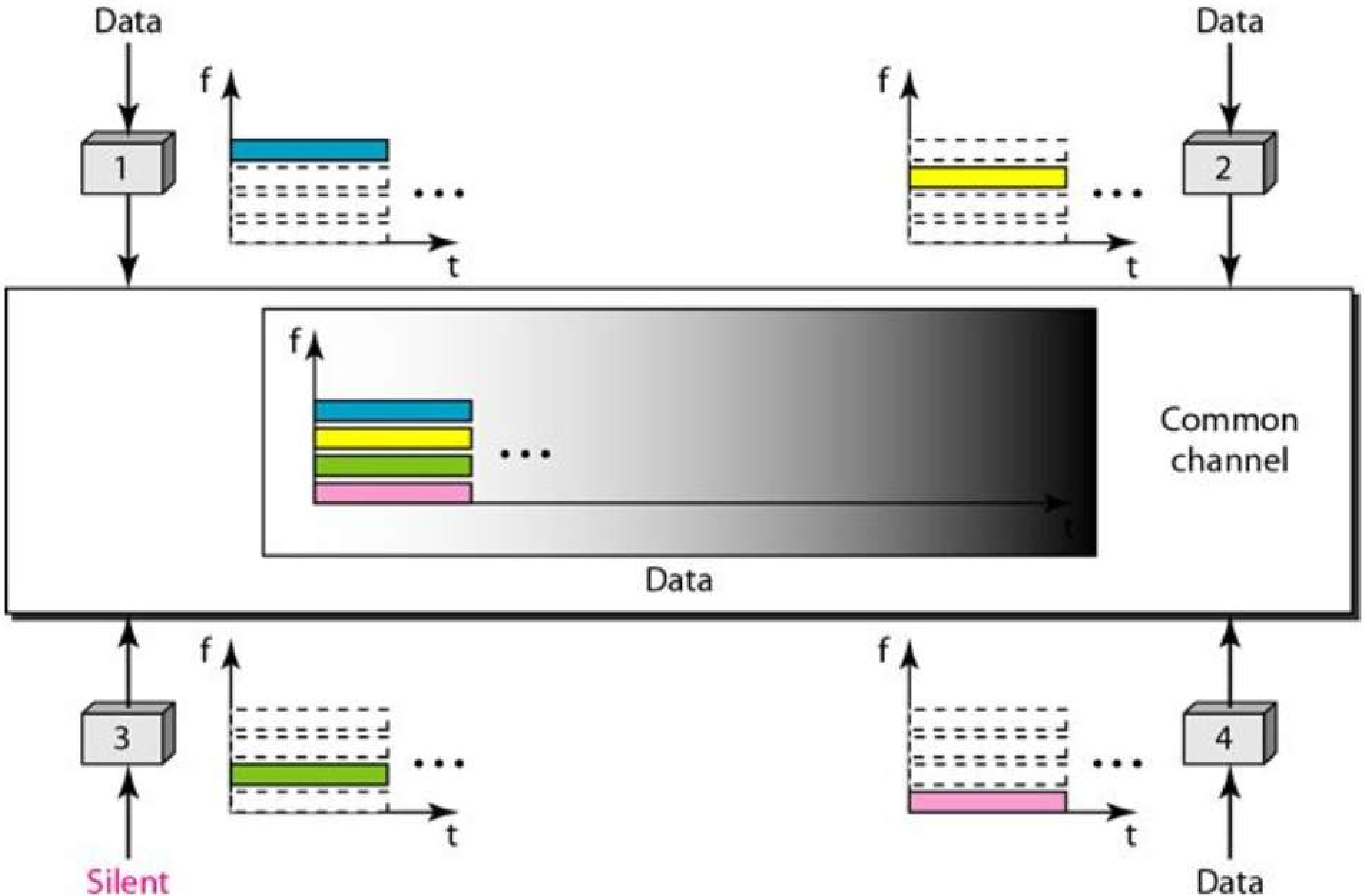
Channelization Protocols

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. The three channelization protocols are FDMA, TDMA, and CDMA.

