Communications Technology

Dr./ Ahmed Mohamed Rabie

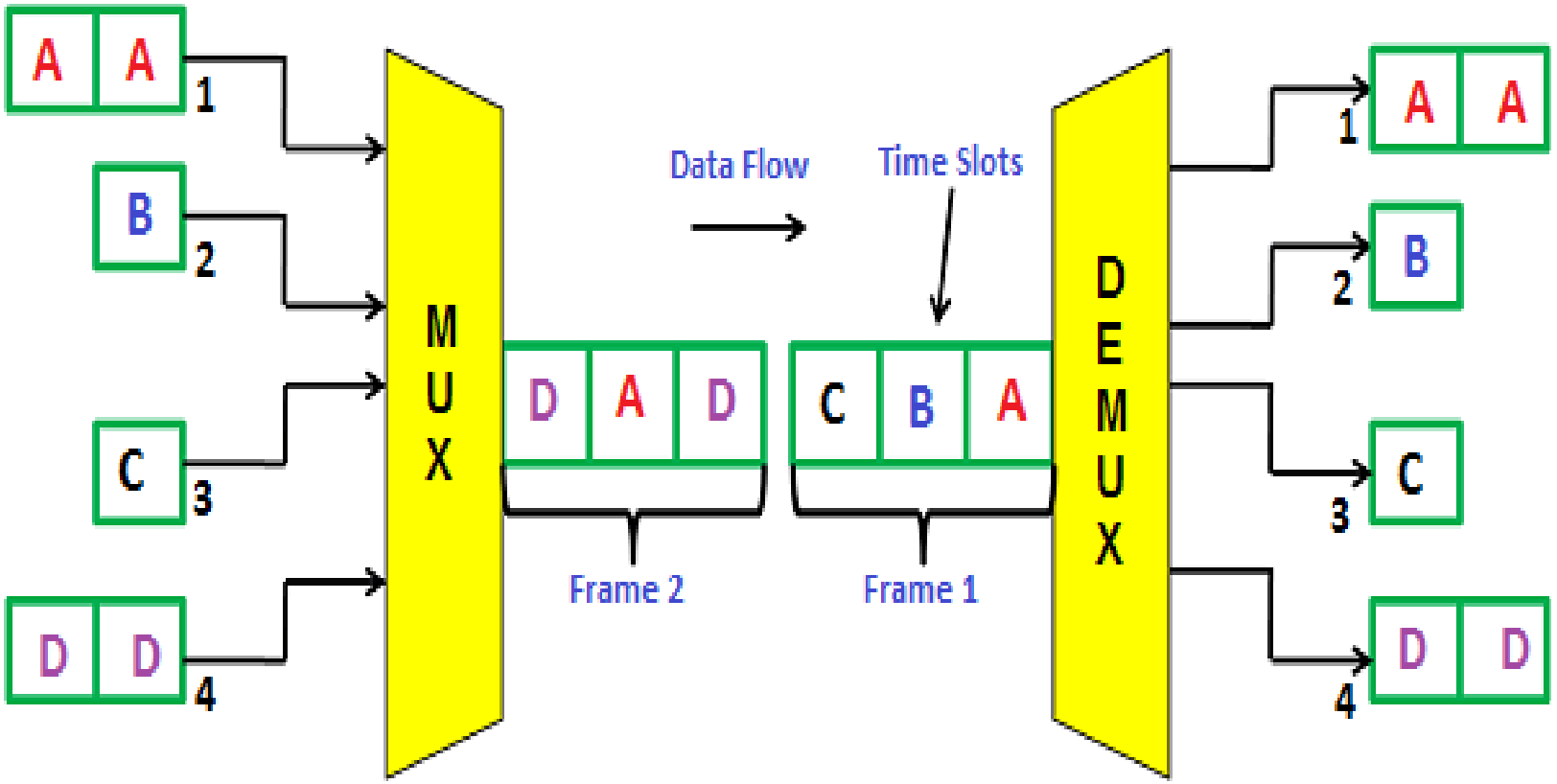
Chapter 1

Introduction

Statistical Time-Division Multiplexing

Statistical (Asynchronous) Time-Division Multiplexing: In synchronous TDM, each input has a reserved slot in the output frame. This can be inefficient if some input lines have no data to send. In statistical time-division multiplexing, slots are dynamically allocated to improve bandwidth efficiency. Only when an input line has a slot's worth of data to send is it given a slot in the output frame.

In statistical multiplexing, the number of slots in each frame is less than the number of input lines. The multiplexer checks each input line in round robin fashion; it allocates a slot for an input line if the line has data to send; otherwise, it skips the line and checks the next line.



Statistical Time Division Multiplexing

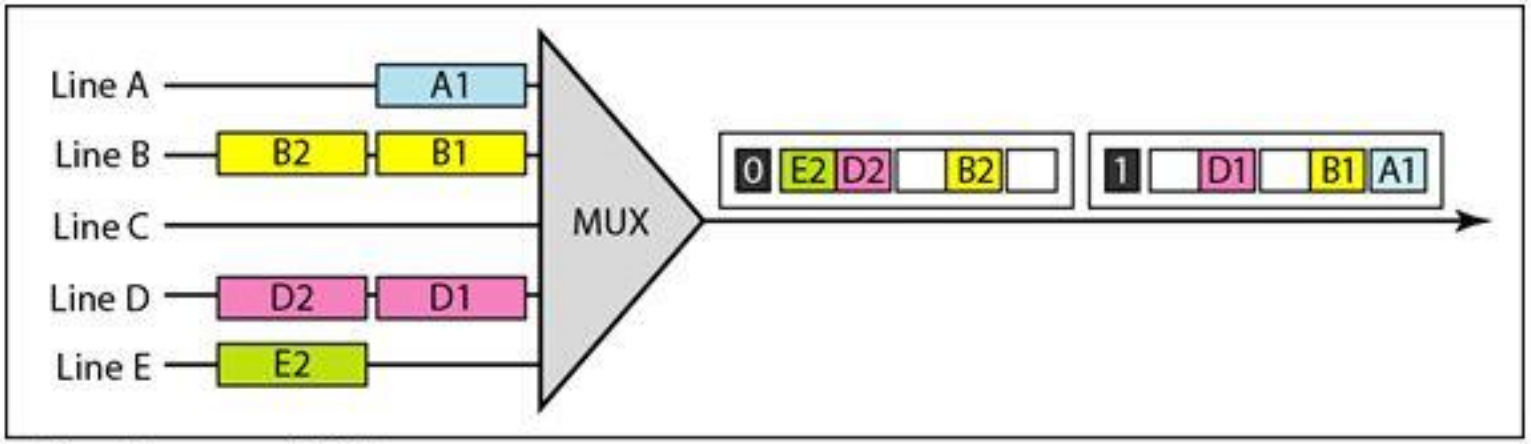
Addressing: A major difference between slots in synchronous TDM and statistical TDM. An output slot in synchronous TDM is totally occupied by data; in statistical TDM, a slot needs to carry data as well as the address of the destination. In synchronous TDM, there is no need for addressing; synchronization and preassigned relationships between the inputs and outputs serve as an address.

We know, for example, that input 1 always goes to input 2. If the multiplexer and the demultiplexer are synchronized, this is guaranteed. In statistical multiplexing, there is no fixed relationship between the inputs and outputs because there are no preassigned or reserved slots. We need to include the address of the receiver inside each slot to show where it is to be delivered.

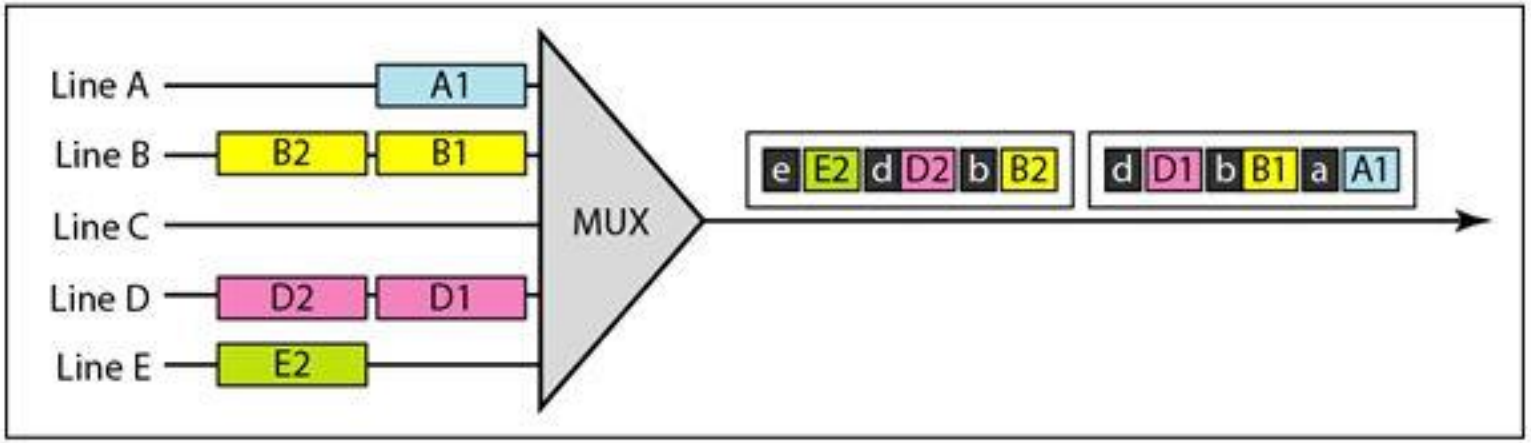
The **addressing** in its simplest form can be n bits to define N different output lines with

$n = \log_2 N = \log N / \log 2$,

For example, for eight different output lines, we need a 3-bit address.



a. Synchronous TDM



b. Statistical TDM

Slot Size: Since a slot carries both data and an address in statistical TDM, the ratio of the data size to address size must be reasonable to make transmission efficient. For example, it would be inefficient to send 1 bit per slot as data when the address is 3 bits. This would mean an overhead of 300 percent. In statistical TDM, a block of data is usually many bytes while the address is just a few bytes.

