ISRO – Telemetry Tracking and Command Network (ISTRAC)

TITLE: NavIC RANGING DATA MONITORING SYSTEM

UNDER THE GUIDANCE OF

RAJESH KUMAR

Scientist/Engineer-SE, NSA, ISTRAC

SUBMITTED BY

PREM. B **Reg No: U22CN156**

STUDENT OF

BHARATH INSTITUTE OF HIGHER EDUCATION AND RESEARCH - BIHER

SELAIYUR, AGHARAM MAIN ROAD, CHENNAI - 600073

ACKNOWLEDGEMENT

If words are considered as symbols of approval and tokens of acknowledgement, then let the words play the heralding role of expressing my gratitude to all those who have directly or indirectly helped during my period of internship.

I am delighted to thank **Dr. A. K. ANILKUMAR, Director, ISRO Telemetry, Tracking, and Command Network (ISTRAC), Bengaluru, Karnataka,** for giving me permission to pursue my Internship in ISTRAC, BENGALURU.

I express my gratitude to **Mr. SANKAR MADASWAMY B, Scientist/Engineer-SG, Manager, HRDD, ISTRAC,** for permitting me to undertake my internship. I wholeheartedly thank **Mr. MANISH CHAWLA, Manager, NDC, INOCTF/NSA** for his valuable guidance and encouragement during the internship.

My sincere thanks also goes to my guide **Mr. RAJESH KUMAR, Scientist/Engineer-SE, Navigation System Area, ISTRAC,** for his mentorship, constructive feedback, and technical insights. His meticulous approach and encouragement have greatly enhanced my understanding and technical proficiency.

I sincerely thank you all whole-heartedly for sparing your valuable time quite often, despite the heavy workload. The discussions had provided me with valuable insights into the workings of ISRO. Finally, I would like to thank my people (my parents, friends and university) for their continuous support in every field of my life.

PREM. B

Reg No: U22CN156

BONAFIDE CERTIFICATE

This is to certify that the Internship report entitled **"NavIC RANGING DATA MONITORING SYSTEM REPORT"** submitted by **PREM B**, bearing **REG NO: U22CN156** of **BHARATH INSTITUTE OF HIGHER EDUCATION AND RESEARCH, CHENNAI, TAMIL NADU**, for partial fulfilment for the award of Degree of Bachelor of Engineering, in **COMPUTER SCIENCE ENGINEERING (CSE)** is a bonafide record of the internship carried out by him under my supervision in the year 2025.

RAJESH KUMAR

Scientist/Engineer-SE,

NSA, ISTRAC

Table of Contents

Table of Figures

1. Abstract

This report presents the design, implementation, and evaluation of the NavIC Ranging Data Monitoring System, a real-time solution for decoding and visualizing binary telemetry data from the IRNSS (Navigation with Indian Constellation) ground segment. The system is architected to acquire data through UDP multicast transmission from NavIC stations, process binary log files via a multi-threaded Flask backend, and present the decoded range measurement and navigation information on a dynamic web-based user interface.

The report details a comprehensive workflow starting with the multicast sender, which reads pre-recorded binary logs and transmits data packets using designated multicast groups. These packets are captured by a multicast receiver that logs the data into station-specific files. A key innovation of this system is the robust handling of packet fragmentation and loss; the backend leverages sequence numbers and fragment identifiers to accurately reassemble fragmented NavIC data packets while employing retransmission logic to mitigate packet loss.

Decoding of binary logs is achieved by parsing structured headers that contain synchronization words, timestamps, message IDs, and error-checking CRCs. The processing pipeline differentiates between range measurement data and navigation data, extracting vital parameters such as pseudo range, Doppler frequency, carrier-to-noise ratios, and lock times. The decoded information is then made available through REST API endpoints to a frontend that supports multiple station views (e.g., Station 1 and Station 2), ensuring that users can seamlessly switch between channels to monitor real-time data.

Rigorous testing was performed under simulated network conditions to validate data integrity, real-time performance, and error handling. The results confirmed that the system reliably decodes and displays data, effectively manages channel isolation, and maintains synchronization despite inherent UDP packet loss challenges.

Future enhancements are outlined, including expanding station support, integrating persistent database storage for historical analysis, and incorporating advanced visualization and AI-based anomaly detection modules. Overall, the NavIC Ranging Data Monitoring System demonstrates a scalable, robust, and efficient framework for satellite telemetry tracking and command network monitoring, contributing significantly to the operational capabilities of ISRO's ground segment infrastructure.

2. About ISTRAC

ISTRAC (ISRO Telemetry Tracking and Command Network) is a ground segment network that provides support for the Indian Space Research Organisation's (ISRO) space missions:

- Telemetry, tracking, and command (TTC): ISTRAC provides TTC services from launch to satellite orbit injection. This includes tracking the launch vehicle from lift-off to spacecraft separation.
- Mission operations: ISTRAC carries out mission operations for remote sensing and scientific satellites.
- Spacecraft control centers: ISTRAC has established spacecraft control centers.
- Deep space network: ISTRAC has a deep space network that provides support for deep space missions.
- Weather radar design and development: ISTRAC designs and develops weather radars for launch vehicle tracking and meteorological applications.
- Search and rescue: ISTRAC provides search and rescue services.
- Space-based services: ISTRAC supports space-based services like telemedicine, tele-education, and Village Resource Centre (VRC).
- Space operations support: ISTRAC provides space operations support for other space agencies.

ISTRAC's headquarters are in Bangalore, Karnataka. It also has ground stations in Lucknow, Sriharikota, Thiruvananthapuram, Port Blair, Brunei, Biak (Indonesia), and Mauritius.

2.1 Satellite Navigation Services

Satellites for navigation services to meet the emerging demands of the Civil Aviation requirements and to meet the user requirements of the positioning, navigation and timing based on the independent satellite navigation system.

