PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100 Feet Ring Road, BSK III Stage, Bengaluru-560 085

Department of Electronics and Communication Engineering

Course: Wireless and Mobile Networking (UE21EC343BA1)

TITLE: MULTI-HOP SATELLITE COMMUNICATION

STUDENT: SOUMYADIPTO DE

SEMESTER – SIXTH SECTION -F

SRN: PES1UG21EC288

PROFESSOR: DR. M. RAJASEKAR

.

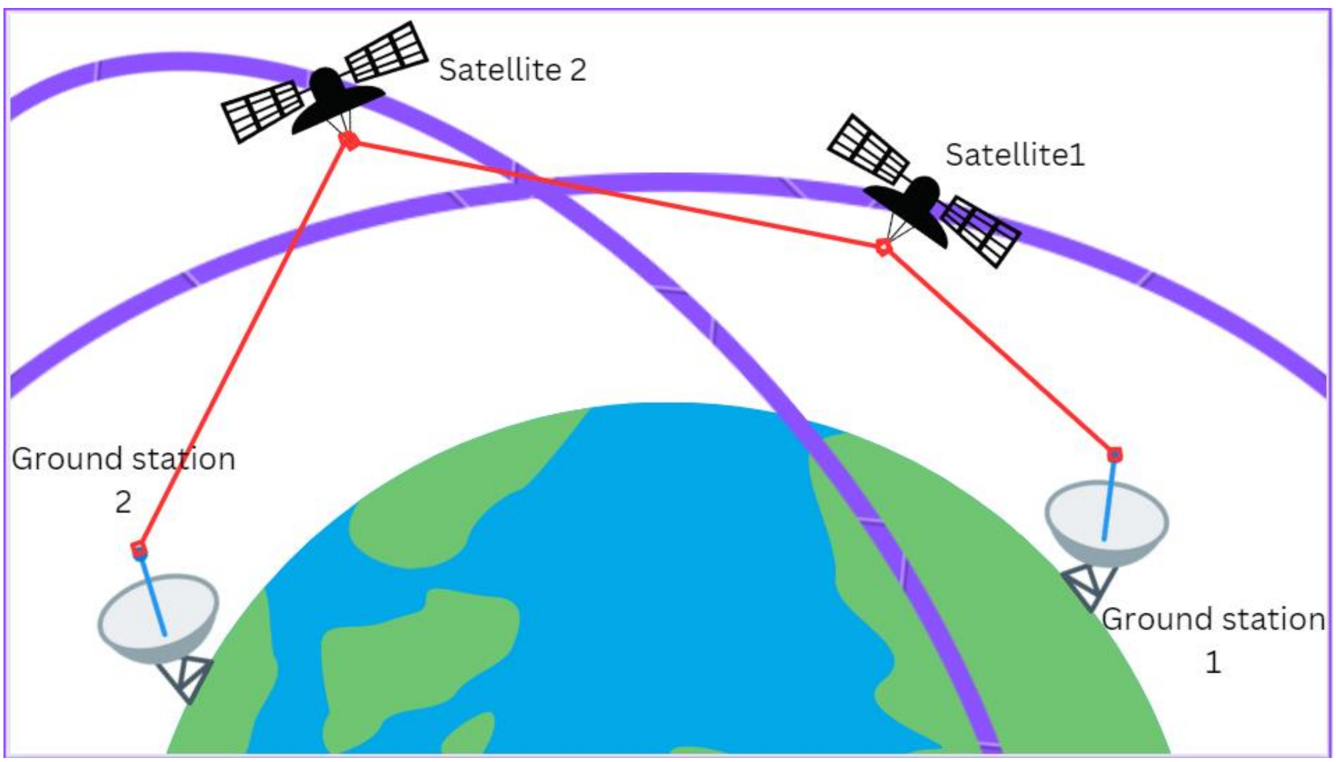
MULTI-HOP SATELLITE COMMUNICATION

AIM: The aim of this project is demonstration of multi-hop satellite communication between two ground stations. The objectives cover 2 parts:

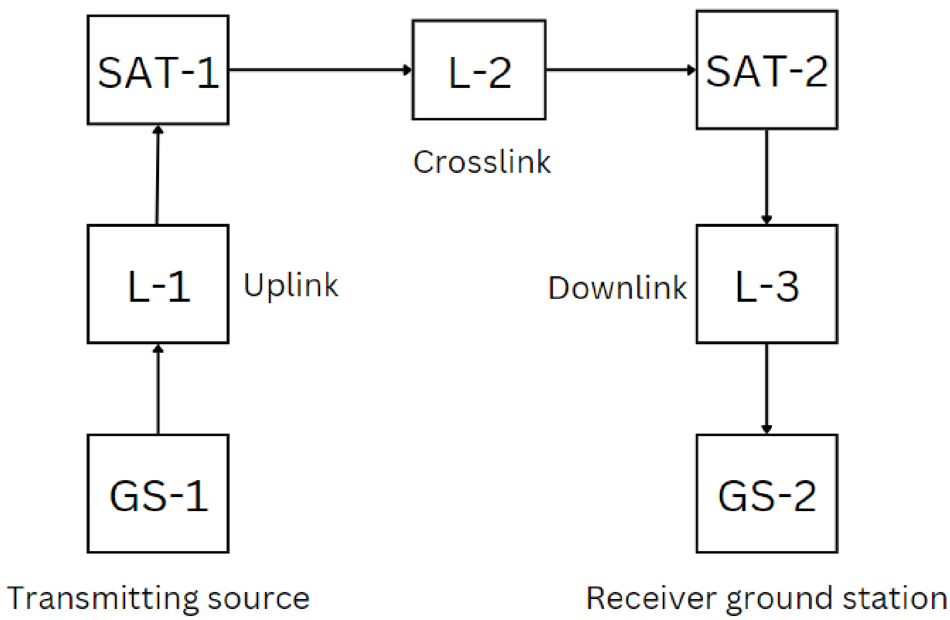
i) Simulating a scenario demonstrating a multi-hop link using two satellites
ii) Link analysis such as availability, latency, and link margin

