

AWS Ground Station과 MATLAB을 활용한 대용량 데이터의 효율적인 위성 통신 및 비용 계산

201827146 조민기



Agenda

①연구 목표

②본론; 연구 과정

③연구 결과

④개선 방안



연구 목표

AWS Ground Station의 사용 시간을 예측

→ 50Mhz 의 Narrowband 선정



Narrowband
(≤54MHz)

Wideband
(>54 MHz)

On
Demand

\$10/min

\$22/min

Reserved

\$3/min

\$10/min

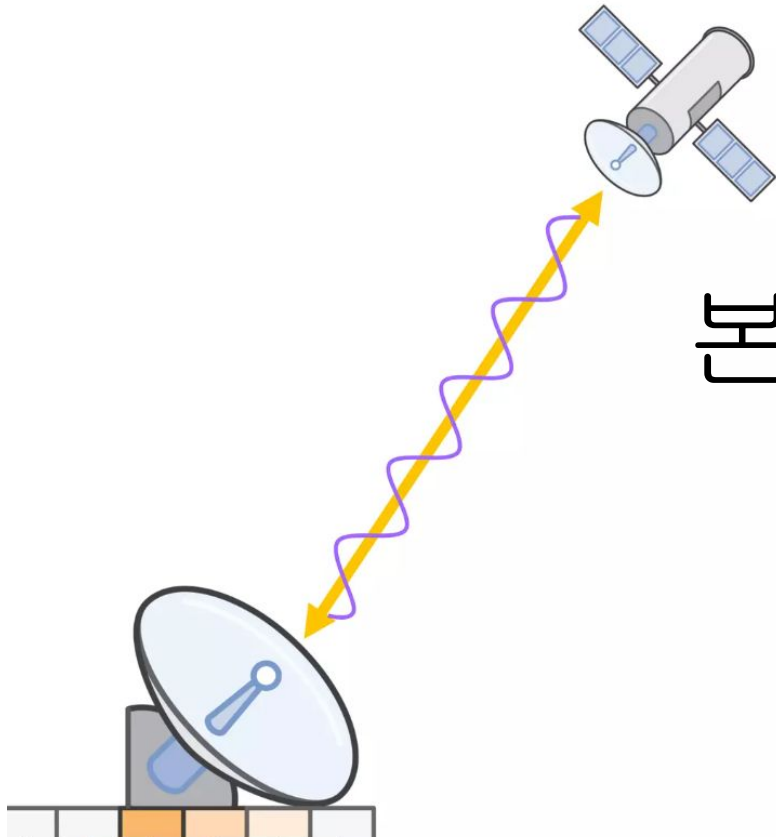
위성이 전송하려는 데이터의 크기(100MB)에 따른

① 다중 지상국(11여개)를 이용하여 가장 빨리 전송하는 시나리오

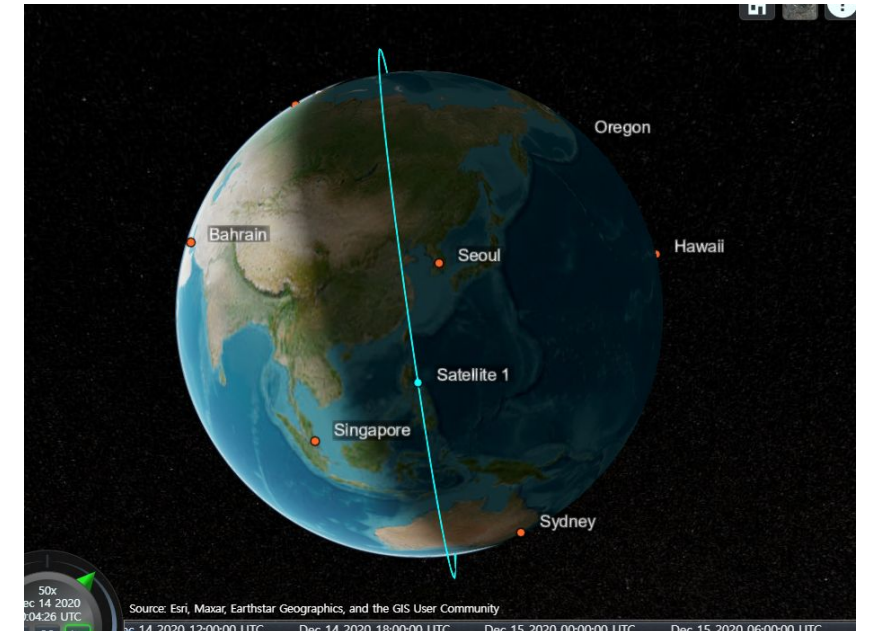
② 단일 지상국을 이용하여 가장 빨리 전송하는 시나리오

에 따른 각 시나리오와 그에 따른 시간 도출

본론 : 연구 과정



AWS Ground Station Location 맵핑



각 도시의 위/경도를 도입, 해발 0m

AWS Ground Station Antenna Spec.

Source : NASA Earth Science Technology Office

AWS Ground Station Specifications Sheet		
For the 5.4m Antenna Systems		
Tracking		
Orbit	LEO > 250 km, GEO	
Inclinations	All	
Pointing	0.05° rms BRE	
Bands	X-Band	S-Band
Polarization		
Receive	Simultaneous LHCP & RHCP	Selectable LHCP/RHCP
Transmit	N/A	Selectable LHCP/RHCP
Frequency		
Receive	7750 to 8400 MHz	2200 to 2300 MHz
Transmit	N/A	2025 to 2120 MHz
G/T	30.5 dB/K	16.0 dB/K
EIRP	N/A	53 dBW
Bandwidth	125 kHz to 500 MHz	125 kHz to 54 MHz
Channels		
Receive	2	1
Transmit	N/A	1
Symbol Rates		
PSK	10 Msps to 600 Msps x 2 channels	7 sps to 5.0 Msps
DVB-S2	10 Msps to 250 Msps x 2 channels	N/A
Digital Interface		
Digital Stream Format	VITA-49 Demodulated / Decoded Data; OR VITA-49 RF-over-IP	VITA-49 RF-over-IP
Modulation	BPSK, QPSK, SQPSK, UAQPSK, OQPSK, 8PSK, 16/32 APSK, MSK, 16QAM	GMSK, BPSK, QPSK, SQPSK, UAQPSK, OQPSK
Forward Error Correction	Convolutional, Interleaving, Reed-Solomon, LDPC, 4D-8PSK-TCM, Viterbi	Viterbi, Reed-Solomon