No Synchronization Bit: There is another difference between synchronous and statistical TDM, but this time it is at the frame level. The frames in statistical TDM need not be synchronized, so we do not need synchronization bits. Bandwidth In statistical TDM, the capacity of the link is normally less than the sum of the capacities of each channel.

The designers of statistical TDM define the capacity of the link based on the statistics of the load for each channel. If on average only x percent of the input slots are filled, the capacity of the link reflects this. Of course, during peak times, some slots need to wait.

Parameter	Synchronous TDM	Statistical TDM
Working	In Synchronous TDM data flow of each input connection is divided into units and each input occupies one output time slot.	In Statistical TDM slots are allotted dynamically. i.e. input line is given slots in output frame if and only if it has data to send.
No. of Slots	In Synchronous TDM no. of slots in each frame are equal to no. of input lines.	In Statistical TDM, No. of slots in each frame are less than the no. of input lines.
Buffers	Buffering is not done, frame is sent after a particular interval of time whether someone has data to send or not.	Buffering is done and only those inputs are given slots in output frame whose buffer contains data to send.
Addressing	Slots in Synchronous TDM carry data only and there is no need of addressing. Synchronization and pre assigned relationships between input and outputs that serve as an address.	Slots in Statistical TDM contain both data and address of the destination.
Synchronization	Synchronization bits are used at the beginning of each frame.	No synchronization bits are used
Capacity	Max. Bandwidth utilization if all inputs have data to send.	The capacity of link is normally is less than the sum of the capacity of each channel.
Data Separation	In Synchronous TDM de-multiplexer at receiving end decomposes each frame, discards framing bits and extracts data unit in turn. This extracted data unit from frame is then passed to destination device.	In Statistical TDM de-multiplexer at receiving end decomposes each frame by checking local address of each data unit. This extracted data unit from frame is then passed to destination device.

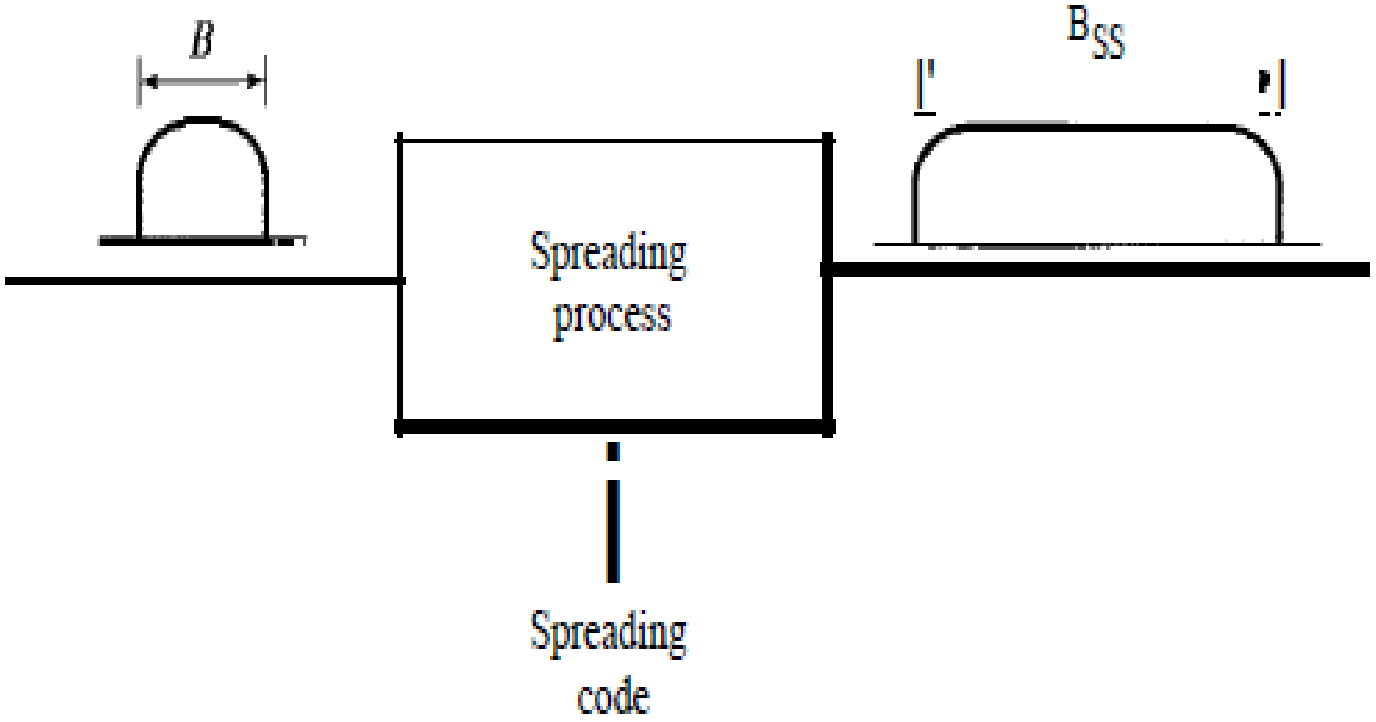
Spreading

SPREAD SPECTRUM: Multiplexing combines signals from several sources to achieve bandwidth efficiency; the available bandwidth of a link is divided between the sources. In spread spectrum, we also combine signals from different sources to fit into a larger bandwidth, but our goals are somewhat different.

Spread spectrum is designed to be used in wireless applications (LANs and WANs). In these types of applications, we have some concerns that outweigh bandwidth efficiency. In wireless applications, all stations use air (or a vacuum) as the medium for communication. Stations must be able to share this medium without interception by an eavesdropper and without being subject to jamming from a malicious intruder (in military operations, for example).

To achieve these goals, spread spectrum techniques add redundancy; they spread the original spectrum needed for each station. If the required bandwidth for each station is B, spread spectrum expands it to B_{ss}' such that $B_{ss} \gg B$. The expanded bandwidth allows the source to wrap its message in a protective envelope for a more secure transmission. An analogy is the sending of a delicate, expensive gift. We can insert the gift in a special box to prevent it from being damaged during transportation, and we can use a superior delivery service to guarantee the safety of the package.