BLOCK DIAGRAM:

The project is a simulation of a scenario wherein two satellites with different orbital planes are used to connect and establish a multi-hop link between two ground stations. The given diagram explains thus:



The block diagram encompasses the satellites, the ground stations, and their links:



The following points are to be noted:

1. Both satellites transmit with a power of 15 dBW.
2. Satellite 1 transmits via crosslink at 30 GHz frequency. Satellite 2 transmits via downlink at 27 GHz. This is an attempt at frequency allocation – certain frequency bands are only allocated for crosslink, and certain for downlink.
3. The type of antenna used throughout the project is Gaussian antenna . It is an antenna whose radiation pattern follows that of a Gaussian distribution.
4. The receiver gain-noise temperature ratio is 3dB/K and required Eb/No is 4 dB.
5. The uplink transmitter sends data to Satellite 1 at a frequency of 30 GHz and a power of 30 dBW.
6. The receiving ground station-2's gain to noise temperature ratio is 3 dB/K and the required Eb/No is 1 dB.

CODE:

```
%% Create Satellite Scenario

startTime = datetime(2024,4,22,9,30,0);   % 19 August 2020 8:55 PM UTC
%startTime = datetime(2020,8,19,20,55,0);
%startTime = datetime(2020,5,12,13,0,0);
stopTime = startTime + days(1);                       % 20 August 2020 8:55 PM UTC
sampleTime = 60;                                      % seconds
%% Launch Satellite Scenario Viewer
sc = satelliteScenario(startTime,stopTime,sampleTime);

%% Add the Satellites
satelliteScenarioViewer(sc);
semiMajorAxis = 10000000;                  % meters
eccentricity = 0;
inclination = 0;                           % degrees
rightAscensionOfAscendingNode = 0;         % degrees
argumentOfPeriapsis = 0;                   % degrees
trueAnomaly = 0;                           % degrees
sat1 = satellite(sc, ...
    semiMajorAxis, ...
    eccentricity, ...
    inclination, ...
    rightAscensionOfAscendingNode, ...
    argumentOfPeriapsis, ...
    trueAnomaly, ...
    "Name","Satellite 1", ...
    "OrbitPropagator","two-body-keplerian");
semiMajorAxis = 10000000;                  % meters
eccentricity = 0;
inclination = 30;                          % degrees
rightAscensionOfAscendingNode = 120;       % degrees
argumentOfPeriapsis = 0;                   % degrees
trueAnomaly = 300;                         % degrees
sat2 = satellite(sc, ...
    semiMajorAxis, ...
    eccentricity, ...
    inclination, ...
    rightAscensionOfAscendingNode, ...
    argumentOfPeriapsis, ...
    trueAnomaly, ...
    "Name","Satellite 2", ...
    "OrbitPropagator","two-body-keplerian");

%% Add Gimbals to the Satellites
gimbalSat1Tx = gimbal(sat1, ...
    "MountingLocation",[0;1;2]);  % meters
gimbalSat2Tx = gimbal(sat2, ...
    "MountingLocation",[0;1;2]);  % meters
gimbalSat1Rx = gimbal(sat1, ...
```

```
    "MountingLocation",[0;-1;2]); % meters
gimbalSat2Rx = gimbal(sat2, ...
    "MountingLocation",[0;-1;2]); % meters

%% Add Receivers and Transmitters to the Gimbals
sat1Rx = receiver(gimbalSat1Rx, ...
    "MountingLocation",[0;0;1], ...      % meters
    "GainToNoiseTemperatureRatio",3, ... % decibels/Kelvin
    "RequiredEbNo",4);                   % decibels
sat2Rx = receiver(gimbalSat2Rx, ...
    "MountingLocation",[0;0;1], ...      % meters
    "GainToNoiseTemperatureRatio",3, ... % decibels/Kelvin
    "RequiredEbNo",4);


gaussianAntenna(sat1Rx, ...
    "DishDiameter",0.5);    % meters
gaussianAntenna(sat2Rx, ...
    "DishDiameter",0.5);    % meters


sat1Tx = transmitter(gimbalSat1Tx, ...
    "MountingLocation",[0;0;1], ...   % meters
    "Frequency",30e9, ...             % hertz
    "Power",15);                      % decibel watts
sat2Tx = transmitter(gimbalSat2Tx, ...