Data Processing	CCSDS frame processing (VCPU/APID), Programmable frame syncs	CCSDS, CORTEX

Pusan National University
Aerospace Engineering



시뮬레이션 환경설정 - 지상국 매핑

```
% 지상국 및 안테나 설정
latitudeArray = [36.34067747841352, 1.4028769359084685, -33.85524717359871, 25.981557777777777, 59.329324777777777, 37.566595444444444, 33.449833333333333, 33.449833333333333, 33.449833333333333, 33.449833333333333, 33.449833333333333, 33.449833333333333];
longitudeArray = [127.67264631378752, 103.78536794765901, 150.9492324832868, 50.543611111111111, 15.871888888888889, 121.48333333333333, 121.48333333333333, 121.48333333333333, 121.48333333333333, 121.48333333333333, 121.48333333333333, 121.48333333333333];
gsNames = ["Seoul", "Singapore", "Sydney", "Bahrain", "Stockholm", "Ireland", "Cape Town", "Cape Town", "Cape Town", "Cape Town", "Cape Town", "Cape Town"];

for i = 1:length(latitudeArray)
    gs = groundStation(sc, latitudeArray(i), longitudeArray(i), "Name", gsNames(i));
    gimbalGs = gimbal(gs, "MountingAngles",[0;180;0], "MountingLocation",[0;0;-5]);

    % S-band Transmitter 설정 (업링크)
    gsTx = transmitter(gimbalGs, "MountingLocation" ...
        ,[0;0;1], "Frequency",2.2e9, "Power",15); % S-band
    gaussianAntenna(gsTx, "DishDiameter",5); % meters

    gsRx = receiver(gimbalGs, "MountingLocation" ...
        ,[0;0;1], "GainToNoiseTemperatureRatio",3, "RequiredEbNo",1);
    gaussianAntenna(gsRx, "DishDiameter",5); % meters

    % 안테나 포인팅 설정
    pointAt(gimbalGs, sat1);
    pointAt(gimbalSat1Rx, gs);

    pointAt(gimbalSat1Tx,gs);
    pointAt(gimbalGs,sat1);
end
```