The Frequency-Division Multiple Access (FDMA):

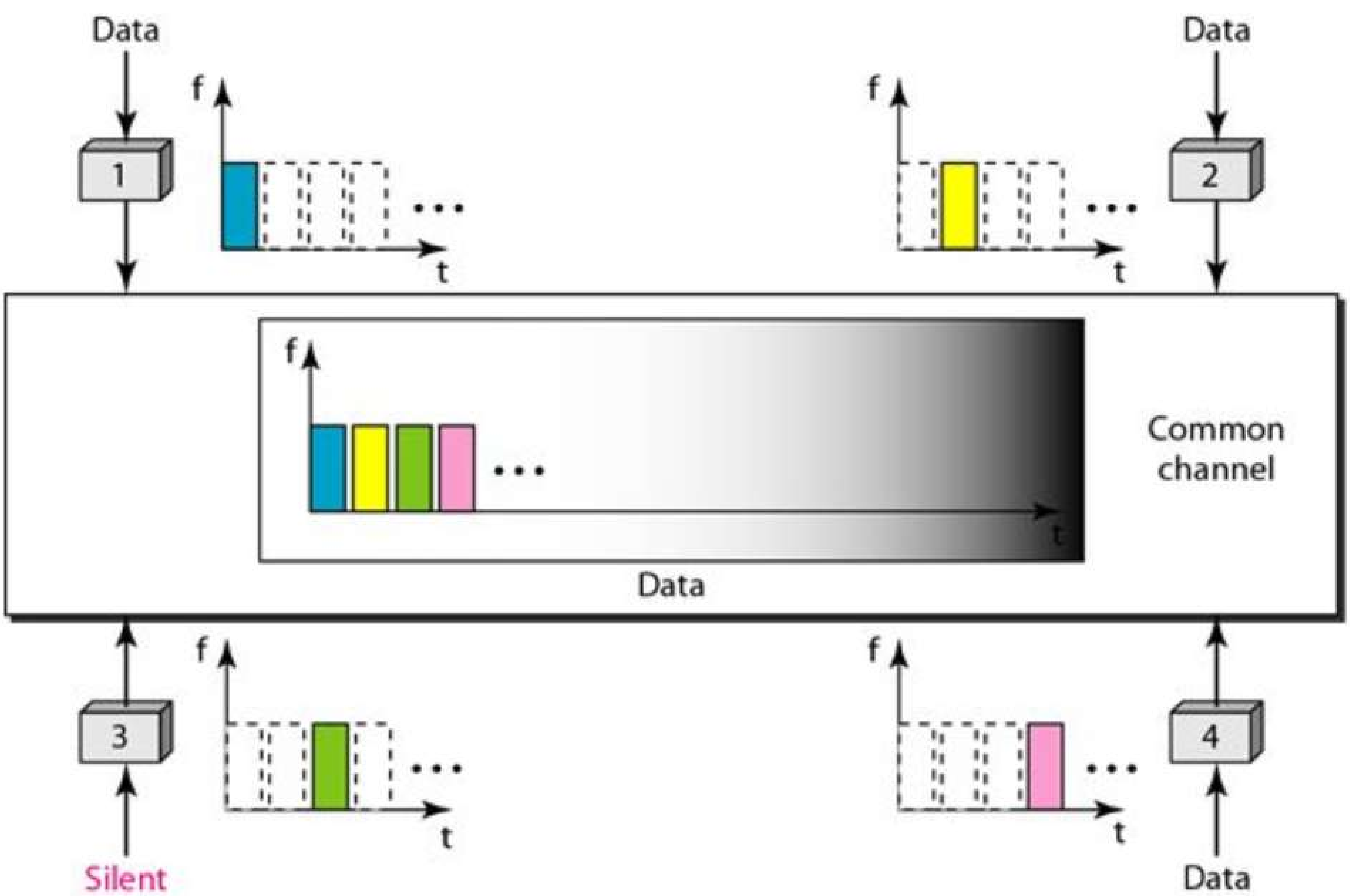
In frequency-division multiple access (FDMA), the available bandwidth is divided into frequency bands. Each station is allocated a band to send its data. In other words, each band is reserved for a specific station, and it belongs to the station all the time. Each station also uses a bandpass filter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



Data
Data
f
t
f
t
1
2
Common
channel
Data
f
t
f
t
3
4
Silent
Data

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. The following figure shows the idea behind TDMA.



Code-Division Multiple Access (CDMA):

CDMA simply means communication with different codes. 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.

