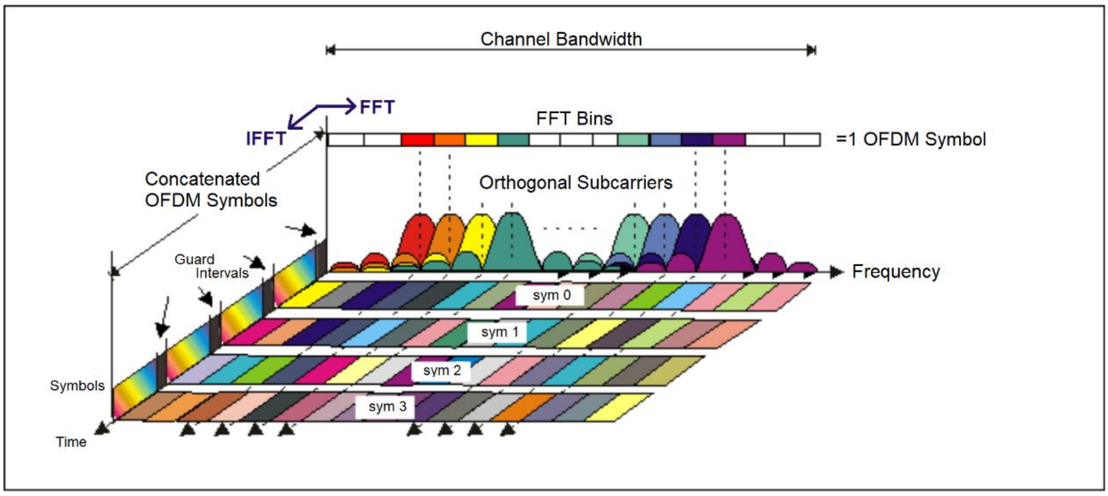
Communication Networks Lab

Mini Project

Simulation of OFDMA using C++



Frequency-Time Representative of an OFDM signal

Name: Animesh Manik Patil

Regd. No: 220907066

Class: ECE-A

Roll No: 6

Table of Contents

1. Introduction to OFDMA ……………………………….

2. Simulation Setup ……………………………………….

3. Data Frames ……………………………………………….

4. Observation ………………………………………………..

5. Scope for Improvement ……………………………….

6. Conclusion ………………………………………………..

7. References …………………………………………….

1. Introduction to OFDMA

Orthogonal Frequency Division Multiple Access (OFDMA) is a multi-user communication technique that extends the principles of Orthogonal Frequency Division Multiplexing (OFDM) to enable efficient simultaneous data transmission between a base station and multiple users. While OFDM divides the available bandwidth into a large number of orthogonal subcarriers for transmitting a single user's data in parallel, OFDMA allows these subcarriers to be split among multiple users at the same time. This parallel access makes OFDMA highly efficient in environments with many users or devices, and it is a foundational technology in systems like LTE, 5G New Radio, and Wi-Fi 6.

In OFDMA, the time-frequency grid is divided into smaller allocation units, commonly referred to as resource blocks (in LTE/5G) or resource units (in Wi-Fi 6). Each of these units consists of a group of subcarriers over a specific time duration. The base station (or access point) dynamically assigns these blocks to users based on real-time factors such as channel conditions, user demand, and quality-of-service requirements. Since the subcarriers are orthogonal, multiple users can transmit data concurrently without causing inter-user interference, as long as synchronization is maintained.

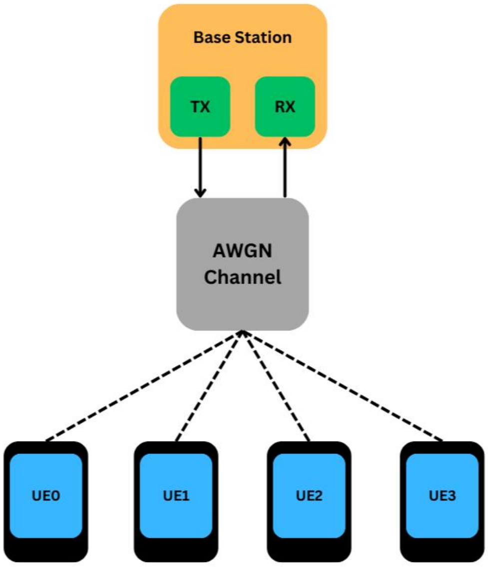
One of the key benefits of OFDMA is its ability to support a large number of users simultaneously, while maintaining low latency and high spectral efficiency. This is particularly useful in high-density environments where users generate diverse traffic profiles — such as streaming, voice, browsing, or IoT data. Unlike older techniques like TDMA or FDMA, which divide time or frequency among users in more rigid ways, OFDMA offers fine-grained resource scheduling and better adaptability to real-time traffic demands. It also improves overall system throughput by allocating more resources to users with better channel quality (e.g., via adaptive modulation and coding).

