**Chapter 12: Multiple Access**

Table of Contents

[12.1 Random Access Protocols 6](#_Toc63258636)

[ALOHA Protocol 6](#_Toc63258637)

[Carrier-Sense Multiple Access 10](#_Toc63258638)

[Carrier-Sense Multiple Access with Collision Detection 12](#_Toc63258639)

[Carrier-Sense Multiple Access with Collision Avoidance 14](#_Toc63258640)

[12.2 Controlled Access 15](#_Toc63258641)

[Reservation 15](#_Toc63258642)

[Polling 15](#_Toc63258643)

[Token Passing 16](#_Toc63258644)

[12.3 Channelization 19](#_Toc63258645)

[Frequency Division Multiple Access 19](#_Toc63258646)

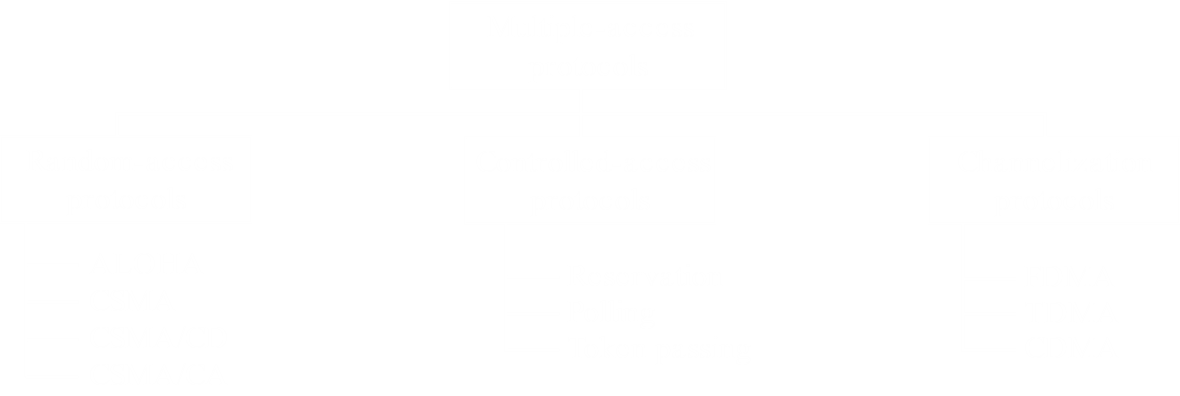
[Time Division Multiple Access 20](#_Toc63258647)

[Code Division Multiple Access 21](#_Toc63258648)

As we discussed earlier, the data-link layer can be subdivided into two parts. The upper layer is called the data link control or logical link control (LLC), which we discussed in the previous chapter. The lower layer is called medium access control (MAC).

When there are multiple users trying to access the same shared resource, there needs to be some technique to organize who will access which resources when. Otherwise, frames from different users will collide. MAC protocols are needed to perform this regulation.

The taxonomy of the different MAC protocols is given below. We will be studying a brief overview of these protocols for now. The details are left for a future course.



The most important of these are the random-access protocols. The other two are not used for medium access control in layer 2, but for other purposes.

* Channelization Protocols – These deal with how users are given different channels to communicate.
  + Frequency Division Multiple Access (FDMA) – Similar to what we saw with multiplexing, the users are given a portion of the total bandwidth available to use the resource. The division is handled centrally. This procedure is used for mobile carriers. Users are able to use the portion allocated to them for the entire duration of their communication. Once the allocated portion of frequency is not being used anymore, it is returned to the central control so it can be reallocated to another user.
  + Time Division Multiple Access (TDMA) – Users get the entire bandwidth of the resource, but only for a portion of the time. Thus, every user is able to use the resource in a round robin fashion. The loop goes so fast that the end user is unable to tell the difference. Mobile carriers actually use a hybrid of TDMA and FDMA, where groups of users are divided using TDMA and between those groups, individual users are divided using FDMA.
  + Code Division Multiple Access (CDMA) – This is a technique used by the CDMA standard, which is completely different from the more common Global System for Mobile Communication (GSM) standard. Every user is given a specific chip that comes pre-installed with the hardware that they are using. This chip is used to encode any data that needs to be communicated. The data itself is thrown into a common medium available to all users. Only the intended user is able to decode the data that is meant for them using their own chip. The shortcoming of the CDMA technique is that as the number of users increases, it is difficult to upscale since the technique relies on hardware.
* Controlled-Access Protocols – These protocols work with ‘master-slave’ (really need a better name for this man) relationships, where there is a controlling authority that controls the entire communication. There are two problems with these techniques. Firstly, if the controlling authority fails, the entire network goes down. Secondly, as the number of users increases, the authority will become way too busy and will be unable to serve everyone.
  + Reservation – Users can make reservations to use the resources.
  + Polling – Users are polled on whether or not they have data to send or whether or not they will be able to receive data that is waiting for them.
  + Token Passing – A token passes through all the users. Whichever user has the token at a given moment is allowed to transmit data.
* Random Access Protocols – In both the above categories, there was a central controller who dictated when we could communicate and when we could not. At a personal level, this is not how things work. If we are making a call, we do not think about whether or not we are scheduled to call at this particular moment or whether it is our turn to use the network. Data comes when data comes. Random access protocols allow this. There is still some scheduling going on, but things move so fast that end users do not feel the difference.
  + ALOHA – Every end user device is thought to be deaf and dumb, meaning we cannot say anything to them and they cannot say anything to us in order to facilitate an arrangement for data transmission. Instead, they just send the data whenever they have the data. If they get lucky, there are empty slots and the data is successfully received by the intended receiver. If they are not lucky, there are collisions and the packet is not sent successfully. It has been determined that the success rate for pure ALOHA is 18%, meaning out of every 100 packets, only 18 are successfully received by the receiver.

To improve upon this, slotted ALOHA was introduced. The total time was divided into slots and end user devices were allowed to send data at the beginning of each slot. So, for example, if the time slots are of 5 minutes and someone wants to send data at 09:03, they have to wait till 09:05 to send it. In reality, the time slots are of microseconds, so users do not notice the difference. Slotted ALOHA has a success rate of 36%.

