WISUN ENVIRONMENTAL MONITORING

KernelKrew

Team 25

By Janya Bhatt, Keshvi Agarwal, Shrawani Nanda, Poojitha J

PROBLEM STATEMENT

- Use a simple environmental monitoring sensor - SI7021 and interface with the existing Wi-SUN network on campus.
- Check signal strengths at various points.

MOTIVATION

- Traditional Wi-Fi or LoRa systems have limitations in range, scalability, and power efficiency.
- Wi-SUN (Wireless Smart Ubiquitous Network) offers a robust, low-power, long-range mesh network ideal for large-scale sensor deployments.
- Integrating a simple sensor like SI7021 helps demonstrate practical data acquisition and transmission over Wi-SUN.
- Enables future expansion into smart campus applications - such as smart energy management, weather tracking, or pollution monitoring.
- Conduct pre-deployment testing for a planned SCRC campus network to provide expected performance data for different Wi-SUN configurations.

HARDWARE

SENSOR:

- SI7021 Temperature & Humidity Sensor - Measures temperature and relative humidity.

MICROCONTROLLERS:

- EFR32FG25: Acts as the sensor node (Fully Functioning Node). Interfaces with SI7021 sensor to read temperature and humidity and supports Wi-SUN communication for sending data to border router.
- EFR32ZG28: Initially configured as the Linux border router for testing purposes. Receives data from multiple FFNs and manages network routing. Connects Wi-SUN network to PC or cloud for data collection and analysis.

MICROCONTROLLERS

EFR32ZG28

EFR32FG25

INITIAL PHASE

1 - Familiarization, Hardware & Firmware Setup

- Explored documentation, installed Simplicity Studio (SSv5), and configured development environment.
- Familiarized with EFR32FG25 (FFN node) and EFR32ZG28 (Border Router) through LED blinking and flashing experiments.
- Learned flashing methods (.bin/.hex/.s37) and debugging workflow.

2 - Network Configuration

- Configured modules as FFN and Border Router, exchanged UDP messages.
- Resolved issues with IP addresses, sockets, and network name mismatches.
- Understood Wi-SUN FAN architecture, join states, and data flow.

INITIAL PHASE

3 - Linux Border Router Setup

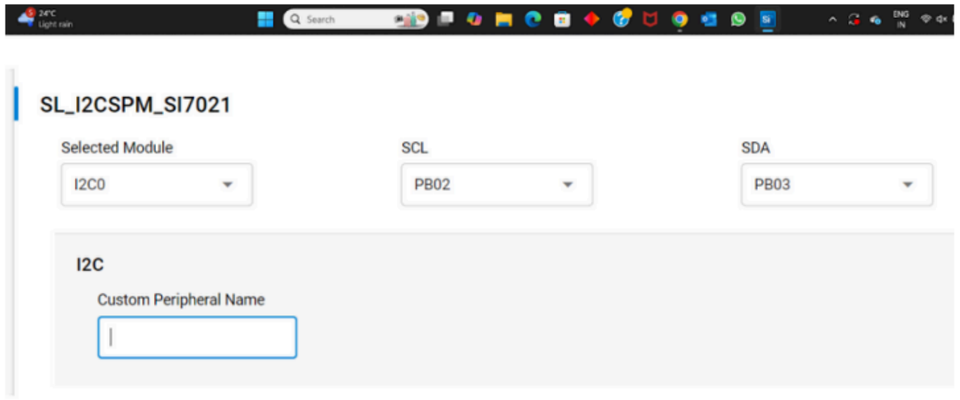
- Flashed RCP image and configured wsbrd settings.
- Installed dependencies, resolved library issues.
- Performed full USB/IP setup to expose radio over network

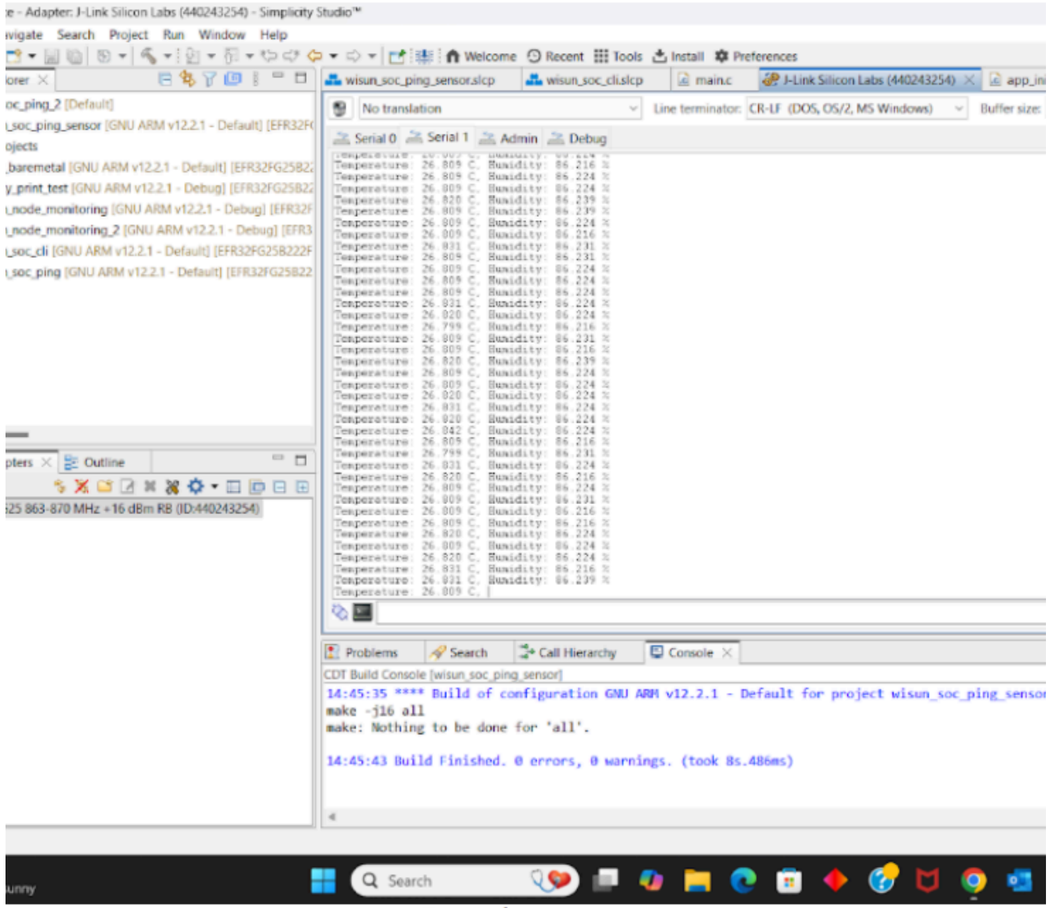
4 - Sensor Integration (SI7021)

- Connected and configured SI7021 via I^2C
- Pin Connections: VCC → 3.3V, GND → GND, SCL → PB02, SDA → PB03.
- Configured I2CSPM driver for SI7021 sensor (Instance: si7021, Module: I2C0).
- Driver Installation: Installed SI70xx sensor driver (no extra configuration needed).
- Modified code in app.c and app_init.c, then successfully printed real-time temperature and humidity on serial console.

