

PNU WiFi 무선랜 서비스 품질 분석 및 개선 연구



201524412 곽민수
201624419 김도형
201724537 이에경

지도교수 김 종 덕

목 차

1. 서론	1
1.1 연구 배경	1
1.2 연구 목표	2
1.3 제약사항	2
2. 연구 내용	2
2.1 연구 설계	2
2.2 데이터 수집 및 정리	3
2.2.1 AP 데이터 분석	3
2.2.2 제외 데이터	3
2.2.3 최대 이용자 수 데이터	4
2.2.4 채널 분포 데이터	6
2.2.5 교내 건물 좌표 및 AP 위치 데이터	8
2.3 MATLAB 시뮬레이션	9
2.3.1 시뮬레이션 도구	10
2.3.2 시뮬레이션 구성	10
2.3.2.1 시뮬레이션 프로세스	10
2.3.2.2 시각화	11
2.3.2.3 환경 설정	11
2.3.2.4 실행	13
2.3.3 시뮬레이션 코드 과정	13
2.4 휴리스틱 알고리즘	15
2.4.1 목표 처리율	15

2.4.2 알고리즘 동작 순서도	15
2.4.3 추가 AP 배치	16
2.4.4 채널 할당	17
2.4.5 AP, STA 연결	20
3. 개발 한계 - 산학협력 프로젝트	22
4. 연구 결과 분석 및 평가	25
4.1 시뮬레이션 실행 시간	25
4.2 연구 결과	25
5. 결론	27
5.1 활용 방안	27
5.2 향후 개발 방향	27
6. 수행 체계	28
6.1 구성원별 역할	28
6.2 개발 일정	29
7. 참고 문헌	29
8. 부록	30

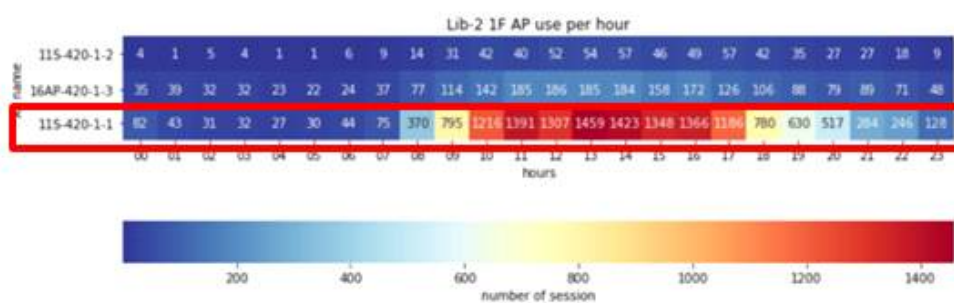
1. 서론

1.1 연구 배경

다양한 스마트기기들이 널리 보급됨에 따라 무선랜 사용이 급증하면서 대규모 무선랜 서비스는 공공장소나 산업 현장에 배치되었다. 무선랜 기술이 보편화 되면서 WiFi망의 사용자들은 많은 양의 데이터와 다양한 서비스를 요구하지만 대규모 무선랜은 사용자의 수요를 효율적으로 대응하지 못해 서비스 안정성이 크게 저하되어 있다.

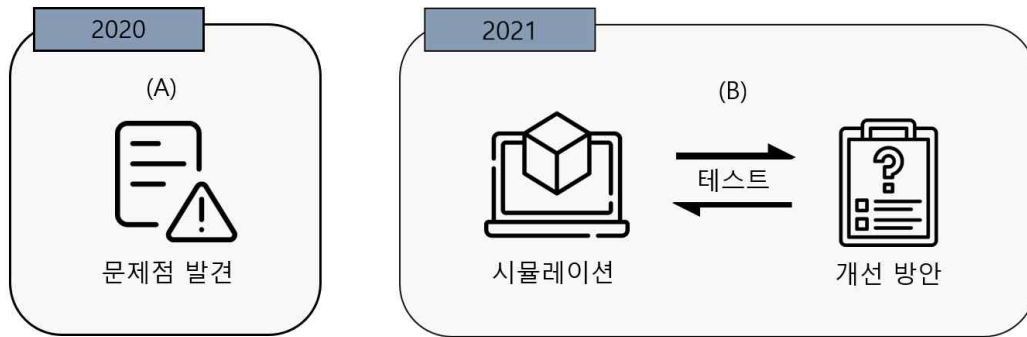
2020년도 전기 졸업과제 “캠퍼스범위 대규모 무선랜 환경의 AP 상태분석”에 따르면 부산대학교 장전캠퍼스에서 운용중인 무선랜 서비스, PNU-WiFi는 약 1,600여개의 AP들을 운용 중이며 약 30,000명의 사용자에게 서비스를 제공하고 있다. 하지만 설문조사 결과, 73.3%의 학생들이 PNU-WiFi 사용에 있어서 불편을 느끼고 있다고 답하였다.

부정적인 설문조사 답변에 대해 이전 연구를 통해 확인한 결과 해당 무선랜 서비스는 자원을 효율적으로 활용하고 있지 않다. 불균형 위치를 찾기 위해 캠퍼스 전체 건물을 용도에 따라 네 그룹으로 분류하였다. 이용자 데이터를 분석하여 각 그룹에서 불균형이 가장 심한 층을 선정했다. AP가 설치된 위치적 특성, 사용 목적 등을 고려했을 때 불필요한 AP가 존재하였다. 뿐만 아니라 PNU-WiFi의 AP들은 모두 서비스 영역을 최대로 설정되어 AP간의 서비스 영역을 간섭하여 서비스 품질을 저하한다. 또한 특정 AP에 과도하게 연결이 몰리는 현상이 있음을 확인할 수 있다.



[그림1] 새벽별 도서관 1층의 과도하게 연결이 몰리는 현상

아래 그림과 같이 이전 연구를 통해 PNU-WiFi의 비효율적 AP운용 문제점들을 확인하였으므로 해당 결과에 대한 추가적인 개선방안 연구가 필요로 하다. 효율적으로 무선랜 서비스를 운용하고 서비스 전체의 품질을 향상시키기 위해서 무선랜 서비스 상태 분석 기술을 개발하고 실제 환경에 알맞은 자원 할당 최적화 전략을 수립할 필요가 있다.



[그림2] 문제점 분석(A)과 개선방안 제시(B)

1.2 연구 목표

본 졸업과제는 PNU-WiFi 서비스의 품질 저하 원인을 AP의 분포와 간섭효과를 분석하여 이를 보완하여 무선랜 서비스의 자원이 고르게 할당되어 사용자들의 데이터 처리율을 향상시키는 것을 목표로 한다. 이에 필요한 AP의 최소 개수를 산정하고 최적의 배치 방법을 제시하는 것을 문제로 정의한다.

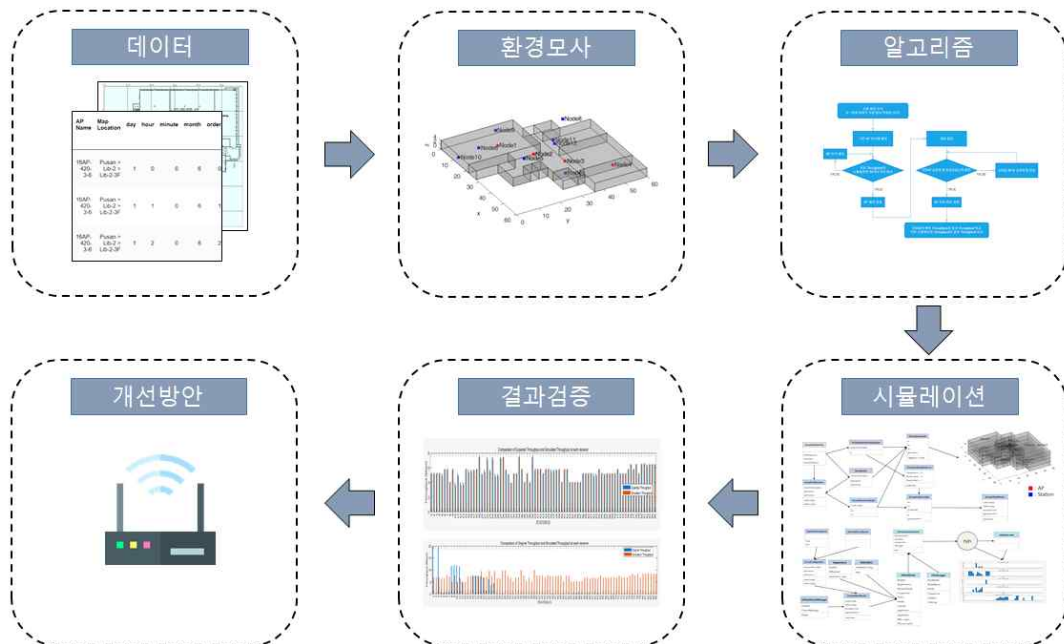
1.3 제약사항

- 모든 STA를 어떠한 AP와 연결할 것
- 모든 STA이 일정 수준의 Throughput을 넘을 것
- AP 커버리지가 건물 면적의 일정 수준 보다 클 것

2. 연구 내용

2.1 연구 설계

설계 구조는 아래 그림과 같이 되어있다. 먼저 실제 도면을 통해 건물 환경을 모사하고 수집한 데이터를 정리 및 적용하여 시뮬레이션을 제작한다. 해당 시뮬레이션과 알고리즘을 통해 최적의 AP 배치를 제안한 후 해당 제안을 검증하기 위하여 다시 시뮬레이션을 통해 실제로 타당한지 확인한다.



[그림3] 전체 연구 설계 구조

2.2 데이터 수집 및 정리

2.2.1 AP 데이터 분석

교내 건물 AP 사용 환경 모사를 위해 정보화본부로부터 수집한 AP 데이터를 받아 분석하고 필요한 데이터를 추출 후 재가공했다. 원본 데이터는 2019년 6월 1일 00:00에서 2020년 3월 31일 23:00까지 1시간 간격으로 수집된 AP의 상태 정보와 무선 접속 정보를 포함하고 있다.

AP 사용 환경을 모사하기 위해서는 시간별 AP에 연결된 최대 이용자 수의 평균이 필요했다. 데이터는 2.4GHz대역에 41개의 열, 5GHz대역에 41개의 열, 두 대역이 공통으로 갖는 19개의 열을 더해 총 101개의 열이 있다. 이 중 AP가 설치된 위치(Map Location), AP 이름(AP Name), 최대 이용자 수(Peak Number of Users), 요일(weekday), 총 4개의 열을 사용해 진행한다. 이후에 AP들의 채널 분포를 파악하기 위해 채널(Channels in Use), 1개 열을 추가로 이용한다.

2.2.2 제외 데이터

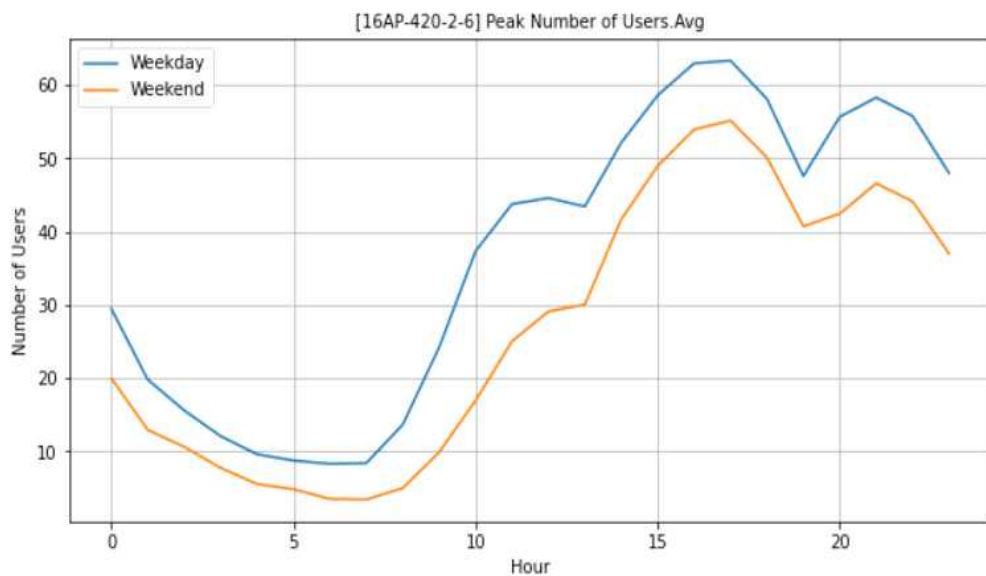
원본 데이터는 장전캠퍼스, 밀양캠퍼스, 양산캠퍼스의 정보를 모두 포함하고 있다. 본 연구에서는 장전캠퍼스 내의 건물들을 대상으로 하므로 Map Location이 Milyang, Yangsan으로 시작하는 데이터는 제외했다. 또한 코로나 바이러스로 캠퍼스 내 네트워크 이용자가 줄어들기 이전인 2019년 12월 31일 23:00까지의 데이터를 사용한다.