11 여곳 GroundStation Data

주어진 안테나 최소 스펙을 활용하여
pointAt 메서드로 인공위성과 지상국
매핑 반복문 수행

궤도 선정

Source : 태양동기-지상반복 궤도를 활용한 군 정찰용 초소형 위성군 설계 Journal of the KIMST, Vol. 23 March, 2020

Table 4. Comparison of Walker-Delta, KOMPSAT series and BB constellation

		WD1	WD2	KOMPSATs	BB
위성군	형태	워커-델타(45°:20/4/1)	워커-델타(45°:16/8/1)	독립적 태양동기궤도	태양동기-지상반복
	위성 수	20	16	3	19
재방문 주기 (분:초)	최소	02:35	14:29	00:22	30:00
	평균	23:54	29:08	279:06	30:00
	최대	121:24	50:43	784:09	30:00
	표준편차	25:33	14:26	201:09	00:00
관측 소요시간 (분:초)	최소	00:17	00:03	00:45	02:21
	평균	02:12	02:16	04:11	02:31
	최대	03:00	03:00	09:40	02:41
	표준편차	00:45	00:43	01:33	00:08
일평균 관측횟수(회)		21.44	18.44	5.07	19

Add the Satellites

Use satellite to add Satellite 1 and Satellite 2 to the scenario

```
semiMajorAxis = 6937800;           % meters
eccentricity = 0;
inclination = 98;                   % degrees
rightAscensionOfAscendingNode = 0; % degrees
argumentOfPeriapsis = 0;           % degrees
trueAnomaly = 0;                   % degrees
```

고도 566.8km

경사각 98도

태양 동기궤도

위성 스펙


Source : SAR microSAT – ICEYEX1 ; eoPortal

S-band 32 kbit/s (업링크)
X-band 50 Mbit/s (다운링크)

안테나 위치 [0m, 0.65m, 0m]

전송할 파일 크기 100MB

최대 전송 대역폭 50Mhz



```
% 위성 안테나 설정
gimbalSat1Tx = gimbal(sat1, "MountingLocation",[0;0.65;0]);
gimbalSat1Rx = gimbal(sat1, "MountingLocation",[0;0.65;0]);

% X-band 송신기 설정 (다운링크)
sat1Tx = transmitter(gimbalSat1Tx, "MountingLocation", ...
    [0;0;1], "Frequency",8.4e9, "Power",15); % X-band
gaussianAntenna(sat1Tx, "DishDiameter",0.5); % meters

% X-band 수신기 설정 (업링크)
sat1Rx = receiver(gimbalSat1Rx, "MountingLocation", ...
    [0;0;1], "GainToNoiseTemperatureRatio",3, "RequiredEbNo",4);
gaussianAntenna(sat1Rx, "DishDiameter",0.5); % meters

% 파일 크기 설정 (예: 10 MB)
fileSizeBits = 1000 * 1024 * 1024 * 8; % 100 MB to bits

% 가정된 대역폭 (예: 50 MHz)
B = 50e6; % Hz
```

시뮬레이션 환경설정 - 시간 설정

```
% 시간대 정보 없이 startTime을 설정
startTime = datetime(2023,12,14,10,0,0, 'TimeZone', 'UTC');
stopTime = startTime + days(1);

sampleTime = 10;
sc = satelliteScenario(startTime,stopTime,sampleTime);
```