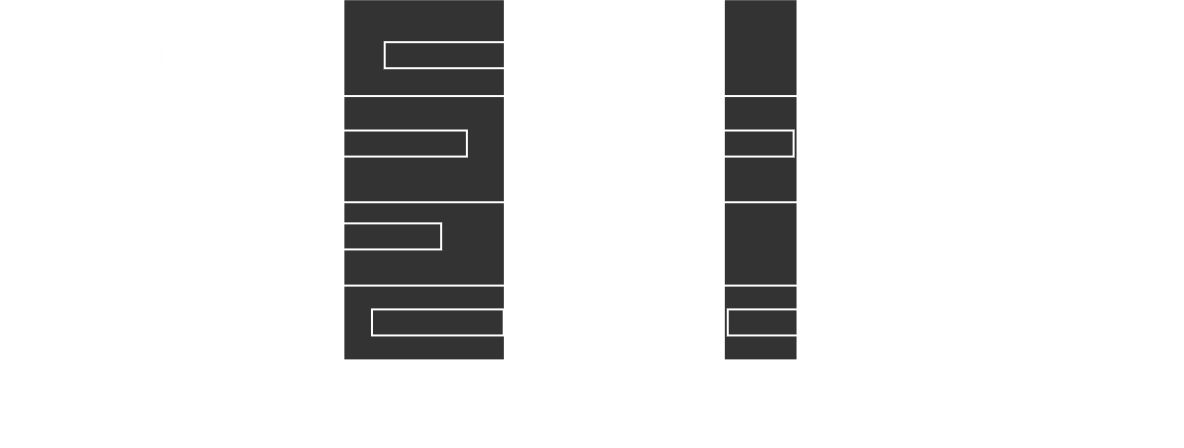
* + Carrier Sense Multiple Access (CSMA) – The fundamental concept here is that users will check to see if the medium is free before transmitting data. This solved lots of problems. However, it was still not possible to detect collisions. It is possible that two different users transmitted data at the same time, both thinking the medium is free, which leads to collisions.
  + Carrier Sense Multiple Access with Collision Detection (CSMA/CD) – This is used in wired LAN.
  + Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) – This is used in wireless LAN.

## 12.1 Random Access Protocols

The main point of random-access protocols is that at any time any station can transmit data. There is no controlling authority. This reflects real life accurately, since in real life, anyone can have data to transmit at any time and will expect that they will be able to transmit then and there. Random-access protocols are also called contention-based methods, since stations essentially fight for control of the transmission medium.

If we have dedicated channels between every single pair of stations, then there would be no access conflicts, or collisions, at all and we would not have any need for protocols. However, this is not practical. In reality, we have to share channels of transmission. This means there is always the possibility that there will be collisions when two or more stations try to transmit data simultaneously. Specifically, if even a single bit from the data being transmitted by one station overlaps with a single bit from the data being transmitted by another station, it is considered to be a collision and neither transmission is successful. This is where protocols come in.

### ALOHA Protocol



The ALOHA protocol is the simplest random-access protocol. Under this protocol, the stations rely purely on luck. Stations transmit data whenever they want and if the receiver receives the data properly, an acknowledgement is sent in return. Otherwise, if an acknowledgement is not received within a predefined amount of time, the data is retransmitted. However, this method is extremely unreliable, with only an success rate.

There are four fundamental questions that each station needs to answer if they want to avoid access conflicts:

1. When can the station access the medium?
2. What can the station do if the medium is busy?
3. How can the station determine the success or failure of the transmission?
4. What can the station do if there is an access conflict?

ALOHAs answers to these questions are less than pleasant, since it simply does whatever it wants and retransmits data if there is a collision.

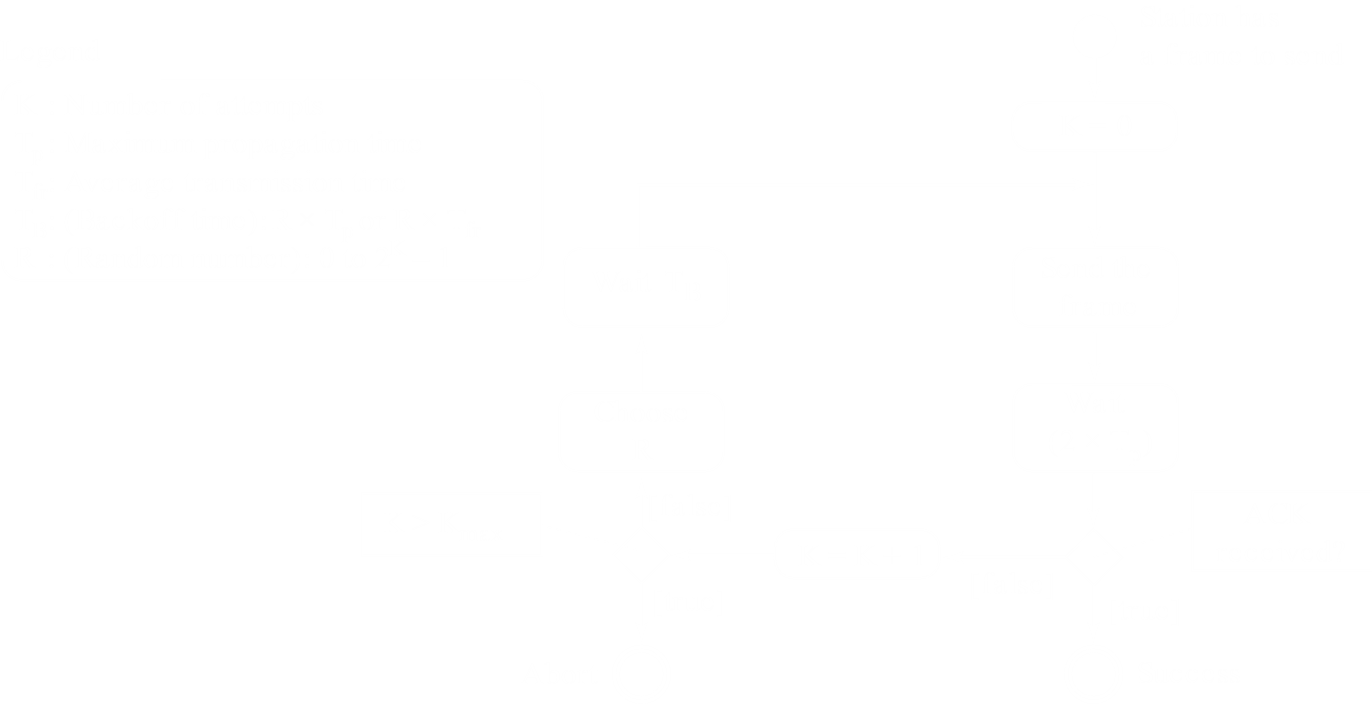
#### Back-Off Strategy

Imagine that multiple stations try to transmit data at the same time, which will obviously result in a collision. Each station has their own timer. If every station waits the same amount of time before retransmitting, it will just result in another collision and the cycle will continue.