2.2.3 최대 이용자 수 데이터

AP Name	Map Location	day	hour	minute	month	order	second	str	weekday	year
16AP-420-3-6	Pusan > Lib-2 > Lib-2-3F	1	0	0	6	0	0	2019-06-01 00:00:00	5	2019
16AP-420-3-6	Pusan > Lib-2 > Lib-2-3F	1	1	0	6	1	0	2019-06-01 01:00:00	5	2019
16AP-420-3-6	Pusan > Lib-2 > Lib-2-3F	1	2	0	6	2	0	2019-06-01 02:00:00	5	2019

[그림4] 새벽별 도서관 데이터의 일부

위 사진은 새벽별 도서관 데이터에서 사용된 열들을 나타낸 사진이다. 데이터가 전체 건물의 정보를 포함하고 있으므로 Map Location을 이용하여 필요한 건물의 데이터만을 추출한다. 다시 AP Name을 이용해 AP별로 데이터를 구분하여 리스트에 저장한다. 리스트에 있는 AP별 데이터에 대해 다음 작업을 수행한다.



[그림5] 새벽별 도서관 2층 AP의 이용자 수

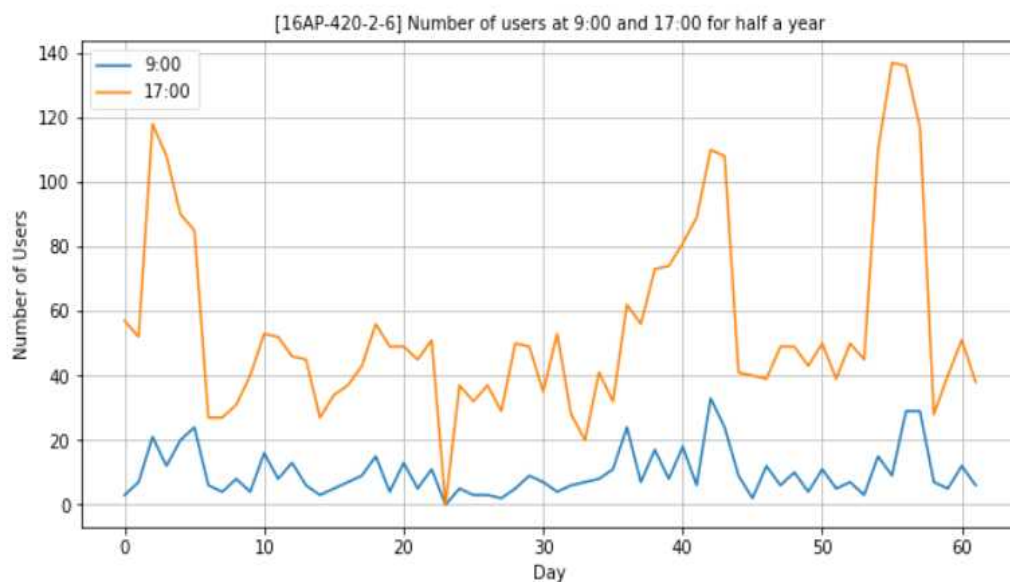
이용자 수는 요일별, 시간별로 주기성을 가지고 있다. 주말보다 평일에 이용자 수가 많고, 오전보다 오후에 이용자 수가 많다. 따라서 데이터를 평일, 주말로 요일을 구분

하여 1시간마다 이용자 수를 평균해서 정리했다. weekday 열은 0(월요일)부터 6(일요일)까지의 값을 가지므로 weekday의 값이 5이거나 6인 데이터를 주말, 나머지 경우의 데이터를 평일로 구분했다.

	a/n/ac	b/g/n
	Signal	Signal
15408	11.0	0.0
15409	1.0	0.0
15410	2.0	1.0
15411	0.0	0.0
15412	0.0	0.0
...
20539	39.0	5.0
20540	40.0	3.0
20541	39.0	3.0
20542	37.0	3.0
20543	18.0	2.0

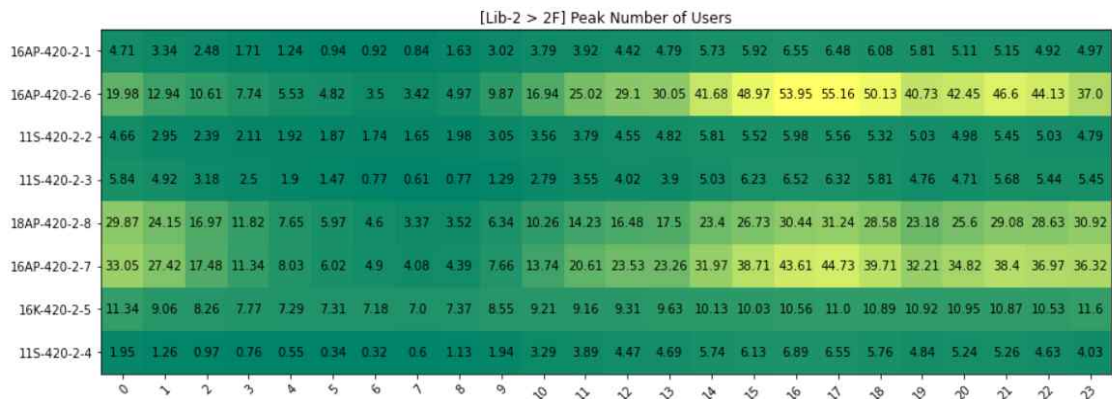
[그림6] Peak Number of Users 추출 데이터 예시

다음으로 Peak Number of Users 열을 이용해 최대 이용자 수를 추출한다. 그러면 위 사진과 같이 두 개로 구분되어 표시가 되는데 이는 주파수 대역에 따라 구분된 것이다. 사진에서 'a/n/ac'는 5GHz대역을, 'b/g/n'은 2.4GHz대역을 의미한다. 두 대역의 선택은 이용자가 아니라 중앙 컨트롤러에 의해 결정되므로 구분하여 보지 않는다. 따라서 두 대역의 이용자 수를 더하여 진행한다.



[그림7] 새벽별 도서관 2층 AP의 9시와 17시의 이용자 수

데이터는 요일을 구분하지 않고 정렬되어 있다. 예를 들어 6월 1일 23시의 데이터 다음에는 6월 2일 0시의 데이터가 나오는 것이다. 따라서 데이터를 24시간 단위로 분할하는 작업이 필요하다. 데이터를 리스트로 바꾸어 24개 단위로 분할하고 이를 다시 데이터프레임 형태로 변환하는 작업을 수행한다. 위의 예시 사진은 24개 단위로 데이터를 분할한 후, 6월부터 12월까지 주말 오전 9시와 오후5시의 이용자 수를 나타낸 그래프이다.



[그림8] 새벽별 도서관 2층의 시간별 최대 이용자 수 평균

마지막으로 구분된 시간대별로 이용자 수를 평균하면 AP별로 평일과 주말, 시간에 따른 최대 이용자 수의 평균을 구할 수 있다. 예시 사진은 새벽별 도서관 2층에 설치된 AP의 주말 최대 이용자 수 평균 데이터의 일부를 가져온 것이다. x축은 시간이며 y축은 AP 이름이 나와있다.

2.2.4 채널 분포 데이터

이후에 AP 사용 환경 모사에 사용하기 위해 채널 분포에 관한 데이터를 정리했다. 802.11b, 802.11g, 802.11n은 2.400 - 2.500GHz 스펙트럼을 사용한다. 802.11a 및 802.11n은 보다 엄격하게 규제되는 4.915-4.925GHz 대역을 사용한다. 각 대스펙트럼은 중심 주파수 및 대역폭을 갖는 채널로 세분화된다. 2.4GHz 대역은 채널 1부터 시작하여 5MHz 간격으로 14개의 채널로 나뉘며 5GHz 대역의 채널 번호는 36번부터 165번까지 매우 다양하다. 각 AP마다 어떤 채널에 집중적으로 이용자가 몰리는지 확인하기 위해 AP 데이터로부터 Channels in Use 열을 사용하여 데이터를 추출, 재가공했다.

데이터는 최대 이용자 수 데이터를 가공할 때와 마찬가지로 요일은 평일과 주말로 구분, 시간은 1시간 간격으로 정리했다. Map Location을 이용해 필요한 건물 데이터를 추출하고 AP Name을 이용해 각 AP에 대해 데이터를 분할한 후 각 AP 데이터에 대해 weekday를 이용하여 평일과 주말로 구분한다.

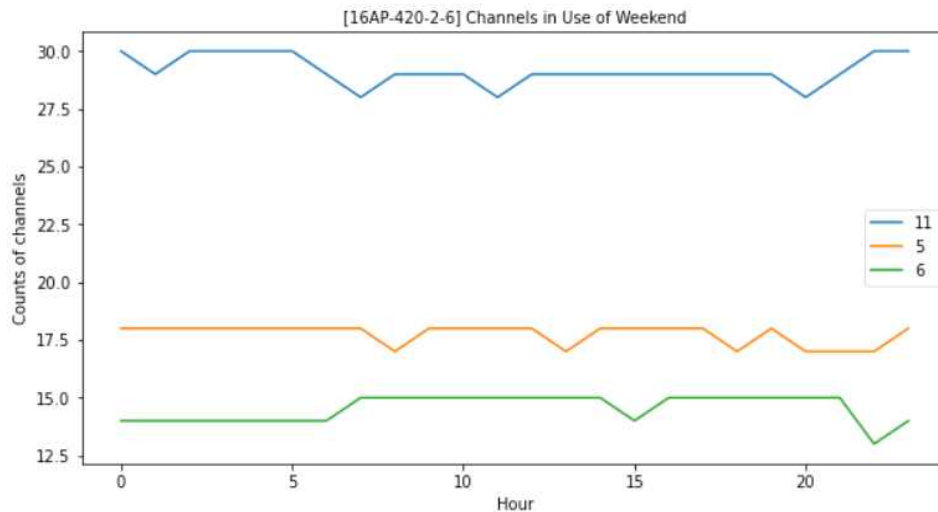
a/n/ac		b/g/n	
15408	['44']	15408	['5']
15409	['44']	15409	['5']
15410	['44']	15410	['5']
15411	['44']	15411	['5']
15412	['44']	15412	['5']
...
20491	['124']	20491	['11']
20492	['124']	20492	['11']
20493	['124']	20493	['11']
20494	['124']	20494	['11']
20495	['124']	20495	['11']

[그림9] Channels in Use 추출 데이터 예시

Channels in Use를 이용해 사용된 채널 데이터를 추출하면 최대 이용자 수 데이터를 정리할 때와 마찬가지로 5GHz 대역과 2.4GHz 대역의 데이터가 각각 표시된다. 두 대역이 사용하는 채널이 다르기 때문에 이번에는 데이터를 합치지 않고 다시 두 개로 분할한다.

채널 데이터도 이용자 수 데이터와 마찬가지로 시간순으로 정렬되어 있다. 따라서 데이터를 리스트로 바꾸어 24개 단위로 분할하고 이를 다시 데이터프레임 형태로 변환하는 작업을 수행했다.

이제 시간대별로 어떤 채널이 얼마나 사용되었는지 데이터가 나타난 횟수를 계산한다. 이때 데이터에 ['1', '3']과 같이 두 채널을 동시에 사용하는 경우가 있었다. 이런 경우 각 채널에 1씩 더해주는 것으로 계산했다. 앞서 언급한 ['1', '3']과 같은 경우 '1'번 채널과 '3'번 채널의 카운트에 각각 1씩 더해주는 것이다.

