NetSim®

Accelerate Network R & D

Cyber

Simulate network attacks on power systems

A Network Simulation & Emulation Software

By

TETCOS LLP

The information contained in this document represents the current view of TETCOS LLP on the issues discussed as of the date of publication. Because TETCOS LLP must respond to changing market conditions, it should not be interpreted to be a commitment on the part of TETCOS LLP, and TETCOS LLP cannot guarantee the accuracy of any information presented after the date of publication.

This manual is for informational purposes only.

The publisher has taken care in the preparation of this document but makes no expressed or implied warranty of any kind and assumes no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information contained herein.

Contact us at

TETCOS LLP
214, 39th A Cross, 7th Main, 5th Block Jayanagar,
Bangalore - 560 041, Karnataka, INDIA.
Phone: +91 80 26630624
E-Mail: sales@tetcos.com
Visit: www.tetcos.com

Contents

1 Introduction

Electric power grids, pivotal to modern life, are recognized as critical infrastructure. Recent advancements in power systems have led to the development of cyber-physical-power-systems (CPPS). These systems merge physical power infrastructure with cyber elements for control, computing, and communication, enabling smart grid technologies through bidirectional electricity and information flows. However, this increased reliance on communication systems introduces new vulnerabilities. Notably, power grids have become prime targets for cyber-attacks, surpassing other critical infrastructures in attack frequency. Recent high-profile incidents highlight the critical need for robust cybersecurity measures. For instance:

- In 2015, the Ukrainian power grid experienced a cyber-attack that led to widespread outages, affecting over 200,000 consumers.
- The 2020 SolarWinds attack, though not directly targeting power grids, exposed vulnerabilities in supply chain security, underscoring the interconnected nature of modern cyber threats.

To address these challenges, it is crucial to test vulnerabilities in both the physical and communication aspects of CPPS. Every communication link represents a potential vector for cyber-attacks. Power system simulators can be used to model the physical components, while NetSim can be used to simulate the communication network. This co-simulation approach facilitates the investigation of CPPS vulnerabilities and the development of defense approaches in a controlled laboratory environment.

1.1 Overview

Electric power grids are the backbone of modern society, ensuring the continuous delivery of electricity essential for daily life and economic activities. With the advent of CPPS, the integration of physical processes with networked computational resources has become a reality. This integration, while offering enhanced functionality and efficiency, also introduces new vulnerabilities to cyber-attacks.

1.2 Importance of CPPS

CPPS represent a convergence of traditional power systems and advanced information technology. This integration allows for real-time monitoring and control, predictive maintenance, and improved resilience of power grids. CPPS are crucial for the smart grid evolution, enabling efficient energy management, distributed generation, and renewable energy integration.

1.3 Objectives of the Manual

This manual aims to provide comprehensive guidance on simulating and analyzing CPPS using NetSim. The manual will cover:

- Interfacing NetSim with real-time power system simulators
- Understanding and implementing various electrical and network protocols
- Simulating network attacks and assessing their impact on CPPS
- Practical examples and case studies to illustrate key concepts

2 Interfacing with Real Time Power System Simulators

2.1 Overview

NetSim can interface with several power system simulators to enhance the study and analysis of cyber-physical power systems (CPPS). This integration allows for comprehensive testing and evaluation of both the physical and cyber components of modern power grids. The supported simulators include:

- OPAL-RT
- RTDS (Real Time Digital Simulator)
- Typhoon HIL
- MATLAB
- Simulink
- SCADA

Additionally, NetSim can interface with:

- Open PMU: An open-source platform for Synchro phasor data.
- Open PDC: An open-source phasor data concentrator.

2.2 Benefits of Integration

Integrating NetSim with these simulators offers several advantages:

- **Enhanced Co-Simulation:** By combining the capabilities of power system simulators with NetSim, researchers can perform detailed co-simulation of power and communication systems.
- **Realistic Scenario Testing:** Researchers can create more realistic testing scenarios that include both the physical behavior of power systems and the network communication dynamics.
- **Improved Security Testing:** The integration allows for thorough testing of vulnerabilities in CPPS, enabling the identification and mitigation of potential cyber-attacks on power grids.
- **Protocol Support:** NetSim supports various electrical protocols necessary for interfacing with power system simulators, which are discussed in the section 3 of this manual.

