Funded by the U.S. National Science Foundation, Grant #2127548

Ship-to-Shore Network Performance Monitoring Using perfSONAR: The R/V Sikuliaq Case Study

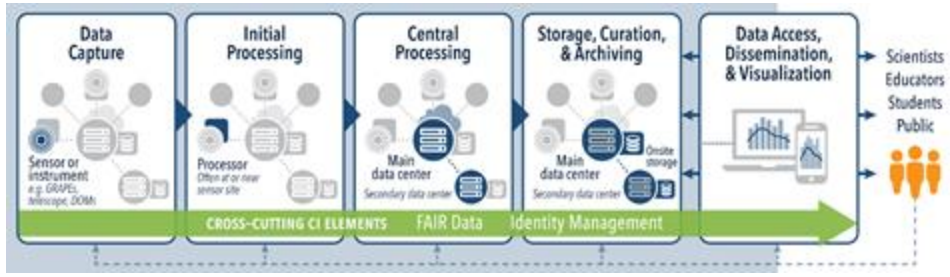
A Collaboration between NSF CI Compass, FABRIC, and the Academic Research Fleet (ARF)

Komal Thareja and Paul Ruth – FABRIC
Anirban Mandal and Ewa Deelman – CI Compass
Julian Race – R/V Sikuliaq, ARF
Christopher Romsos – RCRV, ARF

Funded by the U.S. National Science Foundation, Grant #2127548

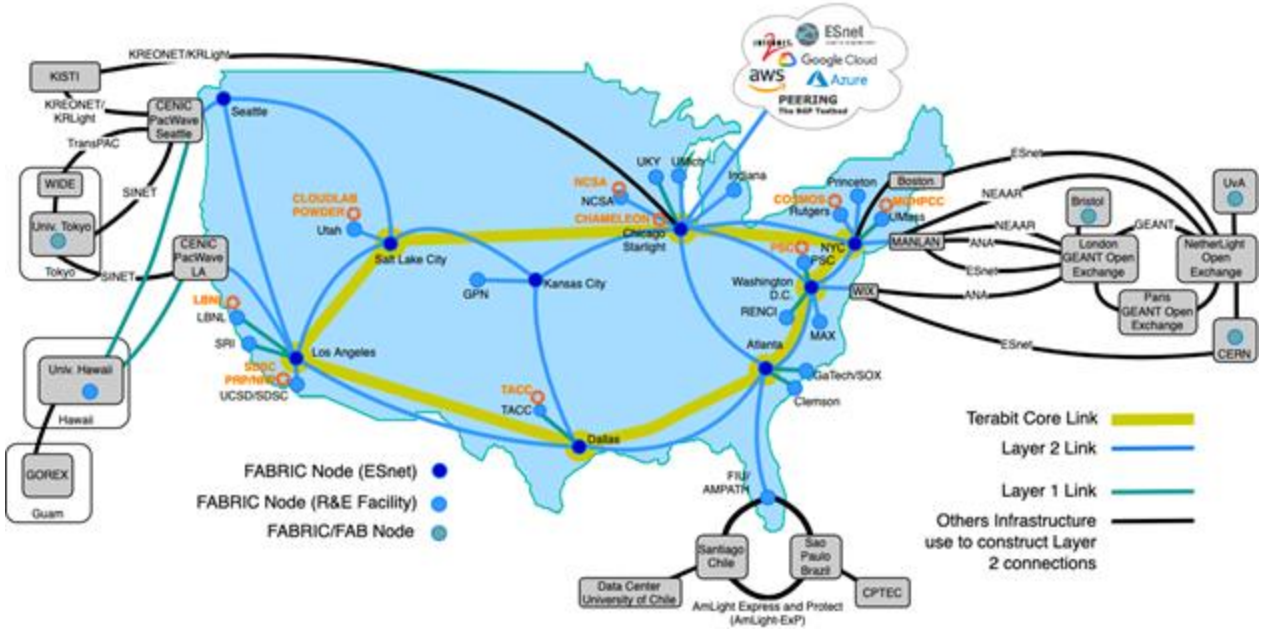
CI Compass provides expertise and active support to cyberinfrastructure practitioners at NSF Major and Midscale Facilities in order to accelerate the data lifecycle



USC Viterbi School of Engineering Information Sciences Institute
THE UNIVERSITY OF UTAH
renci
INDIANA UNIVERSITY
TEXAS TECH UNIVERSITY

CICompass https://ci-compass.org



FABRIC, an NSF Mid-scale network testbed is an International infrastructure that enables cutting-edge experimentation and research at-scale in areas of networking, cybersecurity, distributed computing, storage, machine learning, and science applications.

FABRIC https://fabric-testbed.net/

renci
ESnet ENERGY SCIENCES NETWORK
CLEMSON UNIVERSITY

Funded by the U.S. National Science Foundation, Grant #2127548

Cyber Expedition on R/V Sikuliaq with CI Compass participation (Apr'25)

Goals of the Cruise

- To provide hands-on training in the critical cyberinfrastructure systems and cybersecurity tools that support modern oceanographic research
- To provide early-career researchers with hands-on exposure to CI systems and interdisciplinary team workflows
- To **explore shipboard networking**, data acquisition and management, and remote collaboration tools in action

R/V *Sikuliaq* at its home port at SMC, Seward, Alaska
Photo by Mahedi Hasan, Texas Tech University, CI Compass

Funded by the U.S. National Science Foundation, Grant #2127548

Network Performance Measurement: Why this matters?

- Modern oceanographic missions depend on **real-time, high-throughput data transmission** - from shipboard sensors to shore-based research facilities; and to support science party on-board

- Temporary network disruptions may impact real-time data transmission, but all data remains securely stored onboard, ensuring no loss.

- Researchers and operators need better visibility into various aspects of network performance, e.g. **latency, jitter, packet loss, throughput**, and how these metrics evolve **along cruise routes**.

RV Sikuliaq
by Mahedi Hasan, Texas Tech University

Funded by the U.S. National Science Foundation, Grant #2127548

Network Performance Measurement for Research Vessels

Our Solution Approach

- Integrate perfSONAR, a trusted performance monitoring toolkit.
- Provide a **unified, annotated view of network performance** in relation to ship's location, route, and environmental conditions.
- Possible integration with CORIOLIX, a maritime context framework for weather, location, and vessel telemetry.

Impact

- Improves **operational decision-making** for scientists and engineers in the field.
- Enables **proactive troubleshooting**, enhancing data quality and reliability.
- Lays groundwork for **automated, adaptive resource management** in distributed marine research infrastructure.