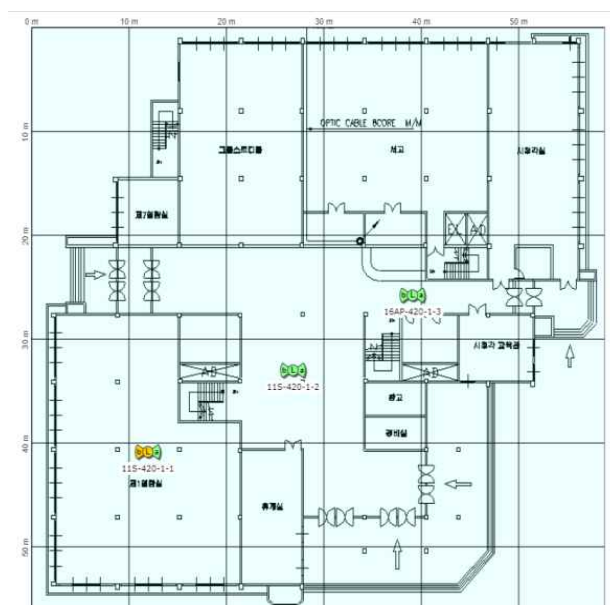


[그림10] 새벽별 도서관 2층 AP 사용된 채널

최종적으로 데이터를 정리한 그래프이다. 새벽별 도서관의 2층 한 AP의 주말 채널 분포에 대한 데이터이다. 시간에 따라 채널이 사용된 횟수를 나타내고 있다. 사용된 채널은 시간에 관계없이 일정한 비율로 사용되고 있음을 알 수 있다.

2.2.5 교내 건물 좌표 및 AP 위치 데이터

본 연구는 부산대학교 장전캠퍼스 내 전체 건물을 대상으로 하고 있다. 이용자가 많아 자원 경쟁이 심한 새벽별 도서관을 우선적으로 선정해 연구를 진행했지만 전체 건물에 대해 연구가 진행되어야 한다. 따라서 교내 전체 건물의 평면도를 받아 건물과 AP 위치를 좌표로 데이터화했다.



[그림11] 새벽별 도서관 1층 도면

시뮬레이션으로 건물을 모델링할 때, 층마다 방과 바닥 정보를 입력해야 한다. 예시 사진은 새벽별 도서관 1층의 도면이다. 왼쪽 아래 점부터 [0, 0, 0]이라 두고 가로는 x 축, 세로는 y축이며, 한 층의 높이는 4로 고정해두었기 때문에 z축은 0, 4, 8, ...의 값을 가진다. 시뮬레이션에서 육면체 방을 만들 때 육면체의 대각선 두 모서리만을 필요로 한다. 따라서 새벽별 도서관 1층의 그룹스터디룸을 만들 때 좌표는 [15, 39, 0], [28, 60, 4]로 둘 수 있다. 시뮬레이션에서 다각형, 삼각형 방을 만드는 방식에 따라 나머지 방의 좌표도 정해준다. 방이 둘러싸고 있는 복도 좌표까지 입력해주면 새벽별 도서관 1층의 건물이 모델링된다. 도면을 정확하게 좌표계에 위치시킬 수 없고 육안으로 좌표를 설정해야 하기 때문에 실제 건물과 어느 정도의 차이는 있지만 무시할 수 있을 정도의 간격이라 가정하고 진행했다.

방 좌표와 마찬가지로 AP 위치도 설정해주었다. 다만 AP는 층의 가운데에 위치되어 있다고 가정하고 있으므로 z값은 AP가 설치되어 있는 층에 따라 2, 6, 10, ... 으로 설정한다. 예를 들어 새벽별 도서관 제1열람실에 설치되어 있는 '11S-420-1-1'의 좌표는 [12, 19, 2]로 설정할 수 있다.

대부분 건물의 각 층은 유사한 구조를 가지고 있었다. 따라서 모든 층의 구조와 가장 유사한 층 하나만을 선정해 방과 바닥 정보를 설정했고, AP 위치 정보는 모든 층에 대해 진행했다. 전체 건물에 대해 위 작업을 수행해 데이터화했다. 아래는 전체 좌표 데이터 중 새벽별 도서관의 건물 좌표와 AP 좌표 예시이다.

```
pt['Lib-2']['Wall Point']
```

```
{'1F': {'R1': [[22, 0, 0], [28, 18, 4]],
'R2': [[34, 18, 0], [40, 22, 4]],
'R3': [[34, 22, 0], [40, 26, 4]],
'R4': [[44, 26, 0], [52, 33, 4]],
'R5': [[8, 38, 0], [15, 45, 4]],
'R6': [[15, 38, 0], [28, 60, 4]],
'R7': [[28, 38, 0], [46, 60, 4]],
'R8': [[46, 36, 0], [60, 60, 4]],
'Hall1': [[8, 33, 0],
[15, 33, 0],
[15, 22, 0],
[22, 22, 0],
[22, 18, 0],
[28, 18, 0],
[28, 10, 0],
[40, 10, 0],
[40, 18, 0],
[34, 18, 0],
[34, 26, 0],
[44, 26, 0],
[44, 33, 0],
[52, 33, 0],
[52, 36, 0],
[46, 36, 0],
[46, 38, 0],
[8, 38, 0]]}}
```

```
pt['Lib-2']['AP Point']
```

```
{'1F': {'11S-420-1-1': [14, 18, 2],
'11S-420-1-2': [27, 27, 2],
'16AP-420-1-3': [38, 35, 2]},
'2F': {'16AP-420-2-1': [15, 35, 6],
'11S-420-2-2': [35, 35, 6],
'11S-420-2-3': [20, 10, 6],
'11S-420-2-4': [35, 15, 6],
'16K-420-2-5': [50, 35, 6],
'16AP-420-2-6': [30, 25, 6],
'16AP-420-2-7': [10, 10, 6],
'18AP-420-2-8': [10, 25, 6]},
'3F': {'16AP-420-3-1': [10, 25, 10],
'11S-420-3-2': [30, 35, 10],
'11S-420-3-3': [25, 10, 10],
'16AP-420-3-4': [10, 10, 10],
'16K-420-3-5': [30, 25, 10],
'16AP-420-3-6': [40, 15, 10]}}
```

[그림12] 새벽별 도서관 건물 좌표(왼쪽)와 AP 좌표(오른쪽)

2.3 MATLAB 시뮬레이션

많은 사용자들이 존재하는 공공장소에서의 WiFi는 다양한 고려사항이 존재한다. 하지만 이러한 요소들은 각 공간의 특성 혹은 시간적 특성 등 여러 가지

경우의 환경들이 있다. 따라서 매번 실제로 세팅을 바꿔가며 실험을 할 수 없기에 컴퓨터를 이용하여 현실과 똑같은 가상의 공간을 만든 뒤, 여러 가지 상황들을 시뮬레이션하여 결과를 미리 예측하고 알고리즘과 개선방안을 검증하는 방법을 필요로 한다.

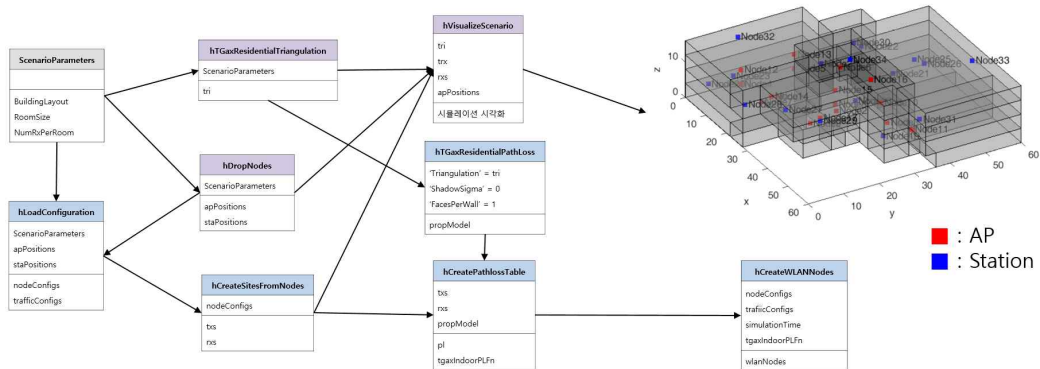
2.3.1 시뮬레이션 도구

시뮬레이션을 진행하기 위해 MATLAB WLAN Toolbox를 사용하고, 교내 환경에 맞게 시뮬레이션 코드를 수정할 필요가 있어 기존 코드에 대한 분석을 진행했다. 기존 시뮬레이션의 주요 기능은 네트워크 및 채널을 구성하여 시나리오를 모델링하는 것과 다중 노드 WLAN 시스템을 시뮬레이션하고 결과값인 처리량, 패킷 손실률 그리고 지연 시간을 그래프로 나타내는 것이다.

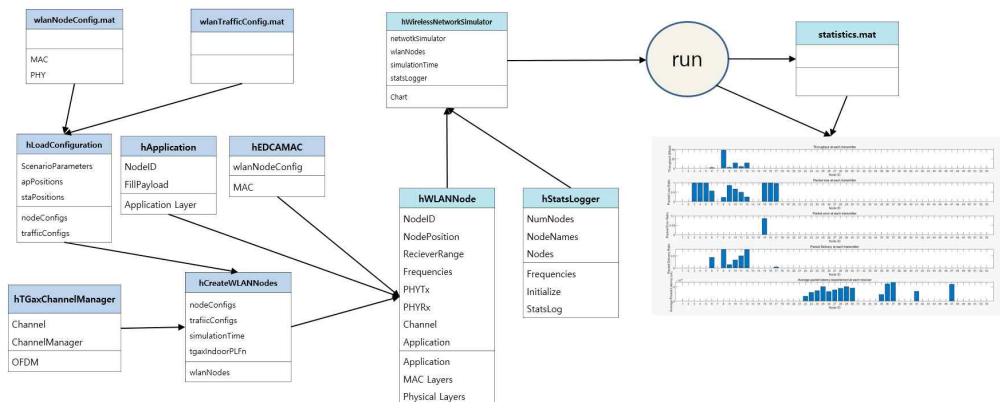
2.3.2 시뮬레이션 구성

2.3.2.1 시뮬레이션 프로세스

- 실제 건물의 물리적 환경 모사 및 시각화
- AP 및 Station에 대한 각 Node 설정
- 시뮬레이션 실행



[그림13] 실제 건물의 물리적 환경 모사 및 시각화 과정



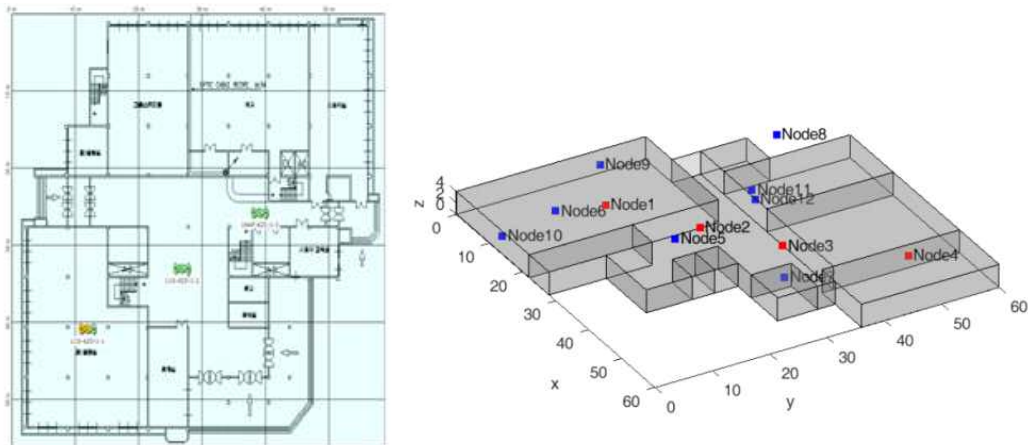
[그림14] Node 설정 및 시뮬레이션 실행 과정

2.3.2.2 시각화

가장 먼저 건물 좌표를 이용해 실제 건물을 모델링하는 기능과 모델링한 건물 내에 AP와 STA들을 표시하는 기능, 시뮬레이션의 결과로 나온 데이터들을 그래프로 시각화하는 기능을 한다.