2023/12/14 AM 10:00 을 기준

+24hours 간의 시뮬레이션

```
% 딜레이 설정
connectionDelay = 0; % seconds

% 전송 시간과 딜레이를 고려한 전체 시간을 저장할 배열
totalTransferTimes = zeros(1, length(latitudeArray));

% SNR (dB) 값 설정
gainToNoiseRatio_dB = 10;
```

커넥션 딜레이 고려 (0s)

SNR 값 (10dB로 가정)

커넥션 딜레이 0s인 이유

단순한 알고리즘으로 *Switching 에 많은 시간을 낭비

시뮬레이션 환경 설정 - 링크 용량 계산

```
% 인터벌의 시작과 종료 시간을 시간 배열의 인덱스로 변환
openIdx = find(timeArray >= openTime, 1, 'first');
closeIdx = find(timeArray <= closeTime, 1, 'last');

% 링크 용량 계산
for t = openIdx:closeIdx
    current_time = timeArray(t);
    if current_time >= openTime && current_time <= closeTime
        C = B * log2(1 + 10^(gainToNoiseRatio_dB/10));
        linkCapacities(i, t) = C;
    end
end
% 링크 배열 업데이트
linkStates(i, openIdx:closeIdx) = 1;
% 총 열린 시간에 현재 인터벌 시간 추가
totalOpenTime ...|
= totalOpenTime + (closeIdx - openIdx + 1) * sampleTime;
```

8641개의 timeIndex 도입

임의 SNR와 샤논 링크용량식 대입

totalOpenTime에 값 업데이트
→ 최종 AWS 요금 계산시 사용

시뮬레이션 환경 설정 – Single_Station 케이스

```
% timeStep별로 각 지상국별 전송 시간 계산
totalDataTransferred = 0;
for t = 1:numTimeSteps
    if linkStates(i, t) == 1 && totalDataTransferred < fileSizeBits
        dataTransferred = min(linkCapacities(i, t) ...
            * sampleTime, fileSizeBits - totalDataTransferred);
        totalDataTransferred = totalDataTransferred + dataTransferred;
    end
end

if totalDataTransferred < fileSizeBits
    totalTransferTimes(i) = null; % 지정한 시간동안 전송 불가능
else
    totalTransferTimes(i) = (find(linkStates(i, :) == 1, 1, 'last') ...
        - find(linkStates(i, :) == 1, 1, 'first') + 1) ...
        * sampleTime + connectionDelay * sum(linkStates(i, :) == 1);
end
end
```