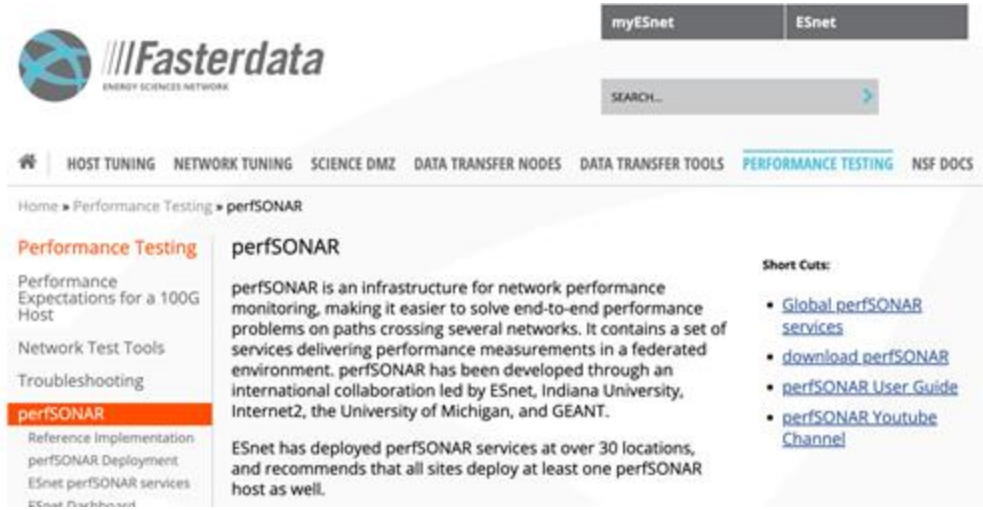
Funded by the U.S. National Science Foundation, Grant #2127548

What is perfSonar?

- perfSONAR (Performance focused Service Oriented Network monitoring ARchitecture) is a **modular, open-source toolkit** designed for **active and passive network measurements**.
- Key Features:
  - Measures **latency, packet loss, throughput, jitter**, and **path changes** across multi-domain networks.
  - Supports **automated, scheduled tests** and **on-demand diagnostics**.
  - Offers **visualization dashboards** and **APIs for integration** with other systems.
- **Trusted & Scalable**
  - Widely adopted across **global R&E networks** (e.g., Internet2, ESnet, GÉANT).
  - Supports **end-to-end performance monitoring** from campus labs to global facilities.

perf20NAR years

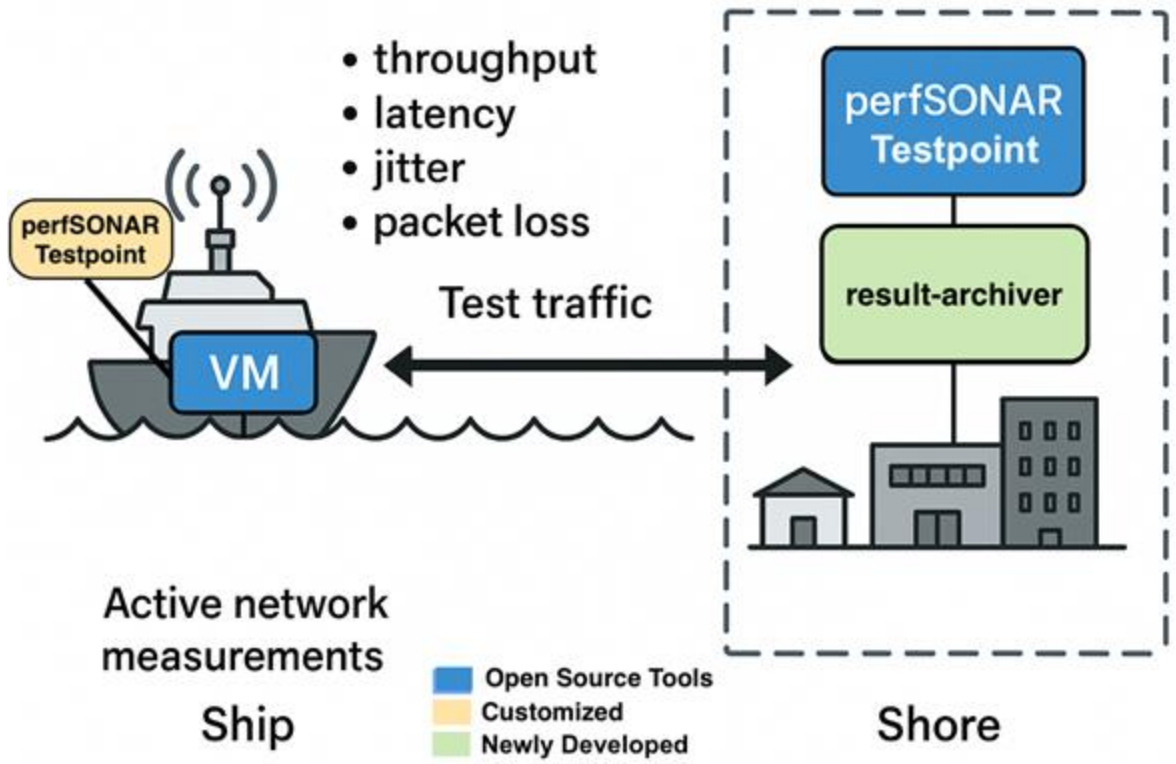
Celebrating 20 Years of perfSONAR



Funded by the U.S. National Science Foundation, Grant #2127548

System Architecture & Deployment

- **Shore-side component:**
  - One or more Virtual Machines (VMs) running the ***perfSONAR Testpoint*** as test endpoints. Can be geographically distributed.
  - Optionally includes a **Result Archiver** service:
    - Captures, stores and **displays** measurement results
    - Can be co-located on the perfSONAR VM or deployed on a **separate VM**
- **Ship-side component:** VM configured to initiate periodic active network measurements
- **Supports adaptive monitoring** of ship-to-shore connectivity during research missions



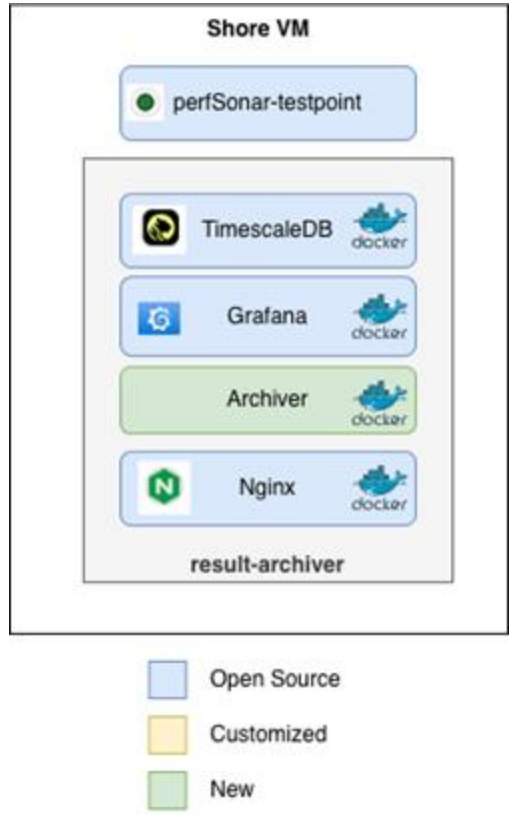
Measure network performance from the perspective of a scientist on-board