2.2 Navigation with Indian Constellation (NavIC)

To meet the positioning, navigation and timing requirements of the nation, ISRO has established a regional navigation satellite system called Navigation with Indian Constellation (NavIC). NavIC was formerly known as Indian Regional Navigation Satellite System (IRNSS).

NavIC is designed with a constellation of 7 satellites and a network of ground stations operating 24/7. Three satellites of the constellation are placed in geostationary orbit, at 32.5°E, 83°E and 129.5°E respectively, and four satellites are placed in inclined geosynchronous orbit with equatorial crossing at 55°E and 111.75°E, with inclination of 29° (two satellites in each plane). The ground network consists of control centre, precise timing facility, range and integrity monitoring stations, two-way ranging stations, etc.

2.3 NavIC System Overview

Navigation with Indian Constellation (NavIC) is an independent and indigenously developed satellite navigation system fully planned, established and controlled by the Indian Space Research Organization (ISRO). NavIC was earlier known as Indian Regional Navigation Satellite System (IRNSS).

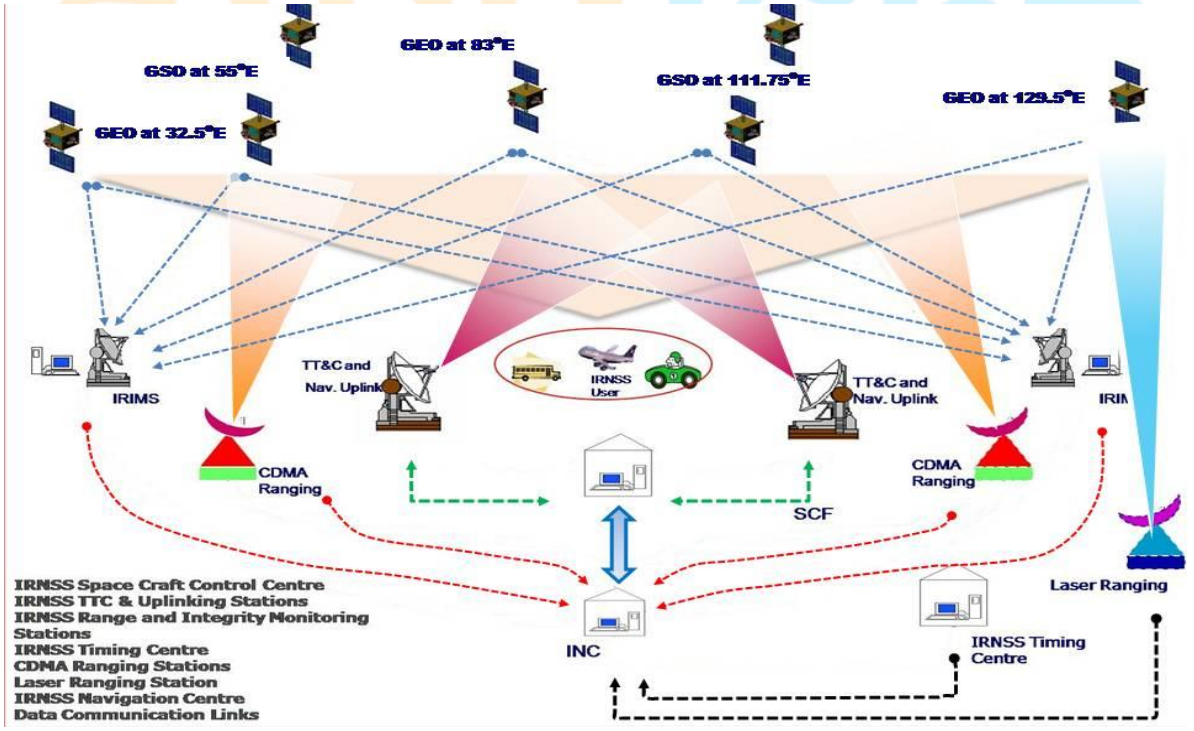


Figure 1 NavIC Architecture

2.4 NavIC Ground Segment

Ground segment is responsible for the maintenance and operation of the NavIC constellation. The ground segment comprises of:

- ISRO Navigation Centre (INC)
- IRNSS Spacecraft Control Facility (IRSCF)
- IRNSS Range and Integrity Monitoring Stations (IRIMS)
- IRNSS Network Timing (IRNWT) Facility
- IRNSS CDMA Ranging Stations (IRCDR)
- Laser Ranging Stations
- Data Communication Network.

2.5 User Segment

The User segment mainly consists of:

- Single frequency NavIC receiver capable of receiving signal in L1 / L5 / S band frequency.
- A multi-frequency NavIC receiver capable of receiving combinations of L1, L5 and S band signals.

A multi-constellation receiver compatible with NavIC and other GNSS signals.

3. Introduction

3.1 Background

The Navigation with Indian Constellation (NavIC), also known as the Indian Regional Navigation Satellite System (IRNSS), forms an integral part of India's satellite navigation capabilities. NavIC comprises three segments:

- **Space Segment:** A constellation of geostationary (GEO) and geosynchronous (GSO) satellites.
- **Ground Segment:** The IRNSS Range and Integrity Monitoring Stations (NAVIC) are critical ground elements designed for continuous one-way ranging and monitoring of the IRNSS satellites.
- **User Segment:** End-user receivers that utilize the positioning, navigation, and timing information.

3.2 Project Objective

The primary objective of this project is to develop a **real-time NavIC Ranging Data Monitoring System**. The system is designed to:

- Acquire and process binary logs containing range measurement and navigation data.
- Decode and parse the data for key parameters such as pseudo range, Doppler frequency, carrier-to-noise ratio, lock time, etc.
- Display the processed data in a dynamic and user-friendly web interface.
- Support multiple stations (e.g., Station 1 and Station 2) within a single backend instance.