단일 지상국으로의 전송 시간 계산

24h 내에 전송 불가할 시에 null

시뮬레이션 환경 설정 – Multi_Station 케이스

```
% 각 시간 단계에서 최적의 링크 선택 및 데이터 전송
for t = 1:numTimeSteps
    if currentTotalTime < totalRequiredTime
        availableLinks = find(linkStates(:, t) == 1);
        if ~isempty(availableLinks)
            % 최대 용량을 가진 링크 선택
            [maxCapacity, bestLinkId] = max(linkCapacities(availableLinks, t));
            bestLink = availableLinks(bestLinkId);

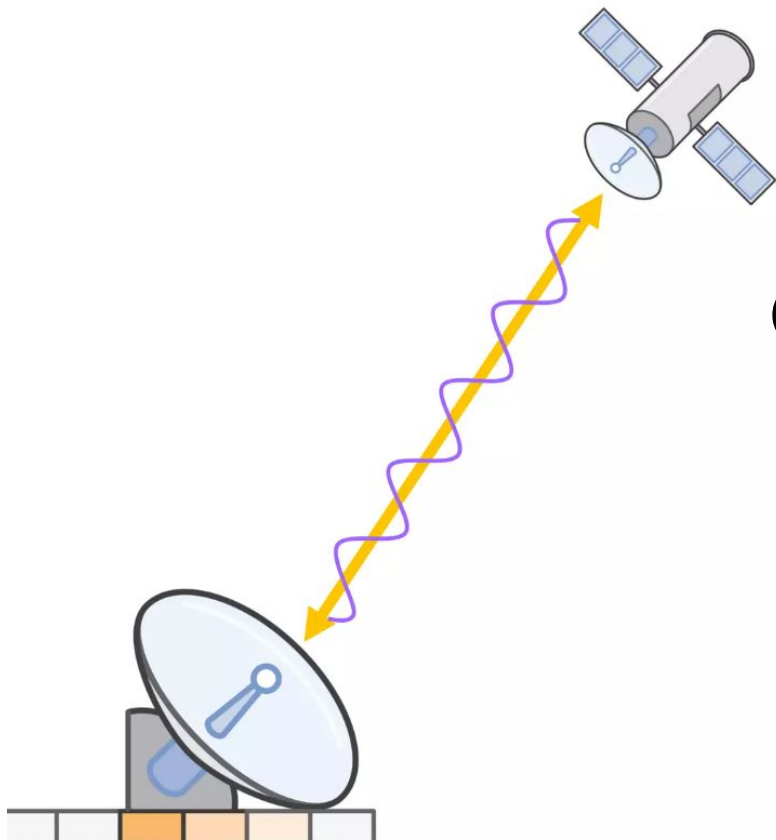
            % 선택된 지상국의 사용 시간 업데이트
            usedTimePerStation(bestLink) = usedTimePerStation(bestLink) + sampleTime;

            % 전체 전송 시간 업데이트
            currentTotalTime = currentTotalTime + sampleTime;
        end
    else
        % 전체 필요한 전송 시간 도달 (EarlyStop)
        totalElapsedTime = t * sampleTime;
        fprintf('총 경과 시간: %d 초\n', totalElapsedTime);
        break;
    end
end
```