기존 시뮬레이션의 시각화 단계에서는 실제 교내 건물들과는 달리 단순한 육면체 건물만을 모델링 할 수 있었다. 그래서 기존 코드에서의 건물 모델링이 3차원 좌표들을 연결해 삼각형을 만들고 그 삼각형을 연결하는 순서에 따라 원하는 면을 만드는 방식이라는 것을 파악하고, 이러한 방식에 따라 원하는 다면체 모양의 방과 각 층에서의 복도, 바닥, 천장을 모델링 할 수 있도록 코드 수정을 하였다.

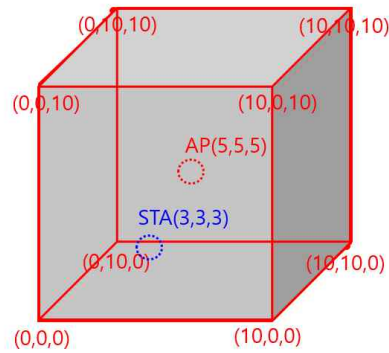
그리고 기존의 건물 모델링에서는 AP와 STA들을 색으로만 구분하고 각 노드 밑에 적힌 Node1, Node2 라는 이름으로는 AP와 STA를 구분하기 힘들었다. 그래서 각 노드들을 더 직관적으로 구분하기 위해 AP와 STA를 구분하여 AP들은 AP명을 그대로 밑에 적어주었고, STA들은 STA별로 STA1, STA2 라는 ID를 적어주었다. 그 후 환경 설정 단계에서 구한 건물 좌표들과 AP 좌표, 랜덤으로 생성된 STA 좌표를 토대로 새벽별 도서관을 모델링 하였다.



[그림15] 새벽별 도서관 실제 도면(왼쪽)과 시각화 예시(오른쪽)

2.3.2.3 환경 설정

환경 설정 부분에서는 모델링할 건물에 대한 환경 설정과 네트워크 시뮬레이션에 사용되는 WLAN Node(AP와 STA)에 대한 환경 설정을 할 수 있다. 실제 건물의 도면을 통해 모델링에 쓰일 건물의 좌표와 건물에 위치한 AP의 좌표를 설정할 수 있다.



[그림16] 건물과 노드의 좌표 설정 예시

또한, 송신/수신 안테나의 환경 설정도 가능하다. 송신 안테나의 설정 요소에는 위치 뿐만 아니라 파워, 등방향성, 주파수 등이 있고 수신 안테나의 경우 위치, 수신감도, 등방향성 등을 설정할 수 있다.

Node와 Traffic에 대한 환경 설정과 Path Loss 설정이 가능하다. Node 환경 설정에서는 각 Node의 MAC 계층과 PHY 계층에 대한 환경 설정을 할 수 있고, Traffic 환경 설정에서는 출발지 노드와 목적지 노드, 패킷 크기, 전송 속도 등의 노드 간 Application Traffic에 대한 설정이 가능하다.

좌표	송신/수신 안테나	Node/Traffic
<ul style="list-style-type: none"> • 건물 • AP • STA 	<ul style="list-style-type: none"> • 파워 • 등방향성 • 주파수 • 수신감도 	<ul style="list-style-type: none"> • Path Loss • Application Layer • MAC Layer • PHY Layer • 패킷 크기 • 전송 속도 • ⋮

[그림17] 기능별 설정 종류

교내 건물을 실제와 유사하게 모델링하기 위해 실제 새벽별 도서관의 도면을 바탕으로 건물 좌표와 AP 좌표를 구했다. 그리고 시뮬레이션 실행을 원하는 시간대(평일/주말, 0~23시간)를 입력한 후 최대 이용자 수를 기준으로 앞서 정리했던 데이터를 받아와 모델링할 건물의 각 층의 AP명에 따라 AP에 연결된 이용자 수를 구해 수신 범위에서 STA들을 랜덤으로 배치했다.

2.3.2.4 실행

네트워크 시뮬레이션을 실행시키는 기능이다. 시나리오 내에서 이루어지는 AP와 STA 사이의 패킷 전송 과정을 담당하며 전송 과정 중 결과에 쓰이는 데이터들을 기록하는 기능을 한다.

다중 노드 WLAN 시스템에서의 네트워크 시뮬레이션 결과 그래프에서 기존 시뮬레이션은 각 AP의 Throughput과 Packet Loss Ratio, 각 STA의 Packet Latency를 그래프로 나타내었다. 그러나 본 과제에서의 제약 사항들 중 모든 STA이 일정 수준의 Throughput을 넘어야 한다는 것과 함께 STA기준으로 여러 결과를 도출하는 것이 최적화 전략을 찾을 때 더 적합하다고 판단해 각 STA의 Throughput, Packet Loss Ratio, Packet Latency에 추가적으로 Packet Delivery Ratio, Packet Error Ratio 그래프를 구현했다.

그리고 이러한 결과를 도출하기 위해 시뮬레이션에서 각 AP에서 전송한 패킷 수, 전송을 성공한 패킷 수, 전송을 실패한 패킷 수, 재시도한 패킷 수, 수신된 패킷 데이터 bytes를 수집했고, 각 AP에서 전송한 패킷 수는 전송 성공한 패킷 수와 전송을 실패한 패킷 수, 재시도한 패킷 수로 이루어져 있다는 것을 확인했다. 따라서 각 STA의 Throughput은 수신된 데이터 bytes에 총 시뮬레이션 시간을 나눠서 구하고, Packet Loss Ratio는 전체 전송된 패킷 수에서 성공한 패킷 수를 뺀 비율, Packet Delivery Ratio는 전체 패킷에서 전송 성공한 패킷의 비율, Packet Error Ratio는 전체 패킷에서 전송 실패한 패킷의 비율, Packet Latency는 해당 패킷이 전송 완료되기까지 시뮬레이션이 진행된 시간에서 패킷을 전송 시작할 때의 시간을 빼줌으로써 패킷 전송에 걸리는 시간을 구했다.

2.3.3 시뮬레이션 코드 과정

먼저 실제 환경과 비슷한 환경을 모사하기 위하여 실제 건물의 도면을 통해 새벽별 도서관의 벽과 바닥에 대한 좌표를 설정하고 각 꼭짓점을 이어 여러 개의 삼각형을 이어 붙인 면을 통해 시각화 하였다. 이렇게 건물의 외벽을 만든 후 실제로 AP가 설치되어 있는 위치와 동일하게 가상공간에도 AP를 위치시켰다.

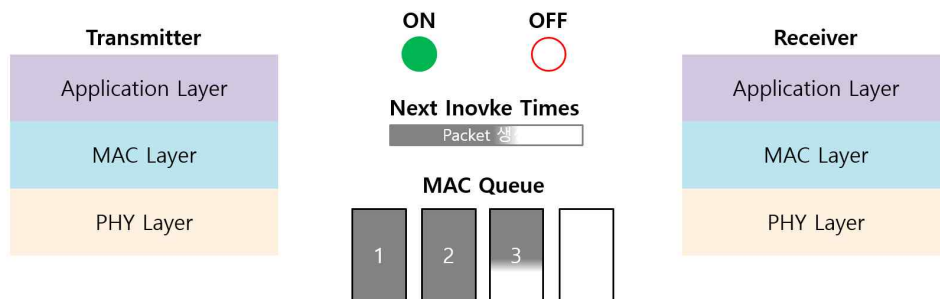
환경 모사를 통해 AP의 위치 설정을 하였다면 그 다음으로 송/수신 안테나를 설정해준다. 안테나 설정이 끝나면 Node와 Traffic에 대해 설정한다. 이때, 설정한 내용들과 환경 모사의 좌표 등을 이용하여 Path Loss를 고려하도록 한다. 마지막으로 데이터 전송을 위해 Application Layer, MAC Layer 그리고 Physical Layer를 생성한다. 각 계층이 만들어지고 나면 node별로 Channel

Manager을 설정해 준다.

모든 설정들이 끝나고 최종적으로 WLAN 노드들이 만들어지고 난 후 네트워크 시뮬레이션이 시작되고 시뮬레이션은 환경 설정에서 만든 각 계층을 Frame이 이동하며 그 과정을 데이터화 한다.

먼저, Application Layer에서 Packet을 생성하게 되는데, 이는 ON-OFF 상태에 따라 ON 상태에서 계속해서 Frame을 만들게 되고 OFF 상태는 더 이상의 Frame을 제작하지 않는다. ON이 동작하는 시간은 따로 설정이 가능하며 OFF는 일정 분포를 가지는 확률로 시간을 가지게 된다. Application Layer은 ON 상태에서 Next Invoke Time이라는 시간 간격마다 계속해서 Packet을 생성하고 생성된 Packet은 MAC Layer의 Queue에 전송하여 대기하게 된다.

MAC Layer은 데이터의 충돌을 막기 위하여 EDCA를 사용하고 있다. 따라서 AIFS시간 동안 대기하며 전송 순서를 판단하고 Back Off시간 동안 추가로 대기한 후 전송을 시작한다. 데이터의 전송이 시작되기 전, Channel Manger에서 AP와 STA의 개수를 통해 H행렬을 만들어 알맞은 전송 방식을 선택하여 준다. 그리고 MAC Layer에서는 PHY Layer의 상태에 따라 전송여부를 판단한다.



[그림18] Application Layer와 MAC Layer의 과정

데이터의 전송이 시작되면 MAC Queue안에 들어있던 Frame을 전송 Buffer(TxBuffer)에 넣어 주게 된다. 이를 추상화를 통해 Waveform으로 변환하여 전송하게 되고 도착한 Waveform은 수신 Buffer(RxBuffer)에 저장되어 Frame으로 다시 변환된다. 이때 전송과정의 시간정보를 각 MAC Layer로 전송하여 Transmitter의 전송 여부 판단과 Receiver의 수신 여부 판단을 진행한다. 위 과정을 각 노드별로 진행하며 마지막으로 그래프를 통해 데이터를 시각화 한다.

2.4 휴리스틱 알고리즘

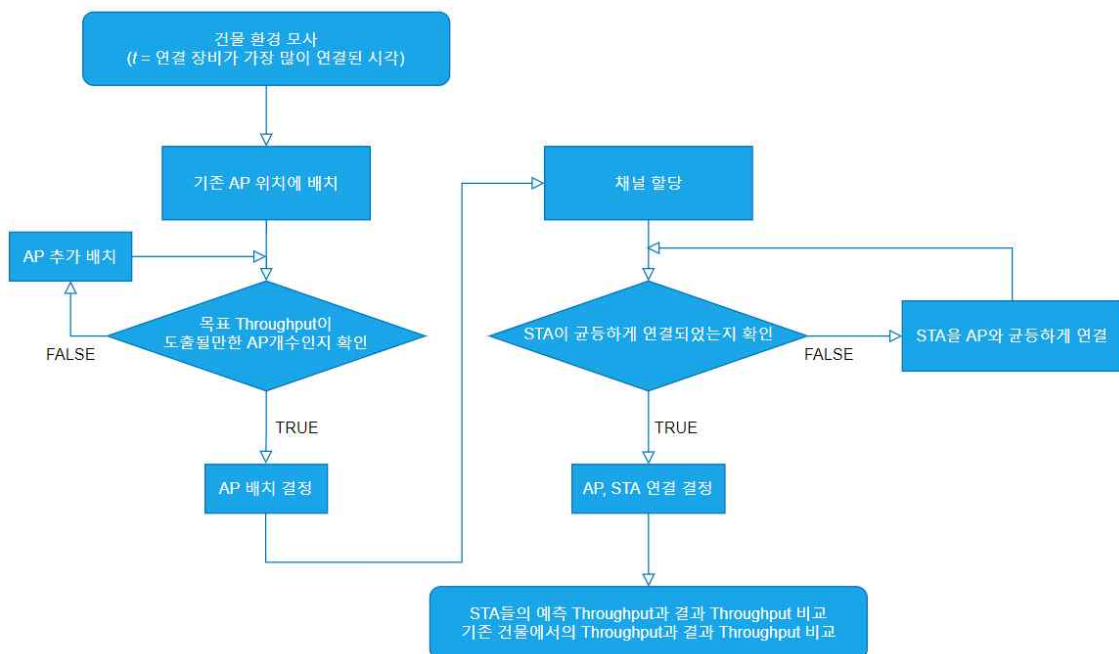
2.4.1 목표 처리율

	Recommended megabits per second
Broadband connection	1.5 Minimum: 0.5
Standard definition (SD)	3
High definition (HD)	5
4K/Ultra HD (UHD)	25

[그림19] 넷플릭스 비디오 시청 요구 전송 속도

위 그림은 넷플릭스에서 제공하는 비디오 시청 시 요구되는 전송 속도이고, 앞서 제시한 본 과제에서의 제약사항 중 모든 STA이 일정 수준의 처리율을 넘어야 한다는 사항에 대해 처리율의 기준을 정하기 위해 위의 자료를 참고했다. 위의 자료에 따르면 비디오를 시청할 때 요구되는 속도는 표준 옵션일 때 3Mbps, HD 옵션일 때 5Mbps 이다. 따라서 본 과제에서는 연결되는 장비가 많은 시간대에서의 처리율을 3Mbps 이상, 그렇지 않은 시간대에서는 5Mbps 이상의 처리율을 각 STA이 가져갈 수 있도록 하는 것을 목표로 했다.