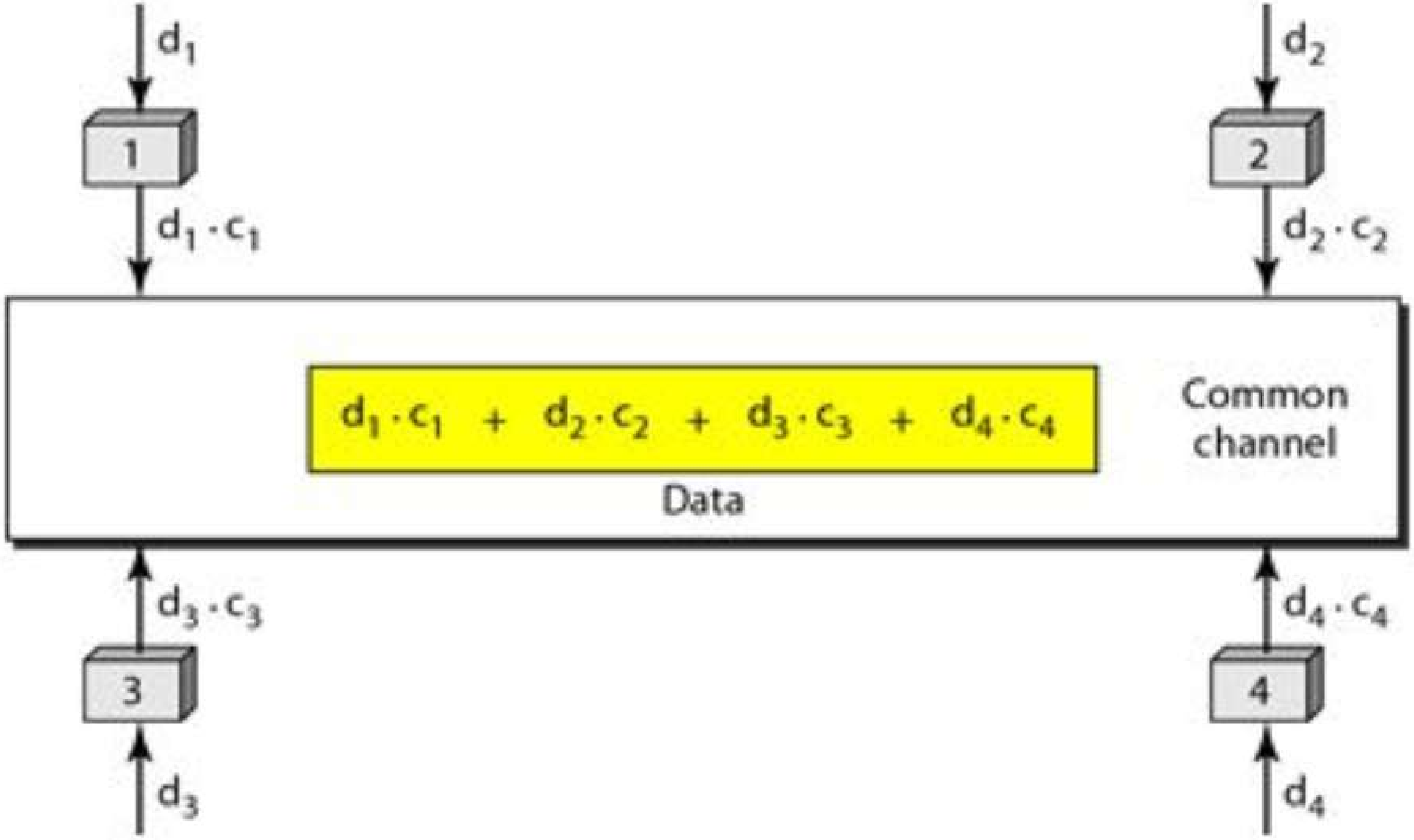
Implementation:

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.