2개 이상의 지상국 경합시에는 높은 LinkCapacity를 우선시함

데이터 전송이 완료 되었다면 EarlyStop

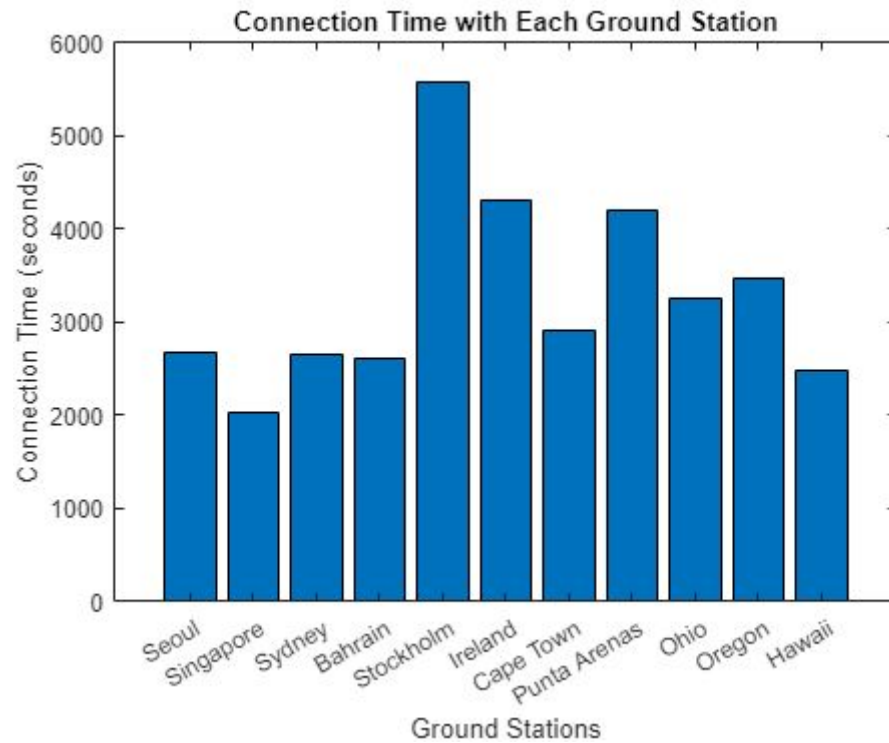
연구 결과



Pusan National University
Aerospace Engineering

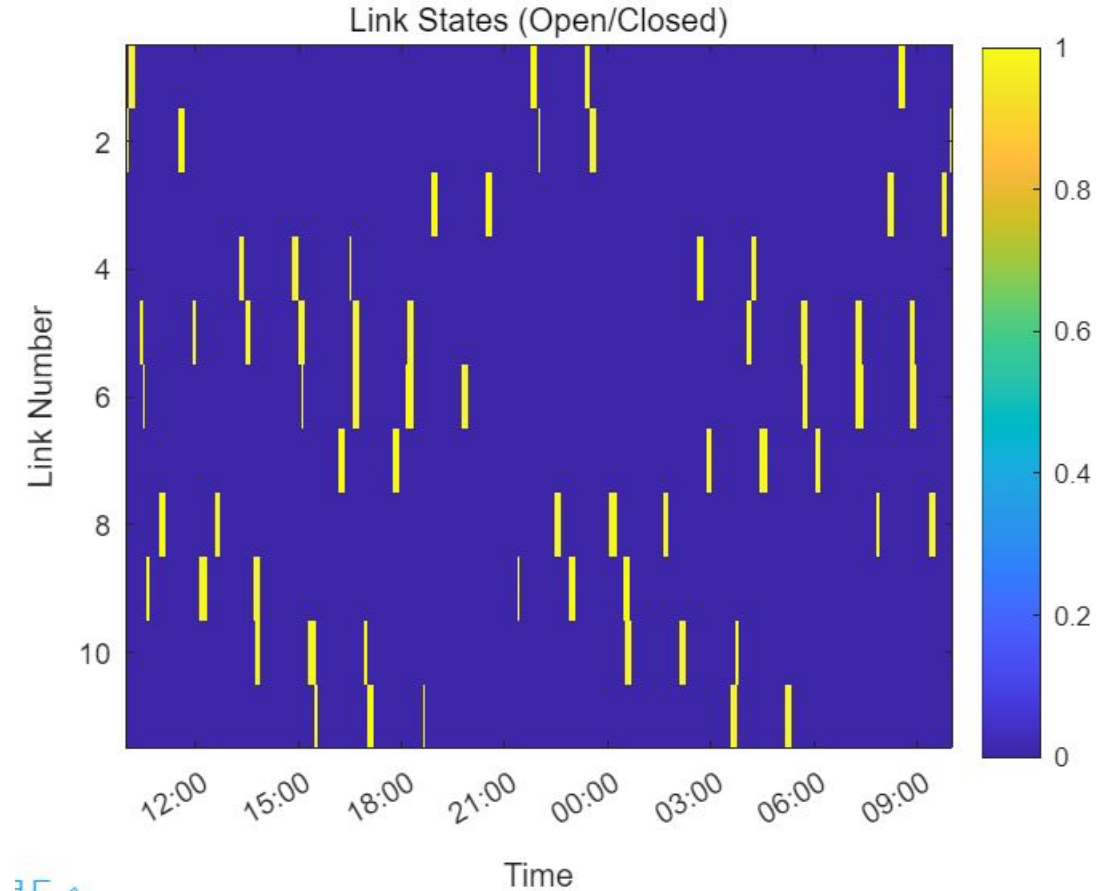


연구 결과 – Available Link States



그래프 1 : 지상국별 연결 가능 시간

Seoul
Singapore
Sydney
Bahrain
Stockholm
Ireland
Cape Town
Punta
Arenas
Ohio
Oregon
Hawaii