SENSOR INTEGRATION

Data sheet to map the pin configurations





Serial output

WISUN COMMUNICATION

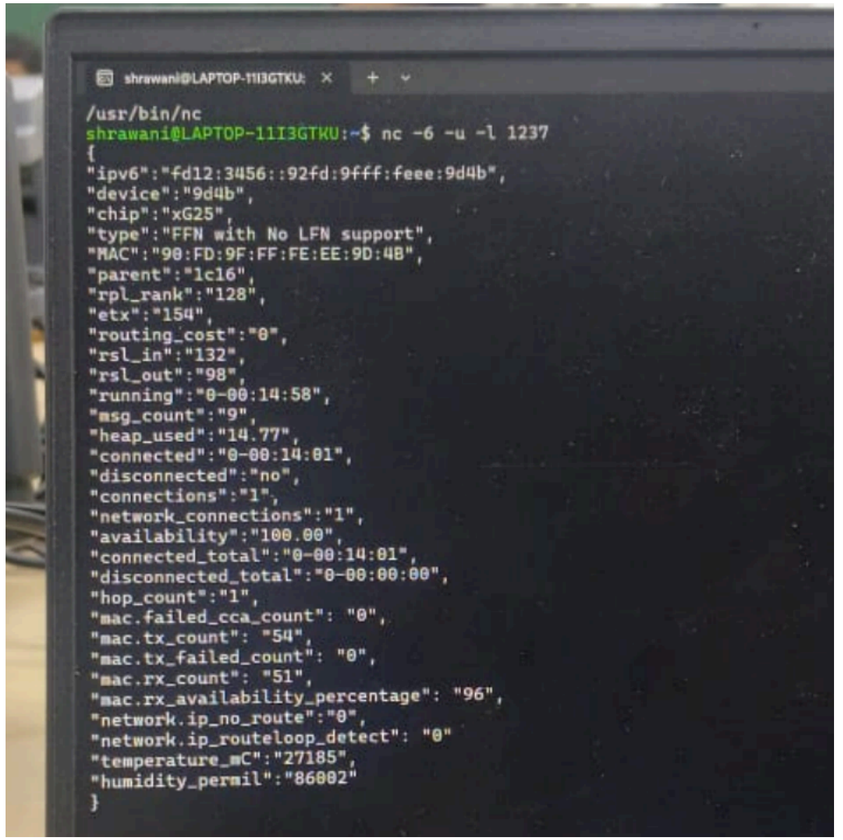
5 - Data Transmission

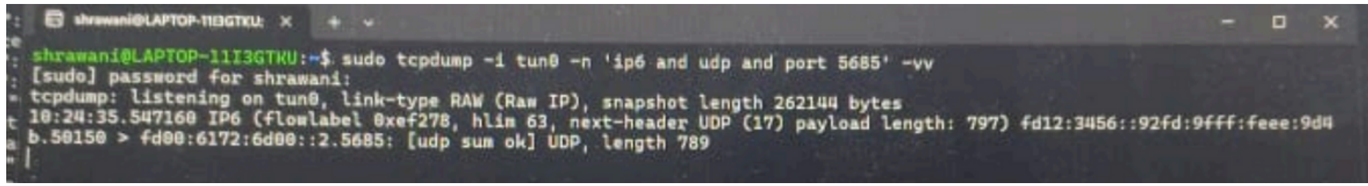
- We modified the Node Monitoring project to send temperature and humidity values in JSON over UDP and CoAP to the border router.
- We created the bootloader-storage-spiflash-single-1024k project, installed lz4 and lzma for lossless data compression, installed ota-dfu.
- We configured the network name, and updated app.c to transmit temperature and humidity over UDP.

- Set CoAP to true, then built and flashed both the bootloader and the node monitoring project (ensuring the bootloader wasn't erased).
- The border router ran on the EFR32ZG28 (RCP module), and the node (EFR32FG25) connected to it.
- JSON data was sent from the node to the border router over UDP and CoAP using:
- sudo tcpdump -i tun0 -n -A 'ip6 and udp and port 5685'
- nc -6 -u -l 5685
- A Python script extracted the temperature and humidity fields and uploaded them to ThingSpeak.

WISUN COMMUNICATION

- Set CoAP to true, then built and flashed both the bootloader and the node monitoring project (ensuring the bootloader wasn't erased).
- The border router ran on the EFR32ZG28 (RCP module), and the node (EFR32FG25) connected to it.
- JSON data was sent from the node to the border router over UDP and CoAP using:
- sudo tcpdump -i tun0 -n -A 'ip6 and udp and port 5685'
- nc -6 -u -l 5685
- A Python script extracted the temperature and humidity fields and uploaded them to ThingSpeak.

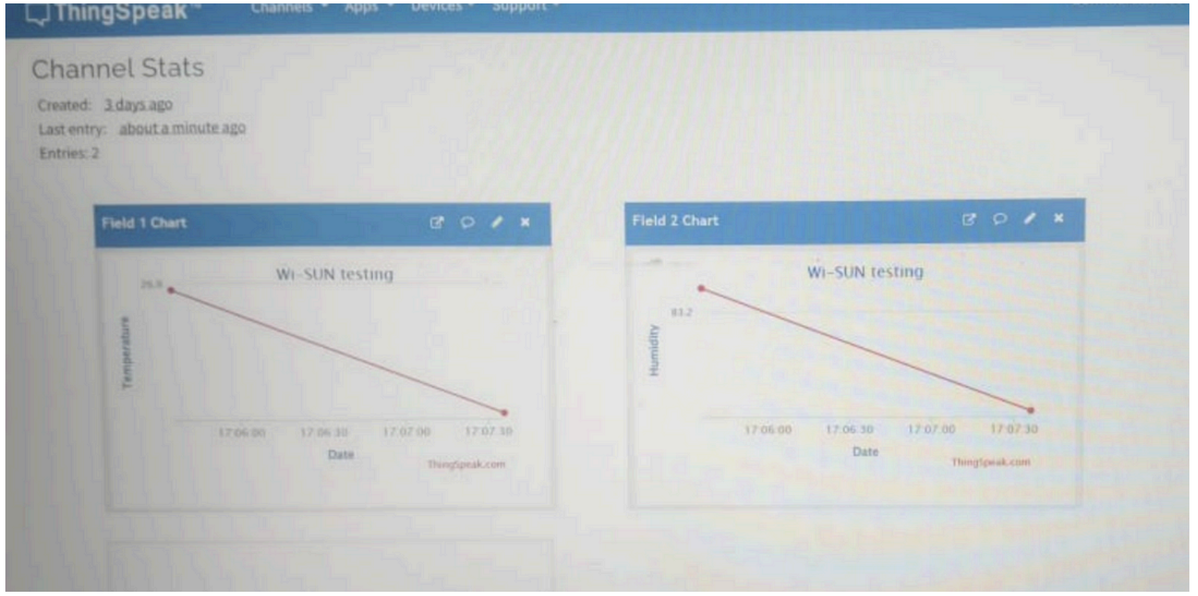




CLOUD INTEGRATION

6 - Cloud Integration & Visualization

- Wrote Python script that requests data from node every 20 seconds via COAP and then parses JSON at Border Router to get values for different parameters like rsl_in, rsl_out, rpl_rank, connected_total, disconnected_total, hop_count, temperature, humidity, etc. and then it uploads this data to ThingSpeak using API keys in real time which helps visualize and analyze data for all parameters.



JOIN STATES

In Silicon Labs Wi-SUN, there are 5 main join states. The sl_wisun_app_core_get_join_state() application call returns the current join state

Join State 0

- DISCONNECTED(0) – SL_WISUN_JOIN_STATE_DISCONNECTED
- Device is not trying to connect. Border Router and already-connected routers are sending PAN Advert frames on all channels.

Join State 1

- SELECT_PAN (1) – SL_WISUN_JOIN_STATE_SELECT_PAN
- Device listens for PAN Advert frames and may send PAN Advert Solicit to speed things up, in order to choose which PAN (network) to join.

JOIN STATES

Join State 2

- AUTHENTICATE (2) – SL_WISUN_JOIN_STATE_AUTHENTICATE
- Security key exchange between the device and the Border Router (can involve an external RADIUS server).

Join State 3

- ACQUIRE_PAN_CONFIG (3) – SL_WISUN_JOIN_STATE_ACQUIRE_PAN_CONFIG
- Device listens for PAN Config frames and may send PAN Advert Solicit to get them faster; it learns network parameters from the Border Router.

Join State 4

- CONFIGURE_ROUTING (4) – SL_WISUN_JOIN_STATE_CONFIGURE_ROUTING
- Routing is set up (with internal sub-steps like parent selection, DHCP, address registration, DAO).

JOIN STATES

- Parent selection: The device compares potential parents (routers it can hear) and chooses the “best” one as its preferred parent for routing toward the Border Router.

- DHCP: The device obtains an IPv6 address from the WiSUN Border Router’s DHCPv6 server (internal or external), so it can participate in IP routing.

- DAO: The device sends a Destination Advertisement Object (DAO) upstream so that routes toward this node are installed in the RPL topology (i.e., others can reach it through the chosen parent).

Join State 5:

- OPERATIONAL (5) – SL_WISUN_JOIN_STATE_OPERATIONAL
- Device is fully connected and can send/receive IPv6 traffic; routers/FFNs start sending PAN Adverts so other nodes can join.

SOME BASIC TERMS

- Multipath: Radio waves may reach the receiver via multiple paths due to reflections from objects such as walls and trees; the signals arrive with different delays and phases, and when they combine they can cause partial or near-total cancellation of the signal (fading).
- Fading: The phenomenon where small movements of the transmitter or receiver, or phase differences between multiple reflected signal paths, cause large variations in link quality, including partial or near-total signal cancellation.

