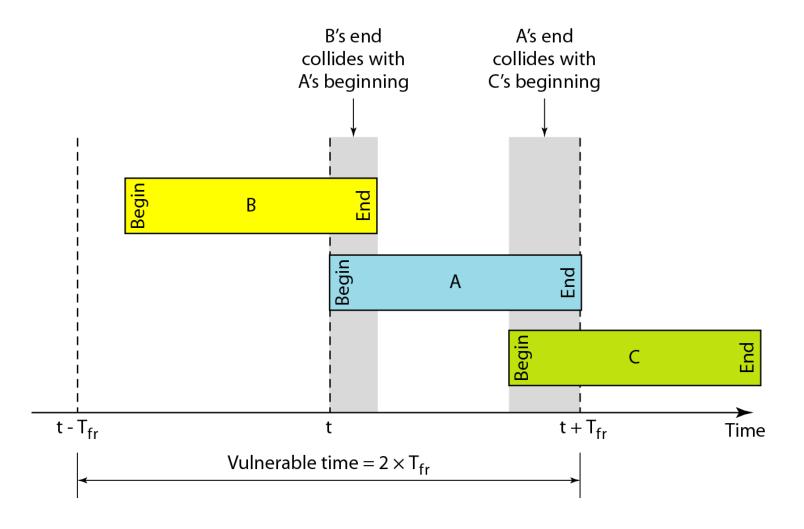
a. For K = 1, the range is $\{0, 1\}$. The station needs to generate a random number with a value of 0 or 1. This means that T_B is either 0 ms (0×2) or 2 ms (1×2) , based on the outcome of the random variable.

Example 12.1 (continued)

- 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 $0, 2, 4, \ldots, 14$ ms, based on the outcome of the random variable.
- d. We need to mention that if K > 10, it is normally set to 10.

Vulnerable time
The vulnerable time, in which there is a possibility
of collision.

Figure 12.5 Vulnerable time for pure ALOHA protocol



Example 12.2

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

Solution

Average frame transmission time T_{fr} is 200 bits/200 kbps or 1 ms. The vulnerable time is 2×1 ms = 2 ms. This means no station should send later than 1 ms before this station starts transmission and no station should start sending during the one 1-ms period that this station is sending.

Note

The throughput for pure ALOHA is $S = G \times e^{-2G}$. The maximum throughput $S_{max} = 0.184$ when G = (1/2).

G transmission-attempts per frame-time.

Example 12.3

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

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

Solution

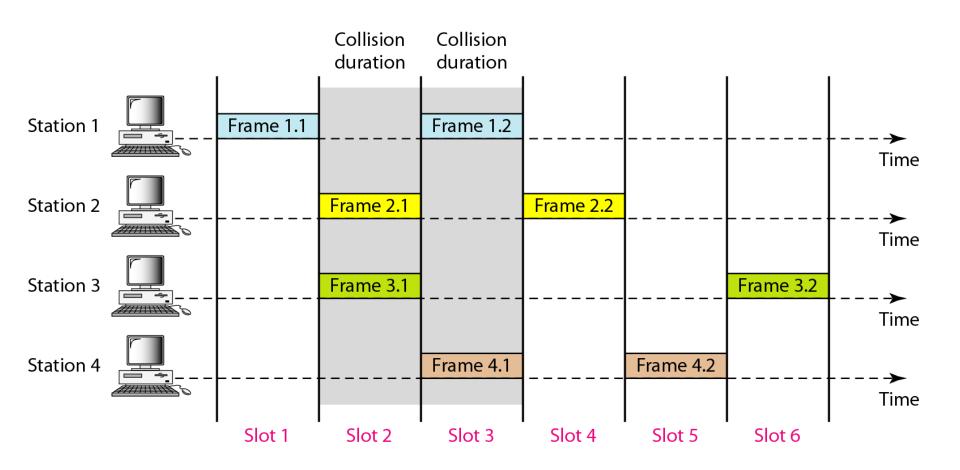
The frame transmission time is 200/200 kbps or 1 ms.