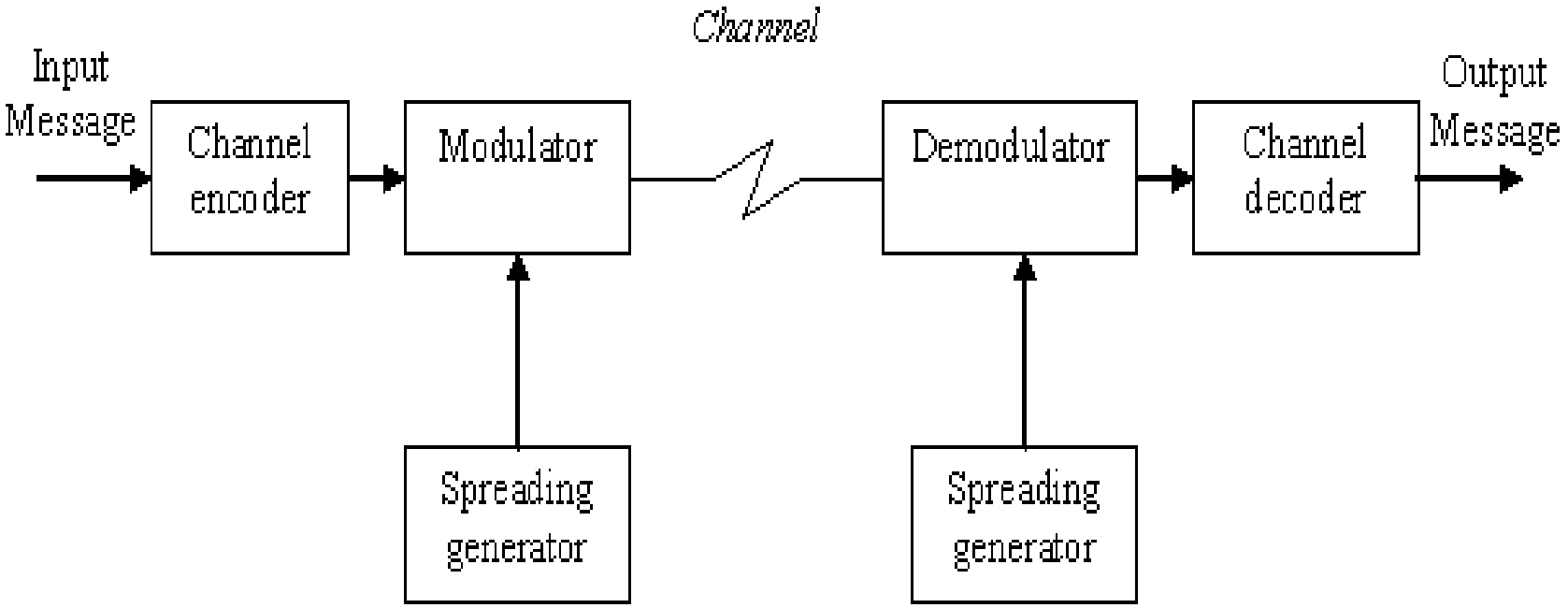
Spread spectrum



The idea of spread spectrum. **Spread spectrum achieves its goals through two principles**:

1. The bandwidth allocated to each station needs to be, by far, larger than what is needed. This allows redundancy.

2. The expanding of the original bandwidth B to the bandwidth B_{ss} must be done by a process that is independent of the original signal. In other words, the spreading process occurs after the signal is created by the source.

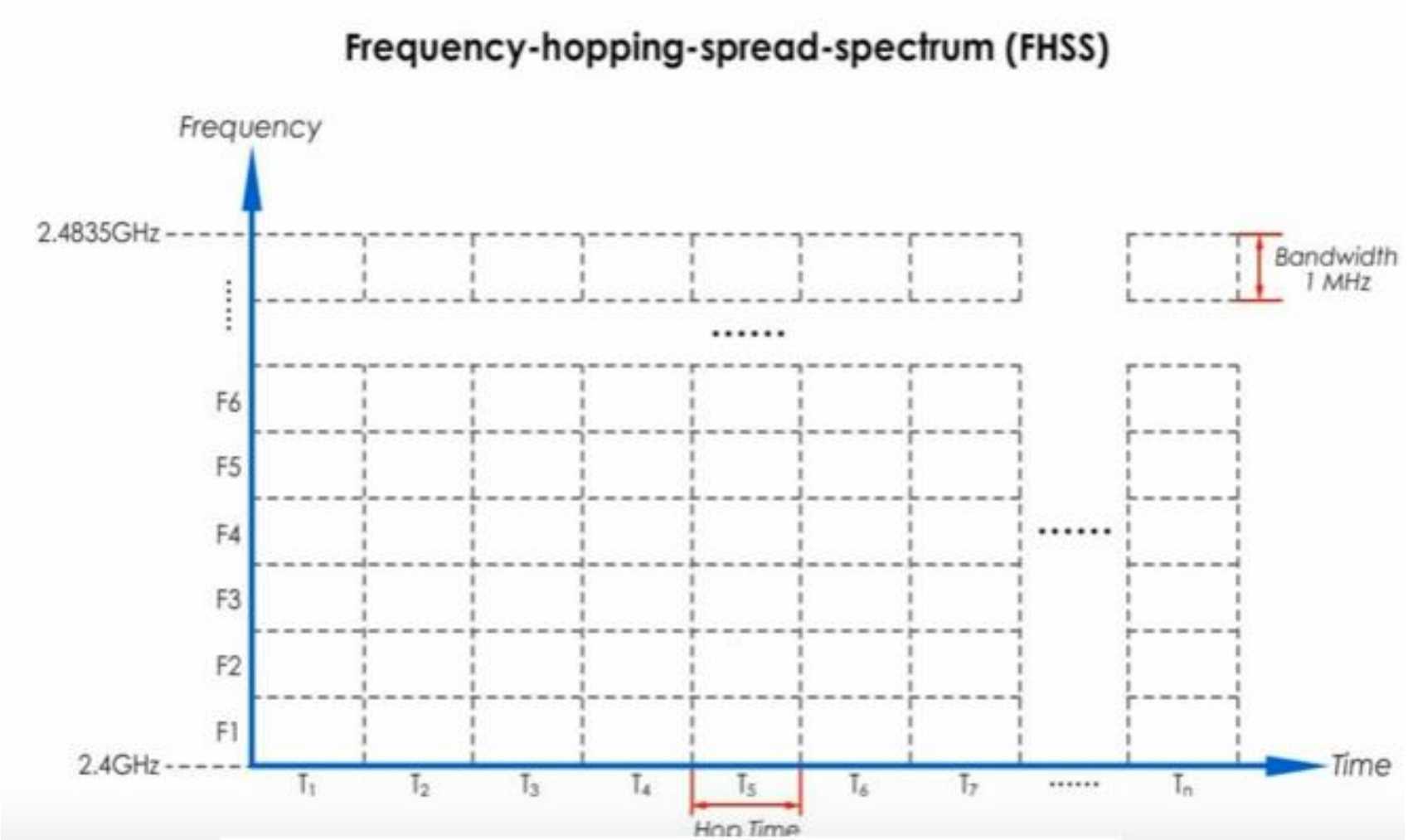


. ***Block diagram of the spread spectrum communication system***

After the signal is created by the source, the spreading process uses a spreading code and spreads the bandwidth. The figure shows the original bandwidth B and the spreaded bandwidth B_{ss}. The spreading code is a series of numbers that look random, but are actually a pattern. There are two techniques to spread the bandwidth: frequency hopping spread spectrum (FHSS) and direct sequence spread spectrum (DSSS).

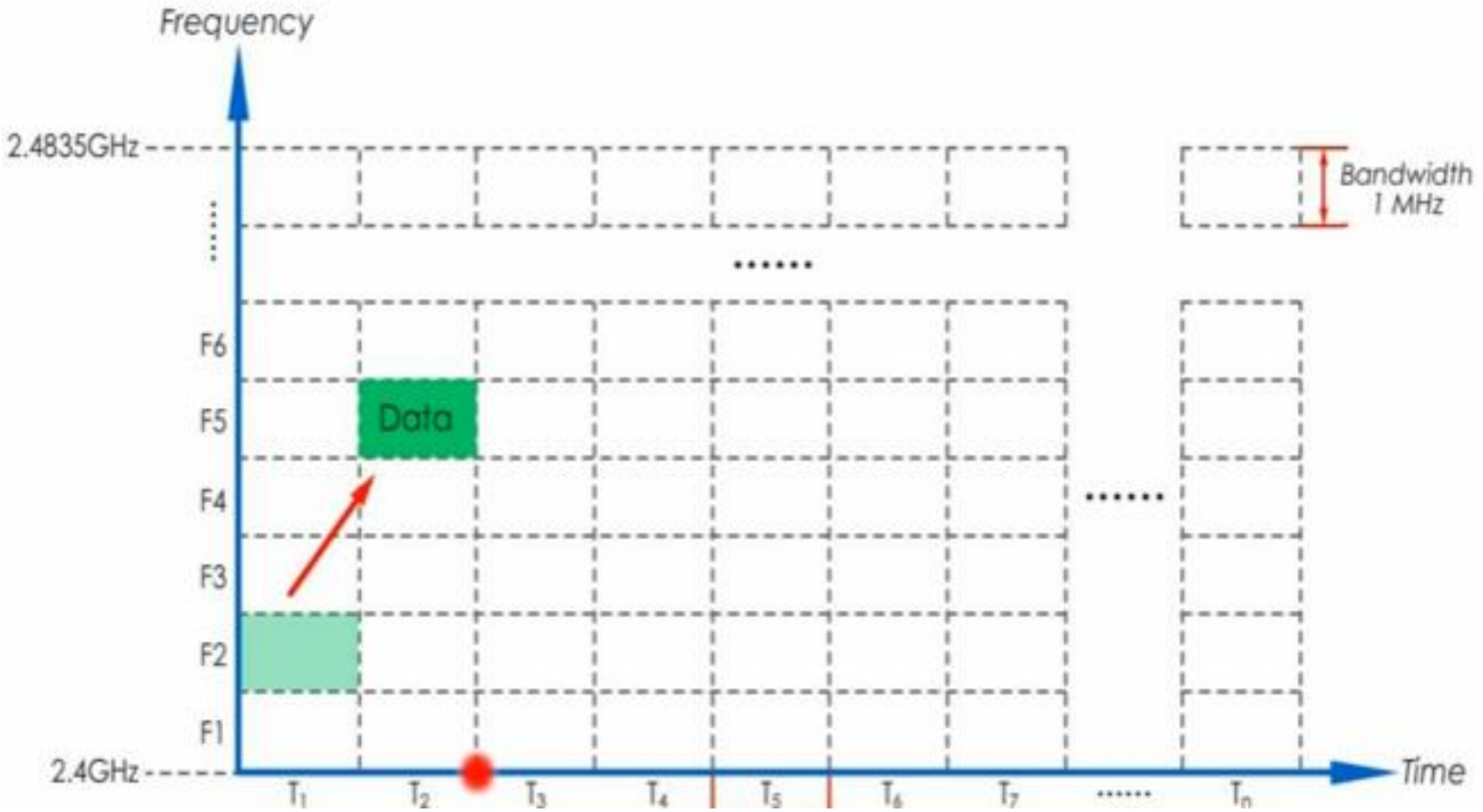
1- Frequency Hopping Spread Spectrum (FHSS): The frequency hopping spread spectrum (FHSS) technique uses M different carrier frequencies that are modulated by the source signal. At one moment, the signal modulates one carrier frequency; at the next moment, the signal modulates another carrier frequency.