2.4.2 알고리즘 동작 순서도



[그림20] 알고리즘 동작 순서도

본 과제에서 사용되는 휴리스틱 알고리즘은 추가 AP 배치, 채널 할당, AP와 STA 연결 알고리즘으로 총 3가지이다. 동작 순서는 건물에 대한 환경모사를 마친 후 기존 AP들을 배치하고 추가 AP 배치 알고리즘을 통해 목표 처리율을 만족시키지 못할 때 추가로 AP를 배치한다. 그리고 AP의 배치가 최종적으로 결정되고 나면 채널 할당 알고리즘을 통해 각 AP의 간섭 여부에 따라 채널을 할당한다. 마지막으로 AP와 STA 연결 알고리즘을 통해 간섭이 존재하는 부분의 STA을 STA이 적게 연결된 AP와 연결함으로써 최대한 한 AP에 STA이 집중되는 현상을 피하면서 AP와 STA 연결을 결정한다. 이러한 알고리즘들을 통해 무선랜 서비스의 자원 할당에 대한 최적화를 마친 후 최적화를 하기 전 환경에서의 처리율과 최적화를 마친 환경에서의 처리율을 비교해보고, 최적화가 이루어진 환경에서 예상되는 처리율과 실제 시뮬레이션의 결과로 도출된 처리율 또한 비교해본다.

2.4.3 추가 AP 배치

추가 AP 배치 알고리즘에서 추가로 AP를 배치하는 조건은 각 STA의 처리율이 목표하는 처리율보다 적을 때 추가 배치하는 것으로 한다. 그리고 각 STA의 처리율은 한 개의 AP에 연결된 STA 개수 ($numSTA$), 간섭이 존재하는 AP 개수($numInterferenceAP$), AP의 *Data rate*를 가지고 아래의 식에 따라 구한다.

$$Throughput\ at\ each\ STA = \frac{Data\ rate}{(numSTA * numInterferenceAP)}$$

그리고 추가 AP의 위치는 5m X 5m 단위의 그리드에서 STA들이 분포된 밀도를 구해 최대 밀도인 위치에 AP를 배치하는 것으로 한다. 따라서 추가 AP 배치 알고리즘의 pseudo code는 다음과 같다.

<i>datarate</i>	AP의 data rate
<i>numAPs</i>	전체 AP의 개수
<i>numSTAs</i>	전체 STA의 개수
<i>expectedThroughput</i>	목표하는 Throughput
<i>gridDensity</i>	5x5 단위 그리드에서의 STA이 분포된 밀도

Input : *datarate, numAPs, numSTAs, expectedThroughput, gridDensity.*

Output : *additional APs*

```

numSTAsPerAP  $\leftarrow$  numSTAs / numAPs
throughputPerSTA  $\leftarrow$  datarate / numSTAsPerAP

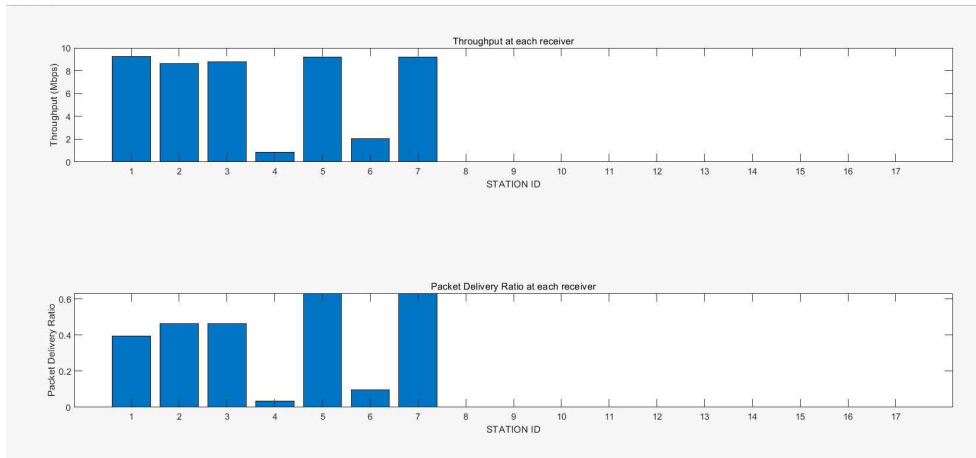
```

```

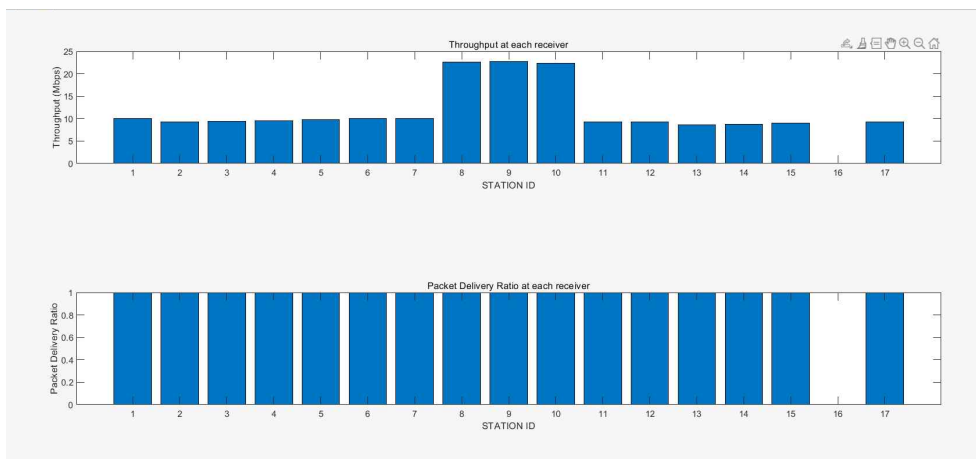
while throughputPerSTA < expectedThroughput do
    Place additional AP at the highest density in gridDensity
    numAPs  $\leftarrow$  numAPs + 1
    numSTAsPerAP  $\leftarrow$  numSTAs / numAPs
    throughputPerSTA  $\leftarrow$  datarate / numSTAsPerAP
end

```

2.4.4 채널 할당

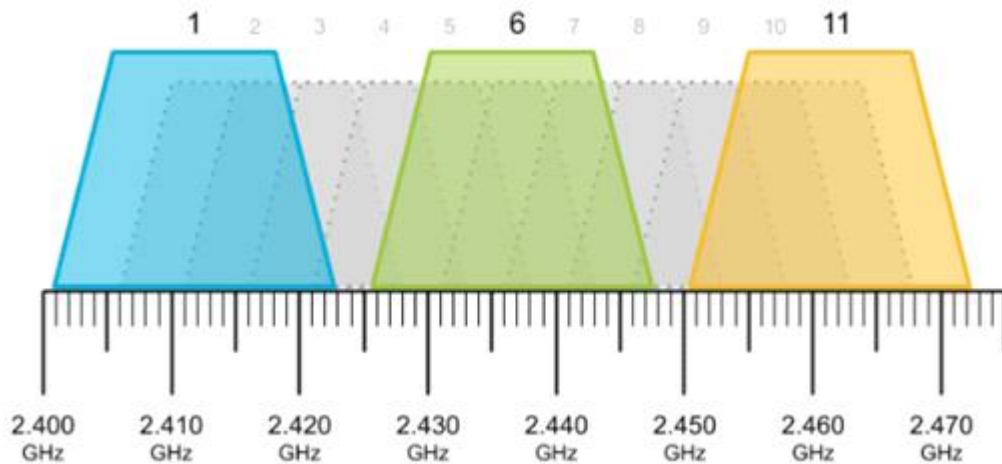


[그림21] 채널 분배를 하지 않았을 때 처리율과 PDR



[그림22] 채널 분배를 했을 때 처리율과 PDR

위의 그림은 채널 분배를 하지 않았을 때와 고르게 할당했을 때의 처리율과 Packet Delivery Ratio를 나타낸다. 두 개의 그래프를 비교했을 때 채널을 고르게 분배했을 때의 처리율과 PDR이 높은 것을 알 수 있다. 따라서 채널 할당 알고리즘은 최대한 채널의 간섭을 피해 고르게 채널을 할당하고 5GHz의 경우 채널 본딩을 고려한 채널 할당을 하는 것으로 한다.



[그림23] 2.4GHz 대역 채널 목록

먼저 AP 간의 거리를 통해 간섭 여부를 확인한 후 간섭이 존재하는 AP들의 목록을 구한다. 그리고 채널 간섭을 피하기 위해 2.4GHz 대역의 경우 위 그림과 같이 1, 6, 11 채널을 순서대로 사용한다.



[그림24] 5GHz 대역 채널 목록

5GHz 대역의 경우 36, 40, 44, 48, 52, 56, 60, 64 총 8개의 채널을 사용하는 데 이를 분배하기 위해 2.4GHz 대역 때와 같이 AP간의 거리를 통해 간섭 여부를 확인한다. 그리고 간섭이 존재하는 AP들의 목록을 구해 각 AP에 간섭이 존재하는 AP 개수에 따라 간섭이 존재하는 AP의 개수에 따라 2개 이하일 때는 80MHz로 4개의 채널을 묶고 3개 이상 4개 이하의 경우에는 40MHz로 2개의 채널을 묶는다. 그 이상의 경우에는 20MHz로 채널을 사용한다. 따라서 채널 할당 알고리즘의 pseudo code는 다음과 같다.