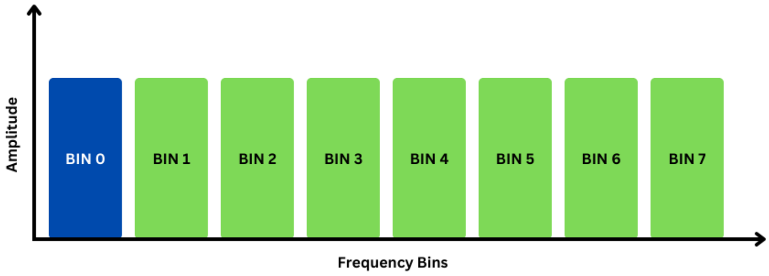
Despite its advantages, OFDMA also presents technical challenges. The most critical is the need for precise synchronization among users, particularly in the uplink. Timing and frequency offsets between users can break subcarrier orthogonality and result in inter-carrier or inter-user interference. This is why uplink OFDMA implementations, such as in LTE and 5G, include sophisticated timing advance and frequency correction mechanisms. Additionally, like OFDM, OFDMA suffers from high peak-to-average power ratio (PAPR), which can affect power efficiency in transmitters, especially for battery-powered devices.

Nevertheless, the flexibility, scalability, and efficiency of OFDMA have made it an essential building block in modern wireless communication. Its ability to adaptively allocate resources, handle diverse traffic types, and serve multiple users in parallel makes it ideal for the demands of next-generation networks. As more devices connect to wireless systems — especially with the growth of IoT, cloud-based services, and mobile streaming — OFDMA provides the framework needed to manage spectrum efficiently and deliver consistent, high-speed performance across a wide range of applications.

2. Simulation Setup

The simulation consists of 2 types of nodes, the Base Station and the User. In this simulation, up to 4 users can subscribe to different OFDMA channel (User IDs 0-3). The channel is assumed to be a Wireless AWGN (Additive White Gaussian Noise) Slow-Fading channel.





The OFDM bandwidth is divided into 8 orthogonal frequency bins, as shown in the figure on the right. Bin 0 (highlighted in blue) is reserved for control codes or codes that specify the type of data being sent (Channel Allocation Request, Data Transmission, Channel Deallocation and Base-Station Response). The remaining 7 bins are used for transmission of data.

Each frequency bin is modulated using QPSK (Quadrature Phase Shift Keying), where phase differences in the transmitted wave can represent 4 values or 2 bits. The overall channel transmission capacity thus turns out to be 2*7 = 14 bits per symbol period.

At the transmitter of every node, the 8-bin frequency domain signal is up-sampled to a 64-length frequency domain representation. The number of bins remains the same, only the resolution of the frequency samples is increased by a factor of 8, i.e. gap between orthogonal subcarriers is equal to 8 frequency samples. This vastly improves the resolution and quality of the final time domain signal. The transmitter then converts the frequency domain signal into a time domain signal using an IFFT (Inverse Fast Fourier Transform).

At the receiver end, the time domain signal is converter back into the frequency domain via an FFT (Fast Fourier Transform). Value of each frequency bin is then correlated to its respective QPSK symbol and the data is decoded.

3. Data Frames

The following set of frames can be sent by users over the channel to the base-station and other users and by base-stations to users as a response or as forwarded data:

(i) Channel Allocation Request

User Dataframe

ALLOC REQ CODE (0)	USER ID (0 - 3)	ALLOC COUNT (0-3)	-	-	-	-	-

Base-Station Response Frame

BASE STATION RESPONSE (2)	USER ID (0 - 3)	ALLOC COUNT (0-3)	START BIN HIGH (0-3)	START BIN LOW (0-3)	-	-	-

User requests for channel allocation, with required number of channels (Alloc Count). The Base Station tries to find the largest contiguous sequence of unused bin and allocates them to the user. The Base Station responds back with the Alloc Count and a Start Bin.