To resolve this, the stations are made to wait a random amount of time before attempting retransmission. In this way, repeated collisions are avoided. This strategy is called ‘back-off’. The strategy is not exclusive to ALOHA, but is used in other random-access protocols as well.

The most common back-off strategy is called Binary Exponential Back-Off. In this strategy, the range of possible wait times doubles every time there is a collision.

#### Procedure for ALOHA



The flowchart above describes the basic procedure for the ALOHA protocol. Specifically, this is for pure ALOHA, which was the original protocol. Later on, a modified version, slotted ALOHA, was invented, which we will be looking into later.

We are using a variable, , to keep track of the number of attempts made to send some data. There is a maximum limit to the number of attempts that will be made, . Normally, .

Once a frame is sent, there is a time-out period for which the sender waits to receive an ACK. This time is given as , where is the propagation time, the time needed to travel from the sender to the receiver. In reality, the time-out period will be a little longer since it will take processing times into account, but for simplicity, we are ignoring this for now.

If an ACK is not received within the time-out period, we will retransmit the frame. Before doing this, we will wait some random amount of time, , due to the previously discussed back-off strategy. This time is determined by the value of . We increment and choose a random number, , between and . This is Binary Exponential Back-Off. The value of is determined by the formula , or ( is the frame transmission time, which is essentially the same thing). Thus, the back-off time will be some multiple of the propagation time.

For the first collision, the value of can be or , since . This gives us a rather poor randomness, since the station will either retransmit immediately or will wait for a time period . If there is a second collision, and can be , , or (notice how the range doubled). This leads to better randomness. As keeps increasing, we will get better and better randomness, which will increase the chances that further collisions will be avoided.

#### Vulnerable Time

In the above scenarios, we have simply assumed that two frames from two sources will be transmitted at the same time, will collide, and both will wait a random amount of time before transmitting again. Since it takes a time of for the time-out period to end, this will result in the loss of a total of time . Sadly, this is the best-case scenario.

Consider that one source is sending a frame and is nearly done sending. There is literally one bit left. At that moment another source begins sending a frame and that one-bit collides. This is the worst-case scenario. This will result in us losing the time taken for the first transmission and for the second transmission, since only after a time period of will the second source realize a collision occurred, and the first source cannot retransmit during this time. Thus, we lose a total of time .

This worst-case time period is the time in which a collision might occur. It is called the vulnerable time. For this entire time, nothing else can be sent and the data that was sent must be retransmitted, meaning the time is wasted. The ALOHA protocol has the highest vulnerable time.

#### Slotted ALOHA

The concept of slotted ALOHA initially came up when people tried to reduce the vulnerable time. It was decided that stations should be given slots to transmit data. This does not refer to individual stations being allowed to transmit at specific times, like in time-division multiplexing. Instead, there are specific start times for each slot. Say every is a slot. Every station is allowed to transmit data at the beginning of a slot. Thus, even if a station has data at say, , it will have to wait till to transmit. If there is data at , the station has to wait till . This time period is actually far lower, to an extent that end-users will not notice it.

The concept of slotted ALOHA helps a lot with vulnerable time. Since each station is given a start time, it is not possible that another station will start transmitting just as one station is about to finish transmitting. Thus, the worst-case collisions will not occur. It is only possible that two stations will start transmitting at the same time. This means, at most, we will lose a time of . This is the vulnerable time for slotted ALOHA.

### Carrier-Sense Multiple Access

In carrier-sense multiple access (CSMA), the basic concept is that the sender should check to see whether the medium is free before sending data. However, collisions can still occur. Perhaps the part of the medium which was sensed was free, but once transmission begins, frames collide in some other part of the medium.

Just the fact that the medium is being sensed made this protocol a huge advancement over the previous techniques. It also has a vulnerable time equal to the propagation time.

CSMA in itself was not implemented well. Instead, two derivations of it, CSMA-CA and CSMA-CD, were more widely adopted.

#### Persistence Methods

The procedures used by stations to sense the medium are called Persistence Methods. There are three methods:

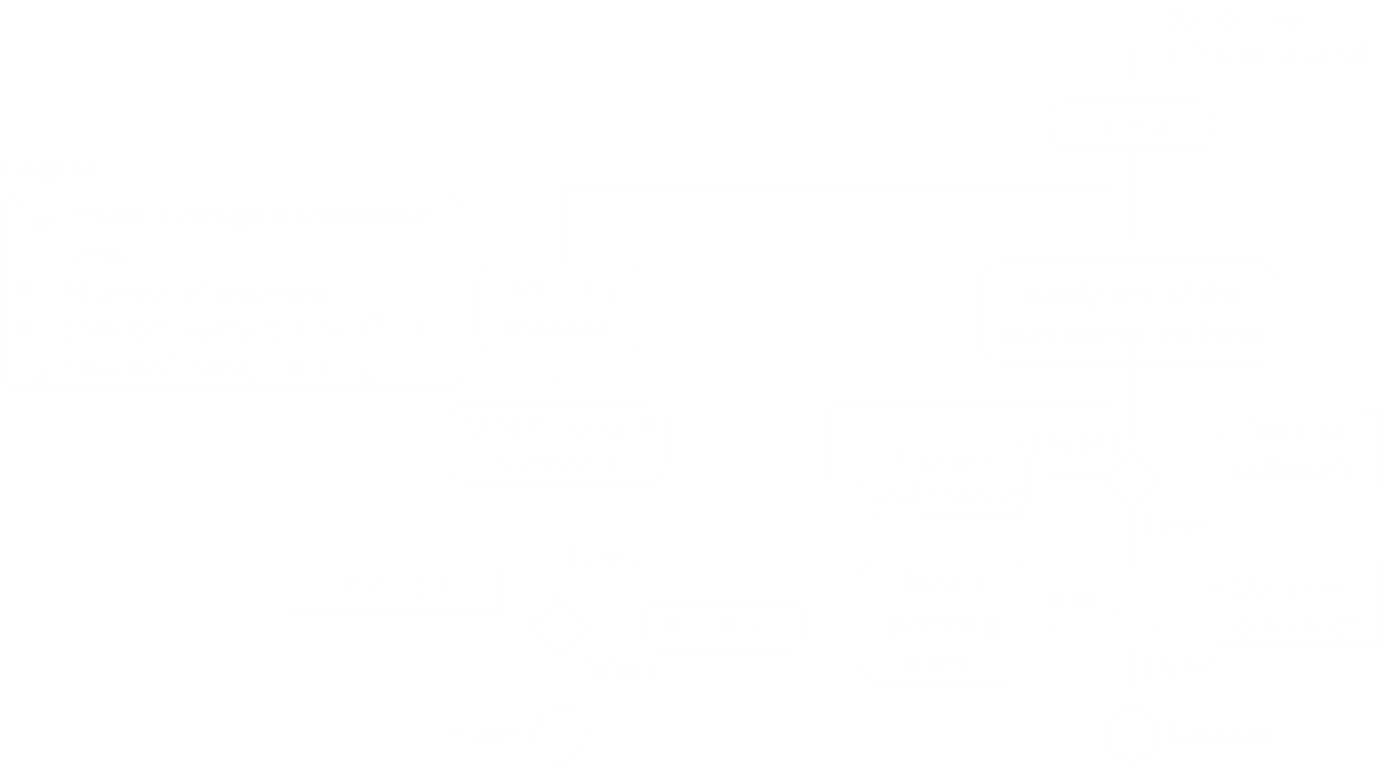
* **1-Persistent Method** – The station continuously senses the medium and sends data as soon as it finds the medium free.
* **Non-Persistent Method** – The station senses the medium, and if it detects that the medium is busy, it waits a random amount of time before sensing the medium again. Consecutive senses wait for longer and longer periods of time. Using this method causes some extra delay since the exact moment when the medium becomes free might not be detected.
* **-Persistent Method** – This method is a combination of the previous two. It continuously senses whether the medium is free, but it does not necessarily transmit as soon as the medium becomes free. Instead, whether or not a transmission will take place depends on probability.

We will transmit immediately with a predetermined probability , and with a probability , we will wait a certain random amount of time. After that amount of time, if the medium is still free, then we will again transmit with a probability and with a probability , we will wait for a random amount of time. Consecutive wait times follow the binary exponential back-off strategy. If the medium is not free when it is sensed, we abandon probability theory and just use the back-off strategy as though a collision occurred.

The part about probability is like picking a random number between and . Say . Thus, if the random number picked is between and , we will transmit and if it is not then we will wait.

### Carrier-Sense Multiple Access with Collision Detection

In CSMA, we had no clue exactly when a collision occurred. CSMA with Collision Detection (CSMA-CD) solves this. The advantage is that, since the station immediately gets to know that a collision has occurred, it can immediately stop the transmission, thus decreasing the time lost. Additionally, the station also sends a small jamming signal telling any receivers to ignore the previous, partially sent frame.



The diagram above shows how the process takes place. It should be simple enough to understand since it is mostly the same as previous methods. The only different part is the collision detection section.

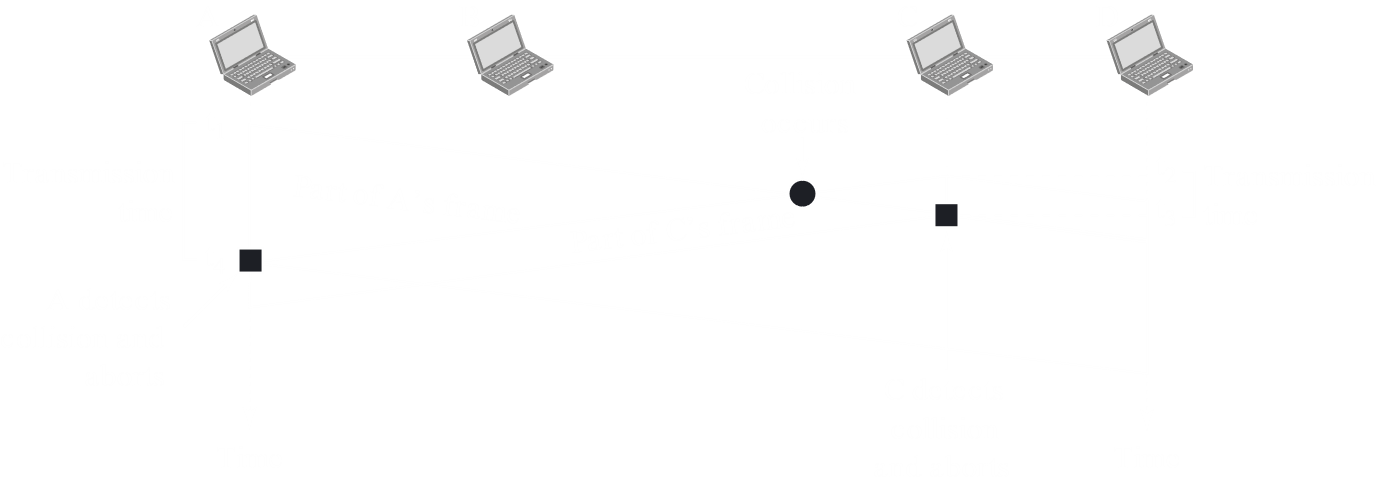
#### Detecting Collisions

Every device has both a transmission and a detection port, both of which work together. The device transmitting data also detects the same data from the transmission medium. If it detects something that it did not transmit, the device realizes that there was a collision.

The actual detection takes place with the help of voltage signals. If the data being transmitted has a total voltage of say , and some collision occurs, the energy in the medium will increase. The increased voltage is what indicates a collision.

CSMA/CD works well with wired LAN because of this. In wired LAN, the voltage in the wires is more or less constant all the time. Any changes are thus immediately detected. In wireless mediums, the voltage changes due to attenuation, so changes due to collisions are harder to detect.