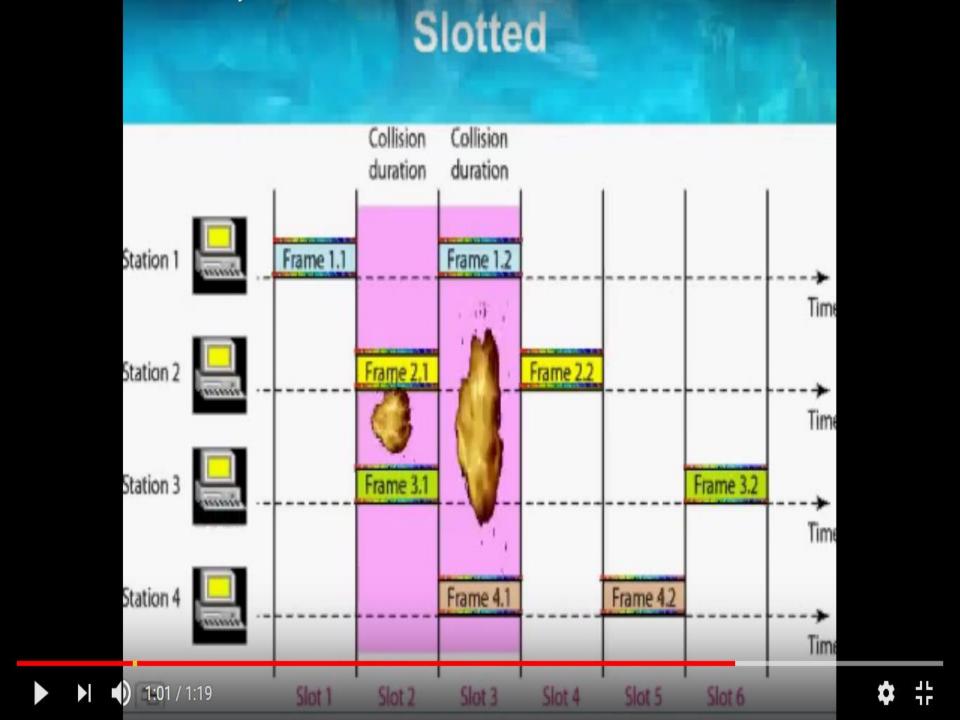
a. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1. In this case $S = G \times e^{-2 G}$ or S = 0.135 (13.5 percent). This means that the throughput is $1000 \times 0.135 = 135$ frames. Only 135 frames out of 1000 will probably survive.

Example 12.3 (continued)

- b. If the system creates 500 frames per second, this is (1/2) frame per millisecond. The load is (1/2). In this case $S = G \times e^{-2G}$ or S = 0.184 (18.4 percent). This means that the throughput is $500 \times 0.184 = 92$ and that only 92 frames out of 500 will probably survive. Note that this is the maximum throughput case, percentagewise.
- c. If the system creates 250 frames per second, this is (1/4) frame per millisecond. The load is (1/4). In this case $S = G \times e^{-2G}$ or S = 0.152 (15.2 percent). This means that the throughput is $250 \times 0.152 = 38$. Only 38 frames out of 250 will probably survive.

Figure 12.6 Frames in a slotted ALOHA network

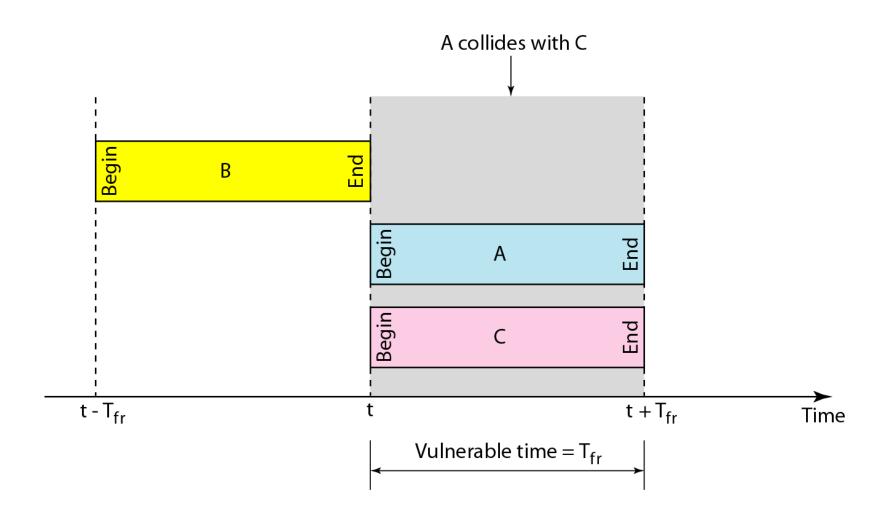




Note

The throughput for slotted ALOHA is $S = G \times e^{-G}$. The maximum throughput $S_{max} = 0.368$ when G = 1.

Figure 12.7 Vulnerable time for slotted ALOHA protocol



Example 12.4

A slotted ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces

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

Solution

The frame transmission time is 200/200 kbps or 1 ms.

a. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1. In this case $S = G \times e^{-G}$ or S = 0.368 (36.8 percent). This means that the throughput is $1000 \times 0.0368 = 368$ frames. Only 386 frames out of 1000 will probably survive.