    "MountingLocation",[0;0;1], ...   % meters
    "Frequency",27e9, ...             % hertz
    "Power",15);
gaussianAntenna(sat1Tx, ...
    "DishDiameter",0.5);    % meters
gaussianAntenna(sat2Tx, ...
    "DishDiameter",0.5);    % meters
halfBeamWidth = 62.7; % degrees

%% Add the Ground Stations
latitude = 12.9436963;          % degrees
longitude = 77.6906568;         % degrees
gs1 = groundStation(sc, ...
    latitude, ...
    longitude, ...
    "Name","Ground Station 1",...
    "MinElevationAngle",20);
latitude = -33.7974039;        % degrees
longitude = 151.1768208;       % degrees
gs2 = groundStation(sc, ...
    latitude, ...
    longitude, ...
    "Name","Ground Station 2",...
    "MinElevationAngle",10);

gimbalGs1 = gimbal(gs1, ...
```

```
    "MountingAngles",[0;180;0], ... % degrees
    "MountingLocation",[0;0;-5]);   % meters
gimbalGs2 = gimbal(gs2, ...
    "MountingAngles",[0;180;0], ... % degrees
    "MountingLocation",[0;0;-5]);   % meters

%% Add Transmitters and Receivers to Ground Station Gimbals
gs1Tx = transmitter(gimbalGs1, ...
    "Name","Ground Station 1 Transmitter", ...
    "MountingLocation",[0;0;1], ...           % meters
    "Frequency",30e9, ...                     % hertz
    "Power",30);
gaussianAntenna(gs1Tx, ...
    "DishDiameter",2); % meters

gs2Rx = receiver(gimbalGs2, ...
    "Name","Ground Station 2 Receiver", ...
    "MountingLocation",[0;0;1], ...        % meters
    "GainToNoiseTemperatureRatio",3, ...   % decibels/Kelvin
    "RequiredEbNo",1);
gaussianAntenna(gs2Rx, ...
    "DishDiameter",2); % meters

%% Set Tracking Targets for Gimbals
pointAt(gimbalGs1,sat1);
pointAt(gimbalSat1Rx,gs1);
pointAt(gimbalSat1Tx,sat2);
pointAt(gimbalSat2Rx,sat1);
pointAt(gimbalSat2Tx,gs2);
pointAt(gimbalGs2,sat2);

%% Add Link Analysis and Visualize Scenario
lnk = link(gs1Tx,sat1Rx,sat1Tx,sat2Rx,sat2Tx,gs2Rx);

%% Determine Times When Link is Closed and Visualize Link Closures
linkIntervals(lnk)
[e, time] = ebno(lnk);
margin = e - gs2Rx.RequiredEbNo;

%% Calculate latency between Satellite 1 and Ground Station 1
[delay_gs1, time_gs1] = latency(sat1, gs1);

% Calculate latency between Satellite 1 and Satellite 2
[delay_sat2, time_sat2] = latency(sat1, sat2);

% Calculate latency between Satellite 1 and Ground Station 2
[delay_gs2, time_gs2] = latency(sat2, gs2);

% Plot latency graph
figure;
subplot(3,1,1);
plot(time_gs1, delay_gs1);
```

```
title('Latency between Satellite 1 and Ground Station 1');
xlabel('Time');
ylabel('Latency (s)');

subplot(3,1,2);
plot(time_sat2, delay_sat2);
title('Latency between Satellite 1 and Satellite 2');
xlabel('Time');
ylabel('Latency (s)');

subplot(3,1,3);
plot(time_gs2, delay_gs2);
title('Latency between Satellite 2 and Ground Station 2');
xlabel('Time');
ylabel('Latency (s)');

%% Modify Required Eb/No and Observe Effect on Link Intervals

gs2Rx.RequiredEbNo = 10; % decibels
linkIntervals(lnk)
[e, newTime] = ebno(lnk);
newMargin = e - gs2Rx.RequiredEbNo;
figure;
plot(newTime,newMargin,"r",time,margin,"b","LineWidth",2);
xlabel("Time");
ylabel("Link Margin (dB)");
legend("New link margin","Old link margin","Location","north");
grid on;

%% Radiation Pattern
%pat1 = pattern(sat1Rx,"Size",3000000);
sat1_pat = pattern(sat1Tx,"Size",3000000);
sat2_pat = pattern(sat2Tx,"Size",3000000);
%pat4 = pattern(sat2Rx,"Size",3000000);
pointAt(sat1,sat2);
pointAt(sat2,gs2);

% Calculate radiation patterns for ground stations
gs1Tx_pat = pattern(gs1Tx, 'Size', 1000000);

gs2Rx_pat = pattern(gs2Rx, 27e9, 'Size', 1000000);


%% Access analysis

name1 = sat1.Name + " Camera";
cam = conicalSensor(sat1,"Name",name1,"MaxViewAngle",90);
ac1 = access(cam,gs1);
fov = fieldOfView(cam([cam.Name] == "Satellite 1 Camera"));
accessIntervals(ac1);

name2 = sat2.Name + " Camera";
```