ERRORS WE FACED

```
wisun udp_server 5002
[Listening: 1]
> [Join state 4: Configure Routing - DHCP]
wisun socket_writeto 1 fd12:3456::92fd:9fff:feee:9dfb 5001 "hello from ffn"
[Wrote 14 bytes]
> [Data sent: 1,2048]
wisun socket_writeto 3 fd12:3456::92fd:9fff:feee:9dfb 5001 "hello from ffn"
[Failed: unable to find the specified socket]
>
```

```
wisun ping fd12:3456::92fd:9fff:feee:9dfb

wisun get wisun.network_name
wisun.network_name = "Wi-SUN Network"

wisun socket_writeto 1 fd12:3456::92fd:9fff:feee:9dfb 1234 "hello from ffn"

wisun socket_writeto 1 fd12:3456::92fd:9fff:feee:9dfb 1234 "hello from ffn"
wisun udp_client fd12:3456::92fd:9fff:feee:9dfb 1234
[Opened: 3]
> wisun socket_writeto 3 fd12:3456::92fd:9fff:feee:9dfb 1234 "hello from ffn"
[Wrote 14 bytes]
[Failed: data sent, error 49 (socket 3)]
>
wisun socket_writeto 4 fd12:3456::92fd:9fff:feee:9dfb 1234 "hello from ffn"
[Failed: unable to find the specified socket]
```

```
> wisun join_fan11
[Failed: already connecting or connected]
wisun get wisun
wisun.join_state = Join state 4: Configure Routing - parent selection (4)
wisun.network_name = "Wi-SUN Network"
wisun.phy_config_type = FAN 1.1 (1)
wisun.regulatory_domain = EU (3)
wisun.operating_class = 1 (unused)
wisun.operating_mode = 0x1a (unused)
wisun.fec = 0 (unused)
wisun.chan_plan_id = 32
wisun.phy_mode_id = 1
wisun.ch0_frequency = 863100 (unused)
wisun.number_of_channels = 69 (unused)
wisun.channel_spacing = 100 (unused)
wisun.crc_type = 4-bytes (2) (unused)
wisun.preamble_length = 56 (unused)
wisun.stf_length = 4 (unused)
wisun.protocol_id = 0 (unused)
wisun.channel_id = 0 (unused)
wisun.network_size = small (1)
wisun.tx_power = 20
wisun.unicast_dwell_interval = 255
wisun.mac_address = 90:fd:9f:ff:fe:ee:9d:4b
wisun.ip_addresses = …
wisun set wisun.network_name "Wi-SUN Network"
```

```
wisun ping fd12:3456::92fd:9fff:feee:9dfb
> wisun ping fd12:3456::92fd:9fff:feee:9dfb
PING fd12:3456::92fd:9fff:feee:9dfb: 40 data bytes
> 40 bytes from fd12:3456::92fd:9fff:feee:9dfb: icmp_seq=1 time=80 ms
```

Some logs: Dealing with errors, checking for values like IP Addresses, successful pings sent.

ERRORS WE FACED

```
shrawani@LAPTOP-11I3GTKU:~/test/wisun-br-linux$ sudo wsbrd -F examples/wsbrd.conf -u /dev/ttyACM0
[DBG ][rout]:      On-link (met 192)
[INFO][addr]: Address added to IF 1: fd12:3456::92fd:9fff:feee:9d4b
[INFO][rout]: Added route:
[DBG ][rout]:  fd12:3456::92fd:9fff:feee:9d4b/128 if:1 src:'Loopback' id:0 lifetime:infinite
[DBG ][rout]:      On-link (met 128)
Successfully registered to system DBus
[INFO][wsbs]: Wi-SUN packet congestion minTh 85, maxTh 170, drop probability 10 weight 32, Packet/Seconds 2
PAN version number update
  domain: IN
  channel plan id: 39
  phy mode id: 0x01
  phy operating modes: disabled
  RCP configuration index: 21
  channel 0 frequency: 865.1MHz
  channel spacing: 100kHz
  channel count: 29
  channel masks:
              advertised   effective
    unicast   --           00:00:00:e0
    broadcast --           00:00:00:e0
    async     --           00:00:00:e0

Nodes join ability:
  rank    FFN1.0    FFN1.1    LFN
    1     no        yes       yes
   >1     no        yes       can[1]
  [1]: neighboring routers must use a channel plan 2 (reg. domain & ChanPlanId)

Wi-SUN Border Router is ready
```

Successfully set up the border router!

- Faced with errors during installing dependencies, read the error logs and installed certain libraries.
- Edited the wsbrd config file to match with the configuration of network type like name, physical layer description, etc.

DEPLOYEMENT JOURNEY

FIELD DATA COLLECTION SETUP

https://earth.google.com/web/@17.44556809,78.34986099,629.38423145a,301.99695778d,30.00000014y,0h,0t,0r/data=CgRCAggBMikKJwolCiExYWZHMVNYQlQyd3Y1aTFqeE9YTld1c2hSMEoyWl92dEMgAToDCgEwQgIIAEoICJLsjsoBEAE?authuser=0

RF MAPPING

Goal: Collect multiple data points on different locations of the campus. The data points included all the main network and environmental parameters:

- Temperature - Measured by SI7021 sensor.
- Humidity - Measured by SI7021 sensor.
- RSL_out: Signal strength transmitted from the node (Outbound).
- RSL_in: Signal strength received by the node (Inbound).
- Hopcount: Number of mesh hops needed to reach the Border Router.
- RPL Rank: Routing “cost” value; higher rank = weaker/longer path.
- Connected_Total: connected_total is the cumulative time the device has been connected since its first connection
- Disconnected_Total: disconnected_total is the cumulative time it has been disconnected since its first connection

Setup and Procedure

Devices & Tools Used

- Wi-SUN Field Node: EFR32FG25 module
- Gateway/Border Router: Provided by SCRC
- Laptop: Power supply + live logging via Simplicity Studio
- Phone: Google Maps & Compass for accurate latitude/longitude
- Measuring Tape: Marking 10 m intervals