3.3 Scope of the Project

This report details the design and implementation of the NavIC Ranging Data Monitoring System, which includes:

- **Data Acquisition:** Using multicast-based data transmission from the reference receivers.

- **Data Logging:** Capturing and storing binary logs at the ground stations.
- **Data Processing:** Decoding and parsing binary log files to extract range measurement and navigation information.
- **Visualization:** Presenting real-time data via a web-based user interface.
- **Multi-Channel Support:** Simultaneous processing of data from different NavIC channels (e.g., Station 1 and Station 2).

4. System Architecture

4.1 High-Level Overview

The NavIC Ranging Data Monitoring System is composed of four major components:

- **Multicast Sender (Data Generation):**
  - Reads binary log files (e.g., for Station 1 and Station 2).
  - Transmits NavIC data via UDP multicast.
- **Multicast Receiver (Data Logging):**
  - Listens to specific multicast groups.
  - Logs incoming binary data into station-specific log files.
- **Backend Server (Flask Application – Processing & API):**
  - Processes the logged binary files.
  - Decodes range measurement data () and navigation data ().
  - Provides a REST API endpoint to supply processed data to the frontend.
- **Frontend (HTML, JavaScript – UI for Visualization):**
  - Fetches data via the API.
  - Dynamically displays range measurement data and navigation logs in tabular format.

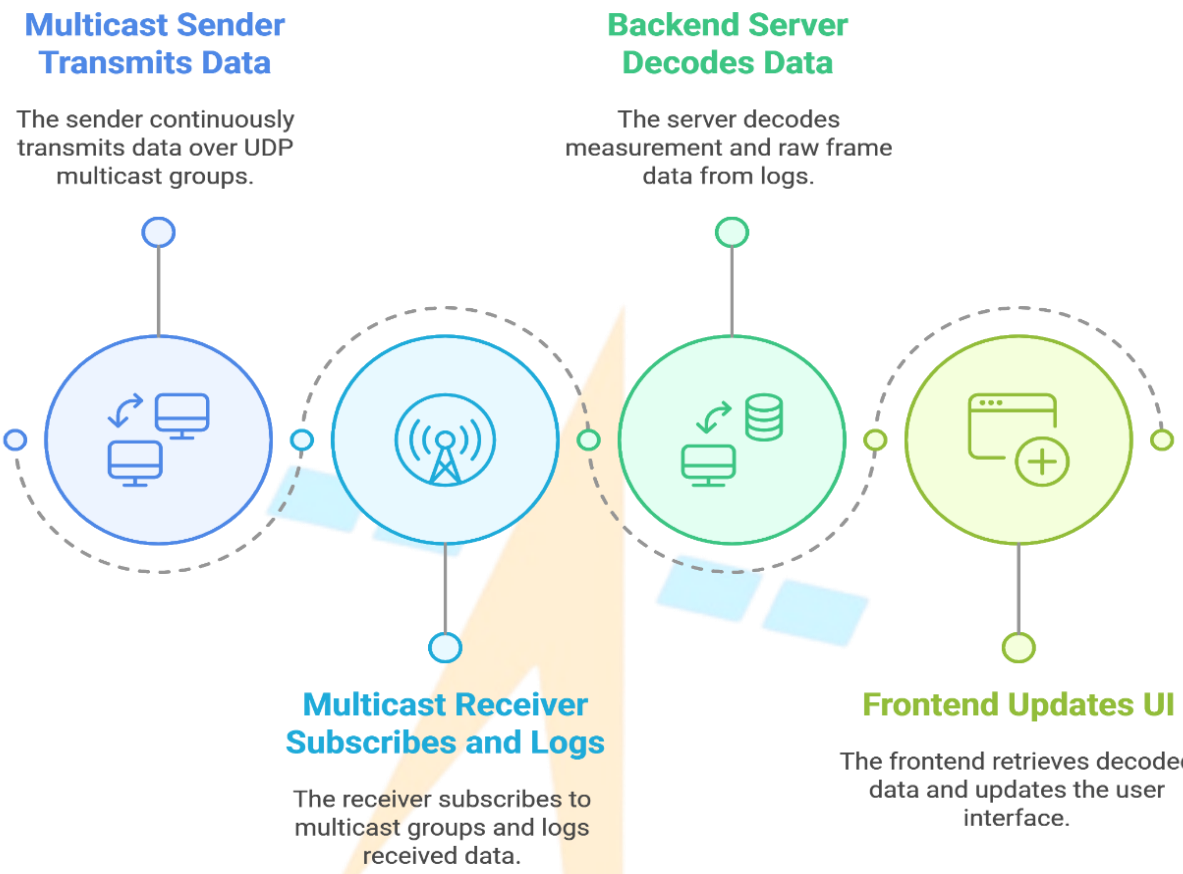


Figure 2 Multicast Data Flow

4.2 Data Flow Explanation

- **Step 1:** The **Multicast Sender** reads pre-recorded binary log files and continuously transmits data over designated UDP multicast groups.
- **Step 2:** The **Multicast Receiver** subscribes to these multicast groups and writes the received data into log files (one for each station, such as Station 1 and Station 2).
- **Step 3:** The **Backend Server** (implemented using Flask) runs multiple threads—each monitoring a log file for a specific station. It decodes the binary data:
    - For **range measurement data ():** The backend extracts parameters like pseudo range, Doppler, carrier-to-noise ratio, etc.

- For **navigation data ():** The backend processes the navigation sub frame details.

➢ **Step 4:** The **Frontend** periodically polls the backend API, retrieves the latest decoded data, and updates the UI, allowing users to switch channels (e.g., Station 1 or Station 2) to view respective data.