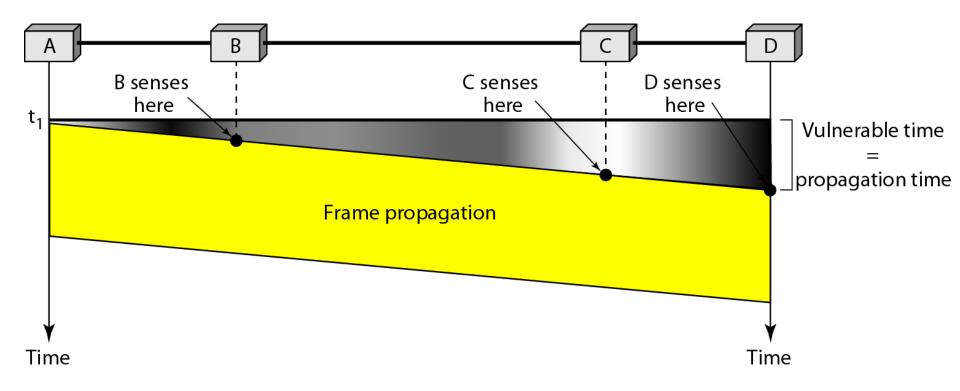
Example 12.4 (continued)

- b. If the system creates 500 frames per second, this is (1/2) frame per millisecond. The load is (1/2). In this case $S = G \times e^{-G}$ or S = 0.303 (30.3 percent). This means that the throughput is $500 \times 0.0303 = 151$. Only 151 frames out of 500 will probably survive.
- c. If the system creates 250 frames per second, this is (1/4) frame per millisecond. The load is (1/4). In this case $S = G \times e^{-G}$ or S = 0.195 (19.5 percent). This means that the throughput is $250 \times 0.195 = 49$. Only 49 frames out of 250 will probably survive.

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

Figure 12.9 Vulnerable time in CSMA



What should a station do if the channel is busy?

What should a station do if the channel is idle?

I-Persistent

The I-persistent method is simple and straightforward. In this method, after the station finds the line idle, it sends its frame immediately This method has the highest chance of collision because---

Nonpersistent

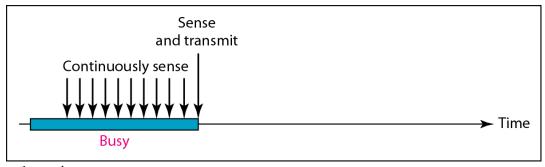
In the nonpersistent method, a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again.

p-persistent

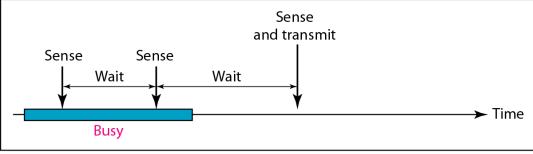
The p-persistent method is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time. The p-persistent approach combines the advantages of the other two strategies.

- 1. With probability p, the station sends its frame.
- 2. With probability q = 1 p, the station waits for the beginning of the next time slot and checks the line again.

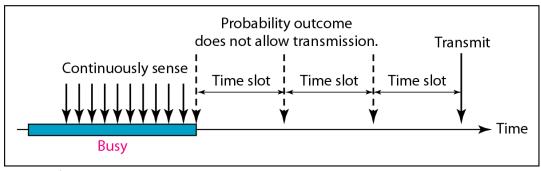
Figure 12.10 Behavior of three persistence methods



a. 1-persistent

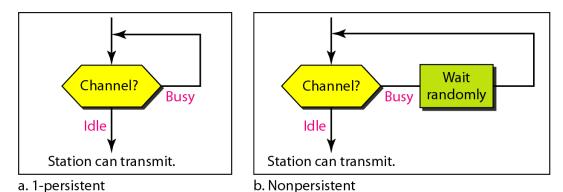


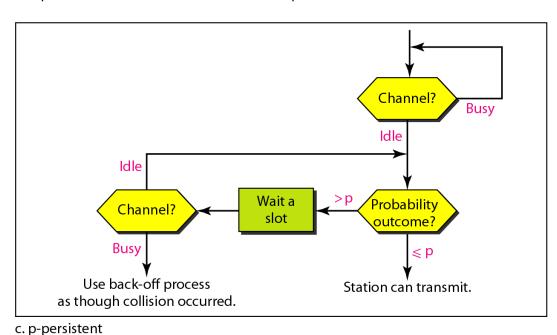
b. Nonpersistent



c. p-persistent

Figure 12.11 Flow diagram for three persistence methods





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

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 12.12 Collision of the first bit in CSMA/CD