#### Frame Sizes



Consider the diagram above. started transmitting at and started transmitting at . There was a collision somewhere in between. This collision was detected by at . Since the collision took place closer to , was able to detect it first. Thus, transmitted from to . However, since is further away, will only get to know about the collision at .

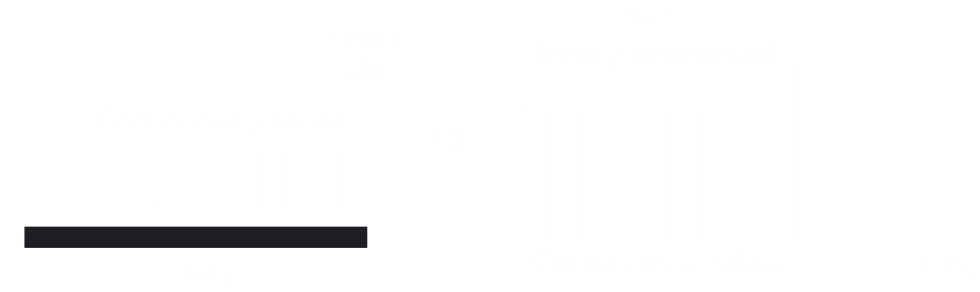
The problem now is that might not be transmitting data anymore at . Perhaps the frame it was sending out was small and it finished sending it out and thus stopped detecting by . Due to this problem, we need to put a restriction on the frame size.

Essentially, the frame size of the data being transmitted needs to be large enough that the transmission occurs for at least twice the propagation time, so it has enough time to send a signal to the other end and also detect any collision. If the actual data is not large enough, padding is added.

### Carrier-Sense Multiple Access with Collision Avoidance

As mentioned before, the techniques used by CSMA/CD are less successful with wireless transmission mediums. We only looked at the problem with detecting energy levels, but there are more reasons as well. Due to such problems, Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) was developed.

The main goal in CSMA/CA is to avoid collisions in the first place. A device will continuously sense the transmission medium to see if it is idle. Once it detects that the transmission medium is idle, it will wait for some period of time, called the interframe space, in case some frame that is far away is actually using the transmission medium. After this, as a precaution, it will wait for an additional small amount of time, called the contention window. Finally, it will send the frame and then wait to receive an acknowledgement.

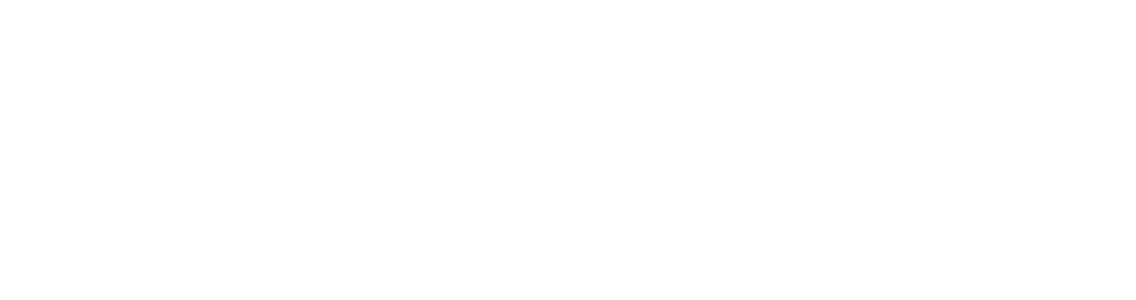


## 12.2 Controlled Access

In controlled access, stations consult one another to check which station has the right to send data. No one can send data without authorization.

### Reservation

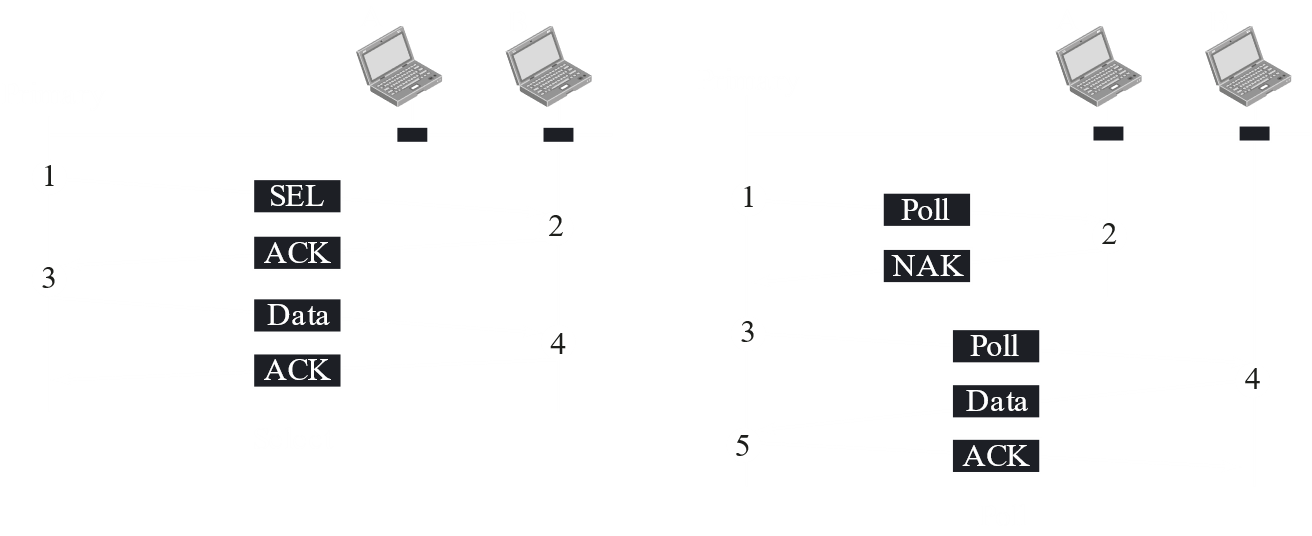
In this method, a station needs to make a reservation before sending data. In each interval, a reservation frame precedes the data frames sent in that interval. The reservation frame has mini-slots for stations in the system. If a station wants to send data, it must make a reservation in its mini-slot. The stations that have made reservations can send their data frames after the reservation frame.



### Polling

Polling works with topologies where one station is designated the primary station and the rest are designated the secondary stations. The primary station controls the link and gives instructions to the secondary stations.