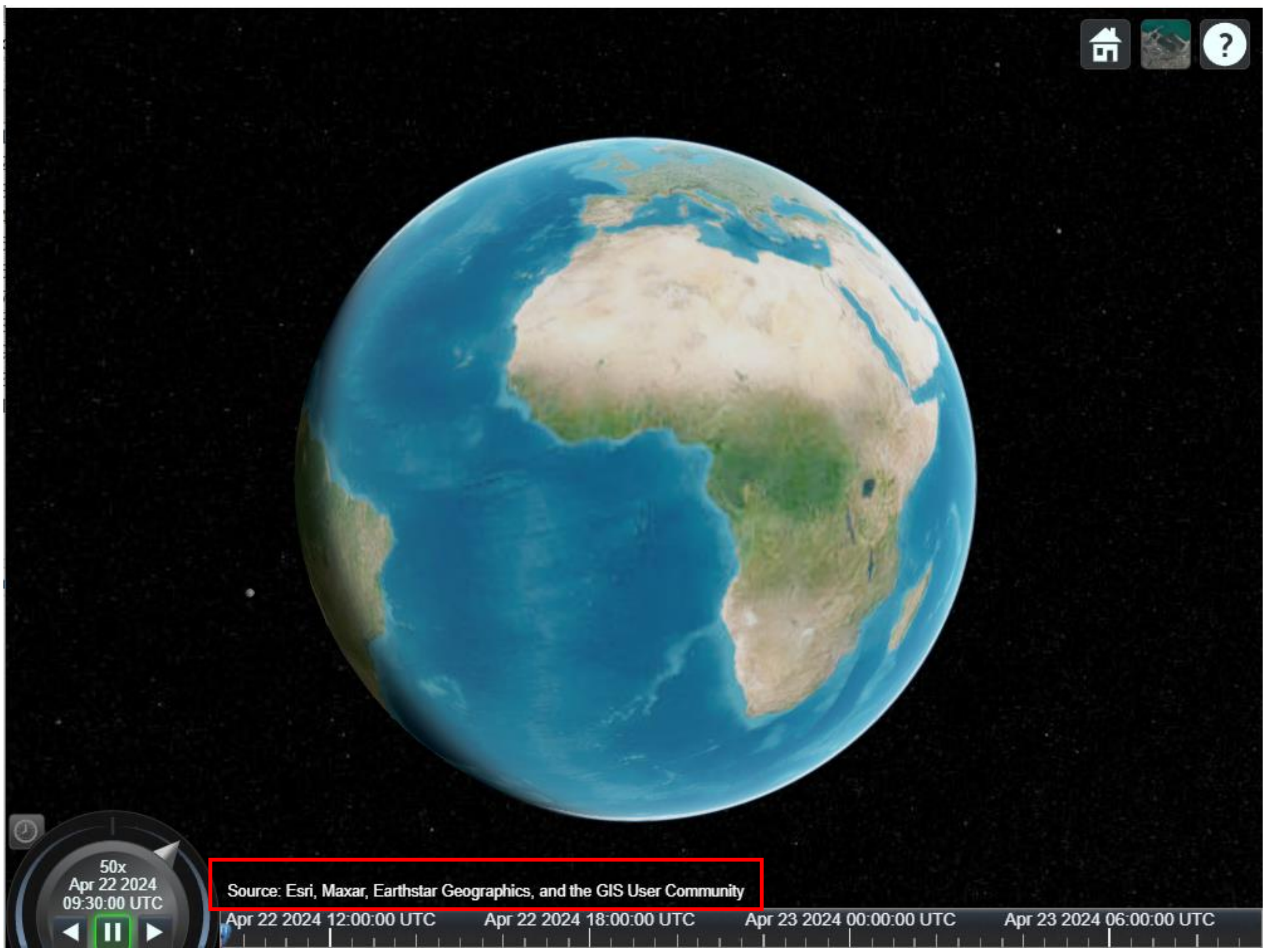
```
cam2 = conicalSensor(sat2,"Name",name2,"MaxViewAngle",90);
ac2 = access(cam2,gs2);
fov2 = fieldOfView(cam2([cam2.Name] == "Satellite 2 Camera"));
accessIntervals(ac2);

%% Eb/No ratio varying with time
time_serial = datetime(time);

e_indices = isinf(e);
e_cleaned = e;
e_cleaned(e_indices) = 0;
figure;
plot(time_serial,e_cleaned, 'LineWidth', 1.5);
title('Eb/N0 vs Time for Multi-Hop link');
xlabel('Discrete Time');
ylabel('Eb/N0 (dB)');
grid on;

%% Each link Eb/No
% Create links separately for gs1Tx and sat1Rx
link_gs1Tx_sat1Rx = link(gs1Tx, sat1Rx);
link_sat2Tx_gs2Rx = link(sat2Tx, gs2Rx);
%link_sat1Tx_sat2Rx = link(sat1Tx, sat2Rx);


% Calculate Eb/N0 for each link
[e_gs1_sat1, time_gs1_sat1] = ebno(link_gs1Tx_sat1Rx);
[e_sat2_gs2, time_sat2_gs2] = ebno(link_sat2Tx_gs2Rx);

% Overwrite -Inf values with 0 baseline
e_1 = e_gs1_sat1; e_2 = e_sat2_gs2;
e_1(isinf(e_gs1_sat1))= 0; e_2(isinf(e_sat2_gs2)) = 0;


% Plot Eb/N0 vs. time for each link
figure;
subplot(2,1,1);
plot(time_gs1_sat1,e_1, 'LineWidth', 1.5,'Color','g');
title('Eb/N0 vs Time for Uplink');
xlabel('Time');
ylabel('Eb/N0 (dB)');
grid on;

subplot(2,1,2);
plot(time_sat2_gs2, e_2, 'LineWidth', 1.5,'Color','r');
title('Eb/N0 vs Time for Downlink');
xlabel('Time');
ylabel('Eb/N0 (dB)');
%legend('gs1Tx - sat1Rx', 'sat1Tx - gs2Rx');
grid on;

%% Plot link establishment
```

```
figure;
bar(time, lnk.linkStatus, 'FaceColor', [0.1 0.9 0.1]);
hold on;
bar(time, ~lnk.linkStatus, 'FaceColor', [0.9 0.1 0.1]);
xlabel('Time');
ylabel('Link Availability');
title('Link Availability Over Time');
legend('Available', 'Unavailable');

% Format x-axis as datetime
xtickformat('yyyy-MM-dd HH:mm:ss');


%% Simulate scenario
play(sc);
```