Frequency-hopping-spread-spectrum (FHSS)

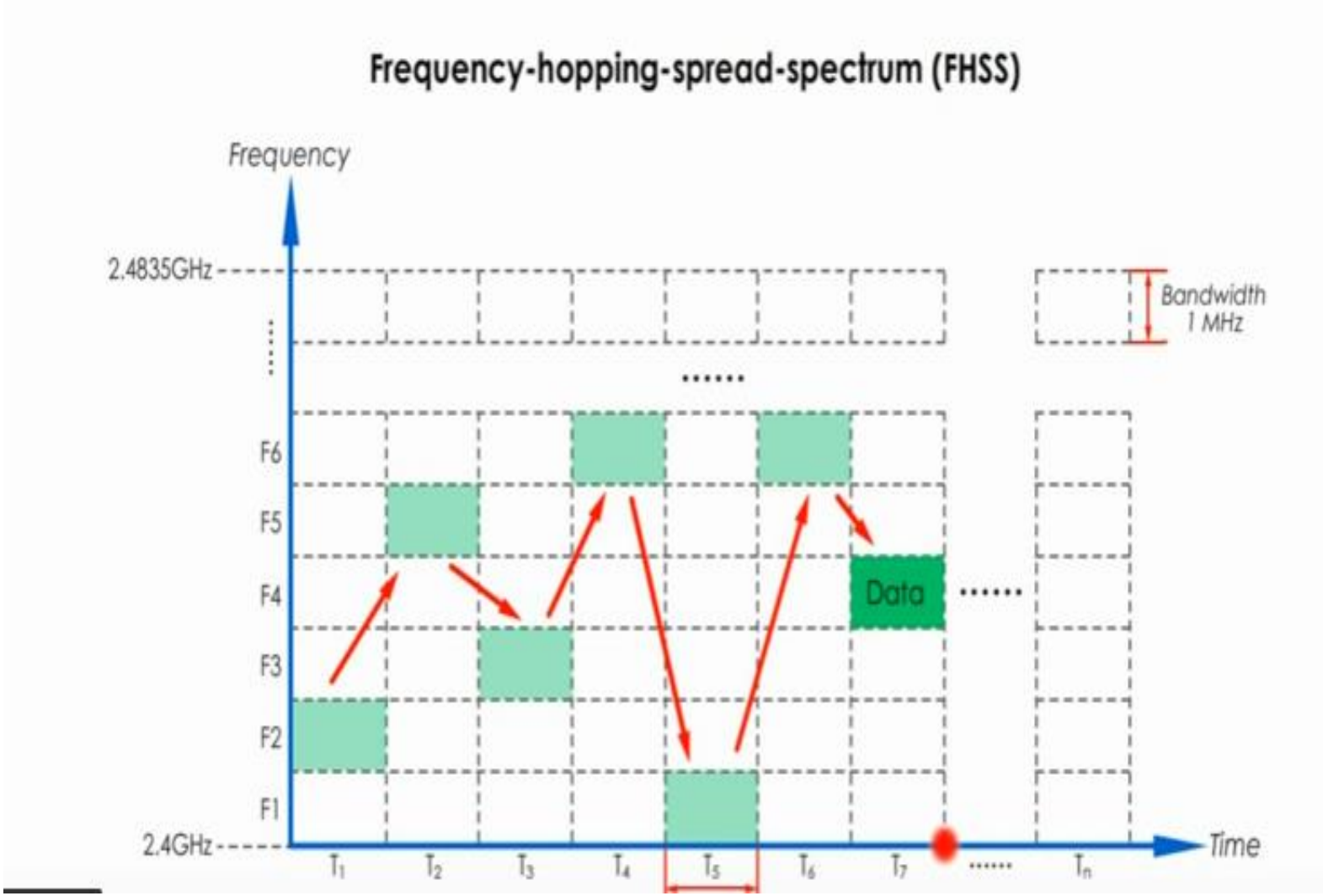


Although the modulation is done using one carrier frequency at a time, M frequencies are used in the long run. The bandwidth occupied by a source after spreading is $B_{pHSS} \gg B$. The general layout for FHSS. A pseudorandom code generator, called pseudorandom noise (PN), creates a k-bit pattern for every hopping period T_h•

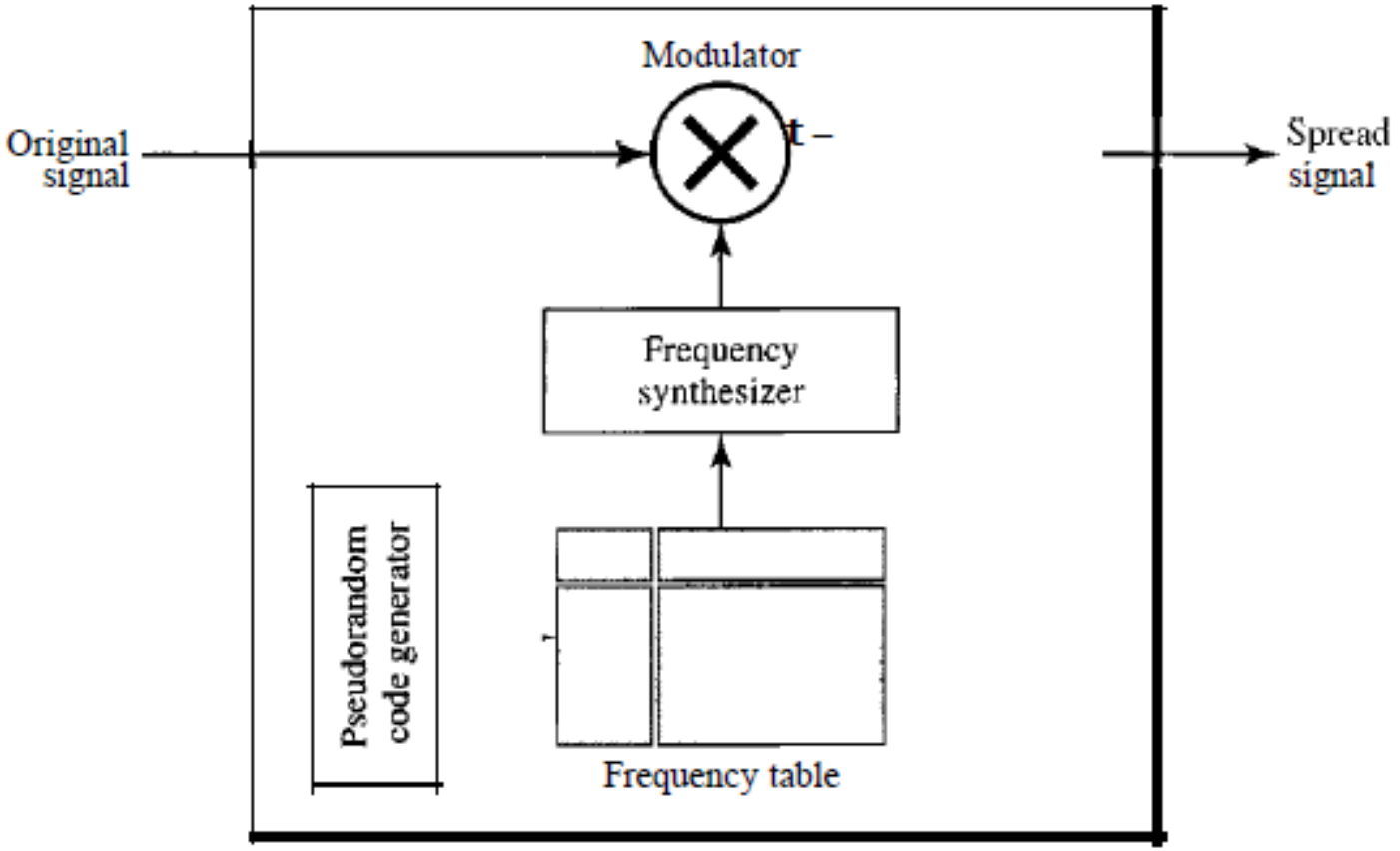
Frequency-hopping-spread-spectrum (FHSS)



The frequency table uses the pattern to find the frequency to be used for this hopping period and passes it to the frequency synthesizer. The frequency synthesizer creates a carrier signal of that frequency, and the source signal modulates the carrier signal.