(ii) Data Transmission

User Dataframe

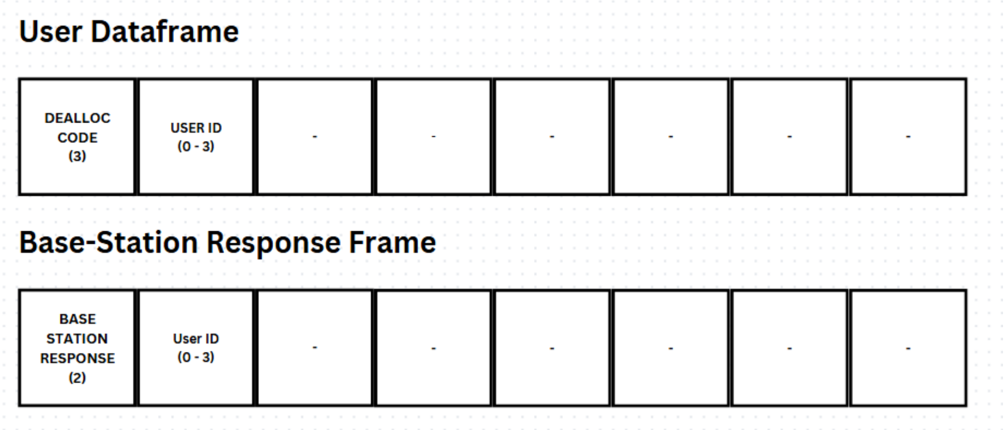
TX DATA CODE (1)	DST ID (0 - 3)	SRC ID (0-3)	DATA BIN (0-3)	DATA BIN (0-3)	DATA BIN (0-3)	DATA BIN (0-3)	DATA BIN (0-3)

Base-Station Response Frame (To DST User)

TX DATA CODE (1)	DST ID (0 - 3)	SRC ID (0-3)	DATA BIN (0-3)	DATA BIN (0-3)	DATA BIN (0-3)	DATA BIN (0-3)	DATA BIN (0-3)

User asks Base-Station to convert and retransmit the data from its allocated bins to the bins allocated to the Destination User. The Data Bins can be employed by multiple users at a time due to the orthogonality of the subcarriers.

(iii) Channel Deallocation



User sends a simple request to deallocate channels for its User ID.

4. Observation

Base-Station side IO

```
PS E:\Animesh\College\6th_Sem\CN_Lab\OFMDMA_Simulation_CPP> make run-base-station

./base_station

Base station simulation started

Access request from user 0 requesting 2 bins.

Allocated 2 bins to user 0 starting at active bin 3

Access request from user 1 requesting 3 bins.

Allocated 3 bins to user 1 starting at active bin 5

Data from user 0 to user 1 => decoded payload = 15

Data from user 1 to user 0 => decoded payload = 15

Deallocation request from user 0

Deallocated bins for user 0

Access request from user 0 requesting 1 bins.

Allocated 1 bins to user 0 starting at active bin 3

Data from user 0 to user 1 => decoded payload = 2
```

User 0 side IO

```
PS E:\Animesh\College\6th_Sem\CN_Lab\OFMDMA_Simulation_CPP> make run-user UID=0

./user 0

User simulation started. user id=0


Commands: req, send, dealloc, read, exit: req

Enter bins requested(1-3): 2

Access request sent.


Commands: req, send, dealloc, read, exit: read

No messages in queue.


Commands: req, send, dealloc, read, exit: read

Message: Response: allocated 2 bins starting at active bin 3


Commands: req, send, dealloc, read, exit: send

You have 2 bins => can send up to 4 bits => max int=15

Destination user(0-3)? 1

Payload(0..15)? 15

Data transmission sent.


Commands: req, send, dealloc, read, exit: read

No messages in queue.


Commands: req, send, dealloc, read, exit: read

Message: Data from user 1, payload=15


Commands: req, send, dealloc, read, exit: dealloc

Deallocation command sent.
```

Commands: req, send, dealloc, read, exit: req

Enter bins requested(1-3): 1

Access request sent.

Commands: req, send, dealloc, read, exit: send

No bins allocated.

Commands: req, send, dealloc, read, exit: read

Message: Response: allocated 0 bins starting at active bin 0

Commands: req, send, dealloc, read, exit: read

Message: Response: allocated 1 bins starting at active bin 3

Commands: req, send, dealloc, read, exit: send

You have 1 bins => can send up to 2 bits => max int=3

Destination user(0-3)? 1

Payload(0..3)? 2

Data transmission sent.