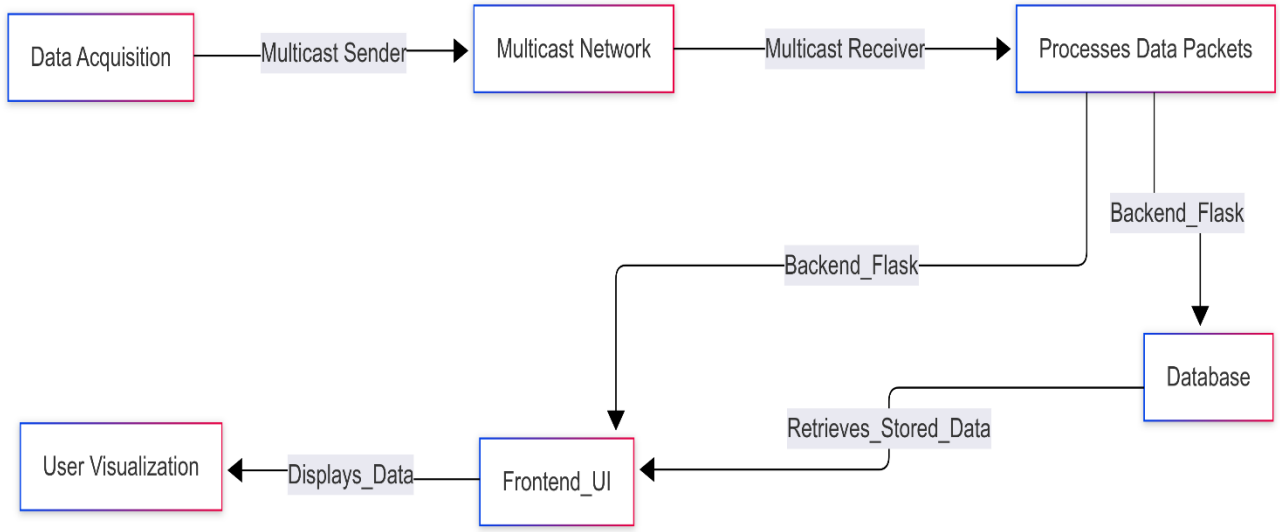


Figure 3 Data Flow Diagram

5. NavIC Data Handling & Binary Log Decoding

5.1 NavIC Log Structure

NavIC logs consist of a **binary header** and a variable-length payload. Key components include:

➢ **Sync Word:** A unique 4-byte pattern (e.g., 0xAACC4756) that identifies the beginning of a log message.

➢ **Message Length & ID:** These fields specify the total length of the message and the type for range measurement data and for navigation data.

➢ **Log Count, Time Status, Week, and Milliseconds:** These fields capture timing and sequencing information.

➢ **CRC:** A 32-bit cyclic redundancy check to ensure data integrity.

5.2 Decoding Process for Range Measurement Data ()

- The decoding function reads the **header** and unpacks it using the Python struct module.
- **Range measurement Data Fields:**
  - **Observations Count:** Indicates the number of satellite observation records.
  - Each observation record (64 bytes) contains parameters like PRN, SV channel, signal channel, hardware channel, pseudo range (PSR), PSR standard deviation, Doppler frequency, carrier-to-noise ratio (C/No), lock time, and channel status.
- The decoded data is stored in a buffer indexed by channel (e.g., “Station 1” or “Station 2”).

5.3 Decoding Process for Navigation Data ()

- Similar to range measurement data, the navigation data is decoded from its binary form.
- **Navigation Fields:**
  - Includes fields such as signal channel, hardware channel, PRN, signal type, parity status, parity failures, and raw data bits/bytes.
- The backend appends these records to a buffer for the corresponding channel.

5.4 Handling Data Fragmentation

- For larger data packets, the NavIC system uses **fragmentation**:
  - A data packet is divided into multiple fragments if it exceeds a certain size.
  - Each fragment contains part of the data along with a fragment number.

- The backend logic uses the **sequence number** and **fragment number** from the header to reassemble the complete data message.

#	Field Name	Description	Format	# Bytes	Binary Offset
1	**header**	Log header	H	0	0
2	**#obs**	Number of data sets to follow (for the tracked channels)	Ulong	4	H
3	**prn**	Satellite PRN tracked	Ushort	2	H+4
4	**svChan**	SV channel	Ushort	2	H+6
5	**sigChan**	Signal Channel	Ushort	2	H+8
6	**HWchan**	Hardware channel number	Ushort	2	H+10
7	**Tracktype**	Indicates tracking type data (See Table 5-10)	Ulong	4	H+12
8	**Psr**	Pseudorange measurement (m)	double	8	H+16
9	**Psr std**	Pseudorange measurement standard deviation (m)	float	4	H+24
10	**adr**	Accumulated Doppler range (cycles)	double	8	H+28
11	**Adr std**	Carrier phase standard deviation (cycles)	float	4	H+36
12	**Dopp**	Instantaneous Doppler frequency (Hz)	float	4	H+40
13	**C/No**	Carrier to noise density ratio (dB-Hz)	float	4	H+44
14	**Reserved**	Reserved	Ulong	4	H+48
15	**Reserved**	Reserved	Ulong	4	H+52
16	**Locktime**	Number of seconds of continuous tracking	double	4	H+56
17	**Channel status**	Channel tracking status (see Table 5-6)	Ulong	8	H+64
18	**Next data set**	Offset = H + 4 + (#previous obs x 56)	variable	-	H + 4 + (#obs x 64)
19	**32-bit CRC**	Checksum validation	Hex	4	H + 4 + (#obs x 64)

Figure 4 NavIC Log Structure

6. Implementation Details

6.1 Multicast Sender – daps_mcast_sender.py

- **Functionality:**
  - Reads binary log files (e.g., Station1_daps_log_065.dat or Station2_daps_log_065.dat).
  - Identifies valid data segments using sync words.
  - Extracts timestamp and message type.
  - Transmit data over two multicast groups for Station 1 and Station 2.
- **Key Considerations:**
  - **Timestamp Synchronization:** Ensures that data is sent in real time.
  - **Fragmentation Handling:** Properly transmits fragmented data as per NavIC protocol.