3 Supported Electrical Protocols

When interfacing with power system simulators, NetSim supports the following electrical protocols:

- **IEEE C37.118 protocol (Synchro phasor Protocol):** Used for real-time exchange of synchro phasor data, enabling precise time synchronization and measurement of electrical quantities across distributed systems.
- **DNP3 (over TCP/IP):** Primarily used in SCADA systems for communication between outstations (e.g., RTUs) and a master station, offering robustness in adverse conditions and support for various data types.
- **Modbus (over TCP/IP):** A widely adopted protocol for communication between devices, allowing simple and efficient data transmission over TCP/IP networks, suitable for monitoring and control applications.
- **Generic Object-Oriented Substation Events (GOOSE), a subset of IEC 61850:** Facilitates high-speed messaging for real-time applications within substations, ensuring rapid exchange of critical information such as status changes and alarms.

- **IEC 60870-5-104 (over TCP/IP):** Standard for telecontrol (telemetry and tele protection) and telecommunication protocols in electrical engineering, enabling reliable data exchange between control centers and substations.

Additionally, NetSim interfaces with any protocol that operates over TCP/IP, ensuring compatibility with a wide range of communication standards used in power systems.

4 Lab Set-up

In this section we discuss how to set up your lab to perform attacks on communication network with real devices connected on a live network.