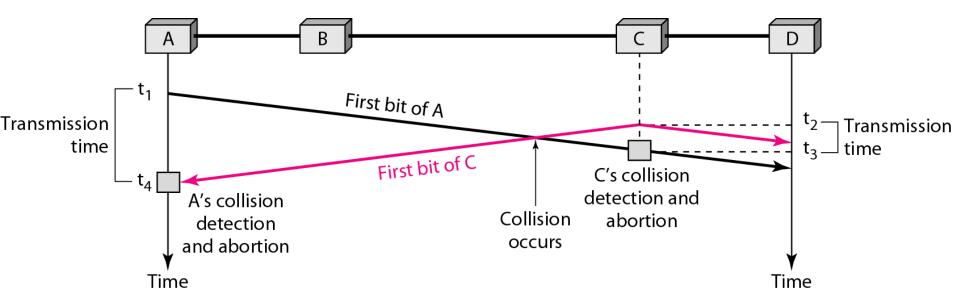
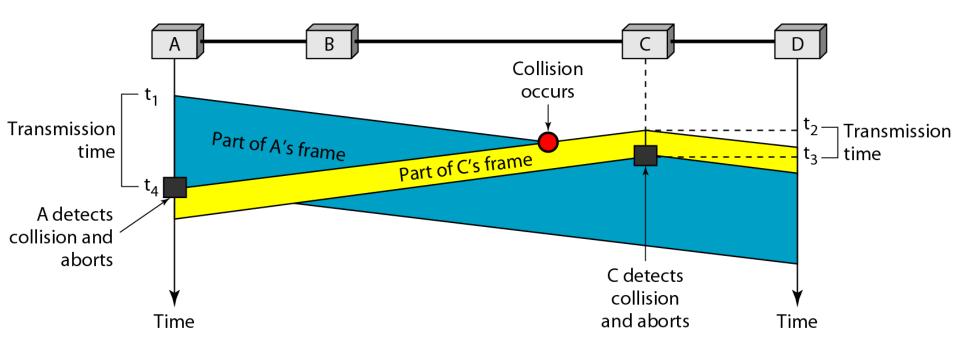


Figure 12.13 Collision and abortion in CSMA/CD



Minimum Frame Size

For CSMA/CD to work, we need a restriction on the frame size. Before sending the last bit of the frame, the sending station must detect a collision, if any, and abort the transmission.

This is so because the station, once the entire frame is sent, does not keep a copy of the frame

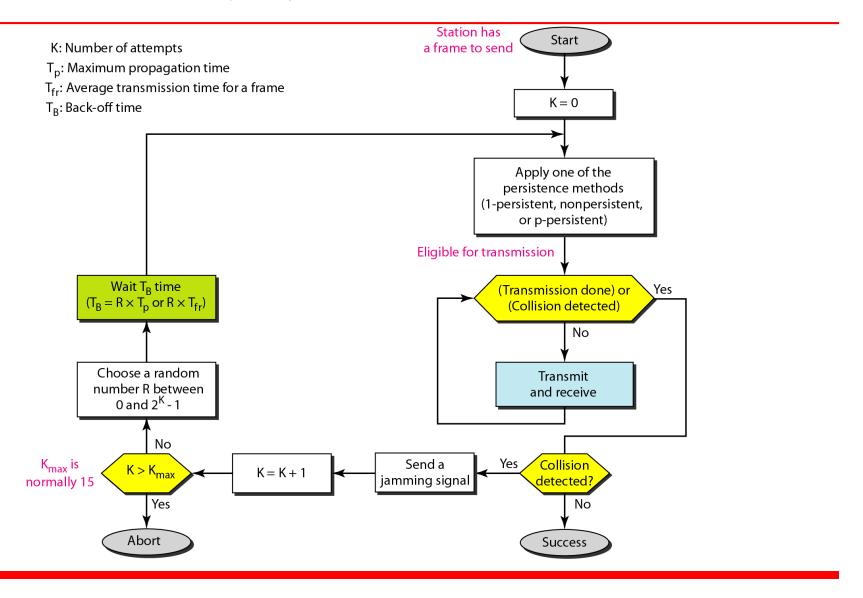
Therefore, the frame transmission time Tfr must be at least two times the maximum propagation time Tp.

A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time (including the delays in the devices and ignoring the time needed to send a jamming signal, as we see later) is 25.6 µs, what is the minimum size of the frame?

Solution

The frame transmission time is $T_{fr} = 2 \times T_p = 51.2 \ \mu s$. This means, in the worst case, a station needs to transmit for a period of 51.2 μs to detect the collision. The minimum size of the frame is 10 Mbps \times 51.2 $\mu s = 512$ bits or 64 bytes. This is actually the minimum size of the frame for Standard Ethernet.