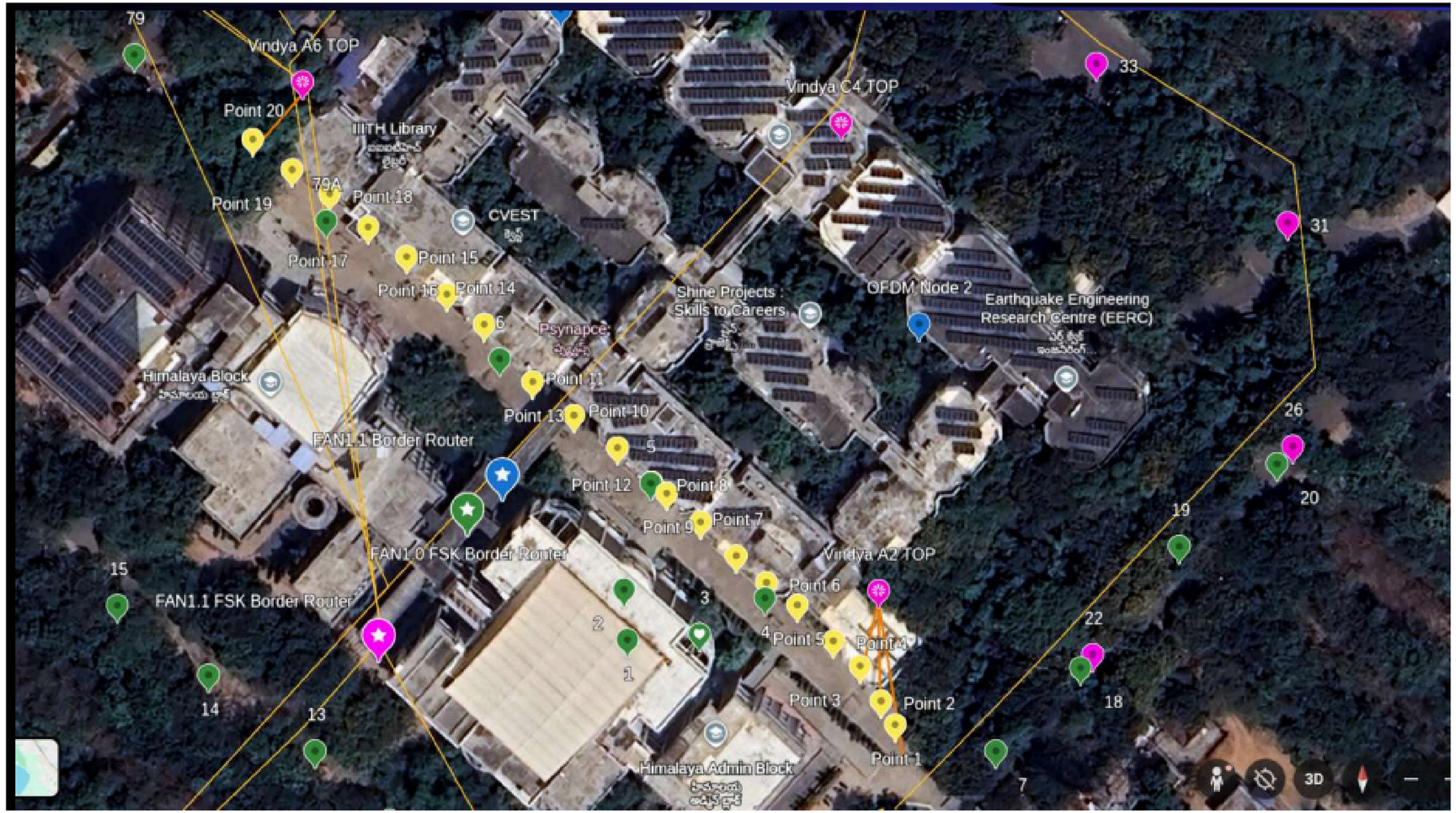
Sampling Details

- Data collected at every 10 m along Research Street
- 30 samples per location for each parameter

Procedure

- Marked measurement points using tape
- Updated configuration to connect the module to the campus network
- Built and flashed the Wi-SUN firmware
- Ran a Python script to send CoAP requests, collect node data, and upload it to ThingSpeak

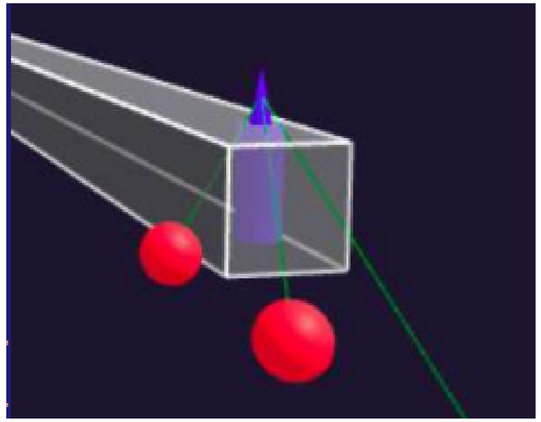
FSK - Research Street



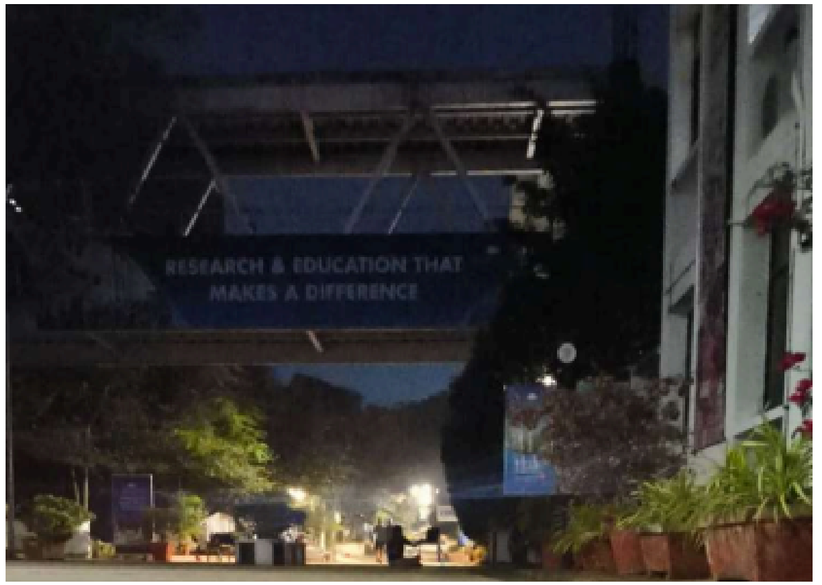
- The yellow points on the research street are the locations where we tested.(20 locations, each at a distance of 10m) .
- The pink points are where the campus deployed nodes are with the FSK modulation, our WiSUN module is supposed to connect with these pink nodes.

FSK - Research Street

Why Connectivity Failed:



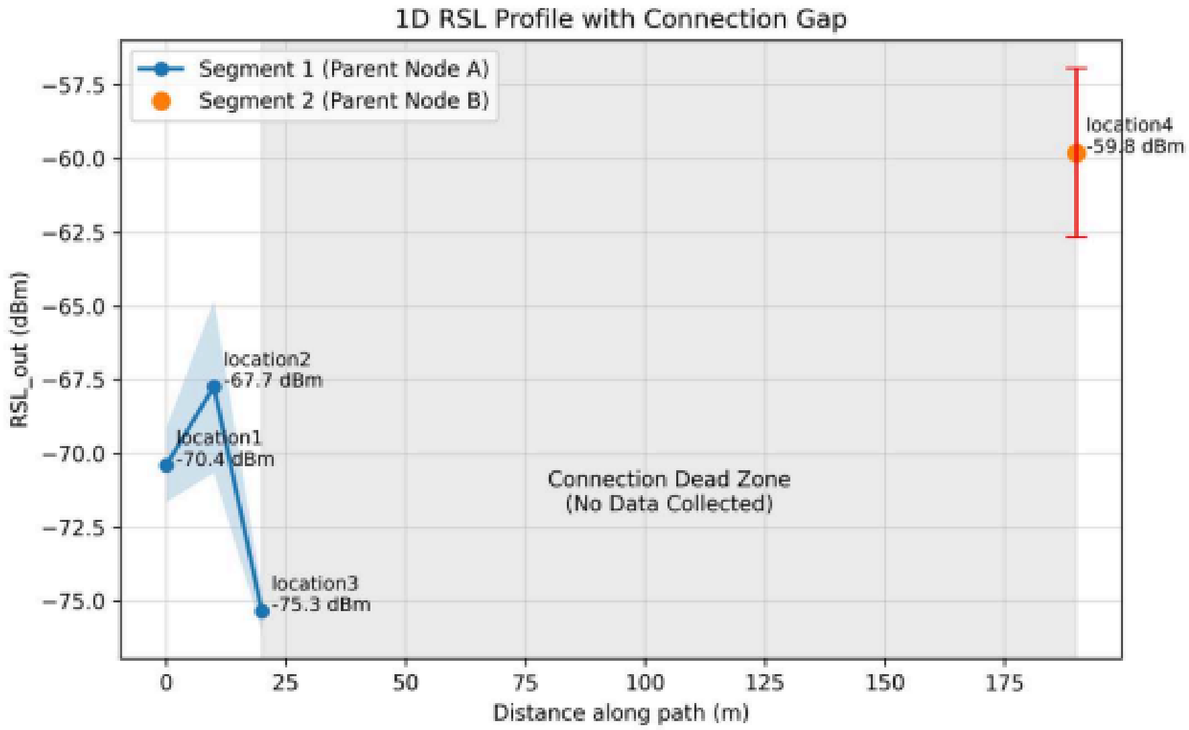
- Severe Line-of-Sight Blockage: Multi-floor Vindhya buildings on both sides create a “RF valley,” blocking rooftop FFNs.
- Metal Obstructions: The metal bridge on Research Street caused reflection + absorption → major dead zones.
- Dense Trees & Foliage: Significant radio absorption and shadowing, especially between Points 3–19.
- Low Node Density: Only 2 FFNs nearby (A2 & A6 rooftops). No alternate path → no mesh fallback → node never found a low-hop parent.
- High RPL Rank (1160–1760): Node failing to find reliable parent; costly routing; weak links.
- Result:
  - Connectivity only at Point 1–3 and Point 20.
  - 16/20 locations had no usable mesh path.



FSK - Research Street

Signal Analysis (Condensed)

- RSL_in and RSL_out highly correlated (ρ ≈ 0.96):
- → Failures caused by RF environment, not device issues.
- Location Clusters:
  - L4 strongest (–85 to –90 dB)
  - L1 moderate (–93 to –96 dB)
  - L2 weaker (–95 to –98 dB)
  - L3 weakest (< –100 dB)
- Environmental Effects: Weather (temp/humidity) had negligible impact.