4.1 Network Setup

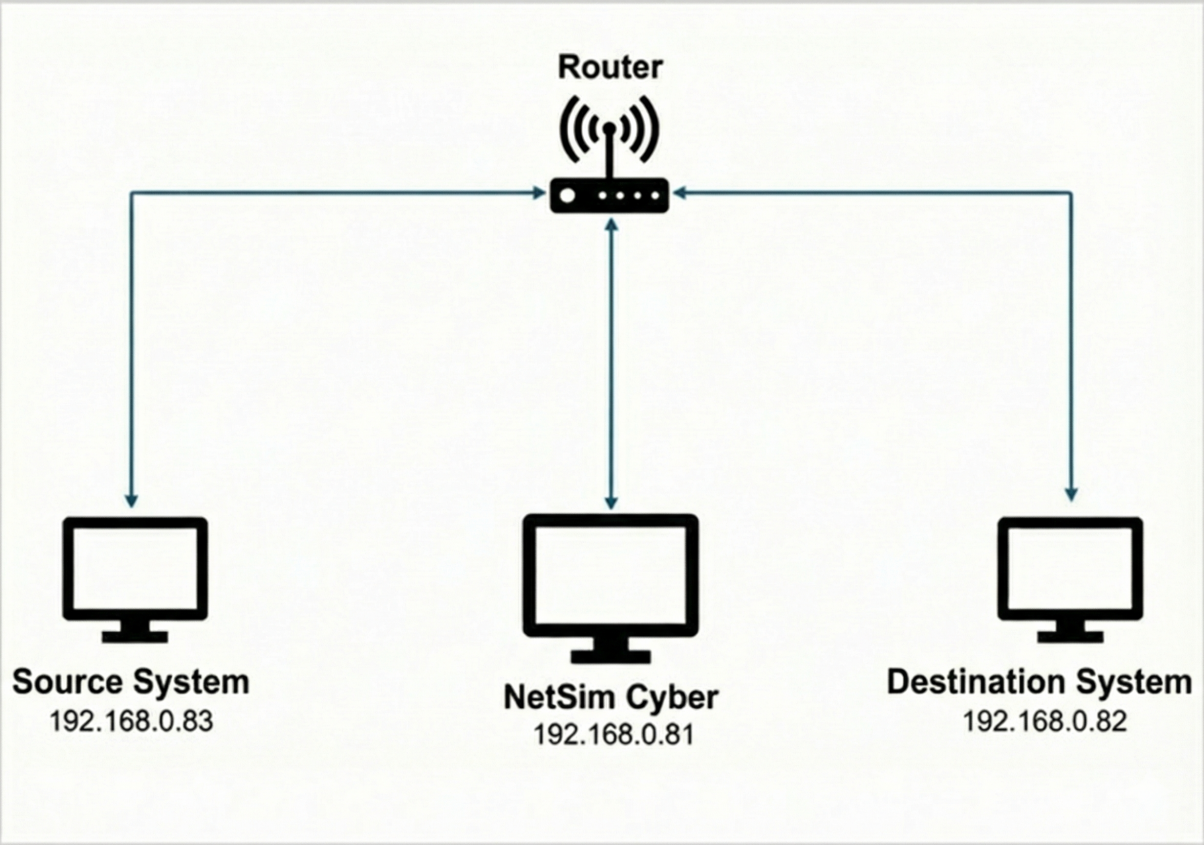

Figure 4-1: *Network Setup for Cyber. The IP addresses shown are for illustration purposes only. Users may configure any IP addresses, and all three devices must be interconnected to ensure communication between one another.*

This network setup consists of three systems: Source, Destination, and the NetSim Cyber system, which serves as the gateway. In this example:

Source System: IP address – 192.168.0.83. This system runs a client program to communicate with the server. Typically, in a use-case this can be the system/IP enabled device generating data and transmitting to its server or destination node. For example, in RTDS with communication module, PMU will be client sending Synchro phasor data can be considered as Client system which connects to PDC.

NetSim Cyber: 192.168.0.81. It emulates a network environment, enabling communication between the client and server. Data flows through the network created in NetSim.

Destination System: 192.168.0.82. This system runs a server program that connects to the source. Typically, in a use-case this can be the system/device which receives the data sent from the client or source node. For example, PDC which receives the data from the PMU is considered as Server System.

4.2 Steps to Set Up Source/Client and Destination/Server System

4.2.1 Method 1: Manual Route Configuration

1. Source system (192.168.0.83)

To add a static route, you'll need to open the command prompt in administrator mode. Once open, input the following command:

```
route add <Destination IP address> mask <Subnet mask> <Gateway IP address> metric 1
```

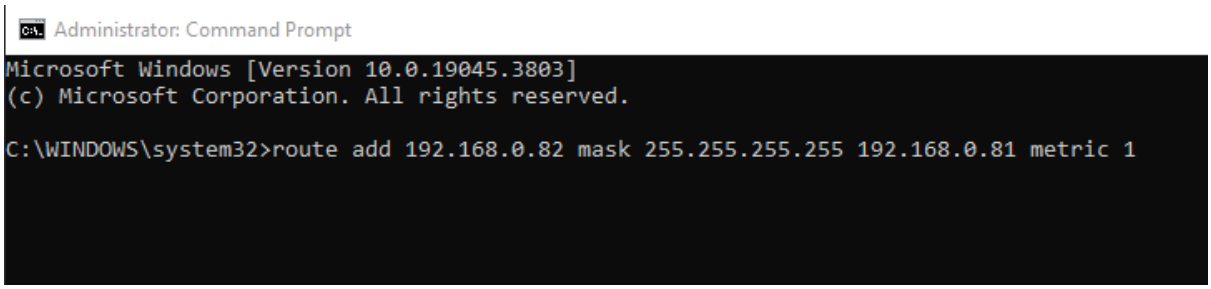

Figure 4-2: *Example for configuring static route in source system.*

Replace `<Destination IP address>` with the IP address of the destination system, `<Subnet mask>` with the appropriate subnet mask, and `<Gateway IP address>` with the IP address of the gateway. This command will effectively add the static route.

2. Destination system (192.168.0.82)

To facilitate communication for TCP traffic, it's essential to configure a reverse route on the destination system as well.

```
route add <IP address of Destination system> mask <Subnet mask> <Gateway IP address> metric
1
```

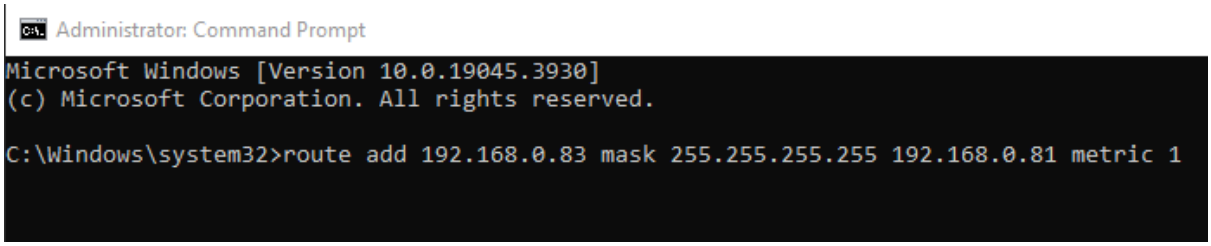

Figure 4-3: *Example for configuring static route in Destination system.*

4.2.2 Method 2: Automatic Configuration using NetSim Cyber Client

As an alternative to manually configuring routes on the source and destination systems, users can utilize the NetSimCyberClient.exe utility to automate the process.