6.2 Multicast Receiver – multicast_rx.py

- **Functionality:**
  - Joins multicast groups to receive data.
  - Uses UDP sockets with appropriate options (e.g., setting TTL correctly with an unsigned byte).
  - Writes received data into station-specific log files (e.g., Station 1_daps_log_065.dat and Station 2_daps_log_065.dat).
- **Key Correction:**
  - TTL value is correctly packed using pack ('B', 255) instead of a signed byte.

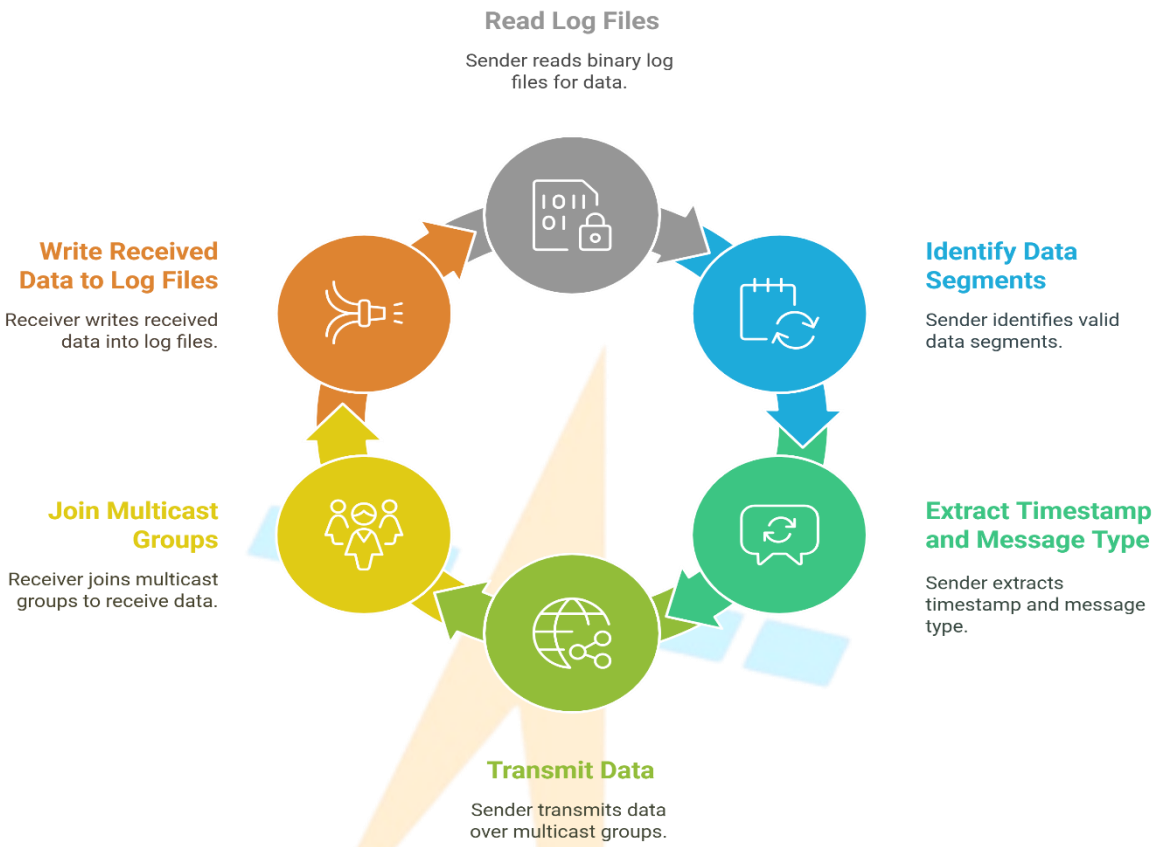


Figure 5 Multicast Sender-Receiver Workflow

6.3 Backend Server – app.py

- **Structure:**
  - Uses **Flask** to serve a web interface and provide REST API endpoints.
  - Launches separate file-reading threads for different channels (e.g., one for Station 1 and one for Station 2).
- **Data Processing:**
  - Each thread reads its assigned log file, decodes range measurement and navigation data, and updates global buffers.
  - The /fetch-data endpoint returns the latest processed data for the selected channel.

- **Error Handling:**
    - Ensures that data for non-active channels is not displayed on the UI.