<i>numAPs</i>	전체 AP의 개수
<i>interferenceAPs</i>	각 AP당 간섭이 존재하는 AP 개수
<i>2.4GHzChannel</i>	2.4GHz 채널 목록
<i>5GHzChannel</i>	5GHz 채널 목록
<i>channelConfig</i>	각 AP당 채널 설정

채널 후보(*channel list* : 5GHz)

$5GHzChannel_1 = [\{36, 40, 44, 48\}, \{52, 56, 60, 64\}]$

$5GHzChannel_2 = [\{36, 40\}, \{44, 48\}, \{52, 56\}, \{60, 64\}]$

$5GHzChannel_3 = [\{36\}, \{40\}, \{44\}, \{48\}, \{52\}, \{56\}, \{60\}, \{64\}]$

채널 후보(*channel list* : 2.4GHz)

$2.4GHzChannel = [\{1\}, \{6\}, \{11\}]$

Input : *numAPs*, *interferenceAPs*

Output : *channelConfig*, *bandwidth*

$2.4GHzChannel \leftarrow [1, 6, 11]$

$5GHzChannel \leftarrow [36, 40, 44, 48, 52, 56, 60, 64]$

for *apIdx* = 1 to *numAPs* **do**

if 5GHz **then**

if *interferenceAPs(apIdx)* < 2 **then**

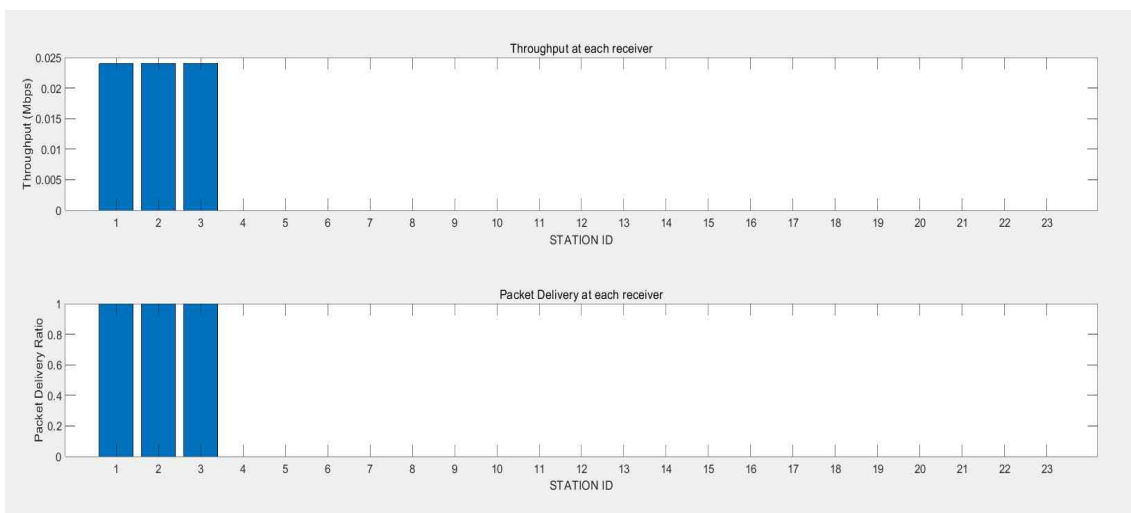
channelConfig(apIdx) $\leftarrow 5GHzChannel_1(1)$

```

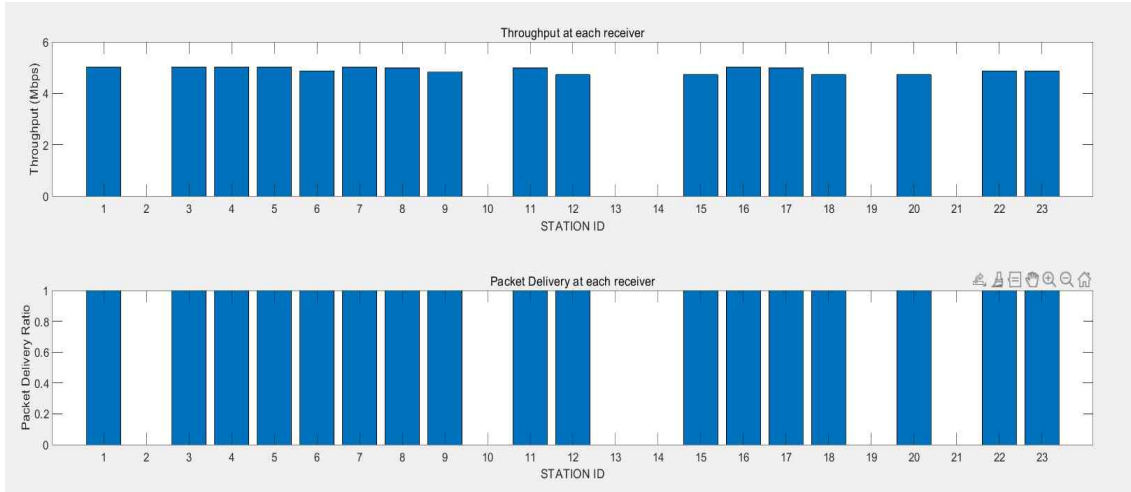
        bandwidth  $\leftarrow$  80MHz
    else if interferenceAPs(apIdx) < 4 then
        channelConfig(apIdx)  $\leftarrow$  5GHzChannel2(1)
        bandwidth  $\leftarrow$  40MHz
    else
        channelConfig(apIdx)  $\leftarrow$  5GHzChannel3(1)
        bandwidth  $\leftarrow$  20MHz
    end
else
    channelConfig(apIdx)  $\leftarrow$  2.4GHzChannel(1)
    bandwidth  $\leftarrow$  20MHz
end
remove subset of allocated Channels.
if channel list =  $\emptyset$  then
    Initialize the channel list to initial state
end
end

```

2.4.5 AP, STA 연결



[그림25] 특정 AP에 STA이 집중되어 있을 때 처리율과 PDR



[그림26] AP와 STA을 균등하게 연결했을 때 처리율과 PDR

위의 그림은 AP와 STA을 균등하게 연결하지 않았을 때와 균등하게 연결했을 때의 처리율을 나타낸다. 두 개의 그래프를 비교했을 때 STA을 균등하게 연결했을 때의 처리율이 약 5Mbps이고 STA이 특정 AP에 집중되어 있을 때 처리율이 약 0.025Mbps로 균등하게 연결했을 때의 처리율과 PDR이 높은 것을 알 수 있다. 따라서 AP, STA 연결 알고리즘은 한 개의 AP에 몰려있는 STA들을 최대한 균등하게 연결하는 것으로 한다. 따라서 간섭이 존재하는 범위 내에 배치되어있는 STA들을 간섭이 존재하는 AP들 중 STA이 적게 연결되어 있는 AP와 연결하여 최대한 STA을 균등하게 연결하며 알고리즘의 pseudo code는 다음과 같다.

<i>numAPs</i>	전체 AP의 개수
<i>numSTAs</i>	전체 STA의 개수
<i>isInterference</i>	각 STA이 간섭이 일어나는 커버리지 내에 있는지 여부
<i>trafficConfig</i>	AP와 STA 사이의 traffic 설정

Input : *numAPs*, *numSTAs*, *isInterference*, *trafficConfig*

Output : *trafficConfig*

for *apIdx* = 1 to *numAPs* **do**

for each *STA* in *trafficConfig* **do**

if *trafficConfig.SourceNode* = *apIdx* **then**

numSTAsPerAP(apIdx) \leftarrow *numSTAsPerAP(apIdx)* + 1

end

```

    end
end

for stIdx = 1 to numSTAs do
    if isInterference then
        Find APID of AP with fewer STAs among interference Aps.
        if trafficConfig.DestinationNode = stIdx then
            previousAPID ← trafficConfig.SourceNode
            numSTAsPerAP(previousAPID) ← numSTAsPerAP(previousAPID)
- 1
            trafficConfig.SourceNode ← APID
            numSTAsPerAP(APID) ← numSTAsPerAP(APID) + 1
        end
    end
end
end

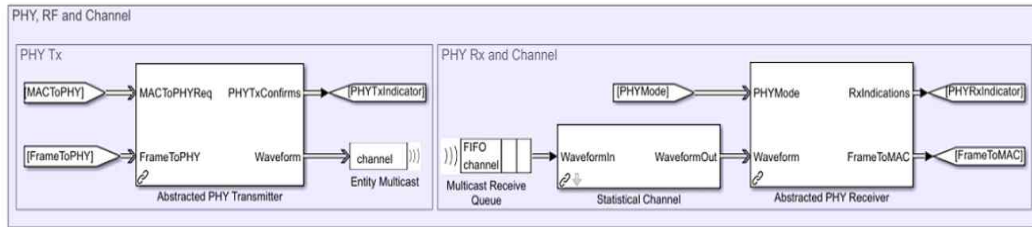
```

3. 개발 한계 – 산학협력 프로젝트

해당 MATLAB Toolbox를 사용하여 Single-User가 아닌 Multi-User 전송방식을 시도하였다. 가장 먼저 단순하게 Node 구성을 HE_MU로 설정하여 기타 다른 설정들이 자동적으로 호환이 되길 기대하였으나 설정 값들을 확인하는 과정에서 Multi-User 방식은 인식되지 않았다. 따라서 Node 뿐만 아니라 각 Layer의 설정 역시 마찬가지로 수정을 진행하였다.

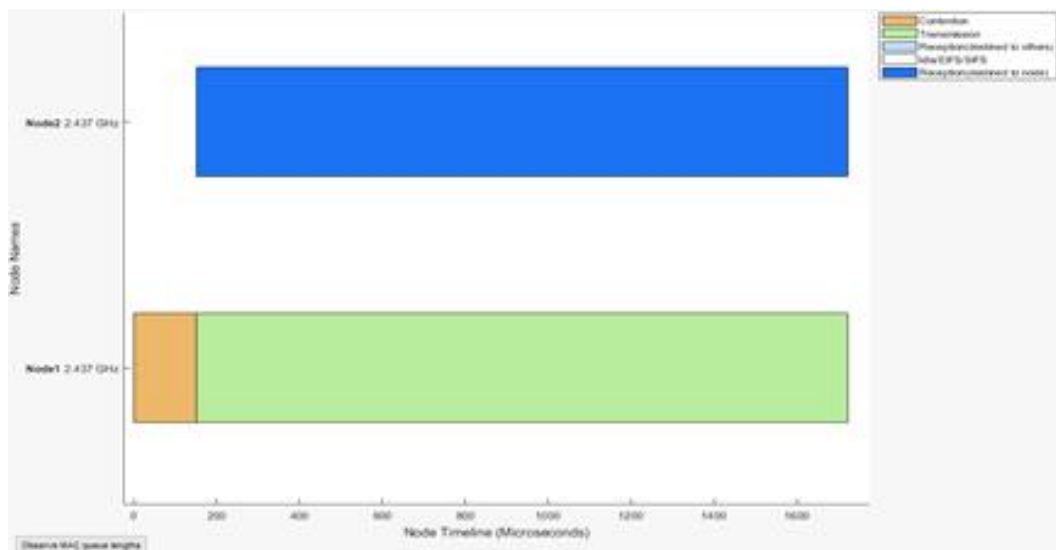
이후 Application Layer의 패킷 생성 과정에서 Invoke Time을 구하지 못하는 오류가 발생하였다. 여러 Layer들의 상호작용에서 구성이 잘 못 되어있다 판단하여 해당 Toolbox의 참조 실험들을 확인하였다. 크게 시각화 과정, Layer 생성과정 그리고 추상화를 통한 패킷 전송과정, 총 세 가지 실험을 기반으로 하고 있다. 따라서 Layer 생성 과정의 Multi-User 부분을 추가 및 수정하였고 기본적인 환경 설정 및 패킷의 생성까지는 성공적으로 구현하였다.

위 방법까지는 성공하여 PHY Layer에서의 정상적인 전송 시도를 확인했다. 아래 그림과 같이 Transmitter에서 waveform으로 변환하여 Receiver로 성공적으로 송신을 시도하였다. 하지만 Receiver에서 받은 waveform들을 Multi-User 방식으로 읽어드릴 수 없었다. 따라서 송신 부분에 대해서는 Multi-User 방식이 정상적으로 작동하였지만 수신 부분에서 오류가 발생하며 시간 관계상 더 이상의 Multi-User 설정을 진행 할 수 없었다.

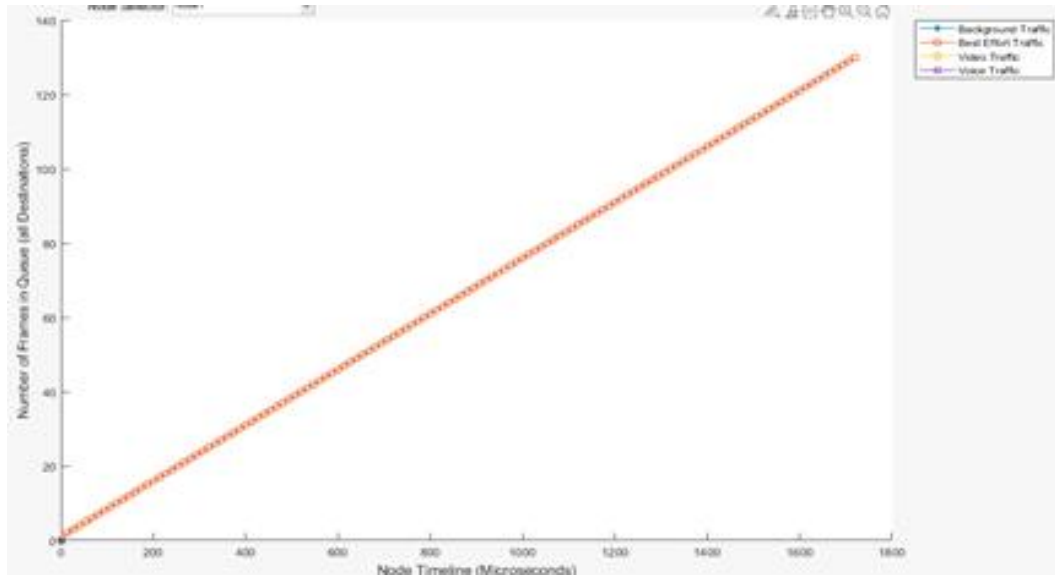


[그림27] 송/수신 추상화 과정

또한 산학 협력 프로젝트 멘토 의견에 의해 새로운 환경에서 시행착오를 줄일 수 있는 시스템 구축을 위하여 이용자 동선 혹은 사용자 패턴 등을 사전 예측하여 반영할 수 있는 모델을 시도 했다. 해당 모델을 위해 강화 학습을 활용해 최적화 전략을 찾아보고 싶었지만 실행 시간이 오래 걸리는 이슈를 완벽하게 해결하지 못해 강화 학습을 활용하기에는 힘들 것이라고 판단하였다.