Funded by the U.S. National Science Foundation, Grant #2127548

Core Components and Enhancements – Shore side

perfSONAR Testpoint (on Shore-based VM) – Off the shelf

- Provides a powerful set of network diagnostic tools (e.g., ping, iperf3, traceroute, owamp, etc.).
- Includes pscheduler, which is responsible for **scheduling and executing network tests**.
- Exposes REST APIs for test initiation and data access, making it extensible for custom workflows.



result-archiver – Updated (pscheduler-result-archiver)

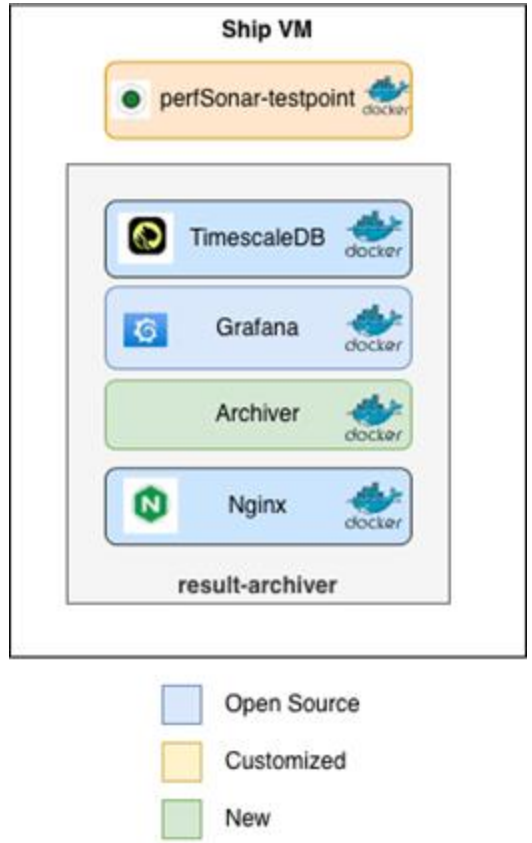
- **REST-based Result Archiver Service:** Ingests, stores, and visualizes pScheduler test results (latency, throughput, RTT, MTU, trace) with idempotent inserts into a TimescaleDB backend, supporting archival and retrieval via documented OpenAPI endpoints.
- **Integrated Visualization Stack**: Includes Grafana dashboards for interactive analysis and NGINX for secure TLS termination and routing.
- The archived data supports **historical analysis**, performance visualization, and correlation with external data like ship location or weather conditions.
- Though it's co-located with perfSONAR on the same VM in our deployment, it can run on a separate host if needed for scalability or separation of concerns.

Funded by the U.S. National Science Foundation, Grant #2127548

Core Components and Enhancements – Ship side

perfSONAR Testpoint – Customized (perfsonar-extensions)

- Built on top of the official perfsonar-testpoint Docker image.
- Extended to include **custom scripts** for running scheduled tests for latency, RTT, throughput, traceroute, etc.
- After each test, the results are **automatically pushed to the local and/or remote pscheduler-archiver** using a REST API, for post-cruise analysis.



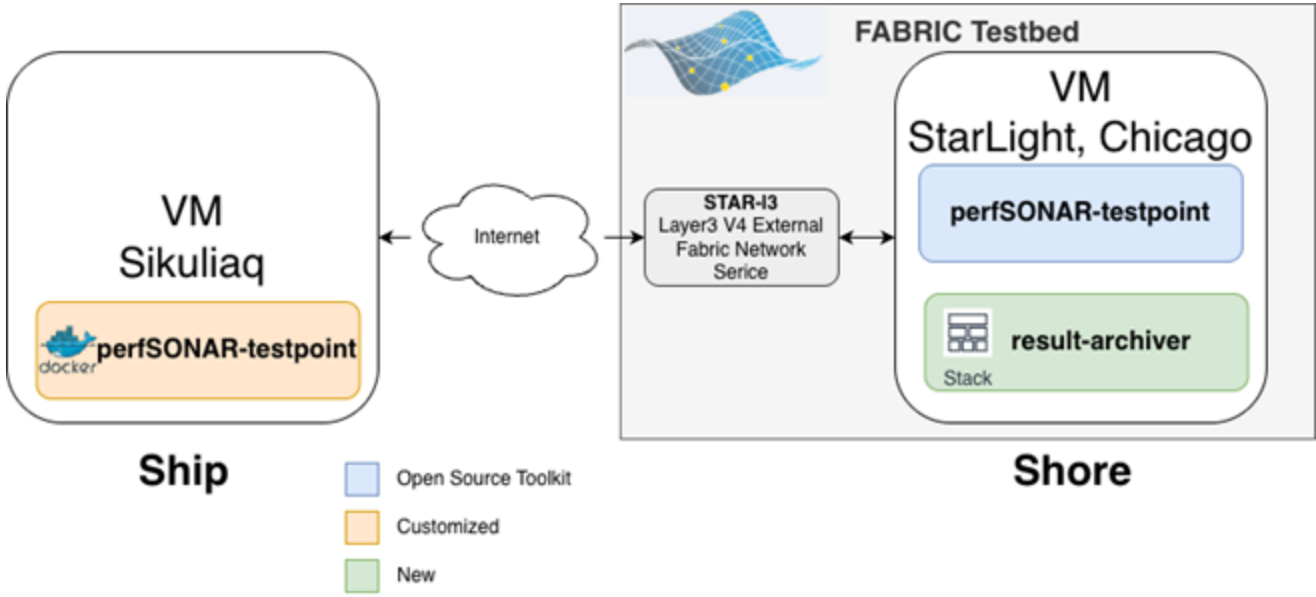
result-archiver – Deployed on Ship

- Runs a custom lightweight pScheduler Result Archiver stack.
- Integrated Visualization Stack: Includes Grafana dashboards for interactive analysis and NGINX for secure TLS termination and routing.