Figure 12.14 Flow diagram for the CSMA/CD



CSMA/CD is similar to the one for the ALOHA protocol, but there are differences.

The first difference is the addition of the persistence process.

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.

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.

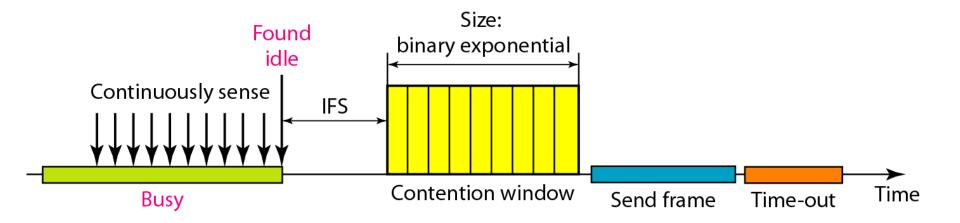
Throughput

I-persistent method the maximum throughput is around 50 percent when G = 1. For nonpersistent method, the maximum throughput can go up to 90 percent when G is between 3 and 8.

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

CSMA/CD is not useful for effective collision detection in wireless mode.

Figure 12.16 Timing in CSMA/CA



Interframe Space (IFS)

First, collisions are avoided by deferring transmission even if the channel is found idle. When an idle channel is found, the station does not send immediately

The IFS variable can also be used to prioritize stations or frame types

Contention Window

The contention window is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time. The number of slots in the window changes according to the binary exponential back-off strategy.

However, if the station finds the channel busy, it does not restart the process; it just stops the timer and restarts it when the channel is sensed as idle. This gives priority to the station with the longest waiting time.

Acknowledgment

With all these precautions, there still may be a collision resulting in destroyed data. In addition, the data may be corrupted during the transmission. The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame.

-

Note

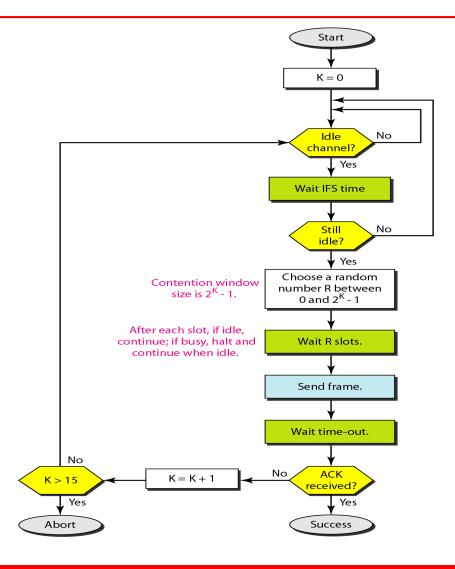
In CSMA/CA, the IFS can also be used to define the priority of a station or a frame.

-

Note

In CSMA/CA, if the station finds the channel busy, it does not restart the timer of the contention window; it stops the timer and restarts it when the channel becomes idle.

Figure 12.17 Flow diagram for CSMA/CA



12-2 CONTROLLED ACCESS

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. We discuss three popular controlled-access methods.

Topics discussed in this section:

Reservation
Polling
Token Passing

Figure 12.18 Reservation access method

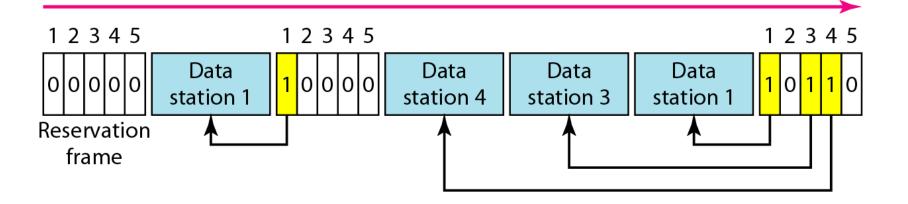


Figure 12.19 Select and poll functions in polling access method

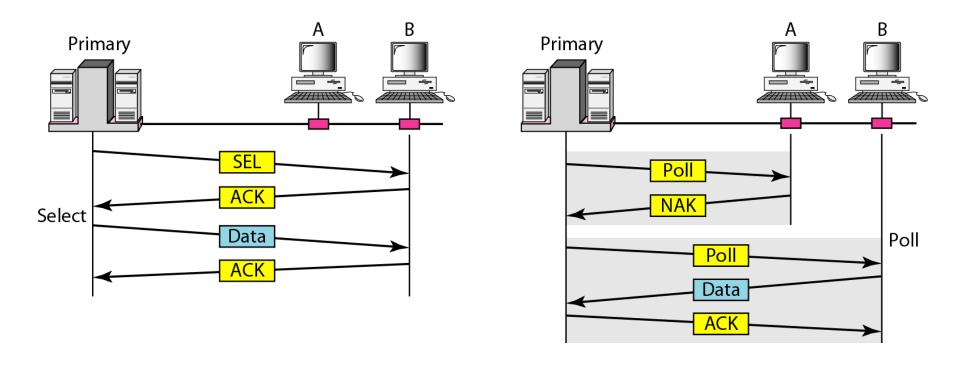
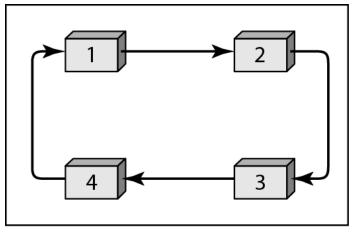
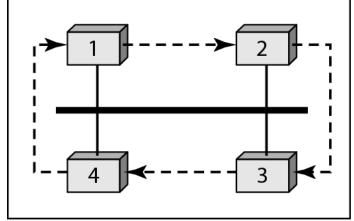


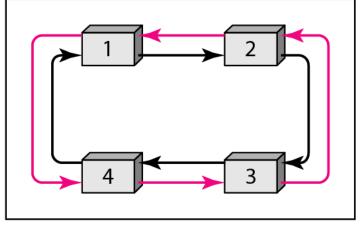
Figure 12.20 Logical ring and physical topology in token-passing access method



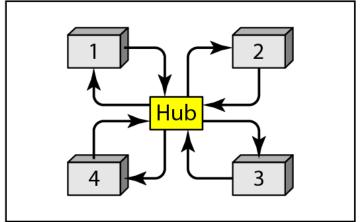
a. Physical ring



c. Bus ring d. Star ring



b. Dual ring



12-3 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.

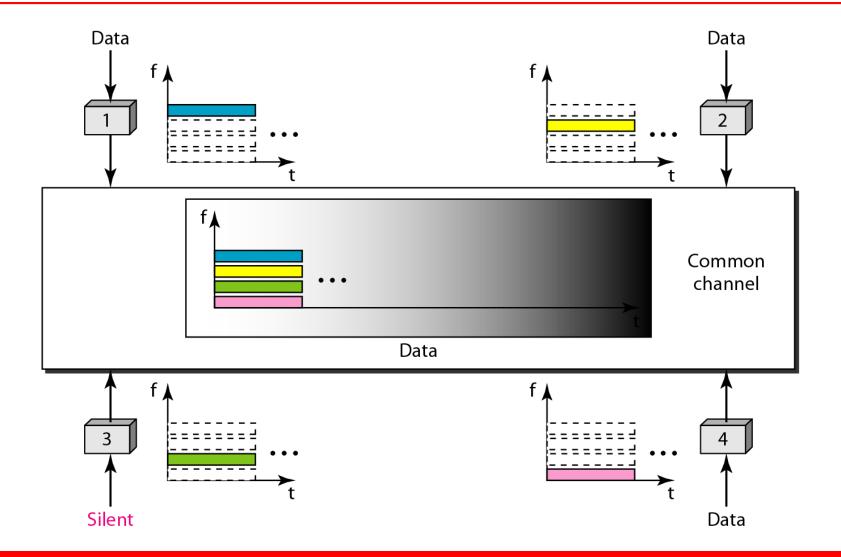
Topics discussed in this section:

Frequency-Division Multiple Access (FDMA)
Time-Division Multiple Access (TDMA)
Code-Division Multiple Access (CDMA)

Note

We see the application of all these methods in Chapter 16 when we discuss cellular phone systems.

Figure 12.21 Frequency-division multiple access (FDMA)