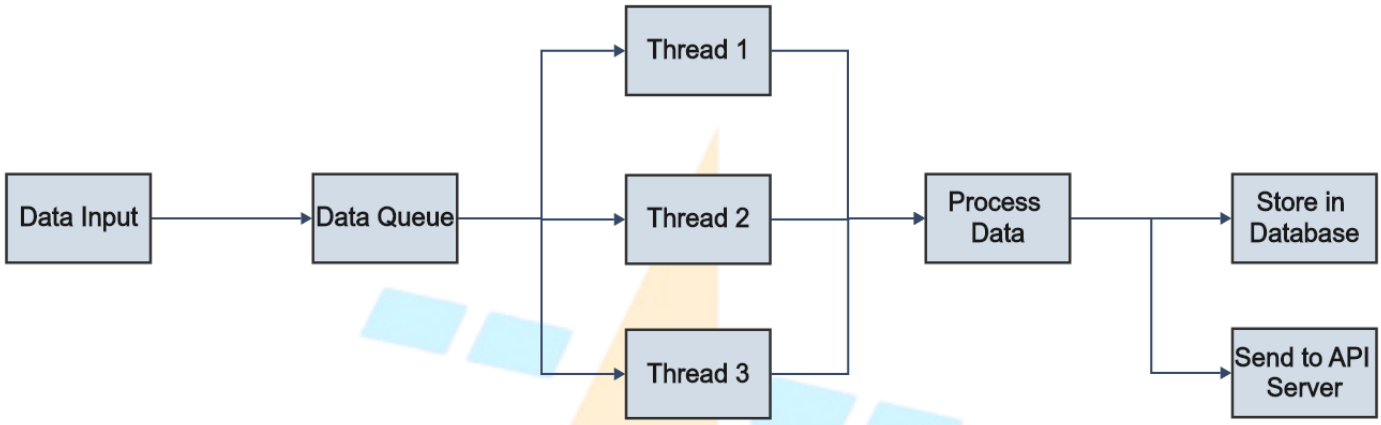


Figure 6 Backend Data Processing

6.4 Frontend – index.html and script.js

- **User Interface:**
    - Presents channel buttons allowing users to select different stations.
    - Contains tabs for displaying **Summary**, **Range measurements**, **Navigation Data**, and **Configuration**.
- **Dynamic Updates:**
    - Uses JavaScript to periodically poll the Flask API.
    - Updates the UI tables based on the fetched data.
- **Channel-Specific Data Handling:**
    - When a channel (e.g., Station 1 or Station 2) is selected, only its data is shown; any residual data from another channel is cleared.

7. Testing and Results

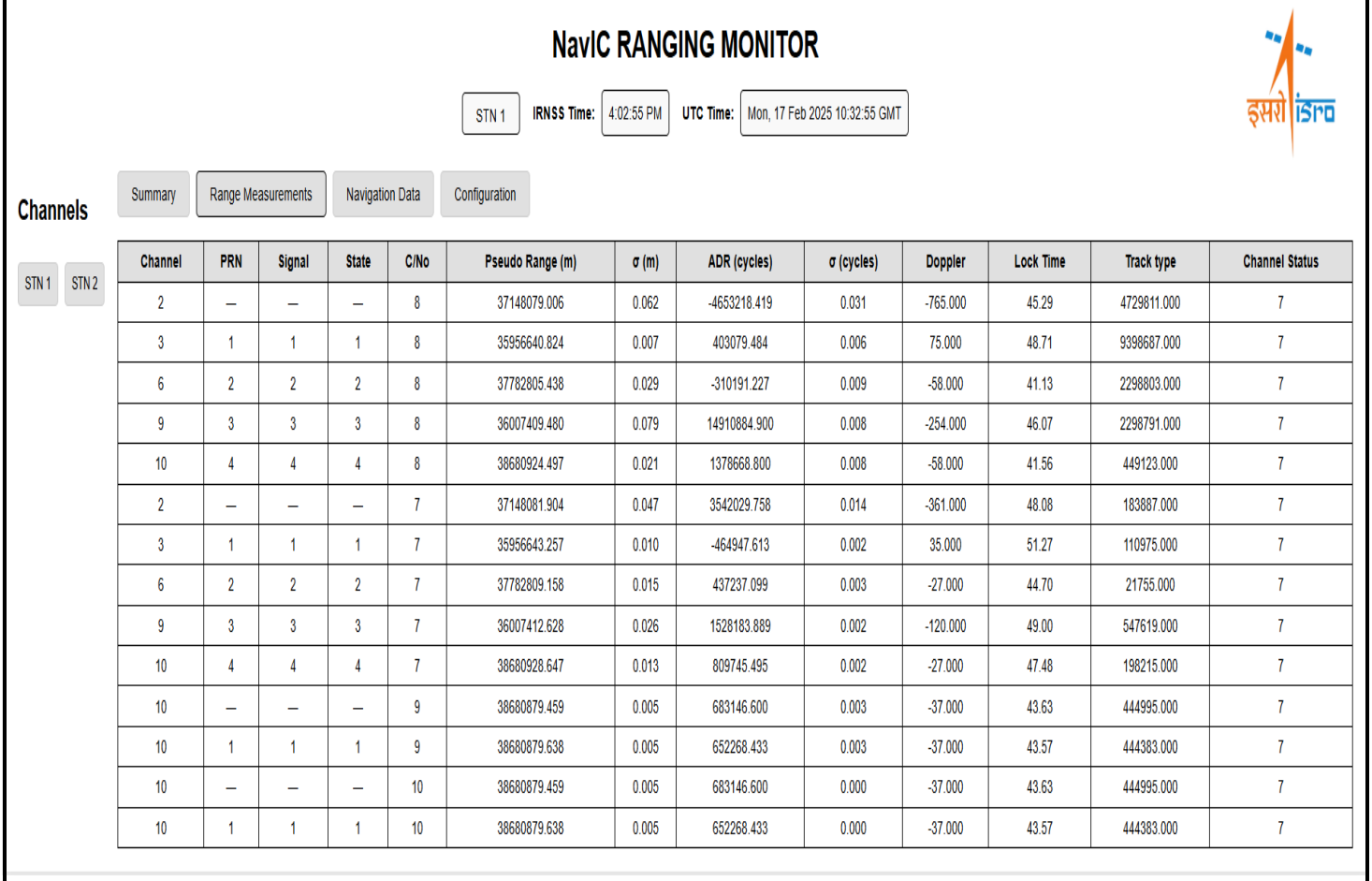
7.1 Testing Methodology

- **Test Environment:**
  - Deployed the system on a test network with simulated NavIC data for both Station 1 and Station 2.
  - Utilized two separate terminals for data sender and receiver, and one for the backend server.
- **Test Cases:**
  1. **Channel Verification:**
     - Verify that clicking on "Station 1" displays only Station 1 data.
     - Verify that clicking on "Station 2" displays only Station 2 data.
  2. **Data Integrity:**
     - Compare the decoded parameters (pseudo range, Doppler, C/No, etc.) with expected values.
  3. **Real-Time Updates:**
     - Confirm that the UI updates every second with new data.
  4. **Fragmentation Handling:**
     - Test with large data packets to ensure that fragmentation and reassembly work as intended.