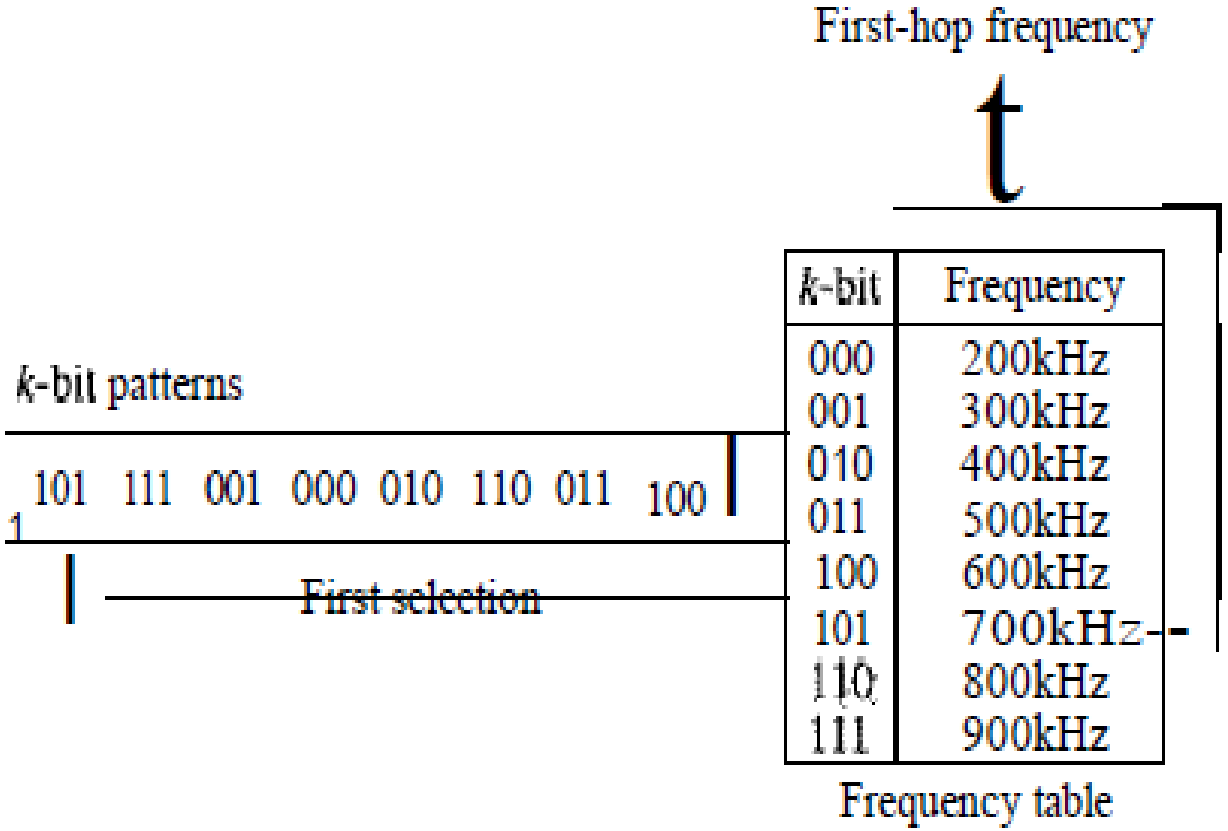

Frequency-hopping-spread-spectrum (FHSS)
Frequency
2.4835GHz
2.4GHz
F6
F5
F4
F3
F2
F1
Data
Bandwidth
1 MHz
Time
T1
T2
T3
T4
T5
T6
T7
Tn

Frequency hopping spread spectrum (FHSS)



Suppose we have decided to have eight hopping frequencies. This is extremely low for real applications and is just for illustration. In this case, M is 8 and k is 3. The pseudorandom code generator will create eight different 3-bit patterns. These are mapped to eight different frequencies in the frequency table.

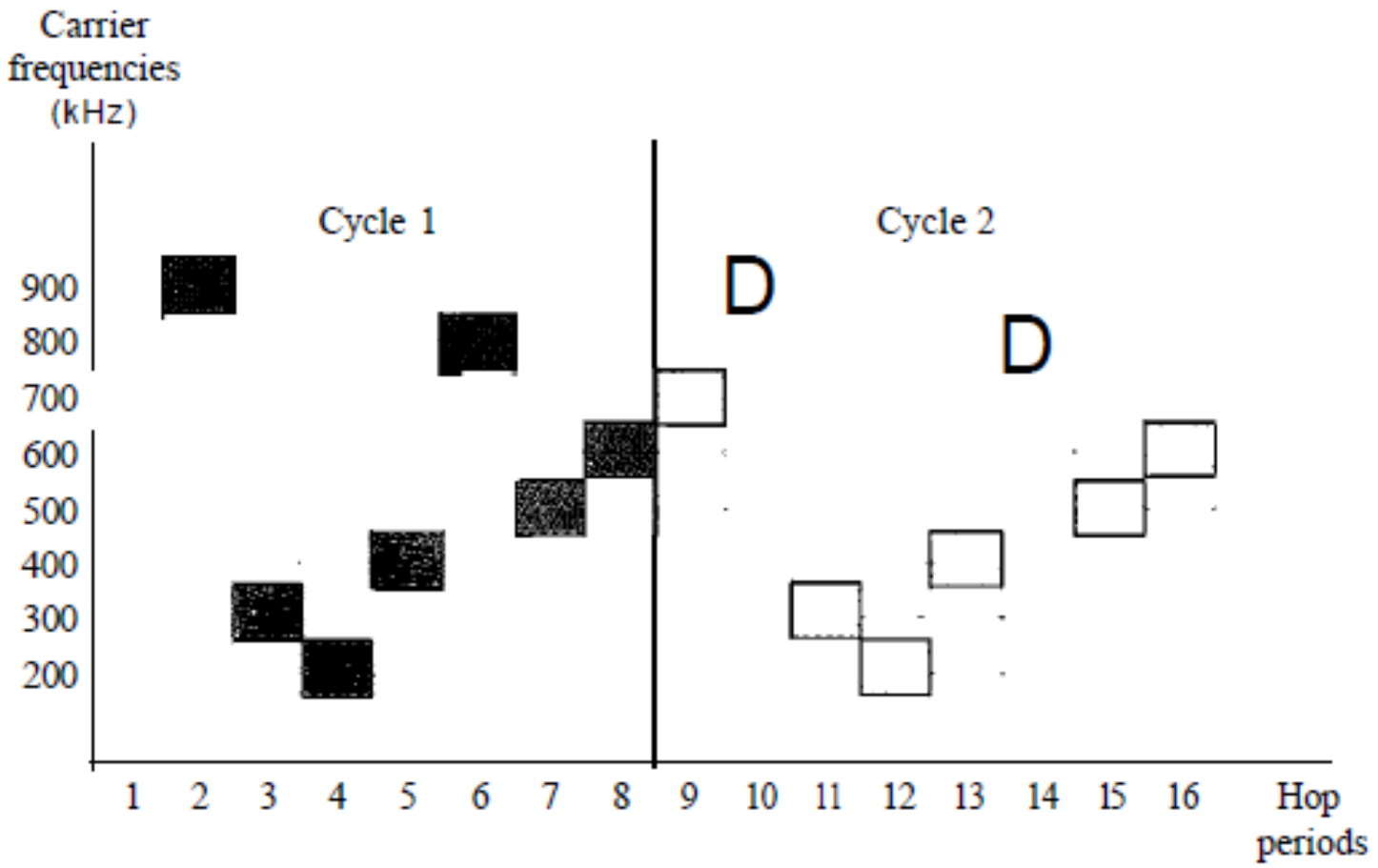
Frequency selection in FHSS



The pattern for this station is 101, 111, 001, 000, 010, all, 100. Note that the pattern is pseudorandom it is repeated after eight hoppings. This means that at hopping period 1, the pattern is 101. The frequency selected is 700 kHz; the source signal modulates this carrier frequency. The second k-bit pattern selected is 111, which selects the 900-kHz carrier; the eighth pattern is 100, the frequency is 600 kHz.

After eight hoppings, the pattern repeats, starting from 101 again. Figure shows how the signal hops around from carrier to carrier. We assume the required bandwidth of the original signal is 100 kHz.