If the primary station wants to receive data, it asks the secondary stations if they have any data to send. This is called the Poll function. If the primary station wants to send data, it tells the relevant secondary station to get ready to receive data. This is called the Select function.



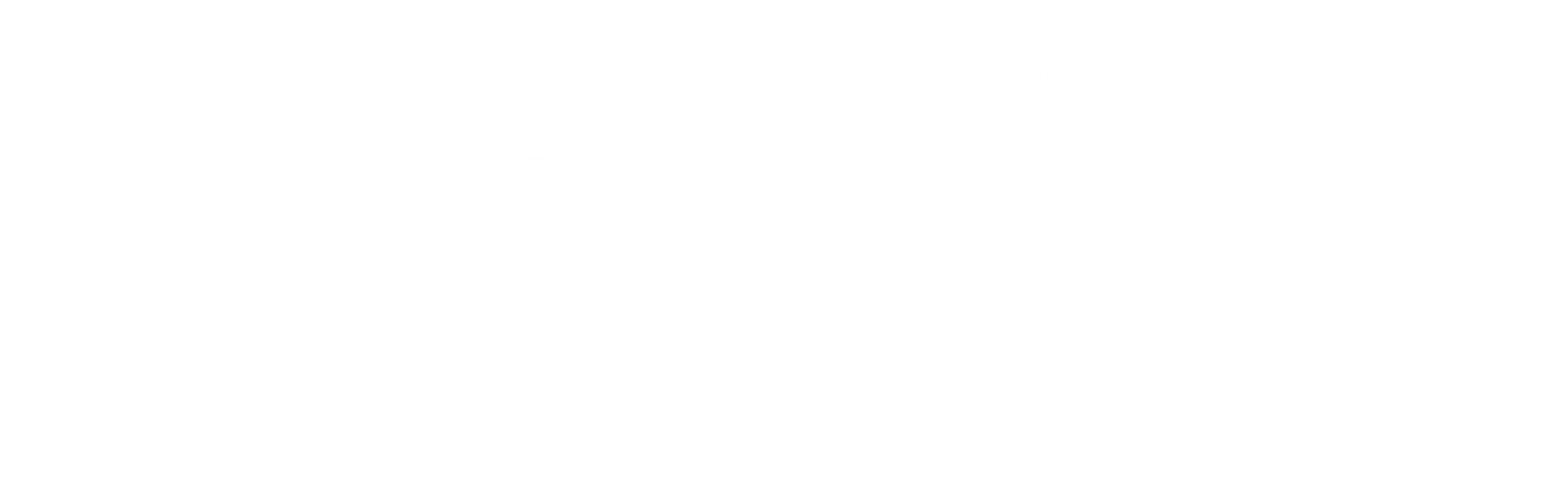
As can be seen in the diagram, the Select function requires an ACK in return, after which the data is sent, which in turn also needs an ACK. The Poll function needs a NAK if no data is available. If there is data, the data should be sent, and the primary station will send an ACK when the data is received.

Note that two secondary devices can still communicate, but they must do so through the primary device.

A major drawback to this method is if the primary device is down, the system is disabled.

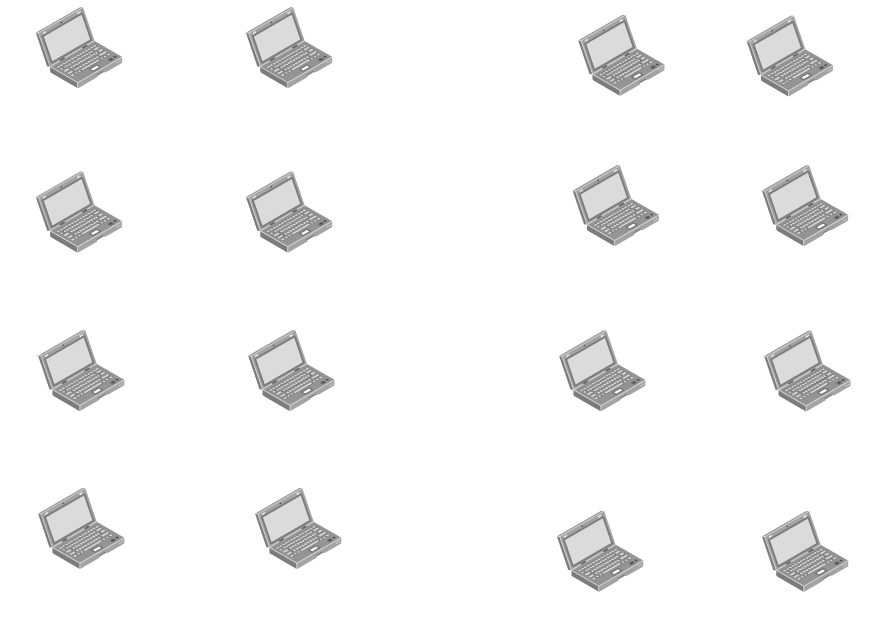
### Token Passing

In Token Passing, the devices are set up in a ring topology and a token, which is a special packet, is passed between them. Whoever has the token is allowed to use the transmission medium. The token is passed on when one device is done sending data.



Token management is also required so as to ensure one station does not take too long with the token and that the token is not lost or destroyed, perhaps because the device holding it failed. Another function of token management is to maintain priority with regards to stations and types of data.

The ring itself can take several forms. It need not be a physically connected ring. A logical ring is acceptable.



In a physical ring, the token is sent to the next station in line. Since the devices are physically connected, other devices on the network cannot see the token. The token does not need an address either, since there is only one place it can go. The problem however, is that if even one of the links fail, the entire system fails.

In a dual ring system, the second ring, called the auxiliary ring, is used in case of emergencies. If one of the links in the main ring fails, the secondary ring is used to create a temporary ring. For this setup to work, each device must have two transmitter and two receiver ports.

In a bus ring topology, which is also called a token bus, the devices are connected to a single bus. Each station knows the address of its successor and the devices create a logical (not real) ring. When the token is being passed, it is sent along with the address of the successor. Only the station that has that address is able to receive the token and gain access to the medium.