Funded by the U.S. National Science Foundation, Grant #2127548

Deployment Context: Sikuliaq Cruise Setup

- During the research cruise, we deployed the testpoint container on an **Ubuntu VM onboard the R/V Sikuliaq**.
- The **shore-side VM** at STARLight Chicago on **FABRIC testbed** acted as the **perfSONAR testpoint** and also hosted the **result archiver**.
- This setup allowed for **continuous ship-to-shore tests** every few hours, even in dynamic network environments.



Data Flow

- perfSONAR testpoint container on Ship VM
  - runs scheduled network tests → sends data to Shore VM.
- Archived data can then be **analyzed**, **visualized**, or **correlated** with ship telemetry via CORIOLIX.

Funded by the U.S. National Science Foundation, Grant #2127548

Deployment Context: Sikuliaq Cruise Installation

Step 1: Provision or Connect to the Ubuntu VM

- Ensure you are connected to the Ubuntu VM onboard the ship.

Step 2: Install Docker based perfSONAR Setup

```
./setup_ship.sh --token AUTH_TOKEN \
  --hosts "23.134.232.50@shore-STAR" \
  --urls "https://localhost:8443/ps,https://23.134.232.50:8443/ps"
```

Note: AUTH_TOKEN can be retrieved from the Shore-side VM.

Funded by the U.S. National Science Foundation, Grant #2127548

Deployment Context: Sikuliaq Cruise Installation

Step 3: Access Grafana

Once the setup completes, you can access Grafana at:

`https://<REPLACE_WITH_SHIP_IP>:8443/`

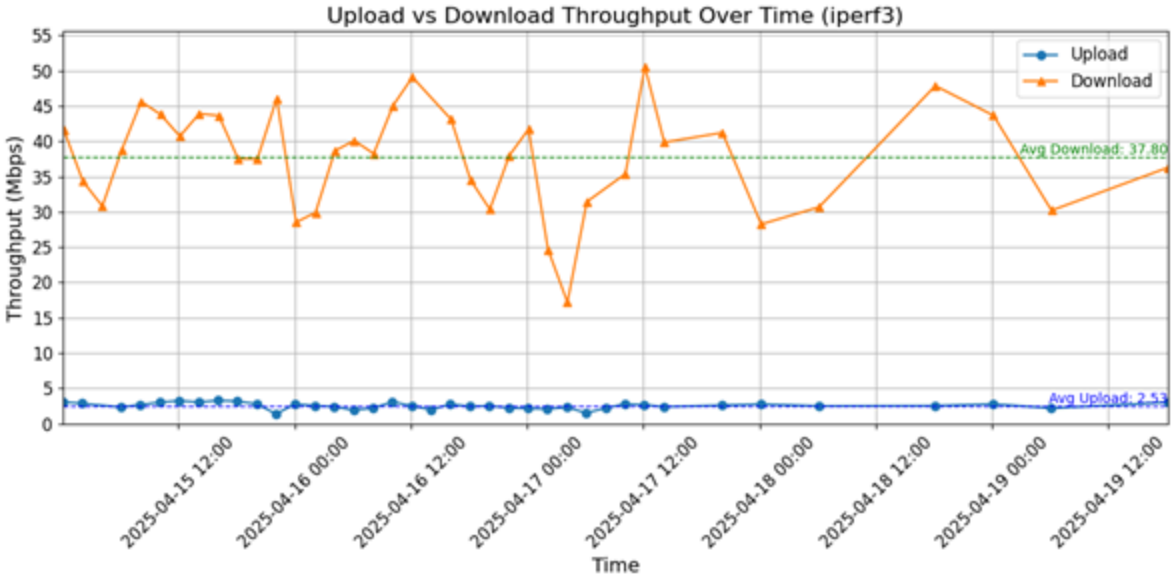
Step 4: Initial Grafana Configuration

- Log in using the default credentials **admin/admin** (you'll be prompted to change the password).
- In the left pane, go to **Data Sources** → **perfsonar-postgres**.
- Under **Authentication**, set the password to change_me, then click **Save & Test**.
  - You should see: *Database Connection OK*.
  - If you see a message saying the password is being updated by another entity, refresh and click **Save & Test** again.
- To view performance data, navigate to **Dashboards** → **pScheduler** → **pscheduler-result-archiver (Pairs)**.