1. If we multiply each code by another, we get 0.

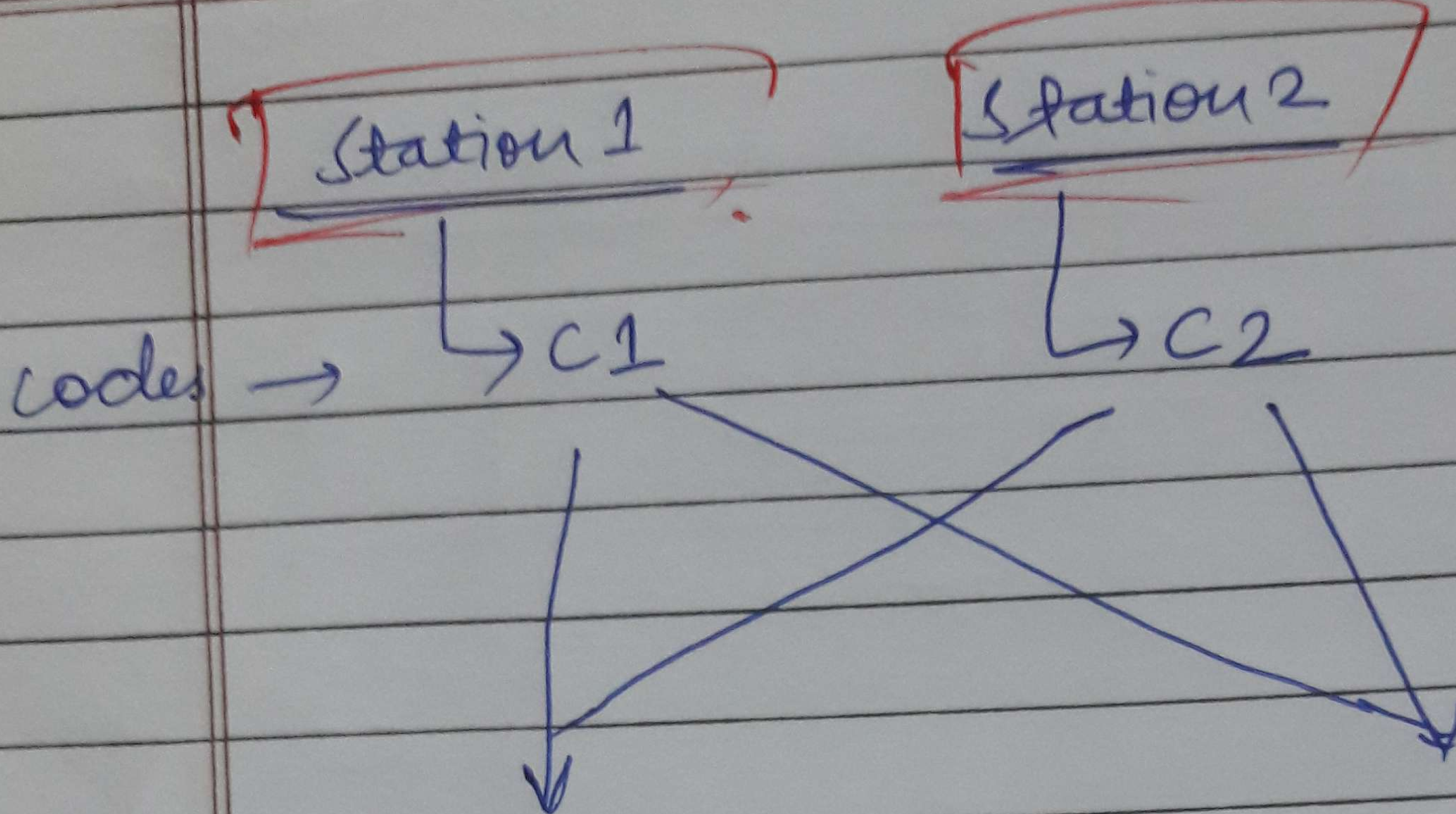
2. If we multiply each code by itself, we get 4 (the number of stations).

With these two properties in mind, how the above four stations can send data using the same common channel, as shown in the following figure.



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. And so on. The data that go on the channel are the sum of all these terms, as shown in the box.

Date



$C1 * C2 = 0$

(tell no. of stations)

↓

C2 × C2 = ~~1~~ 2

C1 × C1 = ~~1~~ 2

Here no of stations are 2 ∴ Ans is 2

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 station1. 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 station1.