FHSS cycles


Carrier frequencies (kHz)
900
800
700
600
500
400
300
200
Cycle 1
Cycle 2
D
D
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Hop periods

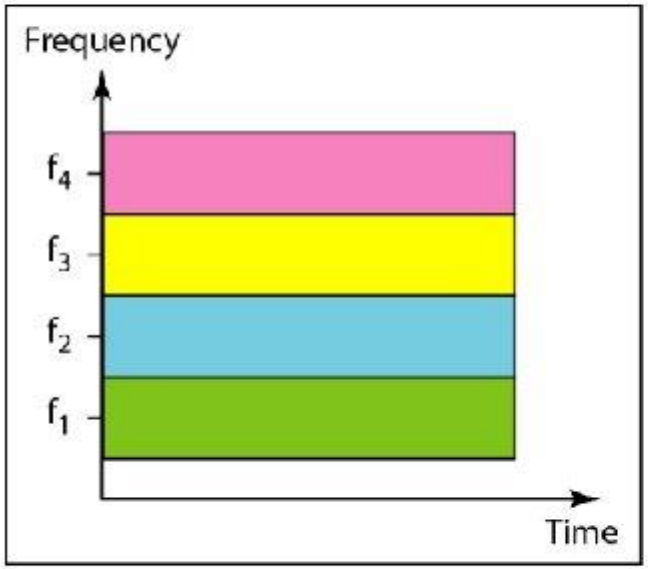
It can be shown that this scheme can accomplish the previously mentioned goals. If there are many k-bit patterns and the hopping period is short, a sender and receiver can have privacy. If an intruder tries to intercept the transmitted signal, she can only access a small piece of data because she does not know the spreading sequence to quickly adapt herself to the next hop.

The scheme has also an antijamming effect. A malicious sender may be able to send noise to jam the signal for one hopping period (randomly), but not for the whole period.

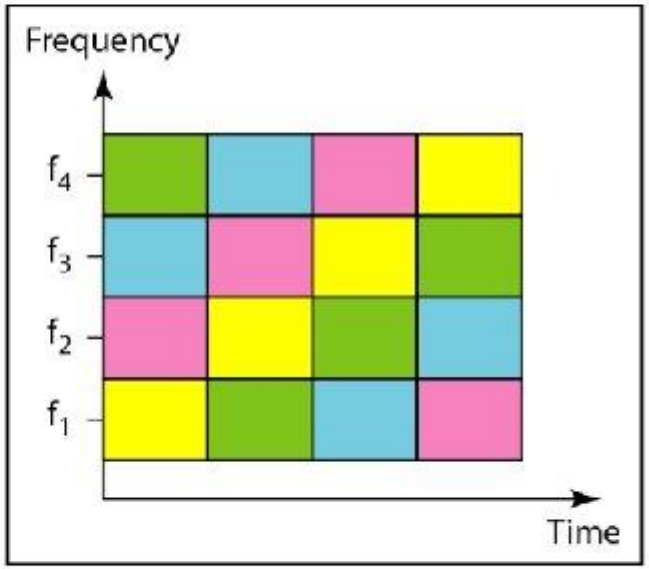
Bandwidth Sharing: If the number of hopping frequencies is M, we can multiplex M channels into one by using the same B_{ss} bandwidth. This is possible because a station uses just one frequency in each hopping period; M - 1 other frequencies can be used by other M - 1 stations. In other words, M different stations can use the same B_{ss} if an appropriate modulation technique such as multiple FSK (MFSK) is used.

FHSS is similar to FDM, as shown in Figure. Figure shows an example of four channels using FDM and four channels using FHSS. In FDM, each station uses 11M of the bandwidth, but the allocation is fixed; in FHSS, each station uses 11M of the bandwidth, but the allocation changes hop to hop.

Bandwidth Sharing



a. FDM

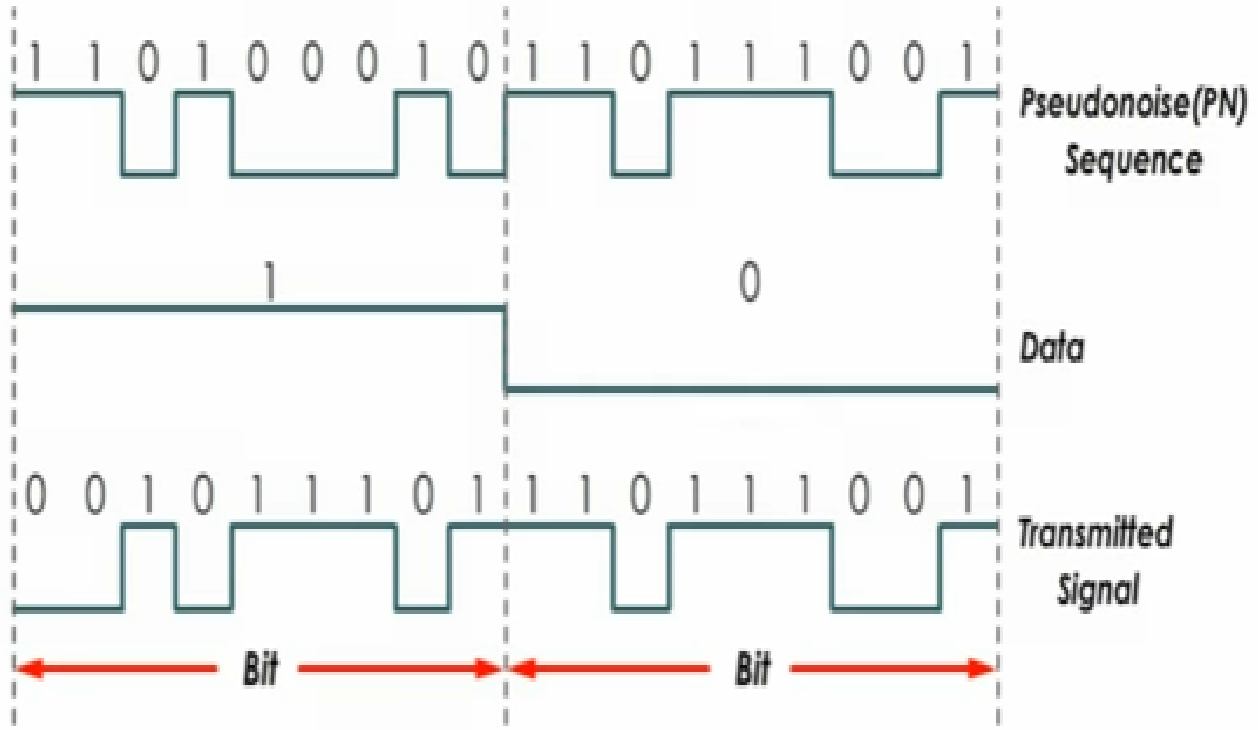


b. FHSS

2- Direct Sequence Spread Spectrum: The direct sequence spread spectrum (DSSS) technique also expands the bandwidth of the original signal, but the process is different. In DSSS, we replace each data bit with n bits using a spreading code. In other words, each bit is assigned a code of n bits, called chips, where the chip rate is n times that of the data bit.

Direct Sequence Spread Spectrum

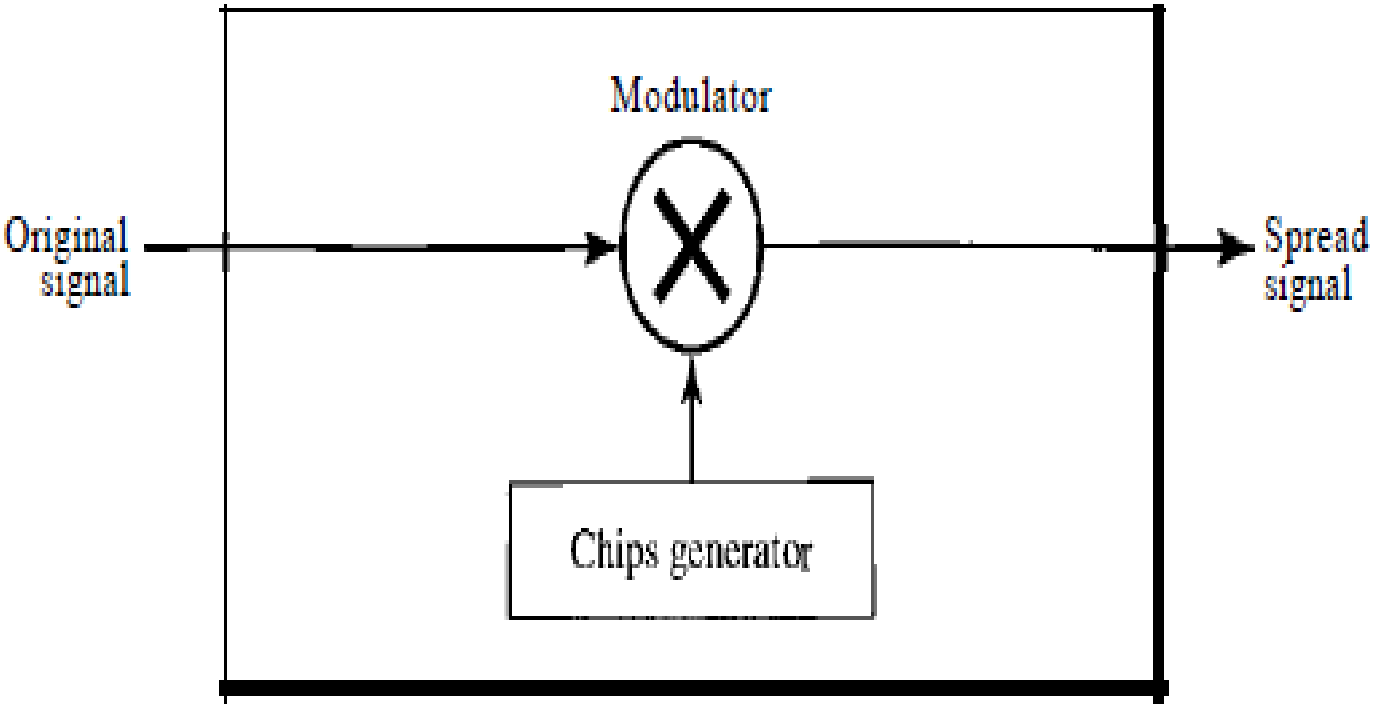
Inputs		Outputs
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0