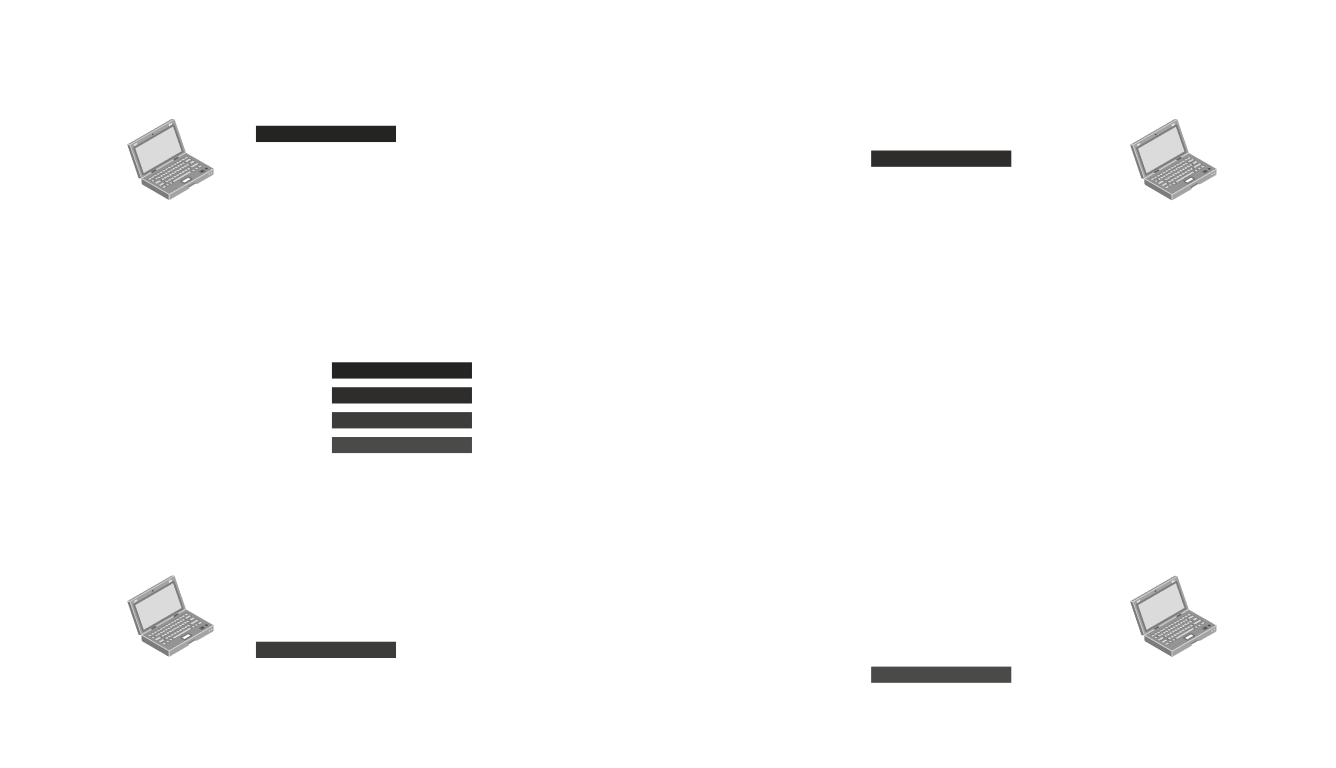
In a star ring topology, there is a hub that acts as a connection between the devices. This setup is advantageous in that if one of the devices fails, the hub can just bypass it. Additionally, adding and removing stations is much easier.

## 12.3 Channelization

Channelization, or channel partition, is a multiple-access method in which the available bandwidth of a link is shared by different stations.

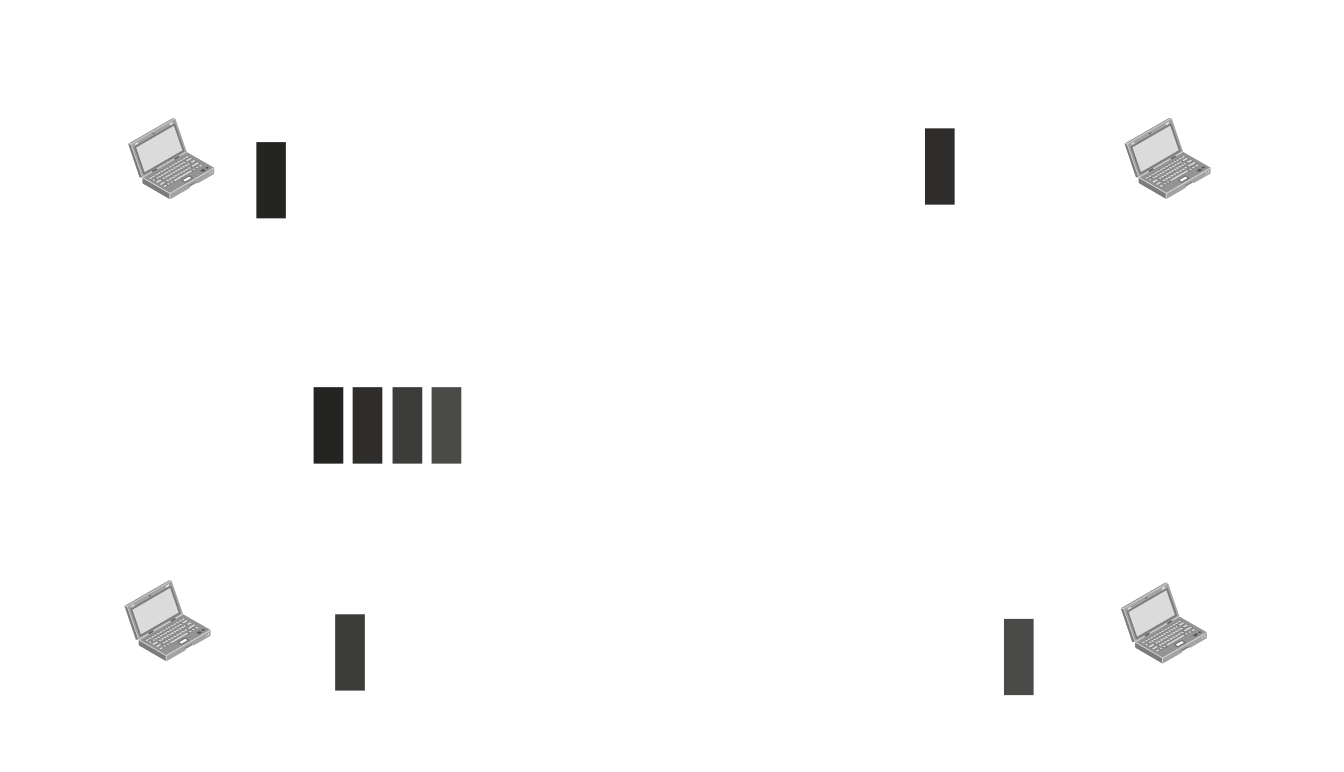
### Frequency Division Multiple Access

In Frequency Division Multiple Access (FDMA), the available bandwidth is divided amongst stations as smaller bands. These are in turn separated by guard bands so as to avoid interference. The advantage of this method is that the band allocated to a station is available for the entire time needed.