Prerequisites: Before proceeding, ensure the NetSim Cyber Suite UI is running on the gateway system (192.168.0.81)

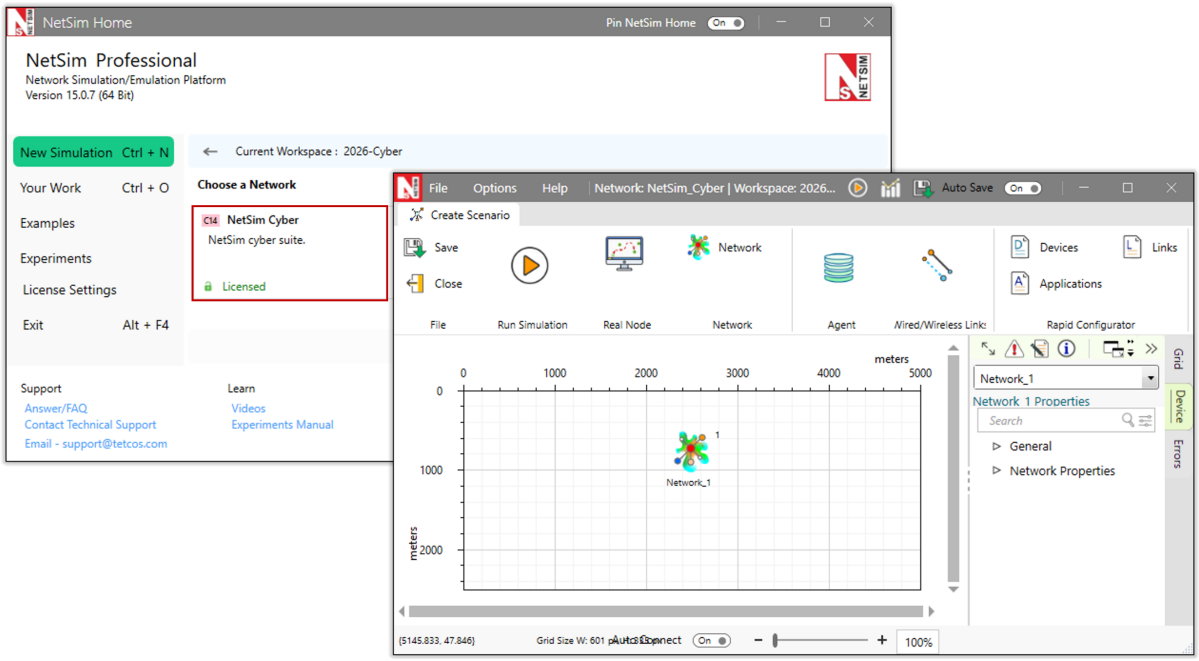

Figure 4-4: *Launching the NetSim Cyber Suite tile on the gateway system (IP: 192.168.0.81).*

Steps:

- Place the NetSimCyberClient.exe utility on both the Source/Client system (192.168.0.83) and the Destination/Server system (192.168.0.82).
- Run the NetSimCyberClient.exe as run as administrator on both systems.