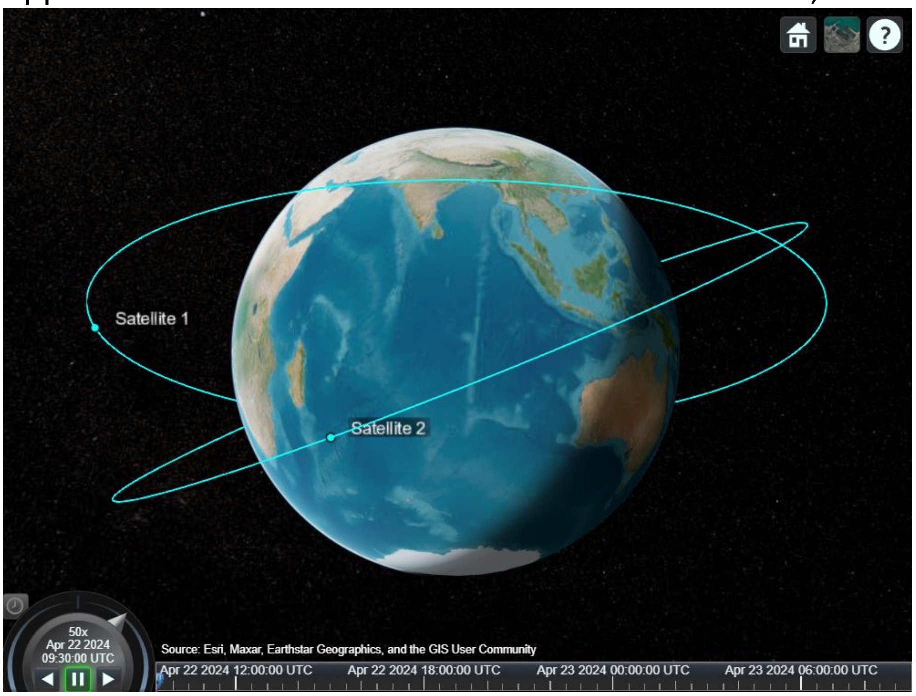
WORKING:

The program is written in MATLAB and explained in several sections, each with their own role to play in the demonstration. It is imperative that the **Satellite Communications Toolbox** is installed for the scenario player to run. The user may only run this program and wait, there are no extra tasks from the user side.

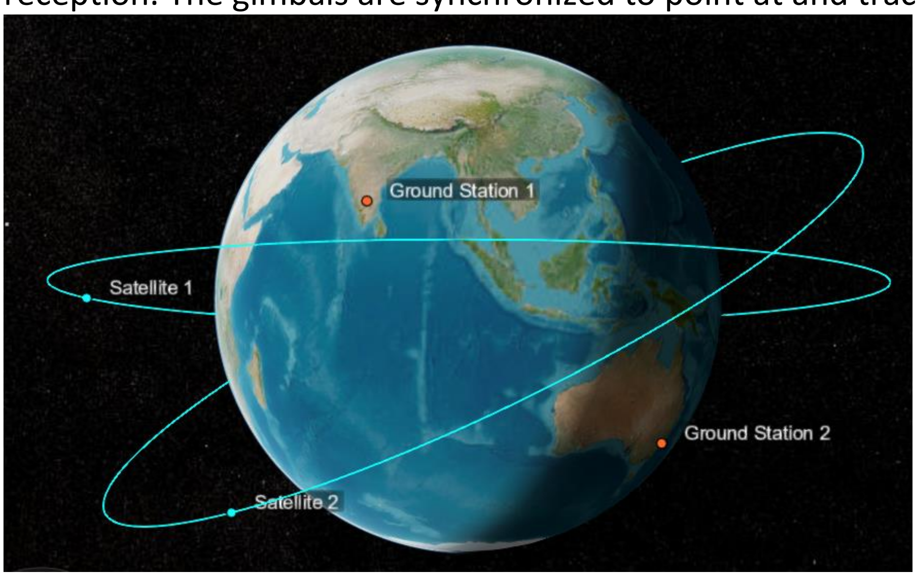
Create Satellite Scenario: A scenario is created with starttime, stoptime, to define the time period of the simulation, and a sample time for recording. **Internet connection is required** as the simulation creates the scenario based on real data acquired from Earthstar Geographics and GIS User Community.



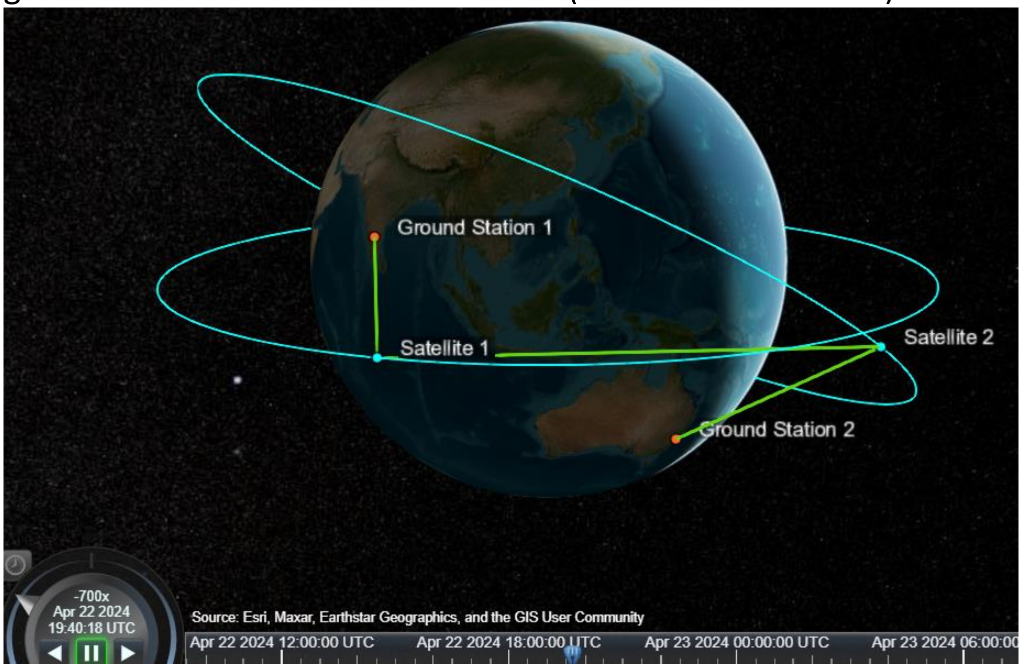
Add Satellites and their Gimbals: Two satellites are added. Their orbital planes are defined. Gimbals are what holds the transmitter and receiver. There are 2 gimbals on opposite sides of both satellites – one for transmitter, other for receiver.