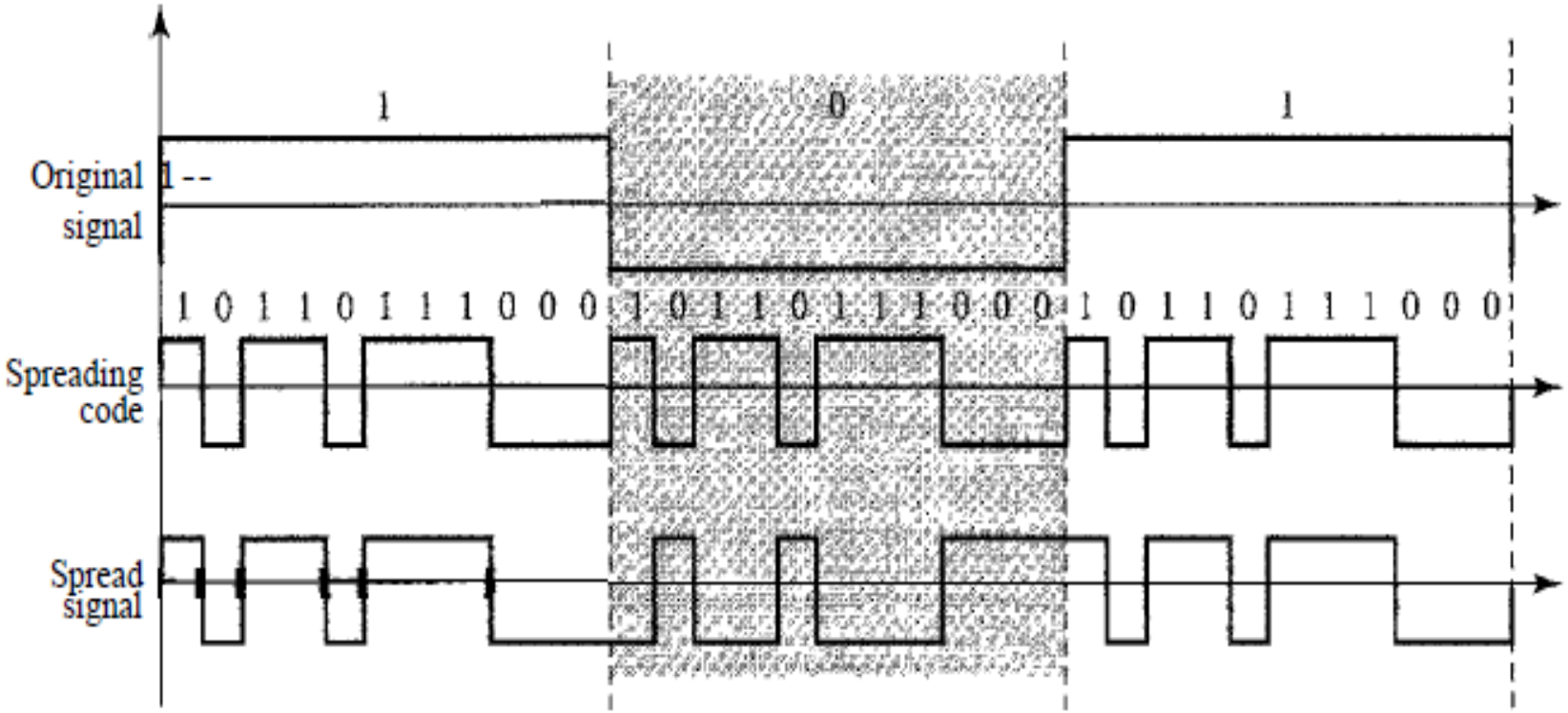
As an example, let us consider the sequence used in a wireless LAN, the famous Barker sequence where n is 11. We assume that the original signal and the chips in the chip generator use polar NRZ encoding. Figure shows the chips and the result of multiplying the original data by the chips to get the spread signal. In Figure, the spreading code is 11 chips having the pattern 10110111000 (in this case).

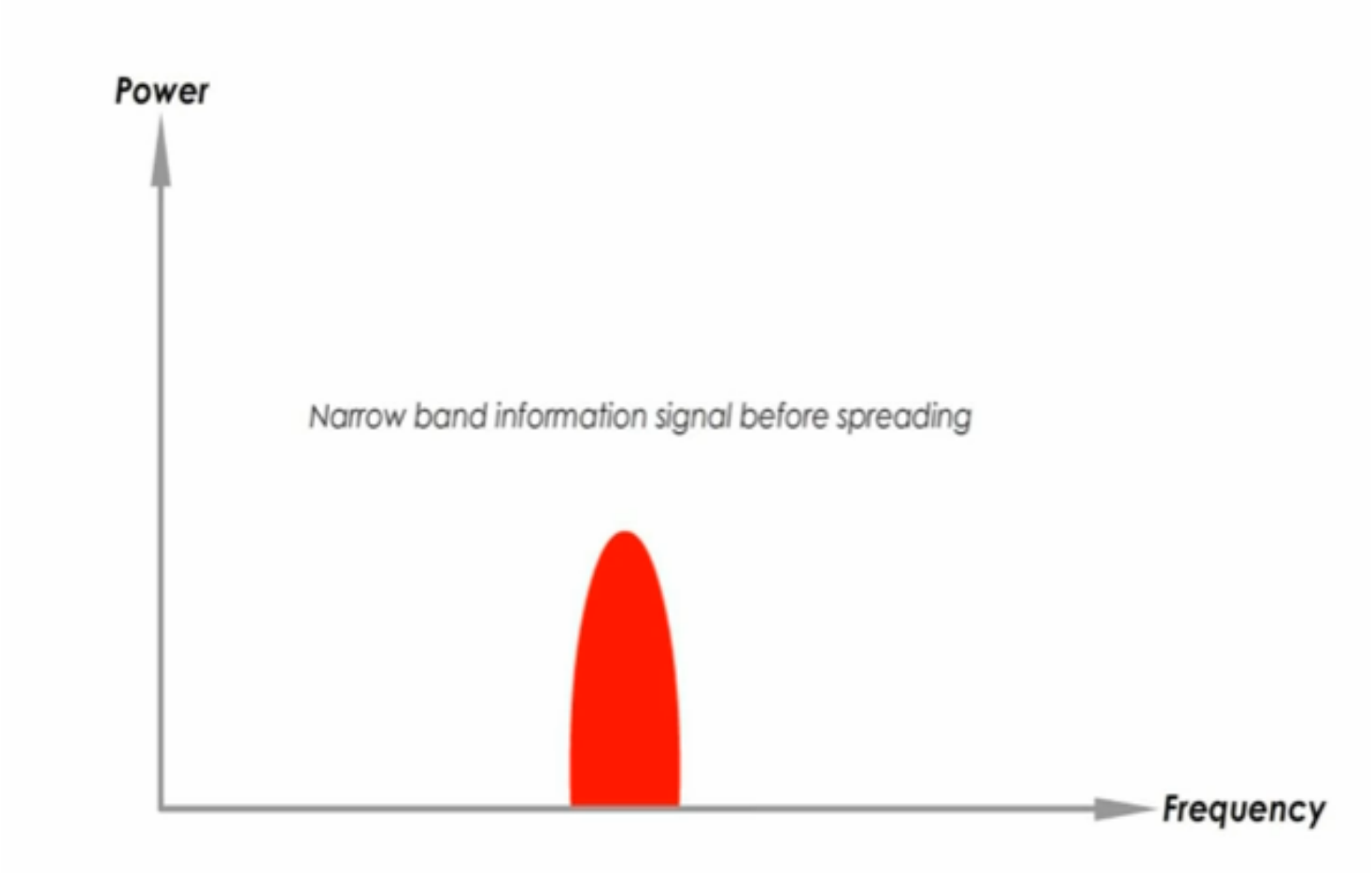
If the original signal rate is N, the rate of the spread signal is 11 N. This means that the required bandwidth for the spread signal is 11 times larger than the bandwidth of the original signal. The spread signal can provide privacy if the intruder does not know the code. It can also provide immunity against interference if each station uses a different code.