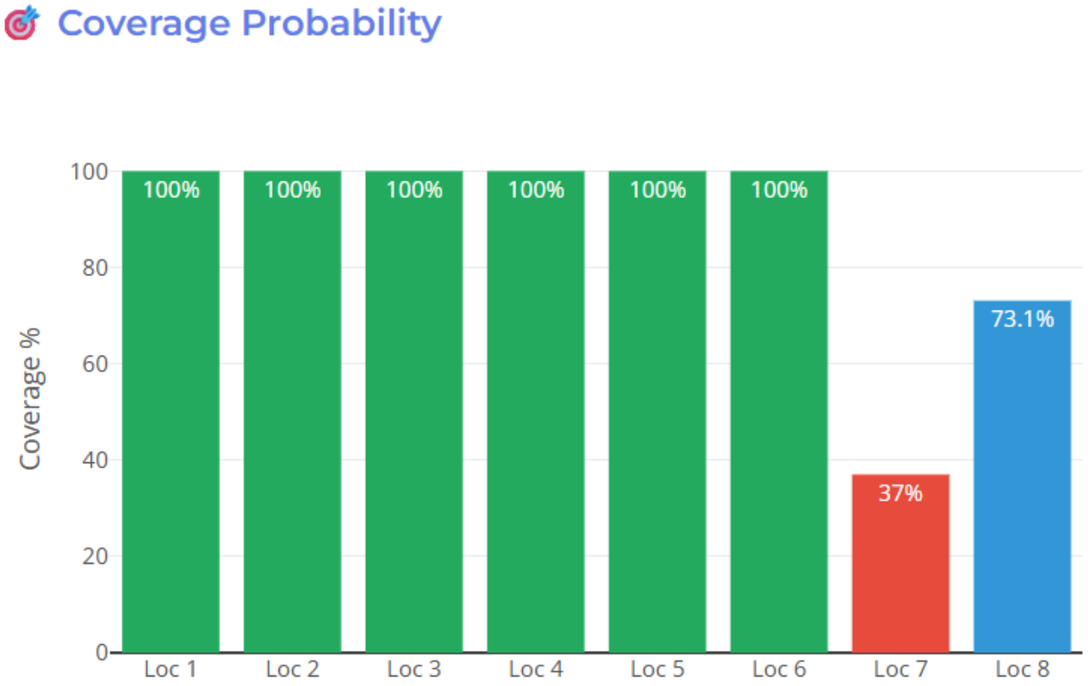
Recommendations

- Increase node density on Research Street (more FFNs for mesh fallback).
- Ensure clear line-of-sight through strategic rooftop / mid-height placements.
- Deploy one node on the metal bridge to bypass obstruction.
- Avoid covering devices (umbrella/sun shielding) & prevent overheating.

FSK - Basketball Street

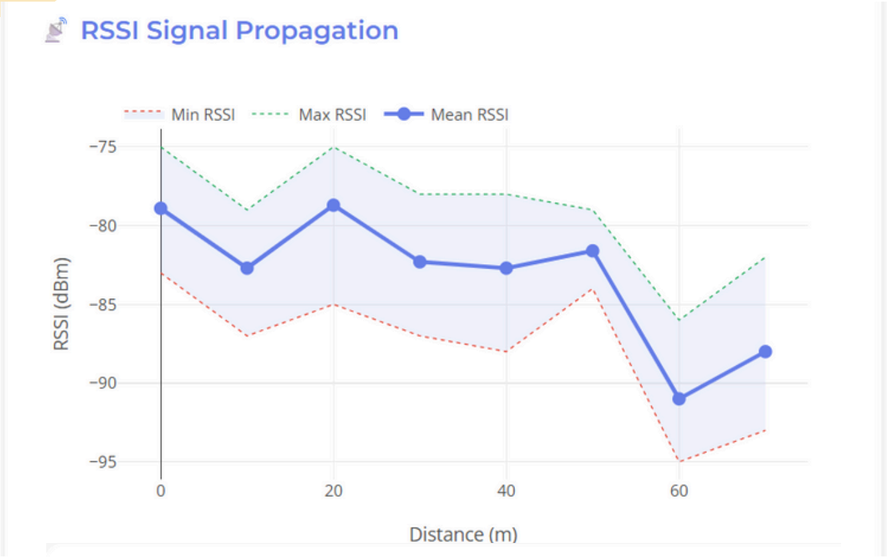
Outcome:

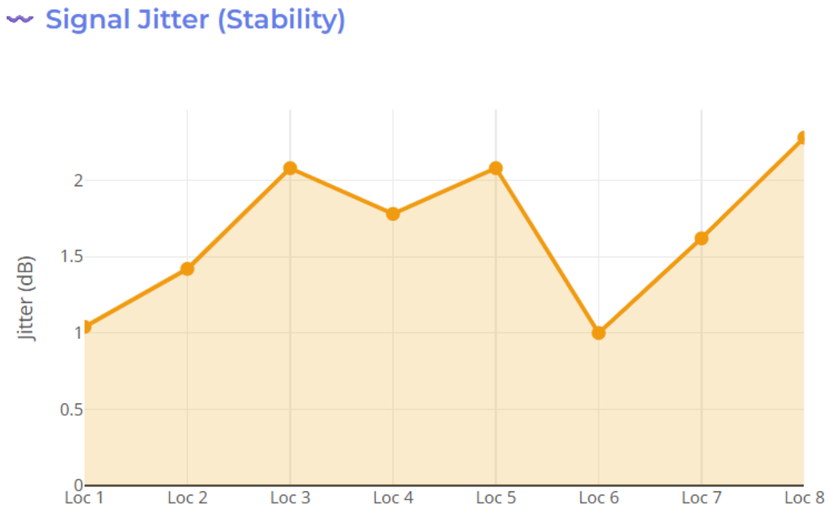
- We tested 12 points along the basketball street, but only 8 locations connected to the FSK network; the remaining points had no link due to severe obstructions.
- The main reasons for non-connectivity were blocked line-of-sight (LoS) caused by the Nilgiri building on one side and PEC building on the other.
- All connected points showed stable data collection, forming the basis of our analysis.
- These results highlight how even small structural changes along the street can significantly alter link quality in Wi-SUN FSK deployments.

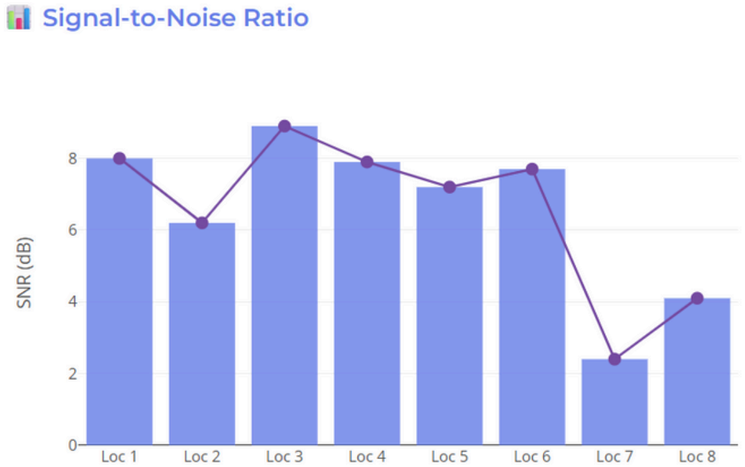


- Locations 7, 8 are near Bakul
- Rest Locations are near PEC ground

FSK - Basketball Street







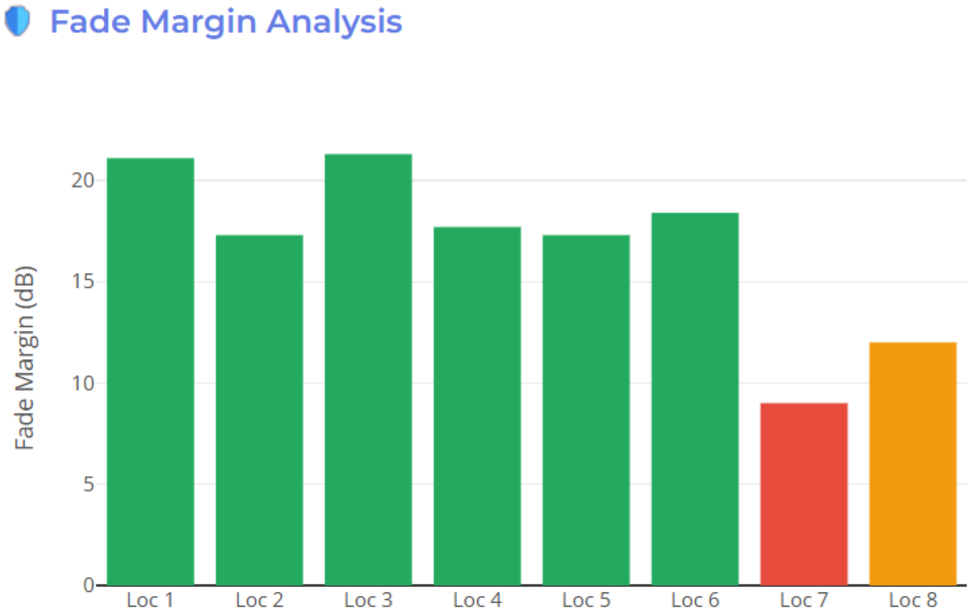
Signal Strength & RF Quality Analysis:

- RSSI across connected points decreased gradually with distance and obstructions, with a noticeable drop near the LoS-blocked region between Nilgiri and PEC.
- SNR (Signal-to-Noise Ratio): Indicates how clean the received signal is; higher SNR means clearer communication.
- Jitter (Signal Stability): Represents short-term fluctuations in signal strength; lower jitter means more stable connectivity.
- Points close to clear LoS zones had strong RSSI and stable SNR; locations near Nilgiri/PEC obstruction showed degraded quality and fluctuating links.

FSK - Basketball Street

Link Reliability & Environmental Impact:

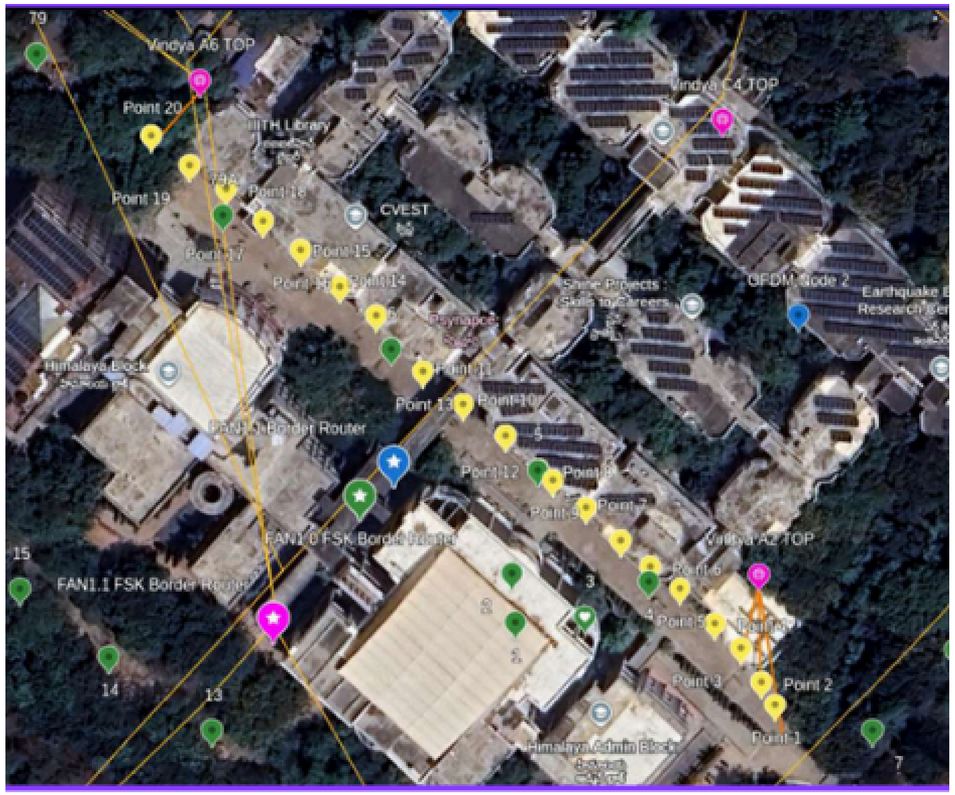
- Fade Margin (Link Reliability): Represents how much “buffer” the signal has before the link becomes unreliable; higher margin means a safer, more resilient connection.
- Despite some obstructions from Nilgiri and PEC, the basketball street performed significantly better than Research Street because most test points open into a large, unobstructed ground, allowing signals to spread more freely.
- The four locations that did not connect correspond directly to areas where Nilgiri and PEC buildings fully blocked the path, creating radio shadow zones.
- The connected locations demonstrated stronger signal strength, cleaner SNR, and smoother jitter patterns compared to Research Street, confirming that environmental openness played a major role in improved link quality.
- Wi-SUN FSK maintains reliable connectivity up to ~70m in line-of-sight conditions. Beyond this, coverage probability drops significantly.



OFDM - Research Street

Deployment Summary and Connectivity Outcome:

- We tested 17 points along Research Street, and OFDM achieved connectivity at almost all locations thanks to excellent line-of-sight (LoS) alignment with the border routers.
- The key advantage here was the placement of the OFDM border routers on rooftops with clear visibility down the street, allowing the signal to propagate with minimal obstruction.
- Environmental factors (temperature & humidity) stayed in a normal range and did not affect signal strength.

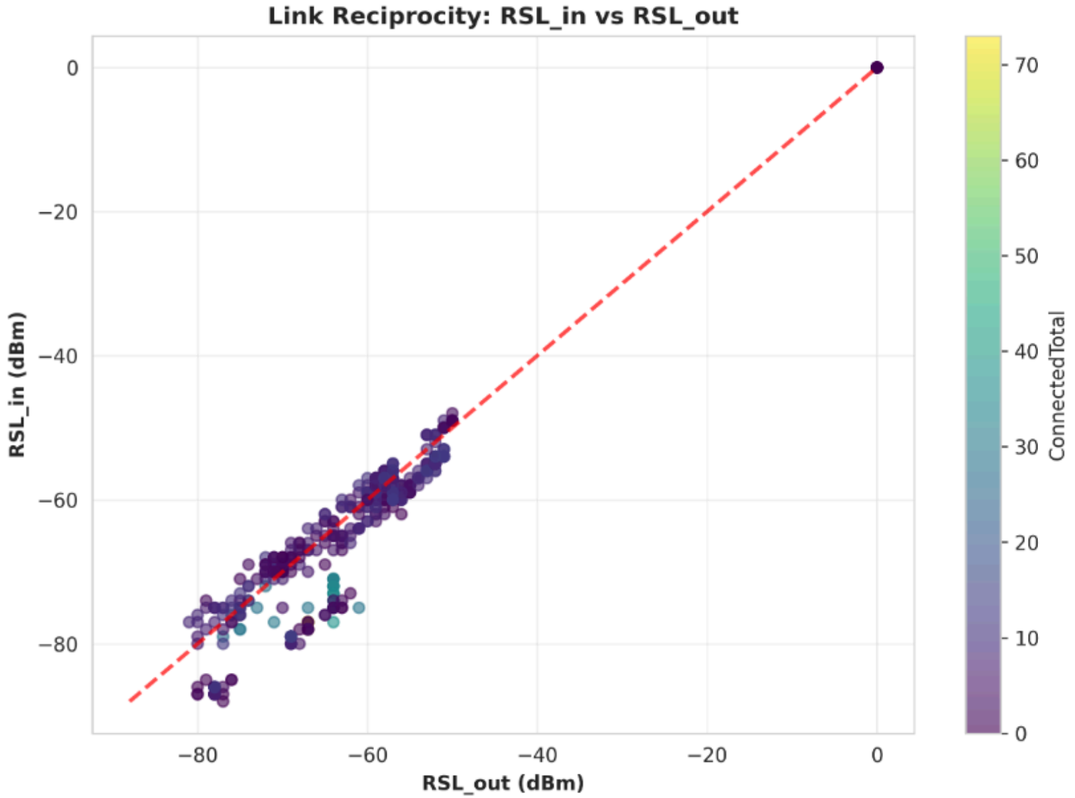


Blue Starred Node is the Border Router

OFDM - Research Street

Signal Strength & RF Quality Analysis:

- RSSI levels remained strong across most test points because the path between nodes and routers was mostly unobstructed.
- RSL_in and RSL_out were almost identical, confirming excellent link reciprocity because both directions experienced the same clear path.



OFDM vs. FSK - Research Street

- Connectivity

OFDM: Achieved full 20/20 coverage — consistent connectivity even in the middle of the street.
FSK: Only 4/20 points connected, mostly at the edges; complete failure in the central section.