Add Ground stations: Two Ground stations are added, one acting as a transmitting station, the other- receiving station. Suitable gimbals are added for transmission and reception. The gimbals are synchronized to point at and track their respective targets.



<u>Link Analysis:</u> The scenario is visualized. At certain timestamps throughout simulation, satellites 1 and 2 are in a position such that a multi-hop link is established between ground stations 1 and 2. Link closure (when the link closes) is also visualized.



5×8 table

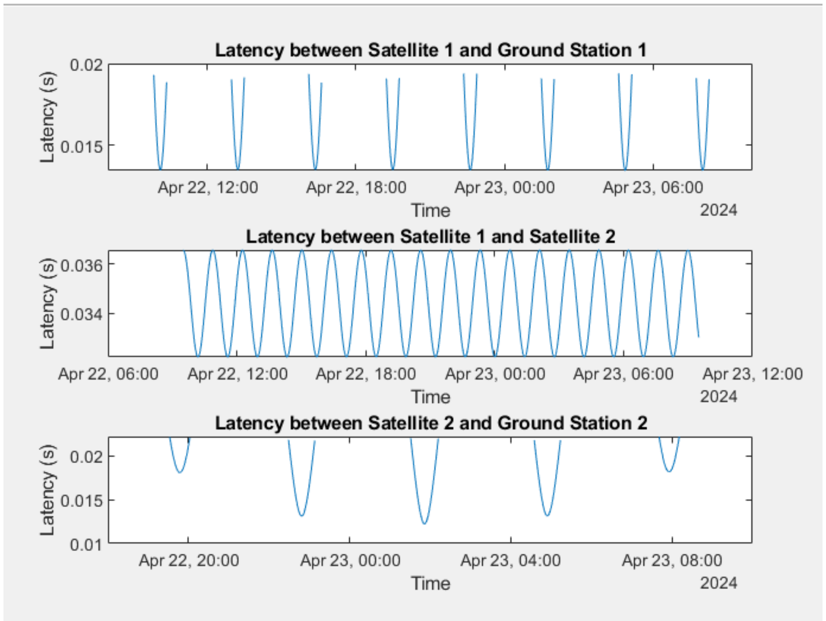
Source	Target	IntervalNumber	StartTime	EndTime	Duration	StartOrbit	EndOrbit
"Ground Station 1 Transmitter"	"Ground Station 2 Receiver"	1	22-Apr-2024 19:32:00	22-Apr-2024 19:45:00	780	NaN	NaN
"Ground Station 1 Transmitter"	"Ground Station 2 Receiver"	2	22-Apr-2024 22:29:00	22-Apr-2024 22:53:00	1440	NaN	NaN
"Ground Station 1 Transmitter"	"Ground Station 2 Receiver"	3	23-Apr-2024 01:31:00	23-Apr-2024 02:00:00	1740	NaN	NaN
"Ground Station 1 Transmitter"	"Ground Station 2 Receiver"	4	23-Apr-2024 04:36:00	23-Apr-2024 05:08:00	1920	NaN	NaN
"Ground Station 1 Transmitter"	"Ground Station 2 Receiver"	5	23-Apr-2024 07:44:00	23-Apr-2024 08:11:00	1620	NaN	NaN

A table of the start and stop times of link closures represent the intervals during which Ground Station 1 can send data to Ground Station 2.

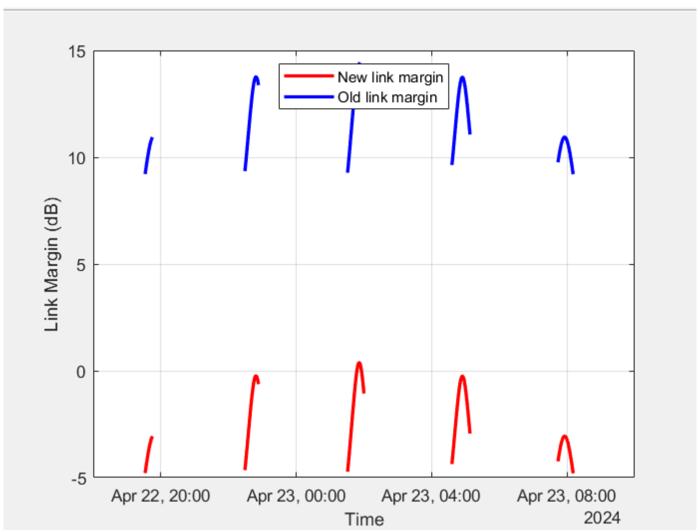
<u>Calculate latency:</u> the latency is calculated between the following:

1. Satellite-1 and Ground station-1
2. Satellite-1 and Satellite-2
3. Satellite-2 and Ground station-2

From the resulting graph, we observe that latency is high initially, which then decreases to a minimum before increasing. The propagational distance is the factor that determines this nature.