그래프 2 : 시간에 따른 지상국별 Link 연결

연구 결과 – Single GroundStation

1	2	3	4	5	6
Source	Target	Inter	StartTime	EndTime	Duration
"Transmitt...	"Receiver 5...	1	2023/12/14 15:28:...	2023/12/14 15:33:00	250
"Transmitt...	"Receiver 5...	2	2023/12/14 16:59:...	2023/12/14 17:11:50	720
"Transmitt...	"Receiver 5...	3	2023/12/14 18:38:...	2023/12/14 18:41:50	220
"Transmitt...	"Receiver 5...	4	2023/12/15 03:35:...	2023/12/15 03:45:30	590
"Transmitt...	"Receiver 5...	5	2023/12/15 05:09:...	2023/12/15 05:20:50	680

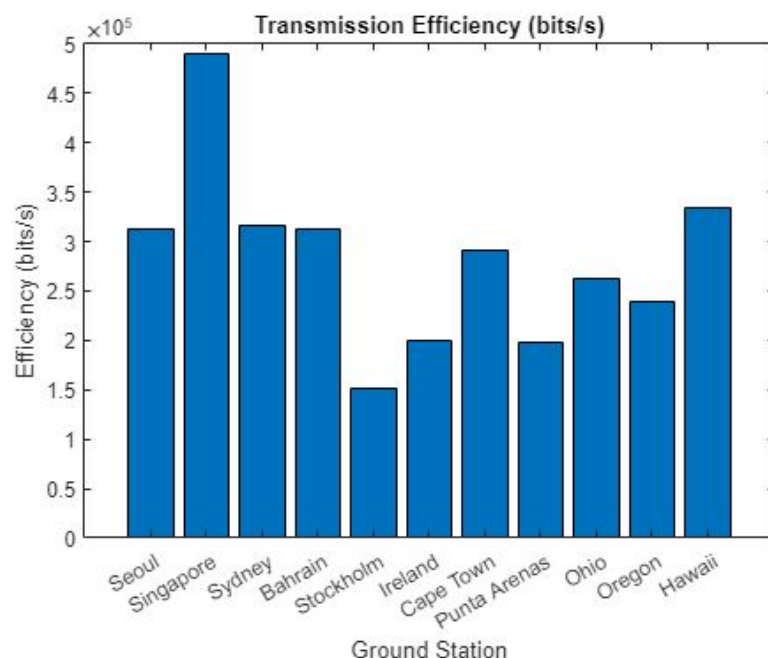


비통신 공전시간

표 1 : Seoul 지상국의 인공위성과 연결
상태

외 10여개의 통신국 정보

연구 결과 – Single GroundStation



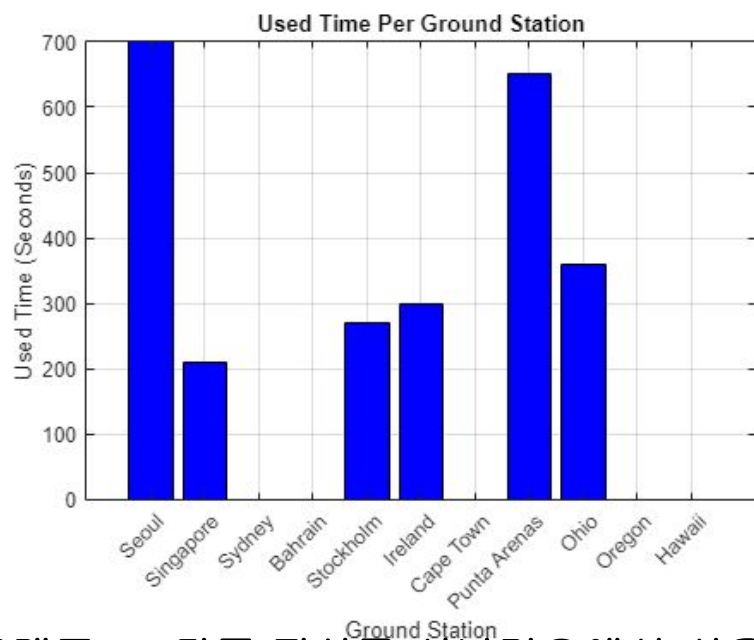
그래프 3 : 지상국 별 전송 효율

도시명	최종 시간(s)
Seoul	81260
Singapore	86410
Sydney	54030
Bahrain	54170
Stockholm	81080
Ireland	81170
Cape Town	50430
Punta Arenas	81290
Ohio	50620
Oregon	50740
Hawaii	49930