$$data = (d1.c1+d2.c2+d3.c3+d4.c4).c1$$

$$= c1. d1. c1+ c1. d2. c2+ c1. d3. c3+ c1. d4. c4= 4d1$$

Station 1 → $c1 \times d1$

Station 2 → $c2 \times d2$

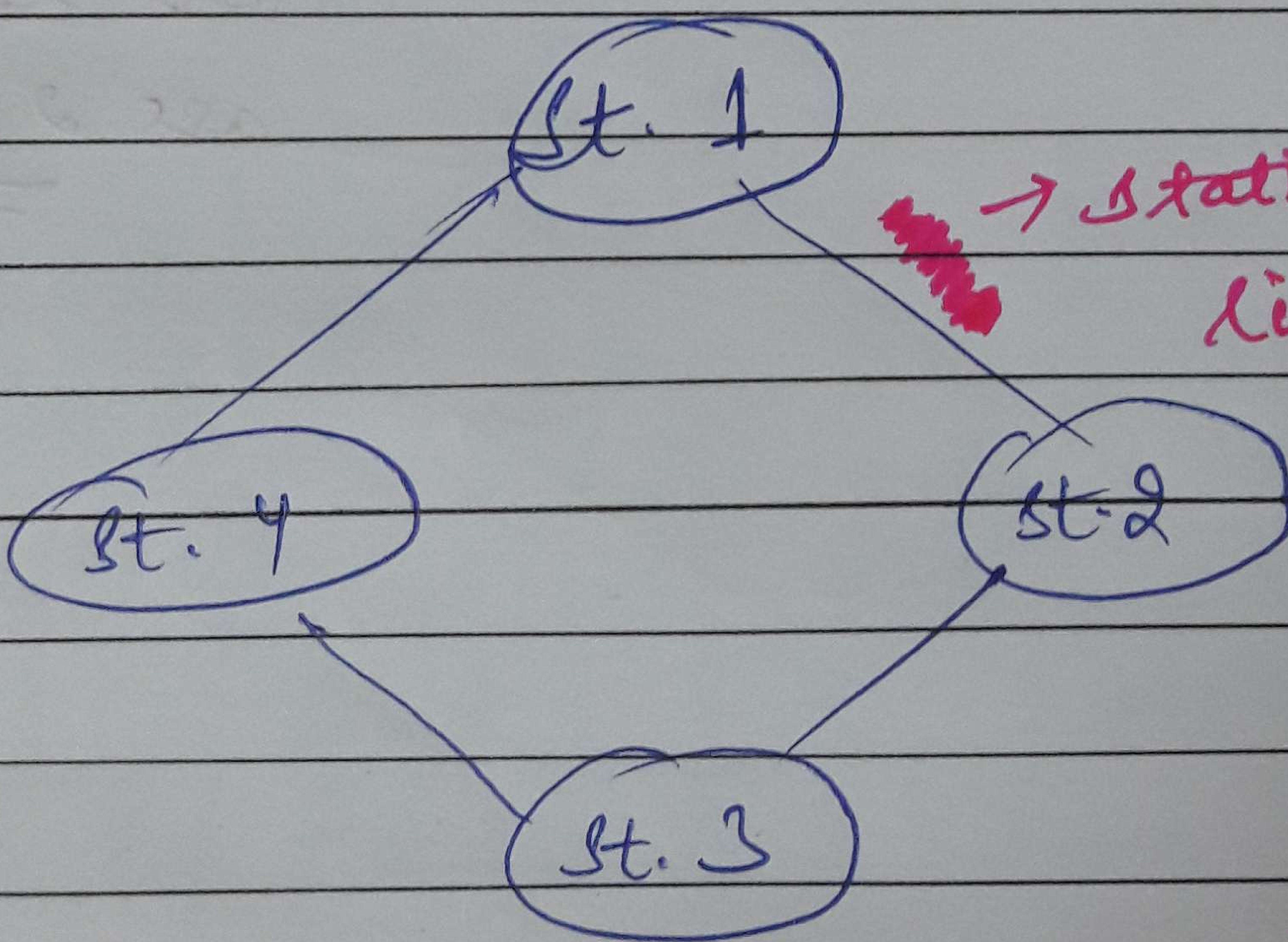
Station 3 → $c3 \times d3$

Station 4 → $c4 \times d4$

$(c1 \times d1) + (c2 \times d2) + (c3 \times d3) + (c4 \times d4)$

↓

on a single channel



→ station 2 wants to listen/hear what station 1 says.

Station 2 Multiplies the complete data by code of st. 1

i.e. $[(c1 \times d1) + (c2 \times d2) + (c3 \times d3) + (c4 \times d4)] \times c1$

$$\Rightarrow (\underbrace{C1 \times C1}_{=4} \times d1) + (\overbrace{C1 \times C2}^{=0} \times d2) + (\overbrace{C1 \times C3}^{=0} \times d3) + (\underbrace{C1 \times C4}_{=0} \times d4)$$

$$\Rightarrow \boxed{4d1} \quad \text{Ans}$$

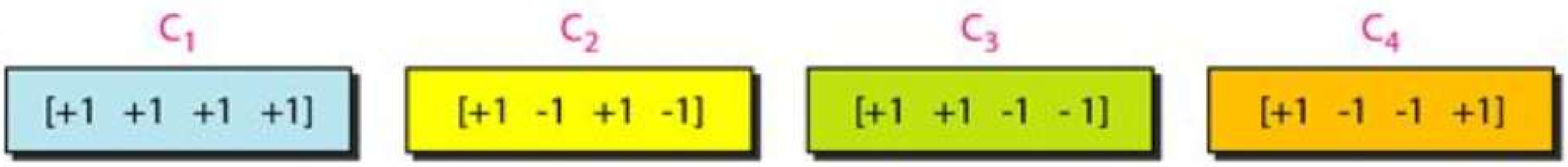
When ~~stat~~ station 2 would hear it then it will divide the obtained data by number of stations i.e. 4.

$$\therefore \frac{4 \times d1}{4} = d1$$

it is the data transmitted by station 1.