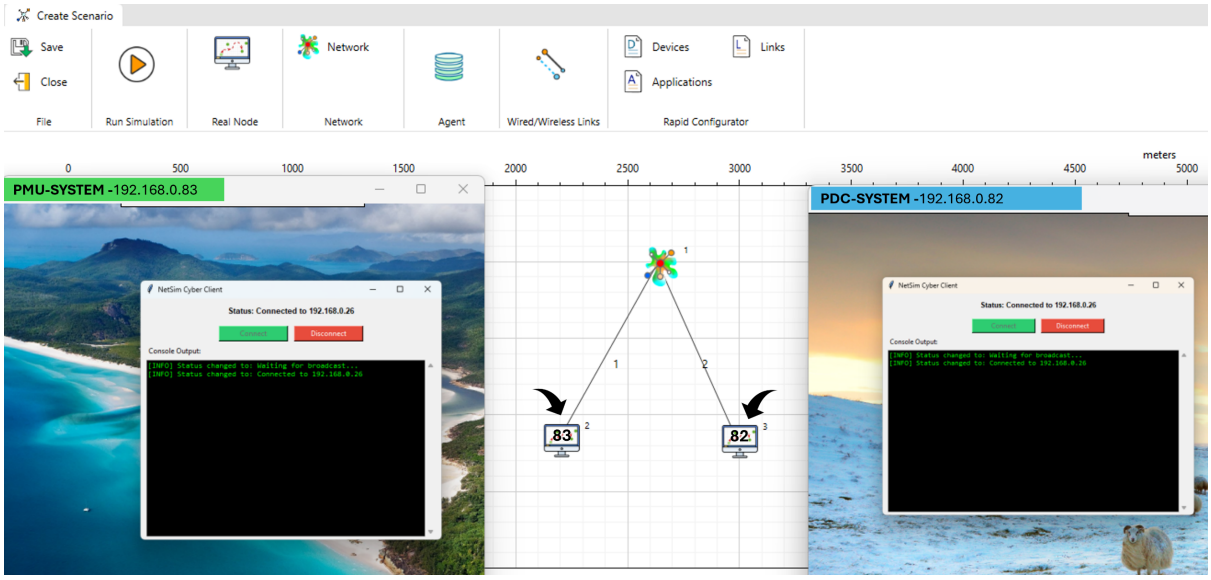

Figure 4-5: *NetSimCyberClient.exe utility interface showing the Connect option and traffic tools, launched with administrator privileges on both source/destination systems.*

Upon running the utility, the nodes will automatically be dropped into the NetSim environment, as seen in the above example layout where distinct nodes connect to the central network cloud. The routing tables on the client and server systems will be updated automatically to direct traffic through the NetSim gateway.

How it works:

- When the NetSim Cyber tile starts on the gateway system, it begins broadcasting a message containing its IP address across the entire network on port 19001 every second.
- NetSimCyberClient utility runs on the client or server system, it starts listening for these broadcast messages on port 19001.
- Upon receiving a message from the network on this port, the utility establishes a connection with the NetSim system.
- It then automatically alters the local route table so that packets are transmitted through the system running NetSim.
- Finally, it starts sending packets containing its own IP address and Device Name to NetSim. When NetSim receives these packets on port 19001, it reads the information and automatically drops a device node with the respective IP address and Device Name into the simulation scenario.

5 Network Attacks

5.1 Overview

In the realm of Cyber-Physical Power Systems (CPPS), network attacks pose significant threats that can disrupt operations, cause financial losses, and even endanger public safety. Understanding the various types of network attacks and their potential impact on CPPS is crucial for developing effective defense mechanisms. This section explores common network attacks, provides detailed descriptions, and explains how to simulate these attacks using NetSim.

NetSim supports network scenarios where the plant, sensor and actuators are connected over:

- Wired or wireless links
- Single and multiple communication links
- And running protocols that could be based on:
  - UDP (No ACKs in the reverse direction)
  - TCP (ACK based)

6 Featured Examples (Simulation)

(Requires NetSim Std/Pro Ver: v15.0 or higher, with Emulator Add-on)

All examples discussed in the manual will have a fixed setup in NetSim.

6.1 IEEE C37.118 Synchrophasor Protocol

This featured example demonstrates a Distributed Multi-Host Co-Simulation designed to validate the impact of cyber-attacks on IEEE C37.118 Synchrophasor data streams.

6.1.1 Protocol Overview

The IEEE C37.118 standard is the dominant protocol for Synchrophasor data transmission in Wide Area Monitoring Systems (WAMS). It defines four distinct message types:

- **Data:** Measurements made by the PMU (Phasors, Frequency, Analog).
- **Configuration:** Machine-readable message describing the data format and calibration factors.

- **Header:** Human-readable descriptive information provided by the user.
- **Command:** Machine-readable codes sent to the PMU for control (e.g., Start/Stop transmission).