표 2 : 지상국 별 100MB 전송까지 걸린 시간

싱가포르 지상국의 효율이 가장 높지만 그래프1에서의 결과와 비교해보면 연결 가능시간이 짧기 때문에 종합적으로 고려해본다면 **하와이 지상국**에 가장 빠른 시간 내에 전송가능

연구 결과 – Multi GroundStation



그래프 4 : 다중 지상국 시나리오에서 사용된 지상국

도시명	시간 (s)
Seoul	700
Singapore	210
Sydney	0
Bahrain	0
Stockholm	270
Ireland	300
Cape Town	0
Punta Arenas	650
Ohio	360
Oregon	0
Hawaii	0

표 3 : 지상국 별 100MB 전송까지 사용된 시간

연구 결과 정리

2023/12/14 AM 10:00 을 기준

+24hours 간의 시뮬레이션

커백션 딜레이 고려 (0s)

SNR 값 (10dB로 가정)

한반도 상공에서 궤도 시작

고도 566.8km

경사각 98도

태양 동기궤도

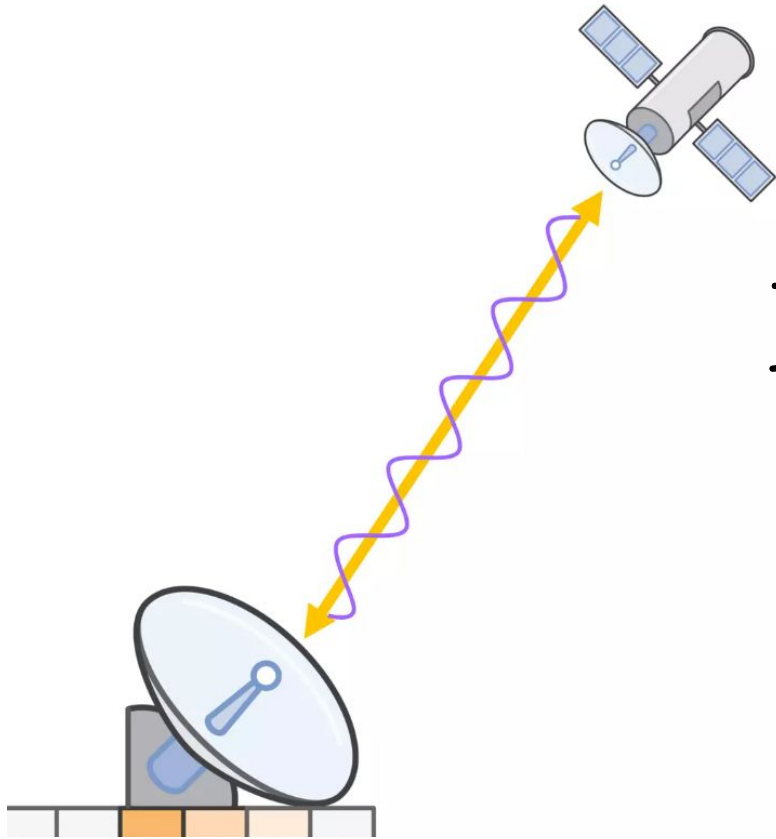
Switching 비용을 고려하여 **다중 지상국 이용시**

- 가장 **빠른 전송을** 원할 경우 4090초 내 전송 가능

Switching 비용을 고려하지 않고 **단일 지상국 이용시**

- 가장 **빠른 전송을** 원할 경우 Hawaii 지상국 이용
- 가장 **적은 비용을** 원할 경우 Singapore 지상국 이용

개선 방안



개선 방안

- ① 2개 이상 링크 경합 시나리오 알고리즘 개선
- ② 임시/가정 값 실제 값으로 대체
- ③ 세밀한 TimeStep (현재: 10초)



감사합니다