Commands: req, send, dealloc, read, exit: exit

User 1 side IO

PS E:\Animesh\College\6th_Sem\CN_Lab\OFMDMA_Simulation_CPP> make run-user UID=0

./user 0

User simulation started. user id=0

Commands: req, send, dealloc, read, exit: req

Enter bins requested(1-3): 2

Access request sent.

Commands: req, send, dealloc, read, exit: read

```
No messages in queue.

Commands: req, send, dealloc, read, exit: read
Message: Response: allocated 2 bins starting at active bin 3

Commands: req, send, dealloc, read, exit: send
You have 2 bins => can send up to 4 bits => max int=15
Destination user(0-3)? 1
Payload(0..15)? 15
Data transmission sent.

Commands: req, send, dealloc, read, exit: read
No messages in queue.

Commands: req, send, dealloc, read, exit: read
Message: Data from user 1, payload=15

Commands: req, send, dealloc, read, exit: dealloc
Deallocation command sent.

Commands: req, send, dealloc, read, exit: req
Enter bins requested(1-3): 1
Access request sent.

Commands: req, send, dealloc, read, exit: send
No bins allocated.

Commands: req, send, dealloc, read, exit: read
Message: Response: allocated 0 bins starting at active bin 0
```

```
Commands: req, send, dealloc, read, exit: read

Message: Response: allocated 1 bins starting at active bin 3



Commands: req, send, dealloc, read, exit: send

You have 1 bins => can send up to 2 bits => max int=3

Destination user(0-3)? 1

Payload(0..3)? 2

Data transmission sent.



Commands: req, send, dealloc, read, exit: exit
```

From the above example we can see that OFDMA channel allocation and signal generation have been successfully modelled. We can further test the robustness of the network by varying the Noise Variance factor.

As the Noise Variance goes from 0.001 to 0.1, the noise immunity of the channel degrades rapidly. At 0.1, the message is completely different from the original message and may end up causing misinterpretation or completely garbage data at the receiver end.

5. Scope for Improvement

The above implementation is a very simplified model of the working of an actual OFDMA system. Areas in which the model can be further developed are:

- Modelling of a realistic fast-fading channel
- Modulation of the time signals at GHz frequencies, as used in 4G/5G technologies
- Adding multipath components to the channel and introducing inter-symbol interference.
- Simulating attenuation of the transmitted waves
- Using empirical models like the Okumura-Hata model to simulate path loss in different environments like Urban, Sub-urban and Rural areas.
- Faster refresh rate of simulation to emulate real time performance (Current model has a refresh rate of 1Hz)
- Dynamic channel allocation based on channel estimation

6. Conclusion

The project successfully demonstrated a simplified OFDMA simulation using separate base station and user programs that communicate via text files. In this implementation, a 64-point FFT/IFFT structure was used with 8 equally spaced active subcarriers, where one subcarrier was reserved for control signalling and the remaining subcarriers were dynamically allocated for data transmission. The system was designed to handle dynamic resource allocation, where users request and receive specific frequency bins, and the base station manages these requests, forwards data transmissions appropriately, and supports deallocation commands.

The implementation integrated QPSK modulation and demodulation to map control and data bits onto the active subcarriers. By encoding key control information—such as message type, user IDs, requested bin count, and starting bin—into carefully chosen subcarriers, the project demonstrated the concept of controlling data flow in a frequency domain. Additionally, the base station re-encodes incoming data from a sender into the frequency bins allocated to the destination user, ensuring that each user decodes data in their own assigned frequency slots. This aspect of the design highlights a fundamental component of OFDMA, where different users are provided with different frequency resources.

Overall, the project illustrates how concepts like FFT/IFFT, modulation, channel noise simulation, and resource allocation can be combined in a practical simulation of an OFDMA system. The separation of code into common signal processing functions and the distinct base station and user modules allowed for modular development and easier debugging. While the simulation remains a simplified model of real-world systems, it provides valuable insights into the process of frequency domain resource management and data transmission in modern wireless communication systems.

Source files for the project can be found here:

https://github.com/animeshpatil/OFDMA_Simulation_CPP

7. References

- Concepts of Orthogonal Frequency Division Multiplexing (OFDM) and 802.11 WLAN - https://helpfiles.keysight.com/csg/89600B/Webhelp/Subsystems/wlan-ofdm/content/ofdm_basicprinciplesoverview.htm
- Hands-on Project: OFDM Signal Transmission and Reception Using Amateur SDR Devices - https://classes.engineering.wustl.edu/~jain/cse574-22/ftp/ofdm_sdr/index.html