Funded by the U.S. National Science Foundation, Grant #2127548

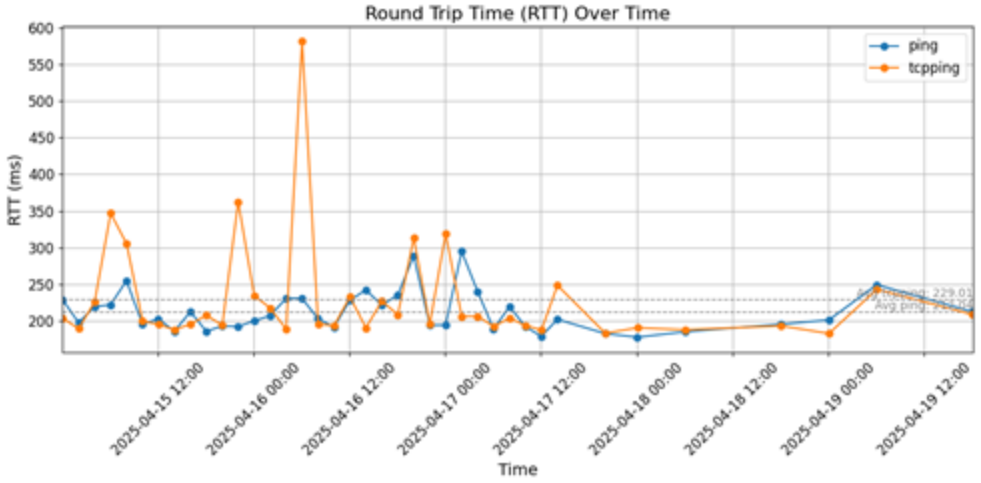
Network Insights: Apr'25 cruise

Throughput Trends: April 14–20, 2025



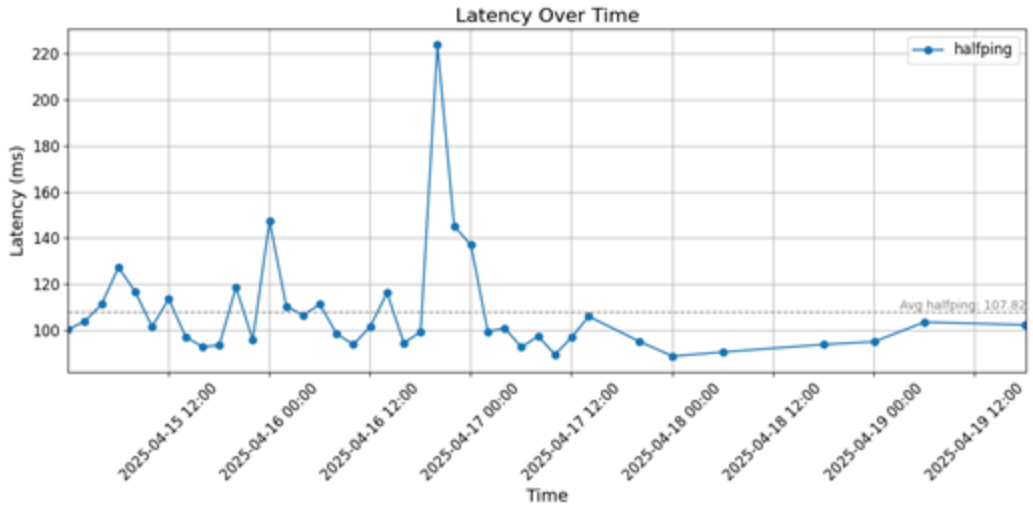
Avg Upload bandwidth (iperf3): 2.53 Mbps
Avg Download bandwidth (iperf3): 37.80 Mbps