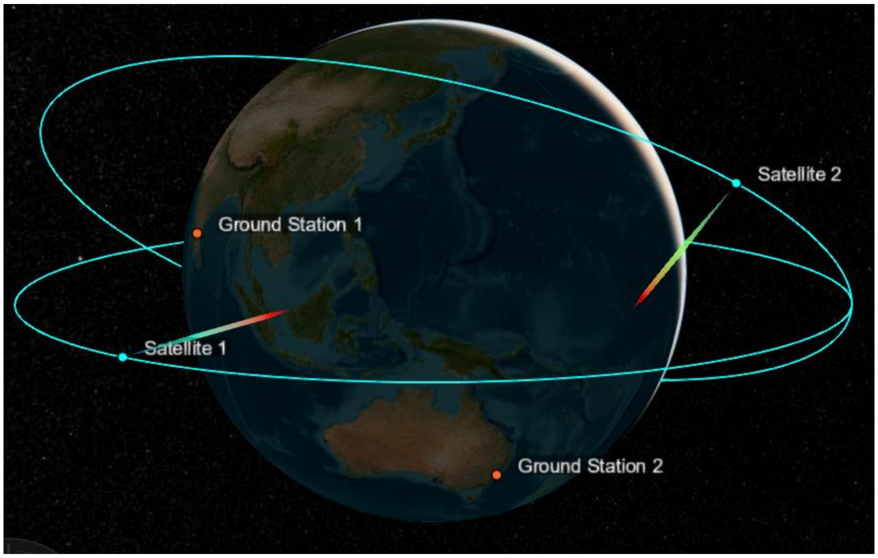
<u>Link Margin at Ground station-2:</u> Changing Eb/No changes number of link intervals. The number of times a multi-hop link established between the 2 ground stations is now 1 because the link quality is reduced. The old and new link margins are plotted.



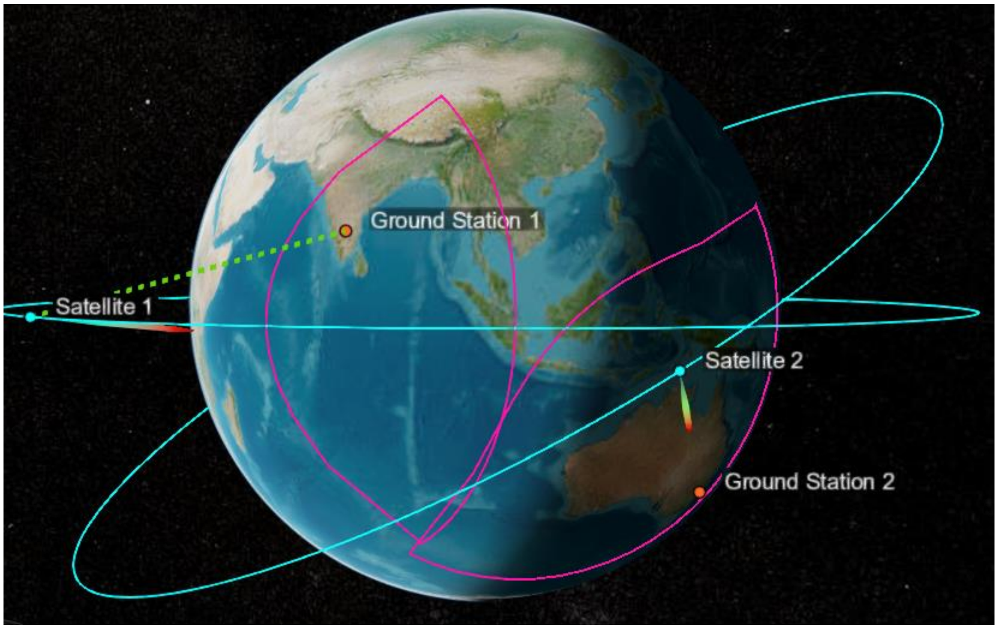
1×8 table

Source	Target	IntervalNumber	StartTime	EndTime	Duration	StartOrbit	EndOrbit
"Ground Station 1 Transmitter"	"Ground Station 2 Receiver"	1	23-Apr-2024 01:48:00	23-Apr-2024 01:55:00	420	NaN	NaN

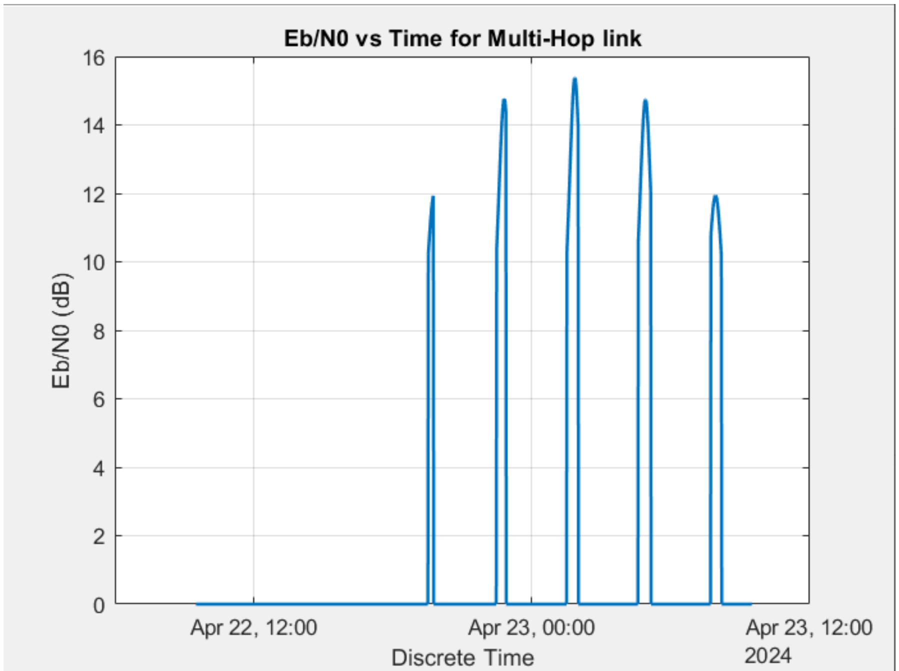
Radiation beam: The radiation beam is displayed for both satellite transmitters.



