Implementation of a Random Delay Processor

Kerem Recep Gür
2448462

1 Implementation Details

1.1 Technologies Used

For implementing the random delay processor, I used **Python** (implementation language), **Scapy** (parsing Ethernet frames), and **random.expovariate()** (generating delays with exponential distribution).

1.2 Key Implementation Features

The most important aspects of my implementation:

- **Random Delay Generation**: Using exponential distribution

```
    delay = random.expovariate(self.delay_value)
    await asyncio.sleep(delay)

```

The key algorithm is the use of exponential distribution for delay generation. With parameter λ (specified by the `DELAY_VALUE` environment variable), the mean delay is $\frac{1}{\lambda}$ seconds.

2 Testing Methodology

To evaluate how different delay parameters affect network performance, I conducted systematic tests:

1. For each delay value, I manually updated the `DELAY_VALUE` in the docker-compose.yml file:

```
environment:
  - DELAY_VALUE=500  # Changed for each test

```

2. Then restarted the container to apply the new delay parameter:

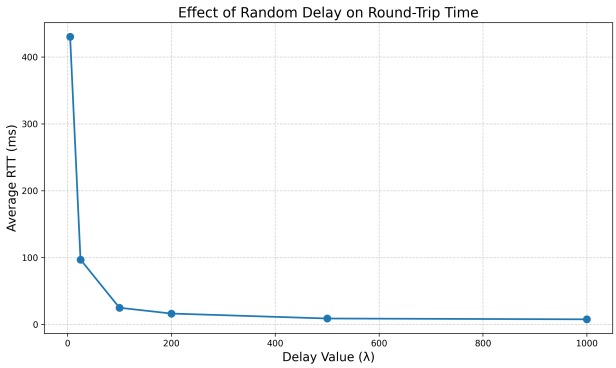
```
sudo docker-compose down
sudo docker-compose up -d

```

3. From the insecure network container, sent 50 ping packets to the secure network:

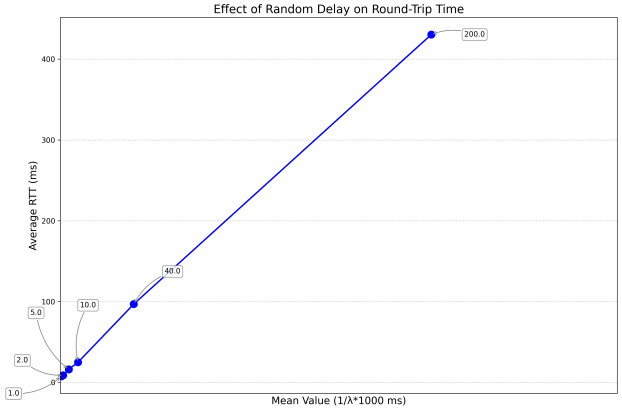
```
sudo docker exec insec ping -c 50 sec

```

3 Results

The ping tests produced RTT statistics for each delay value. These results were analyzed and plotted to show the relationship between the delay parameter (λ) and the average RTT.



(a) Relationship between delay parameter (λ) and average RTT



(b) Relationship between mean value ($\frac{1}{\lambda}$) and average RTT

Table 1: Delay Parameter and Round-Trip Time Measurements

λ	Mean Delay (ms)	RTT (ms)
5.0	200.00	430.32
25.0	40.00	96.95
100.0	10.00	24.91
200.0	5.00	16.24
500.0	2.00	8.83
1000.0	1.00	7.60

Lower values of λ produce longer delays (as the mean of an exponential distribution is $\frac{1}{\lambda}$), resulting in higher RTT measurements. Since delays are applied in both directions (to packets traveling from insec to sec and vice versa), we expect the total RTT increase to be approximately $\frac{2}{\lambda}$ seconds. The experimental results generally confirm this relationship, with some variance due to the random nature of the exponential distribution.

GitHub Repository: `https://github.com/KereMath/middlebox`