Round Trip Time Trends: April 14–20, 2025



Avg RTT (tcpping): 229.01 ms
Avg RTT (ping): 212.04 ms

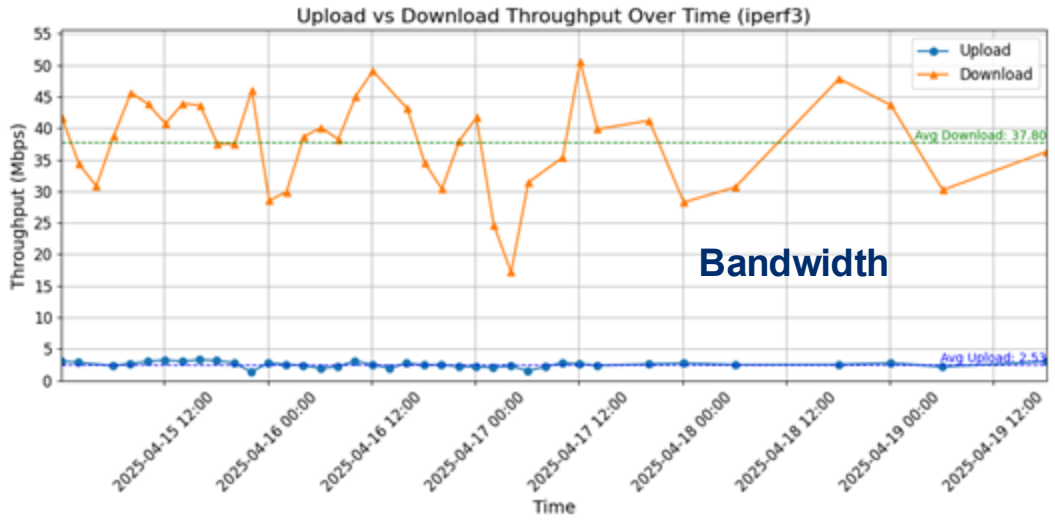
Latency Trends: April 14–20, 2025

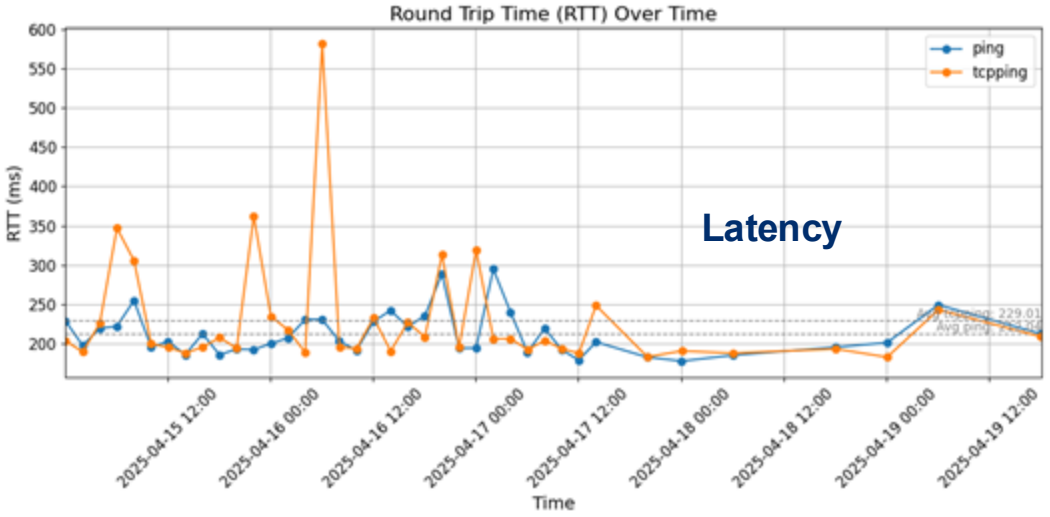


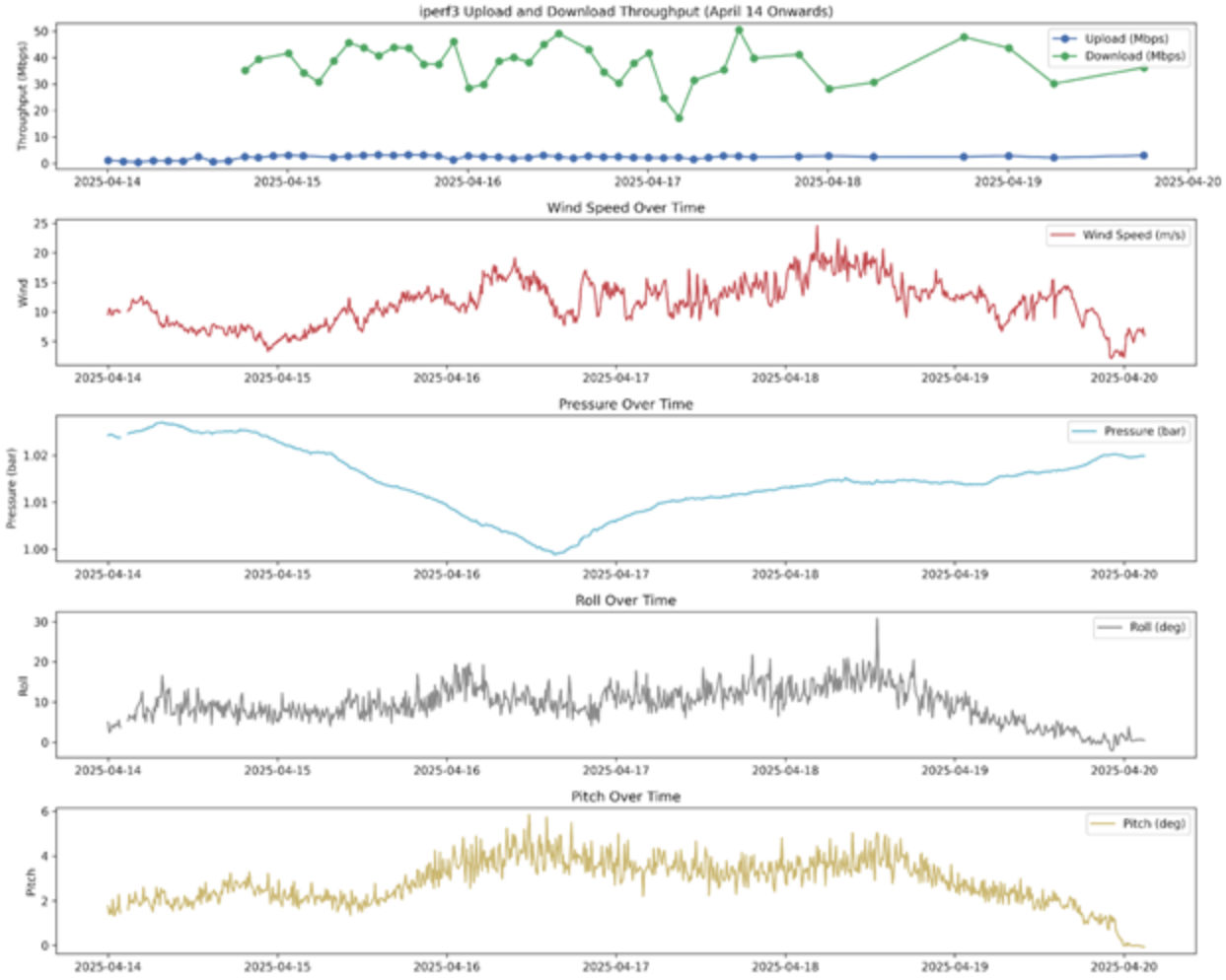
Avg one-way latency (halfping): 107.82ms

Funded by the U.S. National Science Foundation, Grant #2127548

Ship-to-Shore Network Performance Insights: April'25 Cruise from Honolulu, HI to Seward, Alaska







Correlation with environmental parameters (e.g. wind speed, pitch, roll) from CORIOLIX

Funded by the U.S. National Science Foundation, Grant #2127548

Journey - Throughput

Pre-Cruise Baseline Testing – Upload Only – 03/30/2025 – 04/10/2025

- Conducted while docked in Hawaii. Upload bandwidth averaged around 50 Mbps, though fluctuations were observed.

Cruise Start – Upload Only – 04/10/2025

- Testing continued after departure. Bandwidth dropped below 1 Mbps with ongoing instability.
- Determined that short test durations and measuring only upload throughput were insufficient for reliable analysis.

Enhanced Testing Enabled – 04/14/2025

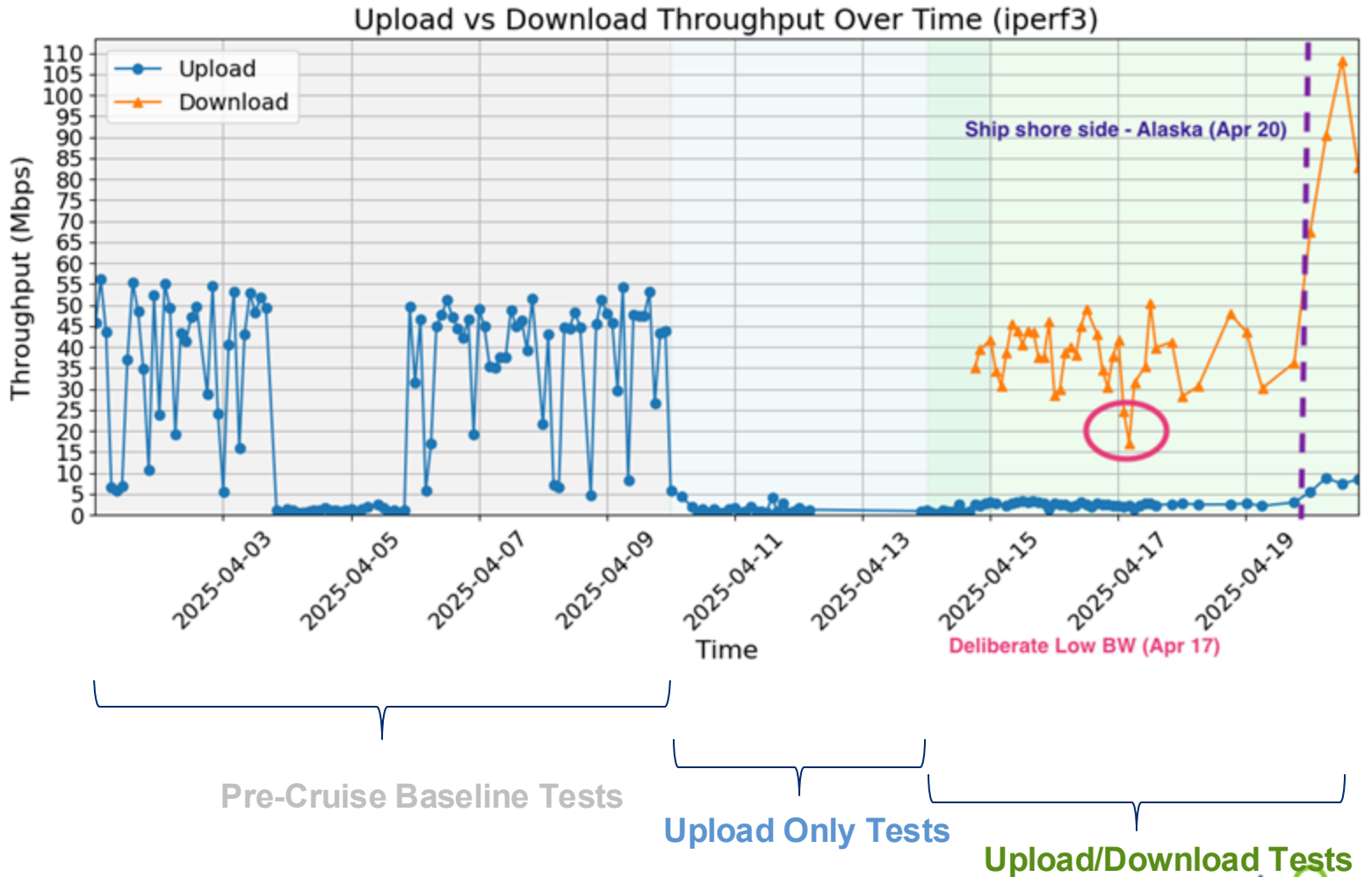
- Increased test duration to 180 seconds and introduced 4 parallel streams to better capture throughput behavior.
- Speed tests were also activated for additional metrics.