### Time Division Multiple Access

In Time Division Multiple Access, the entire bandwidth is given to one station, but for a very short period of time. In a round-robin fashion, each station gets to use the transmission medium. The advantage to this is the higher bandwidth available.



The problem with TDMA is that the sender and receiver need to be synchronized, which can be difficult due to propagation delays. To compensate for this, we could insert guard times or by using synchronization bits at the beginning of each slot.

Keep in mind that, even though FDMA and TDMA may seem similar to FDM and TDM, they are different things. FDMA and TDMA work in the data-link layer. There is no separate work for them in the actual physical layer. In the physical layer, everything just gets mixed together and sent. For FDM and TDM, we saw that the actual physical layer took part in the partitioning.

### Code Division Multiple Access

Over time, the development of chips became very easy. At that time, Code Division Multiple Access (CDMA) became popular. The basic principle of CDMA is that every device is given a unique code, preinstalled with the device. Whenever a device wants to send data, it multiplies the data with its own code and then puts it onto the transmission medium. In the actual transmission medium, all the data from all devices are simultaneously present.

If a receiving device wants to see what data was sent by the sending device, it needs to multiply the data in the transmission medium with the code for the sending device to separate out that data.

#### Coding Theory

The codes being used are also sometimes called chip sequences. The code for each device has two properties:

1. If we multiply the code for one device with the code for another device, the result is .
2. If we multiply the code for one device with itself, the result is , the number of devices in the system.

This multiplication is not just normal multiplication, but a special type of multiplication called an inner product. The sequence of the codes is also a special type of sequence called an orthogonal sequence. When we perform inner product multiplication on orthogonal sequences, the two properties above hold.

Say we have two devices with the following code sequences:

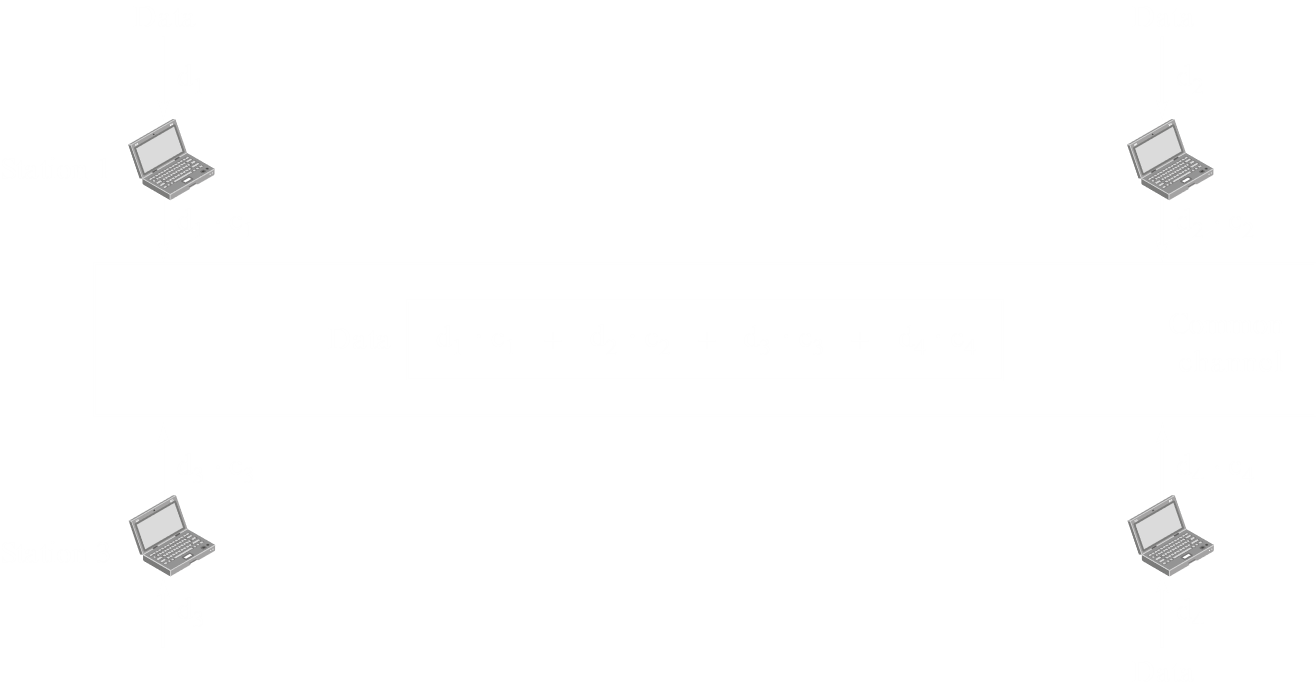
For these codes, the multiplications look like this:

For the second multiplication, the result is since we have elements in the code sequence. There are actually two more stations in this system, which is why the code sequence had elements in the first place.

Example

When sending data, a is interpreted as a bit, a is interpreted as a bit and a is interpreted as silence.

Say we have this situation:



Adding all of these, the result is

This is what is present in the transmission medium.

Now say we want to read what is. To do this, we multiply the code in the transmission medium with .

And then we divide it by , the number of elements in the sequence, which is here.

#### Sequence Generation

We generate sequences for codes using a Walsh table.

We first have to pick a code, . This can be either or , either is fine. Say we pick .

The next step is to apply the following formula:

Thus, if we want to find .

Here, each row is one code. Thus, and .

Similarly,

Thus, the number of codes is always a power of . Even if we have, say, stations, we will still need to generate codes.