- Why This Happens

Line-of-Sight Advantage for OFDM: Border routers on Vindhya rooftops had clear LoS along the street, enabling strong, stable OFDM links.
FSK Blocked by Environment: FSK's narrowband signal could not penetrate or diffract around concrete walls, trees, and the metal bridge in the middle stretch.

- Link Quality

OFDM: Much stronger RSL_in/RSL_out, cleaner SNR, and lower jitter → reliable downlink & uplink.
FSK: RSL_in extremely weak → node could not hear the network → join failures.

- Mesh Behaviour

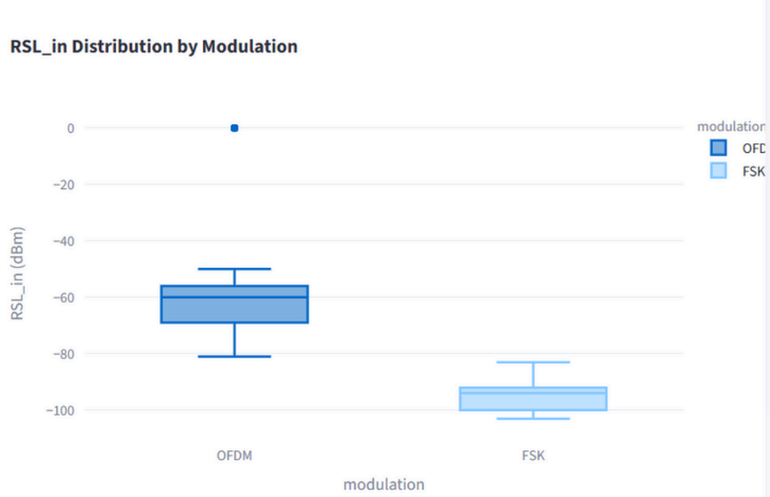
OFDM: Typically 1 hop to routers → low RPL rank → healthy mesh.
FSK: Behaved like a 3–5 hop leaf, with very high RPL rank → routing expensive & unstable.

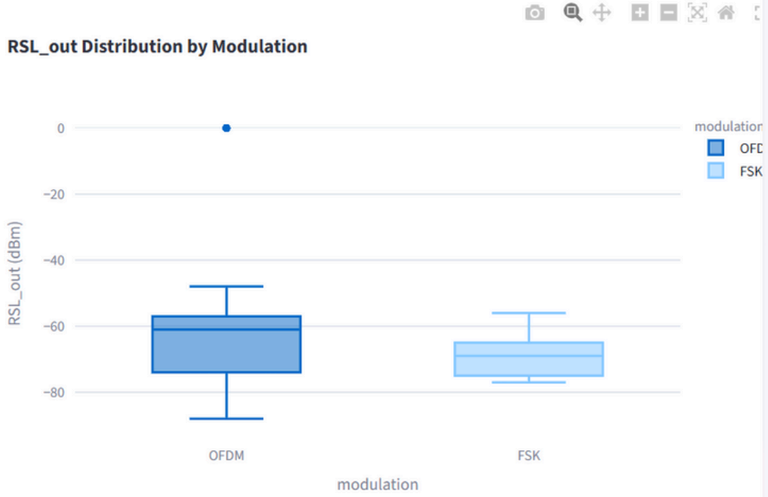
- Core Insight

OFDM survives partial obstructions and multipath due to sub-carrier diversity.
FSK requires clean LoS, and fails completely in canyon-like environments.