Controlled Low Bandwidth Tests – 04/17/2025

Intentionally introduced low bandwidth conditions to validate test sensitivity and system response.

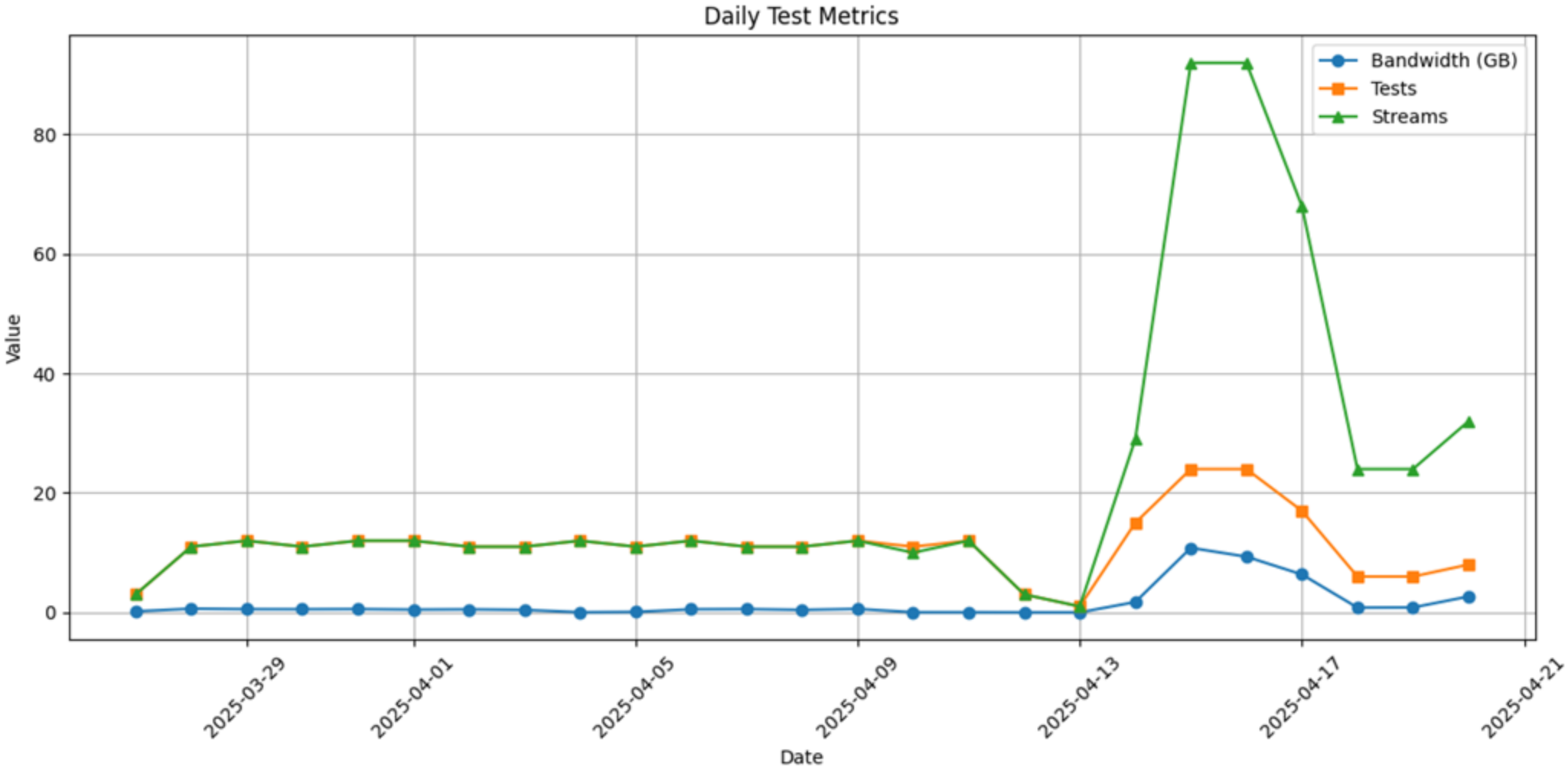
Enhanced Testing Resumed – 04/17/2025

- Resumed with a reduced test duration of 60 seconds.
- Speed tests were disabled in this phase to focus on core throughput measurements.



Funded by the U.S. National Science Foundation, Grant #2127548

Overall Bandwidth Usage During Cruise



Total Bandwidth Used (GB): 38.50 GB

Funded by the U.S. National Science Foundation, Grant #2127548

Post-cruise Feedback and Enhancements

- Plot relevant CORIOLIX metrics of interest alongside network data
  - **Identified a few metrics and experimenting correlations** - Need more deployments and data
- Network monitoring tool enhancements
  - **Ability to configure the frequency of data collection and whether monitoring data should be sent to the shore side**
  - **Enable data collection on the ship and visualization the data on the ship with a dashboard**
  - **Test the new code at next opportunity on Sikuliaq (October cruise)**
- *Explore how to deploy the network monitoring tool on a Raspberry Pi - smaller vessels can just plug in and get the monitoring going - Prototype done and needs testing*
- Test deploying an additional perfSONAR testpoint outside the firewall to measure provider bandwidth before traffic is shaped by the firewall
- Coordinate with HiSeasNet who will be interested in independently verified network data collected on the ship

Funded by the U.S. National Science Foundation, Grant #2127548

Grafana Dashboard Snapshot (from current deployment on Sikuliaq)

Funded by the U.S. National Science Foundation, Grant #2127548

Raspberry Pi 5 Field Kit (16 Units Ready)

- **Goal**: plug-and-play monitoring for small vessels and rapid deployments across the fleet.
- Sixteen Raspberry Pi 5 units are pre-configured with the Dockerized **perfSONAR Testpoint** and **pscheduler-result-archiver** stack — ready for field use.
- On a research vessel, simply plug into power and connect the Pi to the ship's network (Internet/DHCP).
- **Minimal per-ship config (≈ 3 minutes)**: Connect the Raspberry Pi to the internet by setting up the network configuration and it is ready to use.
- **Matches the ship-side VM setup**: scheduled latency / RTT / throughput / trace tests with automatic archiving to shore.