Chips:

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



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,

$$2 \cdot [+1\ +1 -1 -1] = [+2 +2 -2 -2]$$

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]

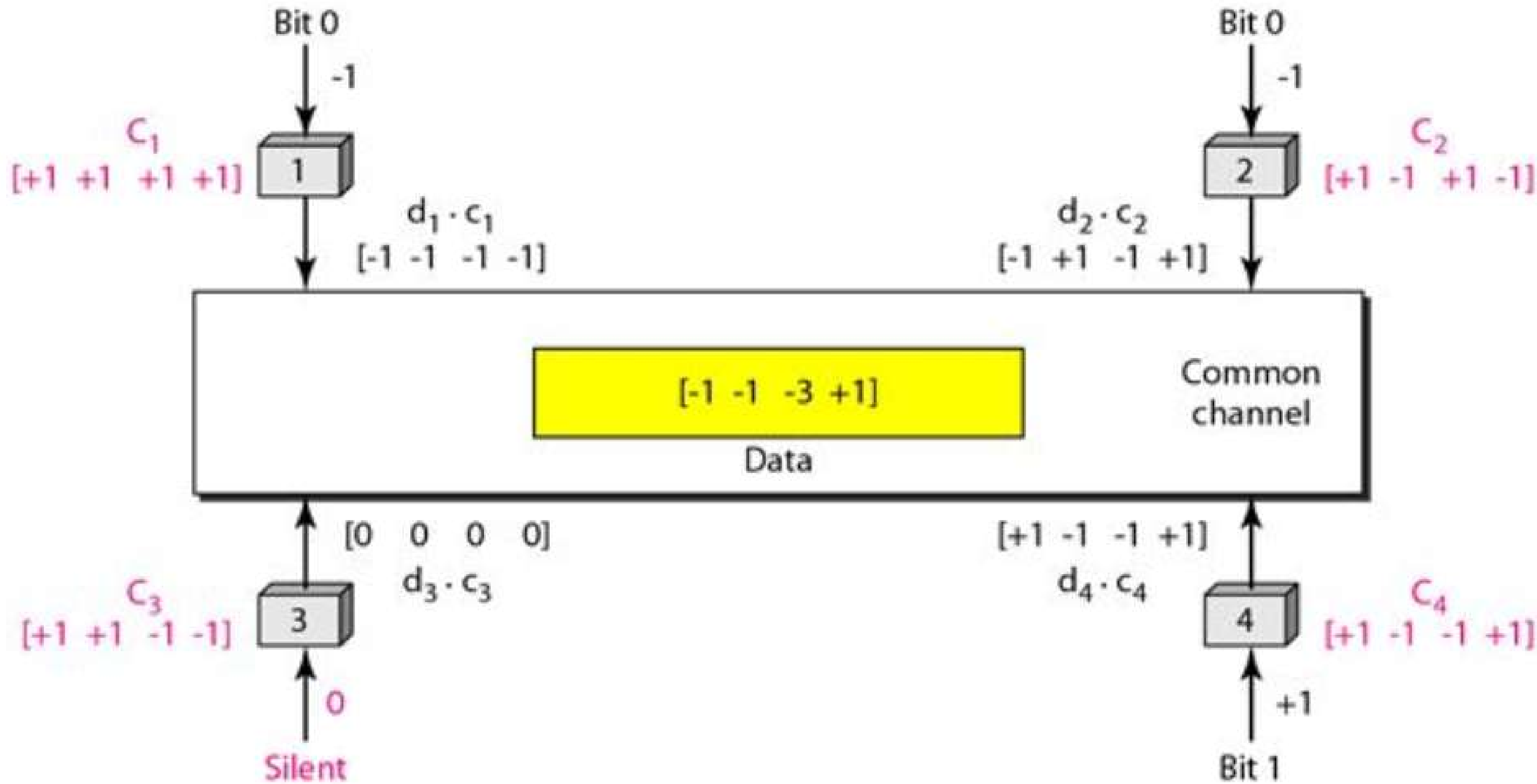
Data Representation:

We follow the following 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.

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. The following figure 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