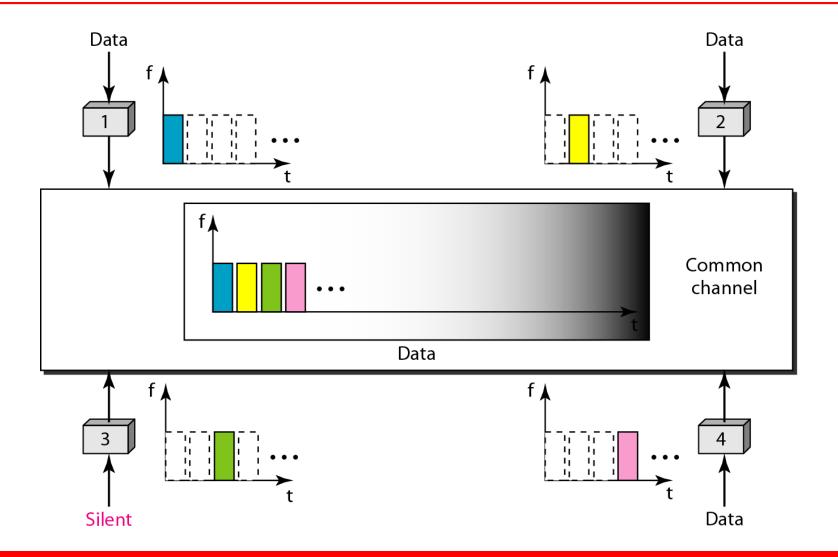


-

Note

In FDMA, the available bandwidth of the common channel is divided into bands that are separated by guard bands.

Figure 12.22 Time-division multiple access (TDMA)



Note

In TDMA, the bandwidth is just one channel that is timeshared between different stations.

Note

In CDMA, one channel carries all transmissions simultaneously.