7.2 Results Analysis

- **Successful Data Reception:**
  - Both Station 1 and Station 2 channels received and displayed the correct range measurement and navigation data.

- **UI Behaviour:**
  - The web interface correctly cleared data when switching channels, ensuring no residual data from the previous channel.
- **Log Decoding Accuracy:**
  - Backend logs confirmed the correct decoding of both range measurement and navigation data.
- **Performance Metrics:**
  - The system's latency and update frequency met the real-time requirements.



NavIC RANGING MONITOR

STN 1 | IRNSS Time: 4:02:55 PM | UTC Time: Mon, 17 Feb 2025 10:32:55 GMT

Channels

STN 1 STN 2

Summary | Range Measurements | Navigation Data | Configuration

Channel	PRN	Signal	State	C/No	Pseudo Range (m)	σ (m)	ADR (cycles)	σ (cycles)	Doppler	Lock Time	Track type	Channel Status
2	–	–	–	8	37148079.006	0.062	-4653218.419	0.031	-765.000	45.29	4729811.000	7
3	1	1	1	8	35956640.824	0.007	403079.484	0.006	75.000	48.71	9398687.000	7
6	2	2	2	8	37782805.438	0.029	-310191.227	0.009	-58.000	41.13	2298803.000	7
9	3	3	3	8	36007409.480	0.079	14910884.900	0.008	-254.000	46.07	2298791.000	7
10	4	4	4	8	38680924.497	0.021	1378668.800	0.008	-58.000	41.56	449123.000	7
2	–	–	–	7	37148081.904	0.047	3542029.758	0.014	-361.000	48.08	183887.000	7
3	1	1	1	7	35956643.257	0.010	-464947.613	0.002	35.000	51.27	110975.000	7
6	2	2	2	7	37782809.158	0.015	437237.099	0.003	-27.000	44.70	21755.000	7
9	3	3	3	7	36007412.628	0.026	1528183.889	0.002	-120.000	49.00	547619.000	7
10	4	4	4	7	38680928.647	0.013	809745.495	0.002	-27.000	47.48	198215.000	7
10	–	–	–	9	38680879.459	0.005	683146.600	0.003	-37.000	43.63	444995.000	7
10	1	1	1	9	38680879.638	0.005	652268.433	0.003	-37.000	43.57	444383.000	7
10	–	–	–	10	38680879.459	0.005	683146.600	0.000	-37.000	43.63	444995.000	7
10	1	1	1	10	38680879.638	0.005	652268.433	0.000	-37.000	43.57	444383.000	7

Figure 7 Station 1 Range measurement Data

NavIC RANGING MONITOR

STN 1 | IRNSS Time: 4:04:58 PM | UTC Time: Mon, 17 Feb 2025 10:34:58 GMT

Channels

Summary | Range Measurements | Navigation Data | Configuration

STN 1 STN 2

SC	HC	PRN	Signal	Parity Status	Parity Failure	Bits	Bytes
–	–	2	8	–	–	292	40
1	1	3	8	–	–	292	40
2	2	6	8	–	–	292	40
3	3	9	7	–	–	292	40
3	3	9	8	–	–	292	40
4	4	10	7	–	–	292	40
4	4	10	8	–	–	292	40