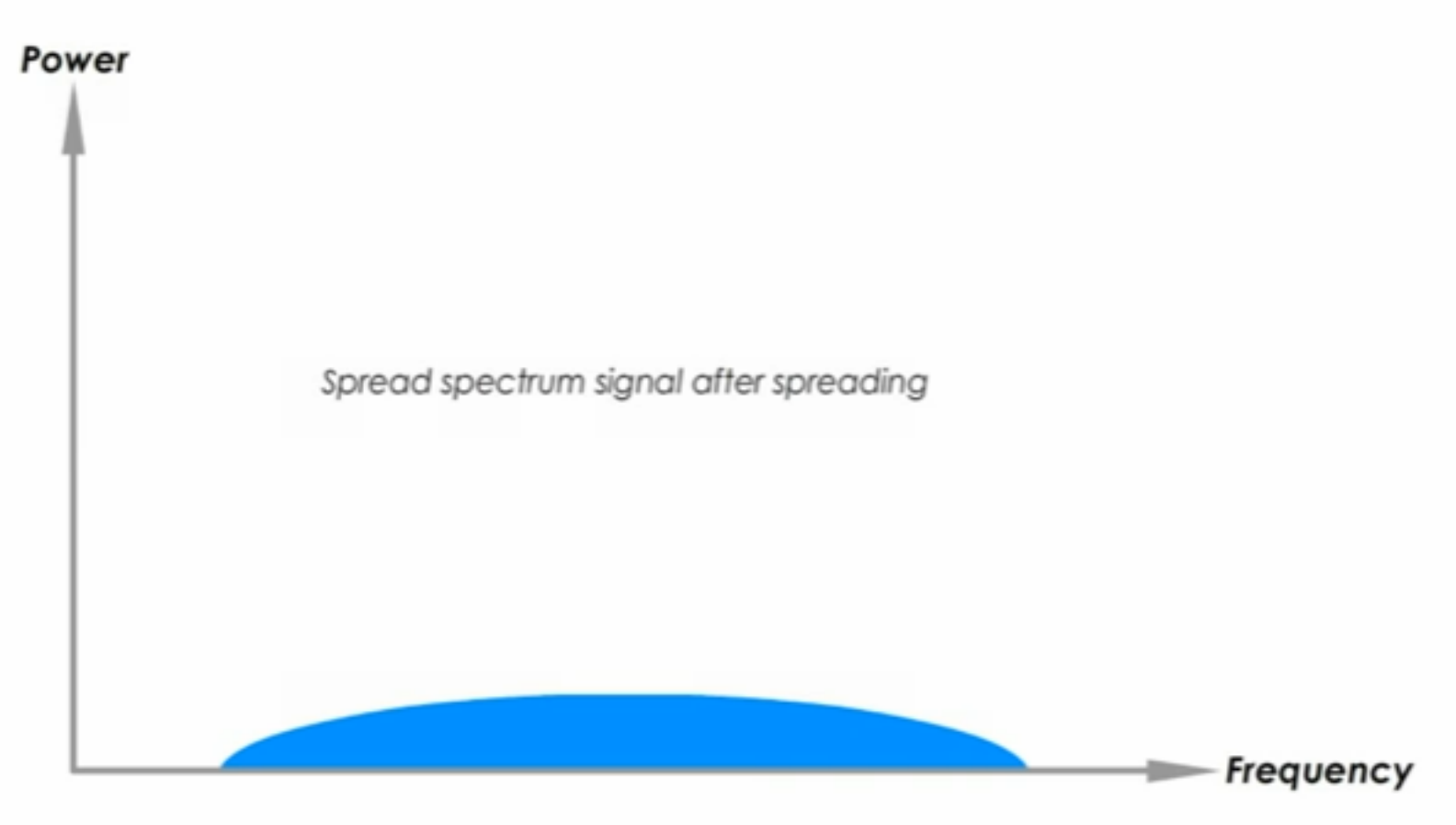
DSSS

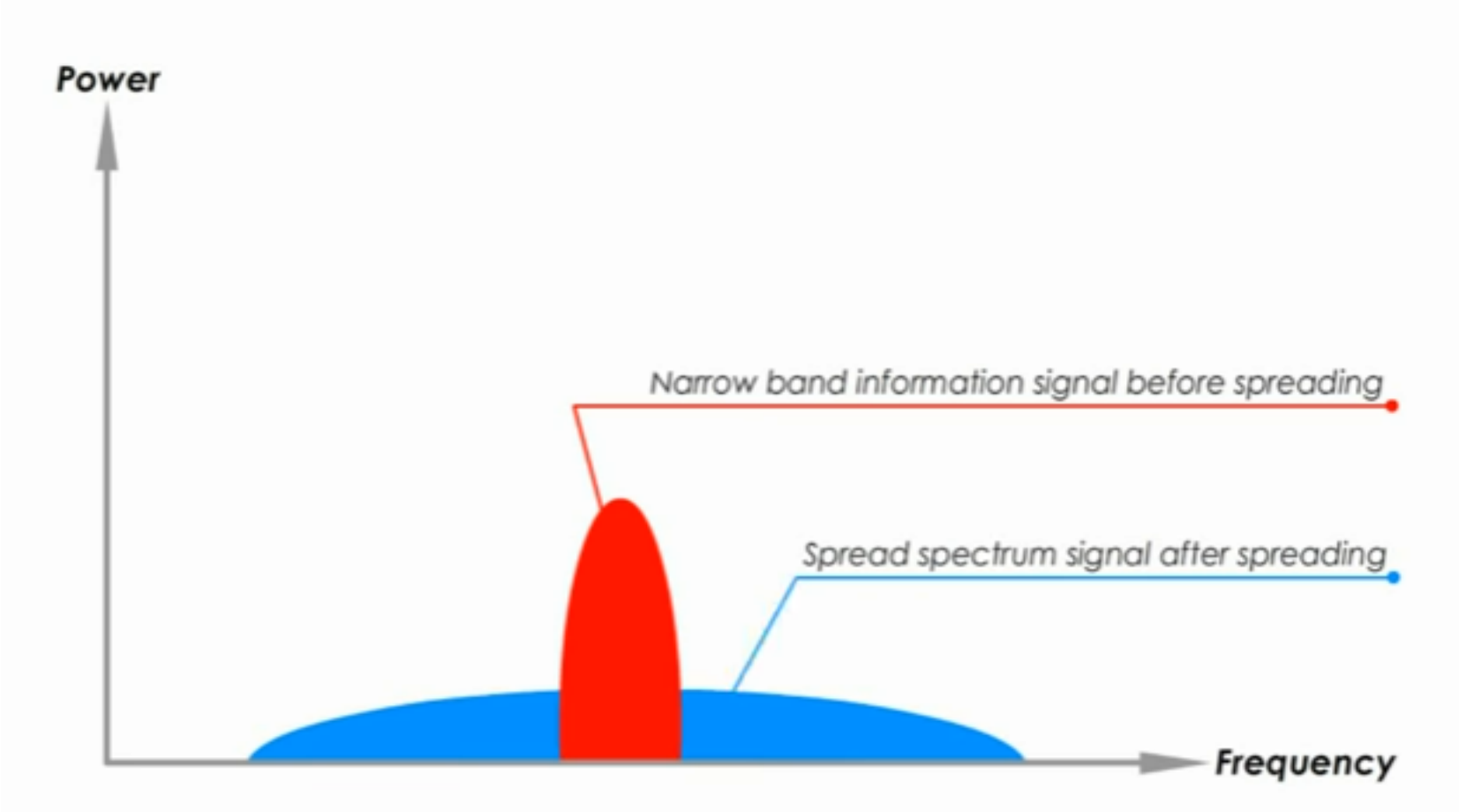


DSSS example




Power
Narrow band information signal before spreading
Frequency


Power
Spread spectrum signal after spreading
Frequency


Power
Narrow band information signal before spreading
Spread spectrum signal after spreading
Frequency

Bandwidth Sharing: Can we share a bandwidth in DSSS as we did in FHSS? The answer is no and yes. If we use a spreading code that spreads signals (from different stations) that cannot be combined and separated, we cannot share a bandwidth. For example, some wireless LANs use DSSS and the spread bandwidth cannot be shared.

However, if we use a special type of sequence code that allows the combining and separating of spread signals, we can share the bandwidth. a special spreading code allows us to use DSSS in cellular telephony and share a bandwidth between several users.