[그림28] Node간의 전송 진행 상황



[그림29] Queue에 대기 중인 프레임 수

실행 시간 이슈의 원인을 찾기 위해 시뮬레이션 과정을 살펴본 결과, 시뮬레이션의 전송 과정을 담당하는 run() 함수에서 전송 진행 상황에 비해 프레임을 만드는 주기가 너무 짧아 전송 중에 계속해서 새로운 프레임을 만들어 Queue에 대기하고 있는 것이 원인이라고 판단되었고, 위 그림의 그래프들에서 프레임을 1개 전송하는 동안 프레임이 계속 생성되면서 Queue에 약 130개의 프레임이 대기 중인 것을 확인할 수 있었다.

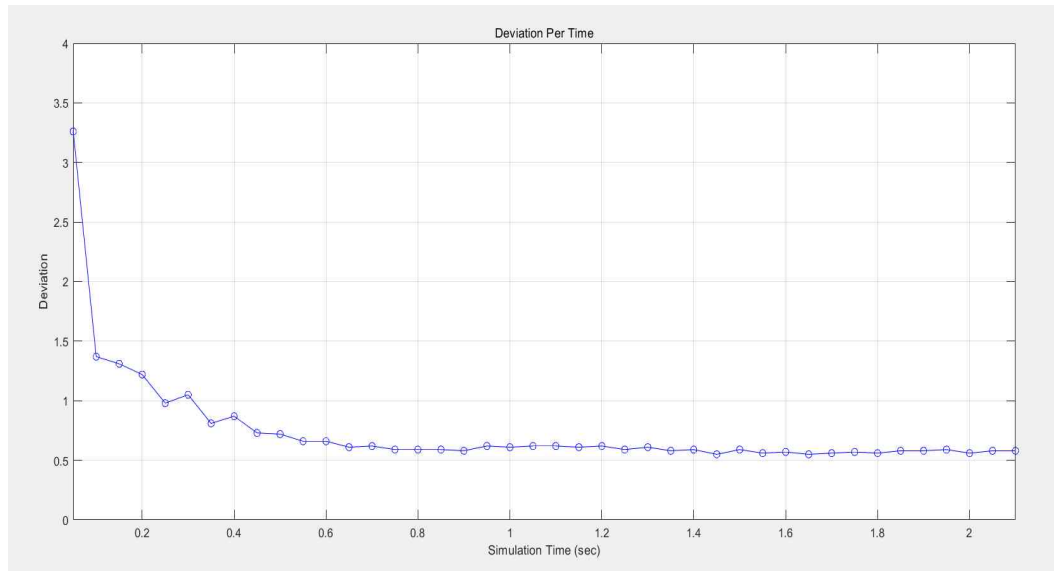
그리고 시뮬레이션이 노드 설정 부분과 전송 과정에서 for문으로 노드 하나 하나의 설정과 전송을 하는 구조로 되어있어서 이것 또한 연결된 STA이 많아 질수록 실행 시간이 오래 걸리는 원인이라고 판단했다.

그래서 이러한 원인들을 해결하기 위해 먼저 프레임 전송 주기를 조절하여 프레임 1개를 전송하는 동안 Queue에 대기하는 프레임 수를 6~7개까지 줄이는 것은 성공했으나 전체 시뮬레이션 실행 시간은 크게 변함이 없었다. 그리고 다수의 STA에 의한 실행 시간을 줄이기 위해서는 병렬 연산을 활용하면 실행 시간을 크게 줄일 수 있겠다고 생각했고, 매트랩에서 지원하는 Parallel Computing Toolbox의 함수들을 통해 병렬 연산을 활용했다. 각 노드들의 설정 부분에서는 노드들의 설정을 독립적으로 만들어 병렬 연산을 활용해 시간을 줄이는 데에 성공했으나 여러 노드들이 서로 엮여있는 전송 과정의 for문에서는 각 노드들을 독립적으로 만드는 데 실패하여 실행 시간을 크게 줄이는 것에는 실패했다. 그래서 강화 학습 대신 휴리스틱 알고리즘을 활용해 최적화 전략을 수립하였다.

4. 연구 결과 분석 및 평가

4.1 시뮬레이션 실행 시간

앞서 말한 개발 한계에서의 실행 시간 이슈로 인해 실행 시간을 최대한 줄일 필요가 있었다.



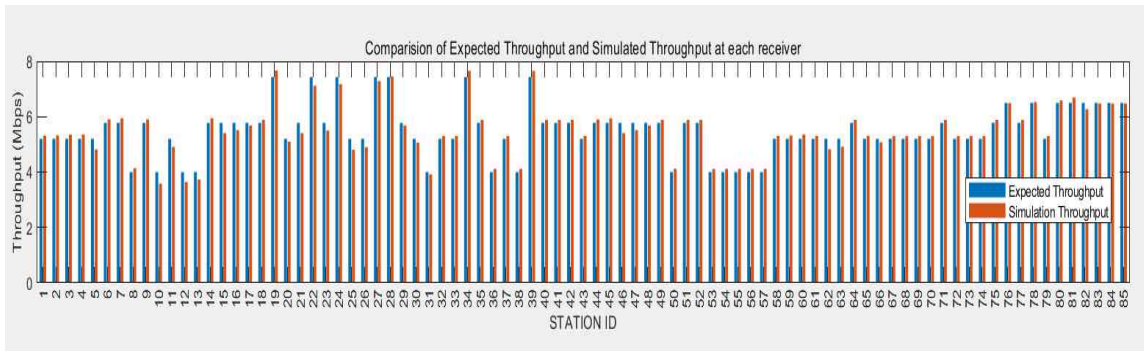
[그림30] 시뮬레이션 시간별 처리율 편차 그래프

그래서 실행 시간을 최소로 하면서 시뮬레이션의 결과를 만족스럽게 가져가기 위해 시뮬레이션 실행 시간별로 STA의 처리율의 편차를 구해 처리율의 편차가 가장 작을 때의 실행 시간이 전송 과정에서 모든 STA이 전송 기회를 어느 정도 보장받는 시간이라고 생각해 위의 그래프와 같이 시간을 점차 늘려가며 처리율의 편차 그래프를 그렸다. 그래프를 보면 0.8초 이후부터 편차가 0.5Mbps 내외로 유지된다는 것을 알 수 있고, 따라서 시뮬레이션 실행 시간을 0.8초로 설정했다.

4.2 연구 결과

무선랜 서비스의 자원 할당 최적화 전략을 수립하는 기존의 다른 연구들에서는 한 공간내에 다수의 STA을 두고 시뮬레이션하는 방식으로 연구를 진행했었다. 그러나 본 과제는 일반적인 육면체 공간이 아닌 실제 건물을 모델링해 환경 모사를 한 후 실제 환경에서의 시뮬레이션을 해봄으로써 좀 더 실제 환경에 적합한 최적화 전략을 제시할 수 있다는 점에서 기존 다른 연구들과는 차별적인 특징을 가지고 있다.

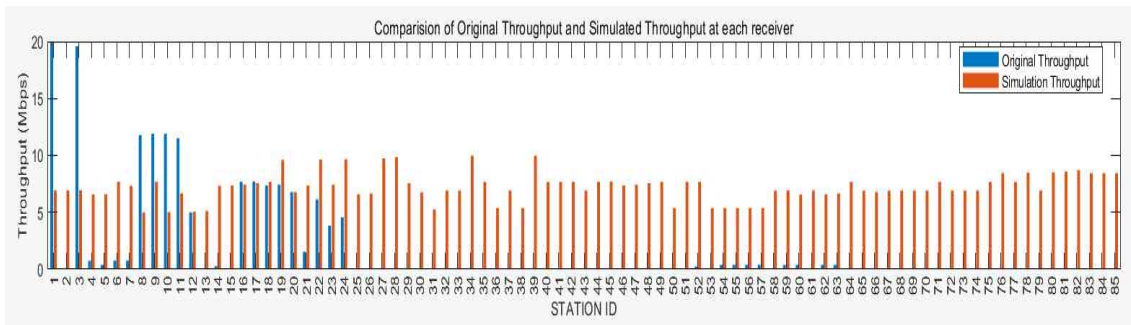
위 그래프는 최적화 알고리즘을 적용했을 때의 예상되는 처리율과 실제 시



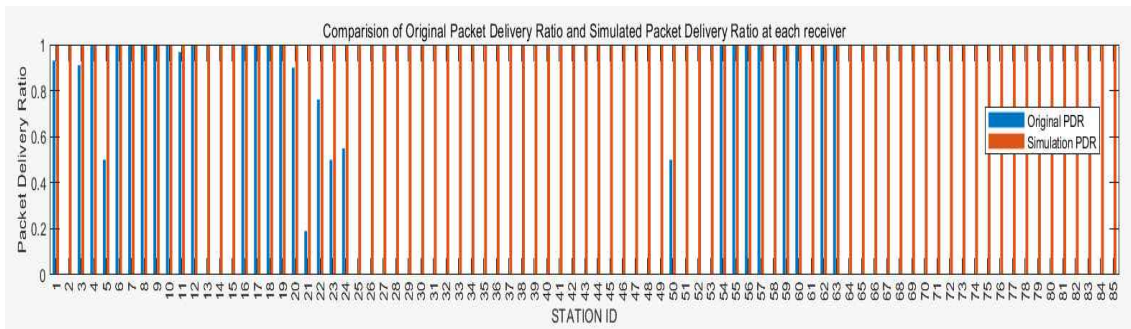
[그림31] 예상 처리율과 시뮬레이션 결과 처리율

물레이션의 결과로 도출된 처리율을 비교한 그래프이다. 예상 처리율은 아래 식에 따라 구했고, 예상 처리율과 결과 처리율의 오차를 계산해 보았을 때 0.14Mbps 정도로 실제 시뮬레이션의 결과 처리율이 예상 처리율과 매우 비슷하게 나온다는 것을 알 수 있었다.

$$\text{Throughput at each STA} = \frac{\text{Data rate}}{(\text{numSTA} * \text{numInterferenceAP})}$$



[그림32] 기존 환경과 최적화 알고리즘 적용 후 환경에서의 처리율



[그림33] 기존 환경과 최적화 알고리즘 적용 후 환경에서의 PDR

위 그래프는 기존 환경에서의 처리율과 PDR, 앞서 제시한 알고리즘들을 적용한 환경에서의 처리율과 PDR을 비교한 그래프이다. 그래프를 보면 기존 환

경에서는 알고리즘에서 제시한 목표 처리율 이상을 가져가는 STA의 수가 전체 STA들 중 15% 미만이라는 것을 확인할 수 있었고, STA들의 평균 처리율을 계산했을 때 1.65Mbps라는 결과가 나왔다. 그리고 알고리즘을 적용해 최적화를 마친 후의 환경에서는 전체 STA들 중 모든 STA들이 목표 처리율 이상을 가져가고 있다는 것을 확인할 수 있었고, 알고리즘에 따라 추가 배치된 AP가 1개임에도 평균 처리율은 기존 환경에서의 평균 처리율보다 약 5배 정도 증가한 8.15Mbps라는 결과가 나왔다. 따라서 본 과제의 최적화 알고리즘이 기존 무선랜 서비스의 개선에 효과적이라는 것을 알 수 있다.