Figure 8 Station 1 Navigation Data

NavIC RANGING MONITOR

STN 2 | IRNSS Time: 4:03:54 PM | UTC Time: Mon, 17 Feb 2025 10:33:54 GMT

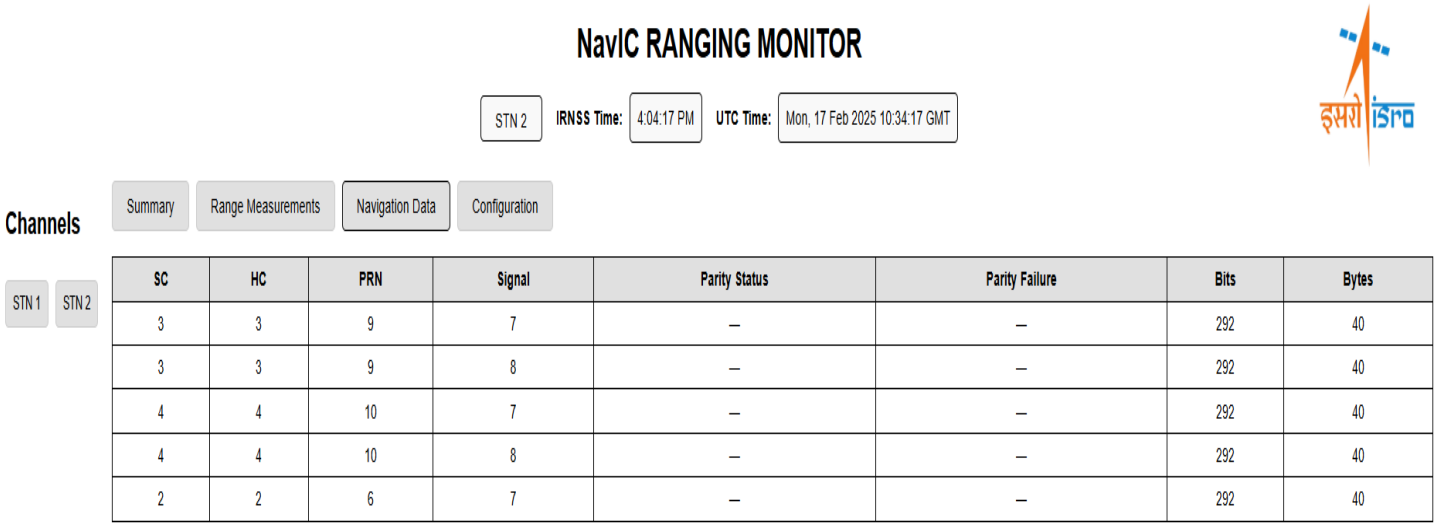
Channels

Summary | Range Measurements | Navigation Data | Configuration

STN 1 STN 2

Channel	PRN	Signal	State	C/No	Pseudo Range (m)	σ (m)	ADR (cycles)	σ (cycles)	Doppler	Lock Time	Track type	Channel Status
2	–	–	–	8	38289288.239	0.075	4846530.390	0.042	-1072.000	43.04	23379.000	7
3	1	1	1	8	36450786.132	0.010	-21647.255	0.010	134.000	46.08	118347.000	7
6	2	2	2	8	38388451.442	0.019	2144292.014	0.011	-71.000	40.84	26211.000	7
9	3	3	3	8	36068520.105	0.038	10242923.128	0.010	121.000	44.69	119511.000	7
10	4	4	4	8	38686508.373	0.016	2991896.435	0.009	-122.000	40.18	33411.000	7
2	–	–	–	7	38289279.009	0.034	4212859.255	0.018	-505.000	48.36	18291.000	7
3	1	1	1	7	36450775.579	0.005	-204380.629	0.002	63.000	51.67	115323.000	7
6	2	2	2	7	38388442.391	0.011	996971.635	0.003	-34.000	45.72	25575.000	7
9	3	3	3	7	36068510.411	0.036	4546606.311	0.002	57.000	48.64	118743.000	7
10	4	4	4	7	38686499.215	0.014	1435688.990	0.003	-58.000	45.91	32355.000	7
10	–	–	–	9	38686463.013	0.006	1925189.620	0.002	-78.000	46.99	32259.000	7
10	1	1	1	9	38686462.870	0.005	1972588.401	0.002	-78.000	46.82	27993.000	7
10	–	–	–	10	38686463.013	0.006	1925189.620	0.000	-78.000	46.99	32259.000	7
10	1	1	1	10	38686462.870	0.005	1972588.401	0.000	-78.000	46.82	27993.000	7

Figure 9 Station 2 Range measurement Data



SC	HC	PRN	Signal	Parity Status	Parity Failure	Bits	Bytes
3	3	9	7	–	–	292	40
3	3	9	8	–	–	292	40
4	4	10	7	–	–	292	40
4	4	10	8	–	–	292	40
2	2	6	7	–	–	292	40

Figure 10 Station 2 Navigation Data

7.3 Debugging and Iteration

- **Issues Encountered:**
  - Initially, Station 1 data was displayed for Station 2 channel selection due to shared buffer handling.
  - Multicast receiver TTL packing error (resolved by switching from signed to unsigned byte).
- **Resolutions Implemented:**
  - Modified buffer management in the frontend.
  - Updated multicast receiver code to use pack ('B', 255).

8. Challenges and Solutions

8.1 Multi-Channel Processing

- **Challenge:**
  - Processing data from both Station 1 and Station 2 in a single backend instance without cross-contamination.

- **Solution:**
    - Implemented multi-threading in the Flask backend (`app.py`) to handle separate log files for each channel.
    - Ensured that API endpoints correctly filtered data by channel.

8.2 Multicast Reception Issues

- **Challenge:**
    - Incorrect TTL value packaging resulted in errors during multicast reception.
- **Solution:**
    - Updated the multicast_rx.py file to pack the TTL value as an unsigned byte pack ('B', 255).

8.3 Data Fragmentation and Reassembly

- **Challenge:**
    - Handling fragmentation in data packets to ensure full message reassembly.
- **Solution:**
    - Utilized sequence and fragment numbers from the NavIC header for proper reassembly.

9. Future Scope

9.1 Expanding Station Support

- **Objective:**
    - Extend the system to support additional NavIC stations beyond Station 1 and Station 2.

- **Approach:**
    - Scale the backend by adding more threads for additional log files.

9.2 Database Integration

- **Objective:**
    - Transition from in-memory buffers to a persistent database.
- **Benefits:**
    - Enables historical data analysis and long-term storage.
- **Potential Databases:**
    - PostgreSQL, MongoDB, or any other NoSQL database.

9.3 Advanced Visualization

- **Objective:**
    - Enhance the user interface with advanced data visualization tools.
- **Approach:**
    - Incorporate charts, graphs, and heat maps for deeper insights.
    - Use libraries such as D3.js or Chart.js for dynamic visualizations.

9.4 AI-Based Anomaly Detection

- **Objective:**
    - Integrate machine learning models to automatically detect anomalies or signal degradations.
- **Approach:**
    - Train models on historical NavIC data.
    - Trigger alerts when abnormal patterns are detected.

10. Conclusion

10.1 Summary of Achievements

- Developed a **real-time NavIC Ranging Data Monitoring System** capable of processing and displaying data from multiple stations.
- Implemented robust **multicast data transmission and reception** mechanisms.
- Achieved accurate **binary log decoding** for both range measurement and navigation data.
- Successfully integrated a **Flask-based backend** with a dynamic **web-based frontend**.

10.2 Impact and Future Enhancements

- The system provides a scalable platform for real-time satellite monitoring.
- Future enhancements include support for additional stations, database integration, advanced visualizations, and AI-based analytics to further improve system performance and reliability.

11. References

1. **ISRO Documentation:**
    - EICD for NavIC Ranging Reference Receiver.
    - IRNSS SIGNAL-IN-SPACE ICD.
2. **Technical References:**
    - Python’s struct module documentation.
    - Flask framework documentation.