Figure 12.23 Simple idea of communication with code

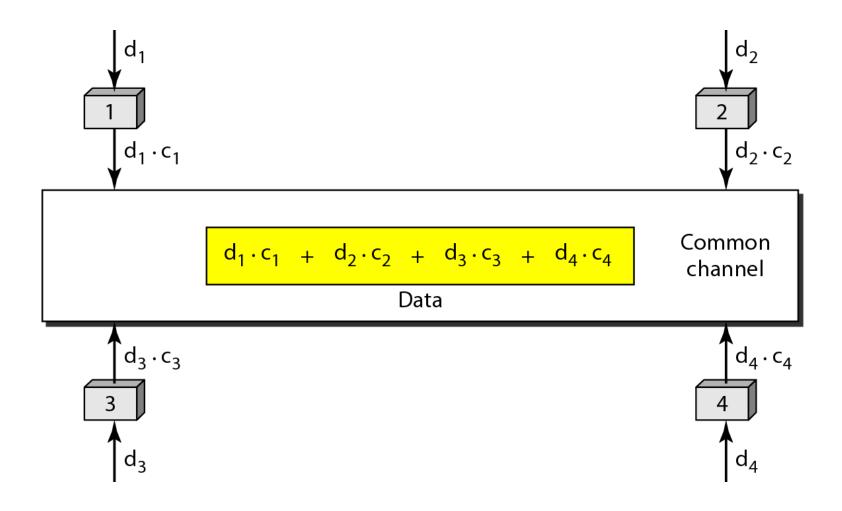


Figure 12.24 Chip sequences

Figure 12.25 Data representation in CDMA



Figure 12.26 Sharing channel in CDMA

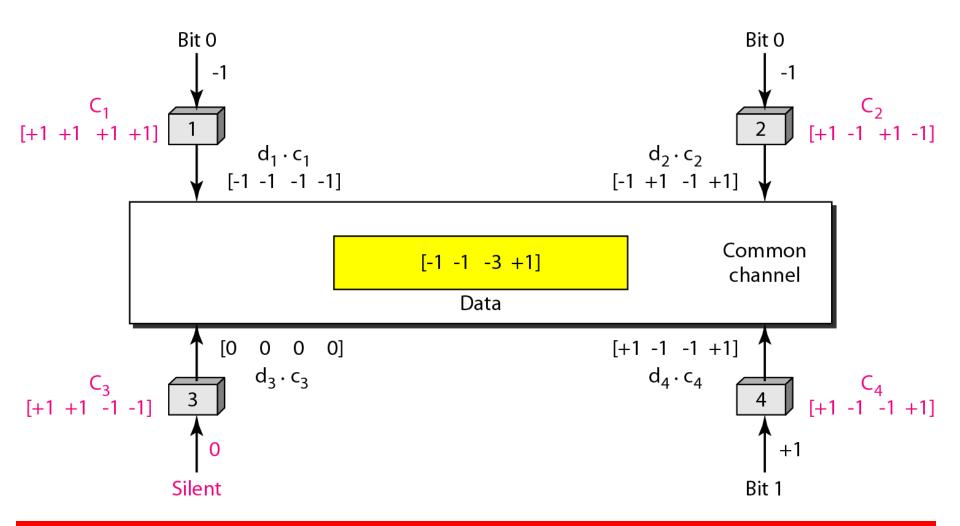


Figure 12.27 Digital signal created by four stations in CDMA

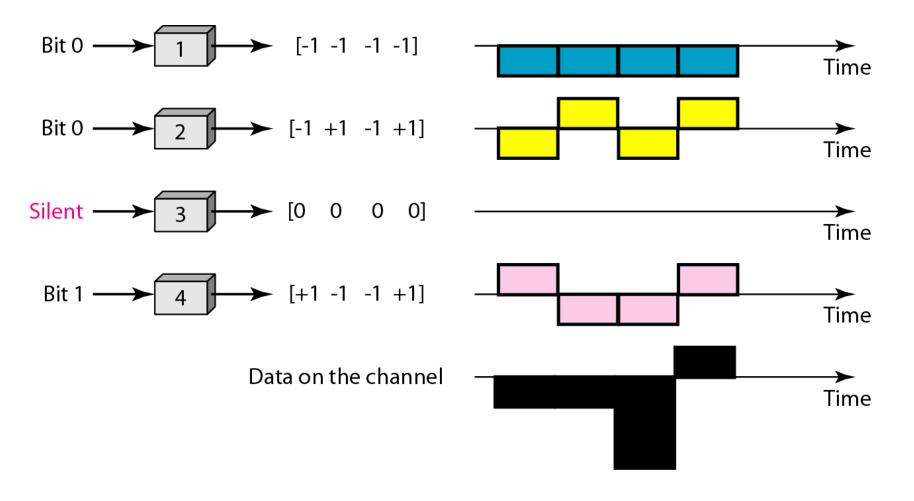


Figure 12.28 Decoding of the composite signal for one in CDMA

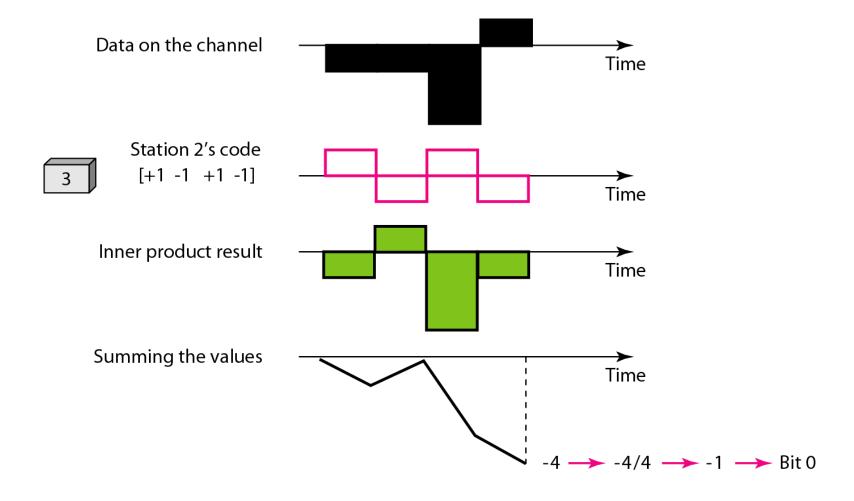


Figure 12.29 General rule and examples of creating Walsh tables

$$W_1 = \begin{bmatrix} +1 \end{bmatrix} \qquad W_{2N} = \begin{bmatrix} W_N & W_N \\ W_N & \overline{W}_N \end{bmatrix}$$

a. Two basic rules

$$W_{1} = \begin{bmatrix} +1 \\ +1 \end{bmatrix}$$

$$W_{2} = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix}$$

$$W_{4} = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix}$$

$$W_{4} = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix}$$

$$W_{1} = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix}$$

$$W_{2} = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix}$$

$$W_{3} = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix}$$

b. Generation of W_1 , W_2 , and W_4

Note

The number of sequences in a Walsh table needs to be $N = 2^{m}$.

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 12.29: 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 -1 -1], and [+1 -1 -1 +1].

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.

Prove that a receiving station can get the data sent by a specific sender if it multiplies the entire data on the channel by the sender's chip code and then divides it by the number of stations.

Solution

Let us prove this for the first station, using our previous four-station example. We can say that the data on the channel

 $D = (d_1 \cdot c_1 + d_2 \cdot c_2 + d_3 \cdot c_3 + d_4 \cdot c_4).$ The receiver which wants to get the data sent by station 1 multiplies these data by c_1 .

Example 12.8 (continued)

$$\begin{aligned} D \cdot c_1 &= (d_1 \cdot c_1 + d_2 \cdot c_2 + d_3 \cdot c_3 + d_4 \cdot c_4) \cdot c_1 \\ &= d_1 \cdot c_1 \cdot c_1 + d_2 \cdot c_2 \cdot c_1 + d_3 \cdot c_3 \cdot c_1 + d_4 \cdot c_4 \cdot c_1 \\ &= d_1 \times N + d_2 \times 0 + d_3 \times 0 + d_4 \times 0 \\ &= d_1 \times N \end{aligned}$$

When we divide the result by N, we get d_1 .