5. 결론

5.1 활용 방안

본 팀이 제시한 결과는 부산대학교 장전캠퍼스의 PNU-WiFi를 연구한 결과이지만 학교뿐만 아니라 사람들이 몰리는 공공장소라면 모두 적용이 가능하다. 학교라는 특성상 네트워크 이용량이 다른 장소에 비해 높을 수 있지만, 다양한 시설들에서 사물인터넷과 더불어 사회의 발전에 따라 네트워크 사용량은 계속해서 늘어난다. 따라서 다양한 곳에서 해당 무선랜 서비스 품질에 대한 연구가 필요로 한다.

5.2 향후 개발 방향

본 연구를 통해 싱글 유저에 대한 통신은 이루어 졌다. 하지만 WiFi6의 특성을 고려하여 더욱 유사한 환경 모사를 위해 멀티유저 설정이 적합하다. 또한 이번 연구의 휴리스틱 알고리즘을 사용하여 강화학습까지 적용한다면, 본 연구결과보다 효과적인 개선방안을 도출해낼 수 있다.

추가로 AP에 접속하는 단말 수량뿐만 아니라 트래픽 사용량에 큰 차이가 있는 단말 종류(스마트폰, PC 등)에 따른 선별이 가능하다면 더욱 효과가 클 것이다. 그리고 제안한 개선방안들을 AP 배치 및 환경 구성에 이용하는 것뿐만 아니라, 사용자들에게 실시간으로 현황을 나타낼 수 있는 툴을 개발하는 것은 추가적인 문제점을 발견하거나 개선하는데 있어 필요로 한다.

6. 수행 체계

6.1 구성원별 역할

이름	역할	
곽민수	• MATLAB을 이용한 부산대학교 교내 건물 환경 구현	O
	• MATLAB을 이용한 부산대학교 교내 AP 환경 구현	O
	• Multi User를 활용한 패킷 전송 구현	X
김도형	• 시뮬레이션 동작 및 결과 시각화	O
	• 시뮬레이션 동작 시간 개선	O
	• 알고리즘 구현	O
이예경	• PNU WiFi 사용 현황 데이터 수집 및 정리	O
	• 시뮬레이션에 실제 데이터 적용	O
	• 부산대 건물 좌표 데이터화	O

6.2 개발 일정

5월		6월				7월					8월				9월			
3주	4주	1주	2주	3주	4주	1주	2주	3주	4주	5주	1주	2주	3주	4주	1주	2주	3주	4주
WiFi 6 및 관련 지식 스터디																		
교내 건물 구현																		
		PNU WiFi 데이터 정리																
		교내 건물 시뮬레이션																
				시뮬레이션 결과 분석														
						중간보고서 작성												
									최적화 알고리즘 구현									
									알고리즘 적용 및 시뮬레이션									
											성능 평가 및 비교							
													문제점 파악 및 오류 수정					
															최종 발표/보고서 준비			

7. 참고 문헌

[1] Y. Kim, H. Yoon, E. Lee and J. Kim, "Analysis of AP status in a large scale wireless LAN environment on campus scope", Journal of Pusan National University : Computer Science and Engineering Technical Report 2020, pp.1-25, 2021. (in Korean)

[2] 802.11ax Multinode System-Level Simulation of Residential Scenario Using MATLAB [Online]. Available:
https://kr.mathworks.com/help/wlan/ug/802-11ax-multinode-system-level-simulation-of-residential-scenario-using-matlab.html#mw_rtc_WLANResidentialScenarioExample_M_006B97DF

[3] Physical Layer Abstraction for System-Level Simulation [Online]. Available:
https://kr.mathworks.com/help/wlan/ug/physical-layer-abstraction-for-system-level-simulation.html?searchHighlight=TGax%20simulation%20scenarios&s_tid=srchtitle

[4] 802.11ax System-Level Simulation with Physical Layer Abstraction [Online]. Available: https://kr.mathworks.com/help/wlan/ug/802-11ax-system-level-simulation-with-physical-layer-abstraction.html?searchHighlight=11ax%20evaluation%20methodology&s_tid=srchtitle

[5] Qiu, Shuwei, et al. "Joint Access Point Placement and Power-Channel-Resource-Unit Assignment for 802.11 ax-Based Dense WiFi with QoS Requirements." IEEE INFOCOM 2020-IEEE Conference on Computer Communications. IEEE, 2020.

[6] Siomina, Iana, and Di Yuan. "Optimization of channel assignment and access point transmit power for minimizing contention in Wireless LANs." 2007 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops. IEEE, 2007.

[7] Riedi, Marcelo, Giovanna G. Basilio, and Marcelo E. Pellenz. "Channel and power allocation algorithm to optimize the performance of large WLANs." Proceedings of the 30th Annual ACM Symposium on Applied Computing. 2015.

[8] Liu, Yongqiang, et al. "Measurement-based channel management in WLANs." 2010 IEEE Wireless Communication and Networking Conference. IEEE, 2010.

8. 부록

❖ wlanNodeConfig (MAC Layer)

- **NodePosition**
 - Description: Position of the node
 - Value: Vector of [x,y,z] format, in units of meters
 - Default: [0 0 0]
- **TxFormat**
 - Description: Transmitting packet format
 - Value: Accepts "NonHT", "HTMixed", "VHT", "HE_SU", "HE_EXT_SU". You can also set the value to the enum type from hFrameFormatsEnum.
 - Default: "HE_SU"
- **Bandwidth**
 - Description: Channel bandwidth
 - Value: Accepts 20, 40, 80, or 160.
 - Default: 20
- **TxMCS**
 - Description: Specifies the modulation and coding scheme (MCS) index that is used for transmitting the frame. This value applies only when RateControl is set to "FixedRate".
 - Value: When TxFormat is set to "HE_SU", accepts numbers in the range [0,11]. When TxFormat is set to "VHT" and NumTxChains is not 3 or 6, accepted range is [0,9]. When TxFormat is set to "VHT" and NumTxChains is 3 or 6, accepted range is [0,8]. When TxFormat is set to "HTMixed" or "NonHT", accepted range is [0,7]. When TxFormat is set to "HE_EXT_SU", accepted range is [0,2]. For "HTMixed", the MCS value input in the range [0,7] is automatically mapped to the index in the range [0,31] based on the NumTxChains value.
 - Default: 7
- **NumTxChains**
 - Description: Number of transmit chains.
 - Value: When TxFormat is set to "VHT" or "HE_SU", accepts numbers in the range [1,8]. When TxFormat is set to "HTMixed", accepted range is [1,4]. When TxFormat is set to "HE_EXT_SU", accepted range is [1,2]. When TxFormat is set to "NonHT", only 1 transmit chain is allowed.
 - Default: 1

-
- **MPDUAggregation**
 - Description: Flag that enables MPDU aggregation when set to true. Applies only when TxFormat is set to "HTMixed" format.
 - Value: A logical scalar (true or false)
 - Default: true
 - **DisableAck**
 - Description: Flag indicating the transmitter does not solicit acknowledgment for the frame.
 - Value: A logical scalar (true or false)
 - Default: false
 - **MaxSubframes**
 - Description: Specifies the maximum number of subframes that can be aggregated in the A-MPDU.
 - Value: Accepts a number in the range [1,256].
 - Default: 64
 - **RTSThreshold**
 - Description: Specifies the threshold length of the frame after which RTS/CTS protection is used for data transmission. Applies only when DisableRTS is set to false.
 - Value: Accepts a number in the range [0,65536].
 - Default: 65536
 - **DisableRTS**
 - Description: Flag that disables RTS/CTS exchange for all the data transmissions when set to true.
 - Value: A logical scalar (true or false)
 - Default: false
 - **MaxShortRetries**
 - Description: Specifies the retry limit for frames less than RTS threshold.
 - Value: Accepts a number in the range [0,32].
 - Default: 7
 - **MaxLongRetries**
 - Description: Specifies the retry limit for frames greater than RTS threshold.
 - Value: Accepts a number in the range [0,32].
 - Default: 7
 - **BasicRates**
 - Description: Set of data rates representing basic rate set.
 - Value: Accepts a vector of data rate values. The values must be from the set [6, 9, 12, 18, 24, 36, 48, 54].
 - Default: [6 12 24]
 - **Use6MbpsForControlFrames**
 - Description: Flag that forces to use 6 Mbps data rate for all control frames ignoring the values in BasicRateSet.
 - Value: A logical scalar (true or false)
 - Default: false
 - **BandAndChannel**
 - Description: Operating band and channel number
 - Value: A cell array of vector in the format [x, y] where x = band, y = channel number. The value of x can be 2, 4, 5, or 6. The value of y can be any valid channel number.
 - Default: [2, 4, 6]
 - **CWMin**
 - Description: Minimum value for the contention window range for each access category.
 - Value: A row vector of size 4, indicating the CWMin in values for four access categories. Each value in the row vector must be in the range [1,1023].
 - Default: [15 15 7 3]
 - **CWMax**
 - Description: Maximum value for the contention window range for each access category.
 - Value: A row vector of size 4, indicating the CWMax values for four access categories. Each value in the row vector must be in the range [1,1023].
 - Default: [1023 1023 15 7]
 - **AIFSSlots**
 - Description: Number of arbitrary interframe space (AIFS) slots for each access category.
 - Value: A row vector of size 4, indicating the number of AIFS slots for four access categories. Each value in the row vector must be in the range [2,15].
 - Default: [3 7 2 2]
 - **RateControl**
 - Description: Rate control algorithm to use
 - Value: Accepts either "FixedRate" or "ARF"
 - Default: "FixedRate"
 - **PowerControl**
 - Description: Power control algorithm to use
 - Value: Accepts only "FixedPower"
 - Default: "FixedPower"
 - ❖ **wlanNodeConfig(PHY Layer)**
 - **TxPower**
 - Description: Transmit power
 - Value: A scalar number representing the transmit power in dBm
 - Default: 15
 - **TxGain**
 - Description: Transmit gain
 - Value: A scalar number representing the transmit gain in dB
 - Default: 1

-
- **RxGain**
 - Description: Receive gain
 - Value: A scalar number representing the receive gain in dB
 - Default: 0
 - **EDThreshold**
 - Description: Energy detection threshold value
 - Value: A scalar number representing the threshold in dBm
 - Default: -82
 - **RxNoiseFigure**
 - Description: Receiver noise figure
 - Value: A scalar number representing the noise figure in dB
 - Default: 7
 - **ReceiverRange**
 - Description: Packet reception range
 - Value: A scalar number representing the range in meters
 - Default: 1000
 - **FreeSpacePathloss**
 - Description: Flag that enables free space pathloss when set to true
 - Value: A logical scalar (true or false)
 - Default: true
 - **PHYAbstractionType**
 - Description: Type of PHY abstraction
 - Value: Accepts either "TGax Simulation Scenarios MAC Calibration" or "TGax Evaluation Methodology Appendix 1"
 - Default: "TGax Evaluation Methodology Appendix 1"
- ❖ **wlanTrafficConfig (Application)**
- **SourceNode**
 - Description: ID of the source node transmitting the packet, at which the application is running. To configure multiple transmitters in the network, this structure must be replicated and the structures should have different SourceNode values
 - Value: A scalar number less than or equal to number of nodes in the network
 - Default: 1
 - **DestinationNode**
 - Description: ID of the destination node to which the packet is intended. To transmit packets destined to two different nodes from the same source node, replicate the structure two times, and the two structures in the array must contain the same value for SourceNode and different values for DestinationNode.
 - Value: A scalar number less than or equal to number of nodes in the network
 - Default: 4
 - **PacketSize**
 - Description: Size of the generated application packets
 - Value: A scalar number in the range [1,2034]
 - Default: 1500
 - **DataRateKbps**
 - Description: Rate at which application packets are generated
 - Value: A scalar number representing the data rate in Kbps
 - Default: 600000
 - **AccessCategory**
 - Description: Access category where 0 represents best-effort traffic (BE), 1 represents background traffic (BK), 2 represents video traffic (VI), 3 represents voice traffic (VO).
 - Value: A scalar number in the range [0,3]
 - Default: 0