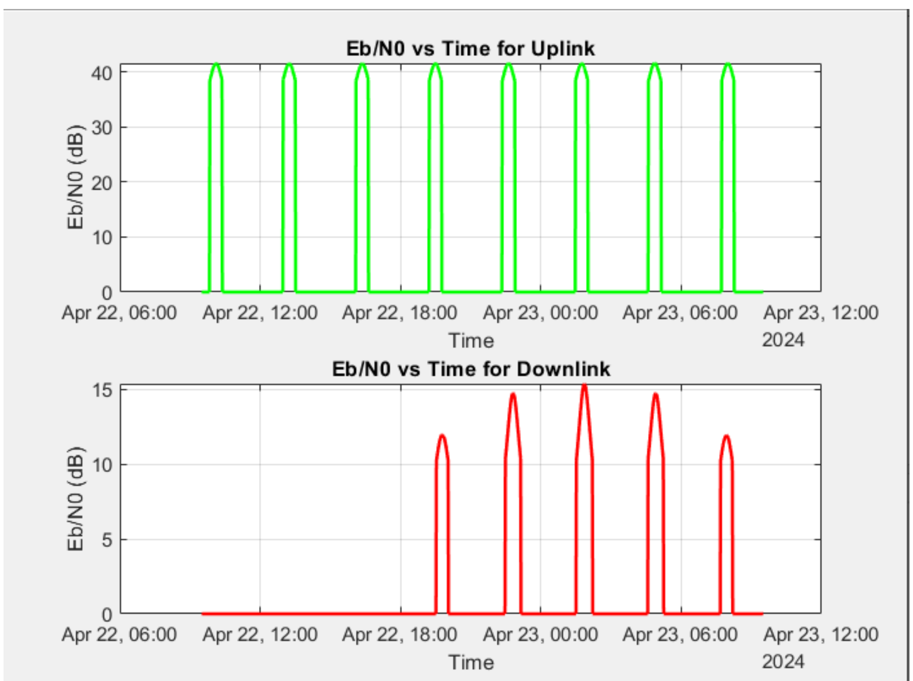
Field view: A camera is attached onboard the satellites to show its field of view.



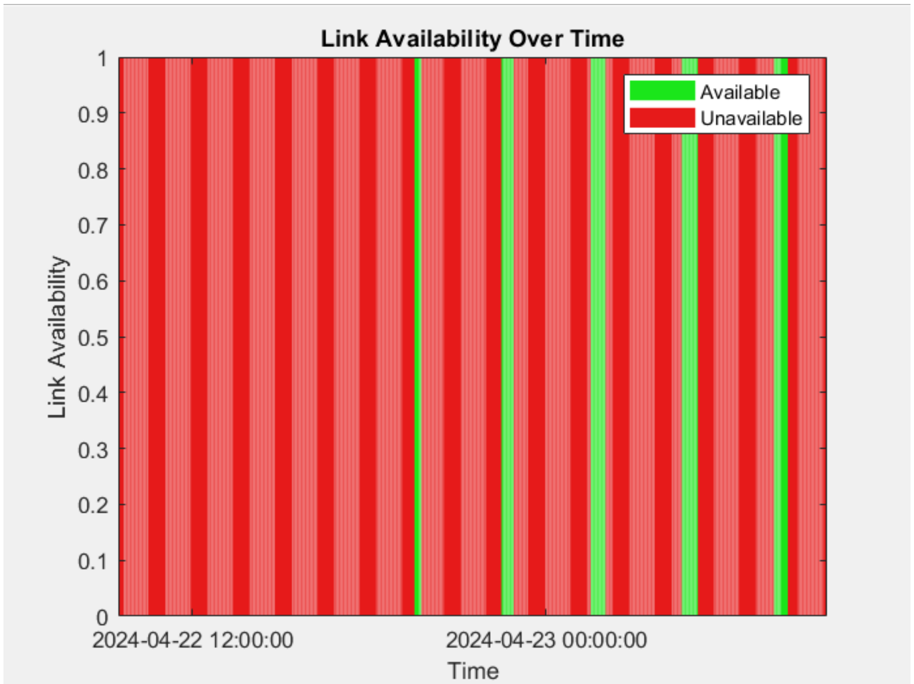
Receiver Signal strength: A graph is plotted to display the signal strength in terms of Eb/No versus time at the receving ground station.



This is expanded further for uplink and downlink as well. The region where both uplink and downlink have signal strength hints at a multi-hop link connection.



Link Availability: A bar graph is plotted to show the time period/ instances where the multi-hop link is available. Green stands as the link being available, while red means no link available.



CONCLUSION:

This project delved into satellite communications. A multi-hop link between 2 ground stations via satellites is simulated. Link analysis is done as in latency, link availability, and receiver signal strength. The future scope of this project can extend to satellite coverage maps, radiation plots and multipath selection with large number of satellites based on optimization algorithms. It was a fruitful experience to work on this project and learn new ideas and methods on MATLAB.