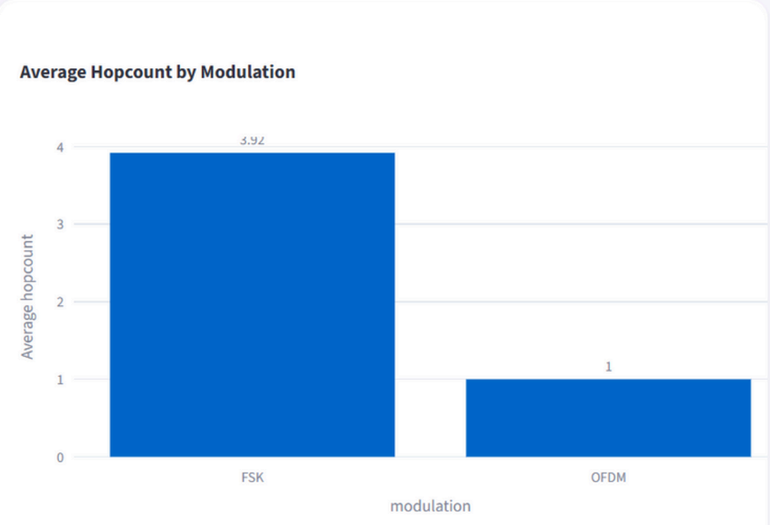
OFDM vs. FSK - Research Street

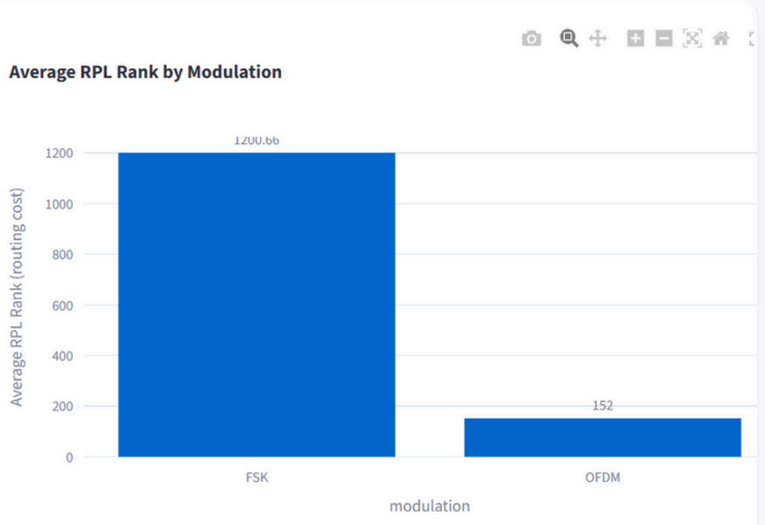
2. Link Quality – RSL_in and RSL_out





3. Mesh Behaviour – Hopcount and RPL Rank

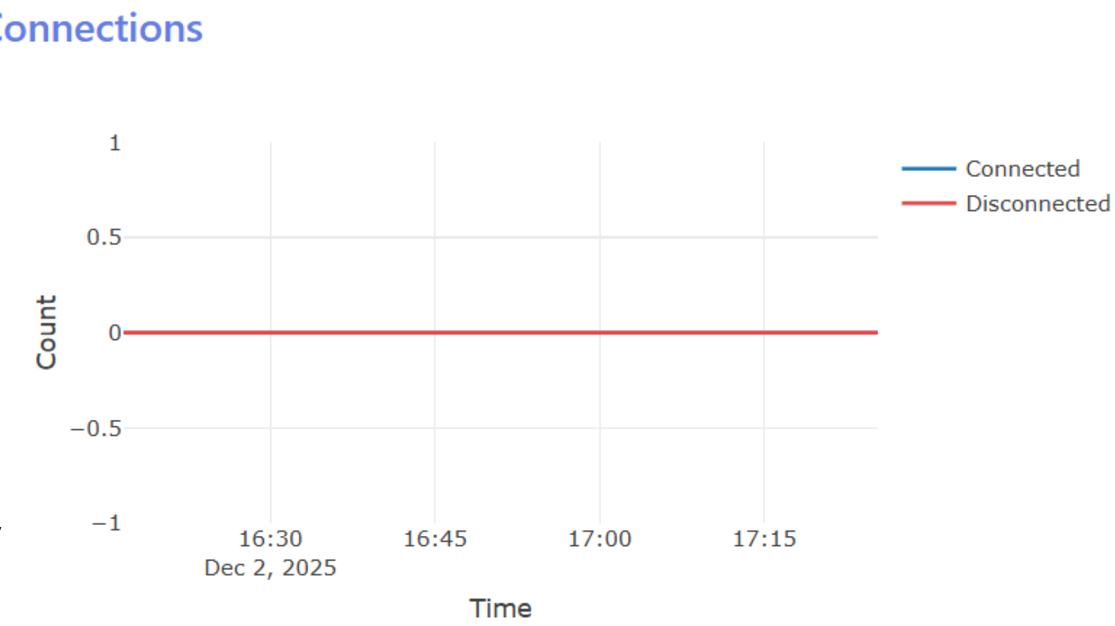


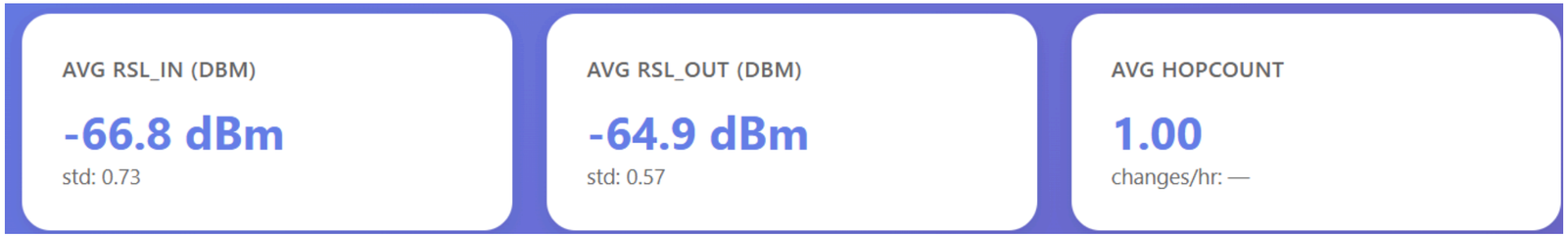


OFDM - Static Deployment

Overall Performance & Reliability:

- The OFDM static node remained highly stable, with strong RSL_in/RSL_out and no mesh fluctuations. The node was deployed on Vindhya A2 terrace, hence no blocked line of sight.
- Hopcount = 1 throughout the test window → node maintained a direct link to the border router.
- Uptime remained high, and PHY health was consistently strong.
- Anomalies correspond to periods where all telemetry values dropped to zero, indicating temporary disconnection, not sensor faults.

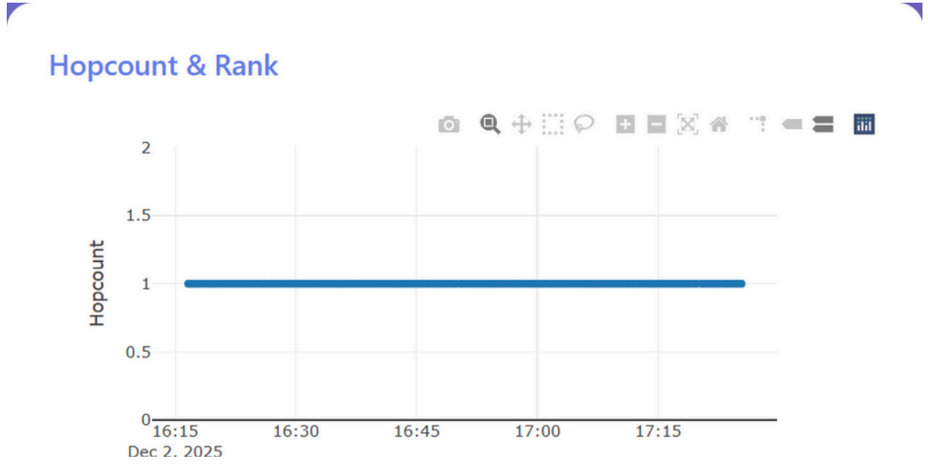


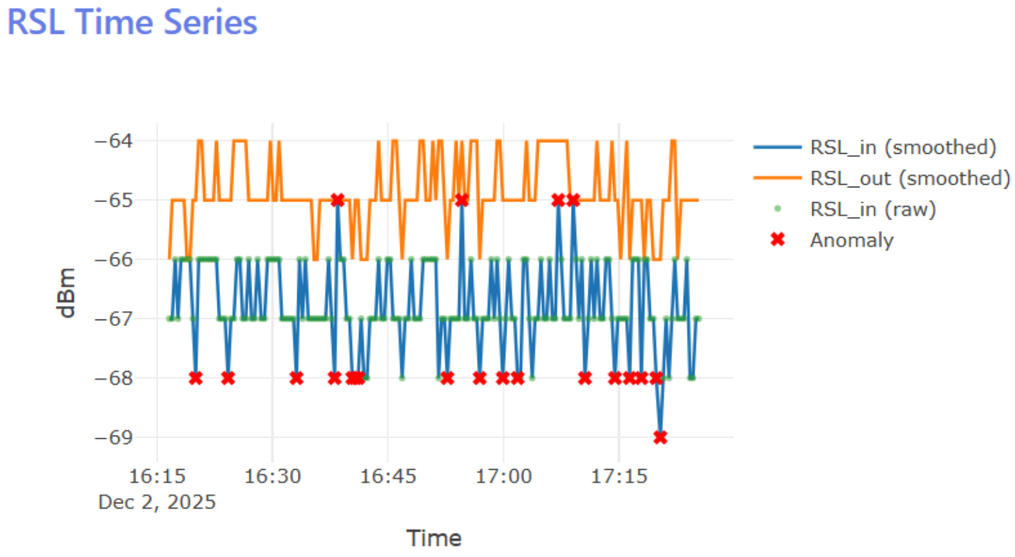


OFDM - Static Deployment

Signal Behaviour Over Time:

- RSL_in and RSL_out stayed within a tight range with minimal jitter, showing highly stable RF conditions.
- The few anomaly points (marked as zeros/red crosses) align with brief link losses, likely caused by temporary RF interference or physical obstruction.
- Hopcount remained constant at 1, confirming strong LoS and stable routing cost.
- No signs of multipath fading or link degradation were observed during the monitored period.

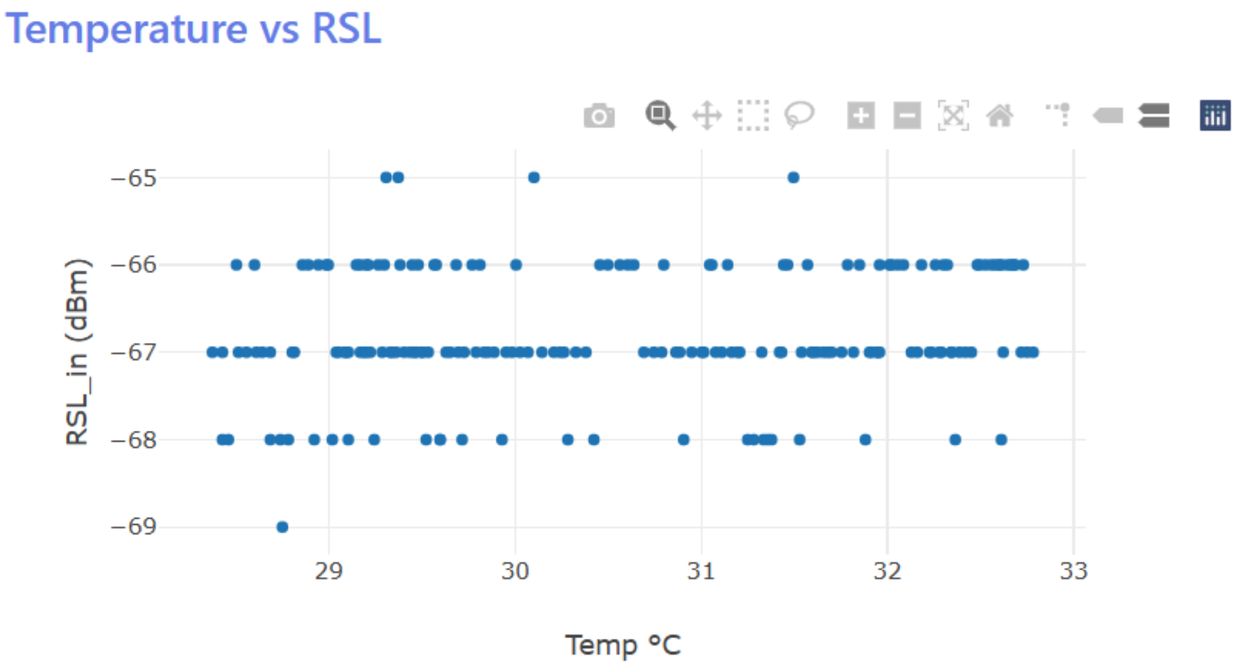




OFDM - Static Deployment

Environmental Effects & Correlation Analysis

- Temperature and humidity varied naturally across the day but had no measurable impact on link quality or connectivity.
- Scatter plots show no correlation between RSL and temperature/humidity.
- Correlation matrix confirms that connecting behaviour is independent of environmental fluctuations.
- This indicates that OFDM links are robust against typical weather variations at the deployment site.



THANK YOU