Funded by the U.S. National Science Foundation, Grant #2127548

Shipboard Activation (per Pi)

Step 1 — Connect the Pi to the internet

- **Purpose:** Pis are configured to run network tests every 4 hours and archive results locally (on-device) and on the shore-side server.
- **Setup:**
  - Connect a **monitor**, **keyboard**, and **power supply** to the Raspberry Pi.
  - Power on the Pi and **log in as `ubuntu`**.
  - **Check network interfaces:**

    ```
    ip link show
    ```

  - Typical interface names: `wlan0` → Wi-Fi `eth0` → Ethernet
    ***Note**: your Wi-Fi interface name (e.g., wlan0)*

  - **Edit Netplan configuration:**

    ```
    sudo vi /etc/netplan/50-cloud-init.yaml
    ```

Funded by the U.S. National Science Foundation, Grant #2127548

Shipboard Activation (per Pi)

- Example configuration shown on the right.
- Configuring WiFi:
  - Update the **access-point name** and **password**
- Apply the configuration

```
sudo netplan apply
```

- Verify connection

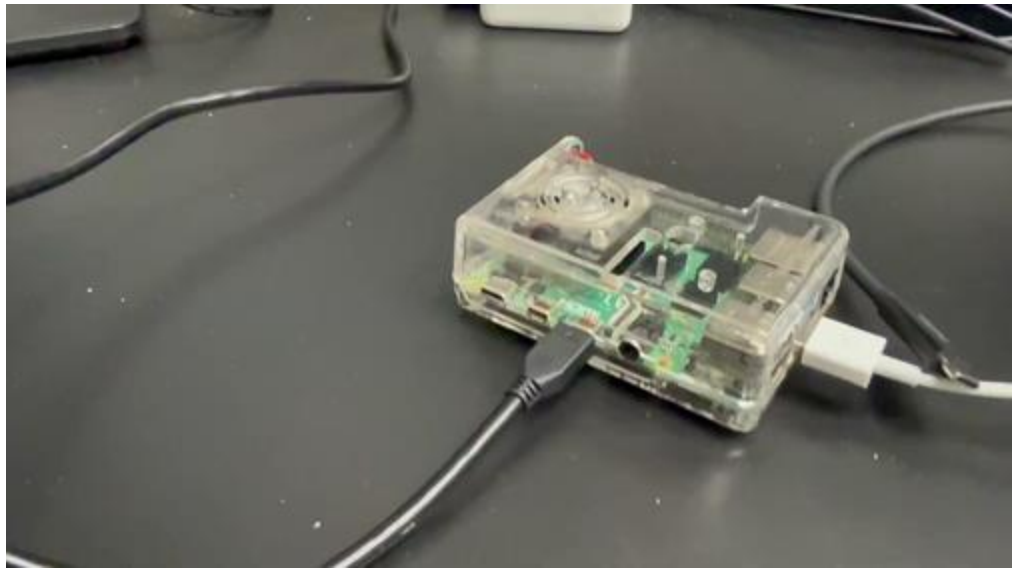
```
ip addr show wlan0

sudo iwconfig wlan0

ping -c 4 8.8.8.8
```

NOTE: users can choose to configure ethernet interface if not using wireless network.

```
network:
  version: 2
  wifis:
    wlan0:
      optional: true
      dhcp4: true
      access-points:
        "TP-Link_0190":
          auth:
            key-management: "psk"
            password: "12345609909888888888888888888888888888888888888888"
```



Instructions for connecting Raspberry Pi to the internet

Funded by the U.S. National Science Foundation, Grant #2127548

Next Steps

- Test network monitoring **tool enhancements** during future cruises on Sikuliaq
- Deploy the **Raspberry Pis on Sikuliaq and other vessels** and test the tool – looking for volunteers
- Test deploying an **additional perfSONAR testpoint outside the firewall** to measure provider bandwidth before traffic is shaped by the firewall - on a future Sikuliaq cruise
- Engage with **HiSeasNet** for providing independently verified network performance data collected on the research vessels
- Experiment with CORIOLIX metrics with wide data collection across the fleet
- Upcoming publication on the work:
  - K. Thareja, A. Mandal, J. Race, P. Ruth, C. Romsos, and E. Deelman, *"Ship-to-Shore Network Monitoring: The Research Vessel Sikuliaq Experience"*, IEEE Internet Computing, 2025. [In publication, expected Nov 2025]

Funded by the U.S. National Science Foundation, Grant #2127548

Thank you !!

NSF CI Compass project (NSF Award 2127548)
University of Alaska Fairbanks/Sikuliaq Oceanographic Technical Services of the U.S Academic Research Fleet (NSF Award 2421881)
NSF FABRIC Testbed (NSF Award 2330891)

Contact: anirban@renci.org or deelman@isi.edu

Funded by the U.S. National Science Foundation, Grant #2127548

Shipboard Activation (per Pi)

Step 2 — Trigger a test manually (Optional)

- Run the following command as ubuntu user:

```
docker exec -ti perfsonar-testpoint /usr/bin/python3 /usr/src/app/periodic.py --hosts
23.134.232.50@shore-STAR --output-dir /data --archiver-urls
https://localhost:8443/ps,https://23.134.232.50:8443/ps --auth-token
f196e527459fb54937672345ba1ea4240d7d8209b0dee210752572e4ada93dc1 --reverse
```

- Verify the measurement is captured on the Local Grafana Instance or FABRIC Grafana Instance

Step 3 — Update configuration to use a different Shore instance (Optional)

- Log in to the Raspberry Pi as the ubuntu user.
- Run the following commands to update the configuration and restart the containers:

```
cd perfsonar-extensions/docker
sed -i 's@HOSTS=.*@HOSTS=<new_shore_ip>@<new_shore_name>' .env
docker compose -f docker-compose-testpoint.yml up --force-recreate -d
```

Funded by the U.S. National Science Foundation, Grant #2127548

Bootstrapping Ship-Shore Measurement Infrastructure

Ship Node Setup

```
git clone https://github.com/kthare10/fabric-recipes.git
cd perfsonar/node_tools
./setup_shore.sh --token {ARCHIVER_TOKEN}
./setup_shore.sh --token {ARCHIVER_TOKEN}
```

Shore Node Setup

```
git clone https://github.com/kthare10/fabric-recipes.git
cd perfsonar/node_tools
./setup_ship.sh --token {ARCHIVER_TOKEN}  --hosts <shore_ip>@<shore_name>  --urls
https://localhost:8443/ps,https://<shore_ip>:8443/ps
./setup_ship.sh --token {ARCHIVER_TOKEN}  --hosts <shore_ip>@<shore_name>  --urls
https://localhost:8443/ps,https://<shore_ip>:8443/ps
```

ARCHIVER_TOKEN

- Can be **any string** chosen by the operator. Recommended to generate securely, for example: openssl rand -hex 32
- The **same token value must be used** on both the **ship** and **shore** nodes.

Note: setup*.sh command should be executed **twice** —

- First run: installs Docker
- Second run: launches the PerfSONAR containers