Phase 1 — Random Delay Injection and RTT Measurement

1. Introduction

In this phase, we aimed to implement a simple middlebox-like processor that injects random delays into Ethernet frames and to measure how these delays affect Round Trip Time (RTT) for ICMP ping packets. We began by forking the *middlebox* repository, then developed our own processor (main.py) using Python and the NATS messaging system (nats.io). Our processor subscribes to two topics—inpktsec and inpktinsec—where incoming Ethernet frames are received via NATS. Each frame is parsed using the scapy library, a random delay is applied, and the frame is then republished to the corresponding output topic.

2. Methodology

- **Random Delay Generation:** We used an exponential distribution to generate random delays with random.expovariate(1 / 5e-6). This approach yields a range of delay values that reflect a Poisson process, mimicking real-world latency fluctuations.
- **Integration with NATS:** Our code connects to the NATS server and continuously listens for new Ethernet frames on *inpktsec* and *inpktinsec*. After injecting the random delay, the frames are published back to *outpktsec* or *outpktinsec*.
- **RTT Logging:** To measure the network performance impact, we created a separate script (log_rtt.py) that runs ICMP pings from within a Docker container. After 20 ping attempts, the script parses the RTT values, computes their average, and appends it to a file (rtts.txt). A “stop signal” file is then created to inform the main processor to exit gracefully.
- **Plotting:** Finally, plotting.py reads the mean delays recorded in delays.txt and the average RTT values from rtts.txt. It plots them on a scatter plot with the x-axis as **Mean Random Delay (ms)** and the y-axis as **Average RTT for Ping Packets (ms)**. Annotations and dashed lines are drawn to help visualize each measurement.

3. Results

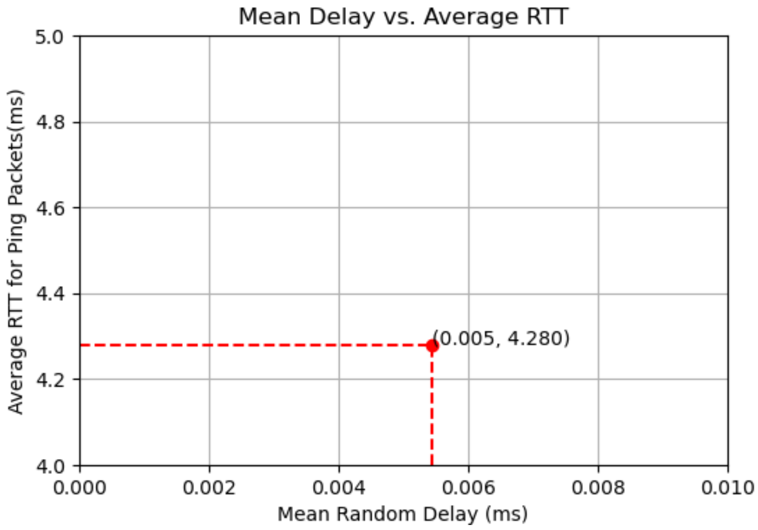
In our test scenario, we obtained a single data point where the mean random delay was approximately 0.005 ms, leading to an average ping RTT of around 4.28 ms. The figure demonstrates the relationship between the measured mean delay on the x-axis and the resulting RTT on the y-axis. As expected, even a small artificial delay can noticeably influence overall RTT.

4. Conclusion

This phase successfully demonstrates a basic middlebox function that adds random delays to Ethernet frames using NATS. We verified our approach by measuring how these injected delays affect ICMP ping RTTs. Our single data point shows that a mean delay of 0.005 ms corresponds to an RTT of roughly 4.28 ms, indicating the feasibility of extending this framework to more complex or realistic network scenarios.

5. GitHub Repository

GitHub repository link - https://github.com/rab-ai/middlebox


Mean Delay vs. Average RTT
Average RTT for Ping Packets(ms)
Mean Random Delay (ms)
(0.005, 4.280)