Frame Structure regardless of the message type, all transmissions follow a common binary frame structure as defined in the standard. This structure ensures that information—whether it is real-time data or configuration files—can be consistently parsed by the Phasor Data Concentrator (PDC).

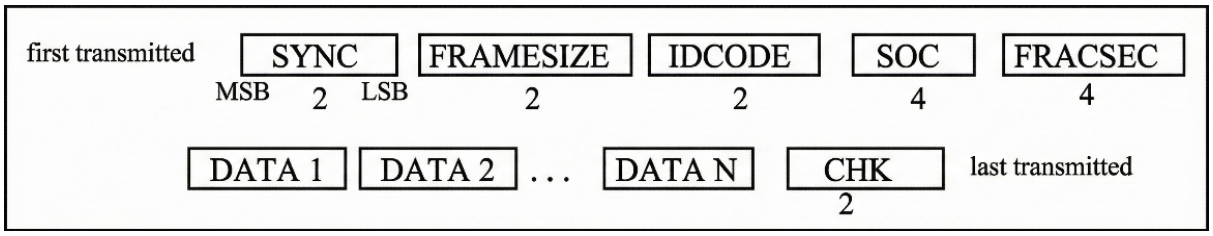

Figure 6-1: *Generic frame transmission order showing the Synchronization word (SYNC), ID Code, Timestamp (SOC/FRACSEC), Payload (Data), and Checksum (CHK).*

6.1.2 NetSim Cyber Architecture (3-System Setup)

This experimental setup is configured as a Distributed Multi-Host Co-Simulation. The architecture involves three distinct physical systems interconnected via a Local Area Network (LAN). NetSim Cyber functions as a real-time network platform (Man-in-the-Middle) to intercept, buffer, manipulate, and forward traffic between the Generation Tool and the Monitoring Tool.

The Physical Topology:

- **System 1 (Source):** Runs the Python PMU Simulator.
- **System 2 (NetSim Cyber):** Runs NetSim with the Cyber module.
- **System 3 (Destination):** Runs OpenPDC.

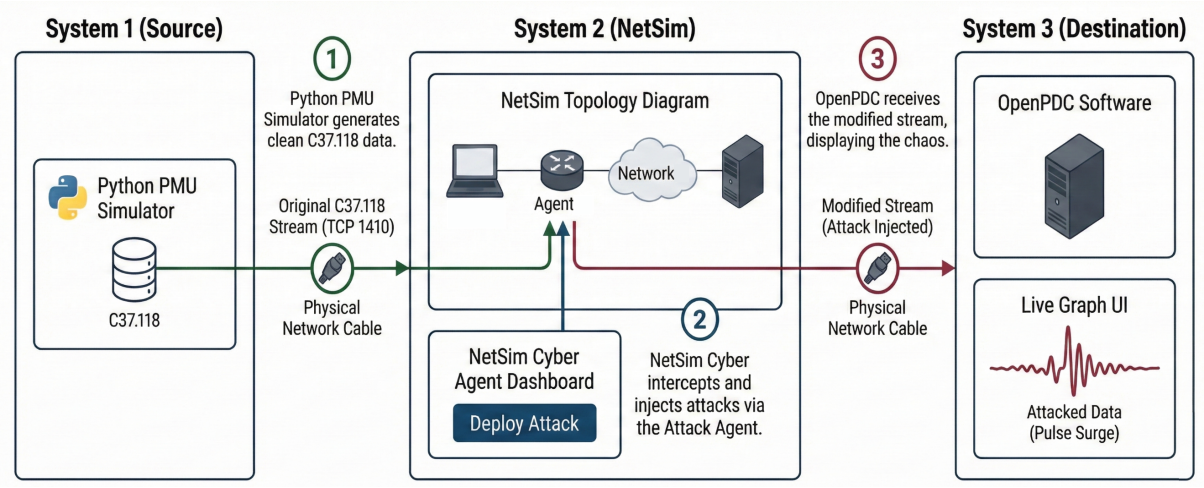

Figure 6-2: *Distributed Multi-Host Co-Simulation. The Attack Agent within the virtual network executes custom Python scripts to inject specific anomalies into the IEEE C37.118 data stream before it reaches OpenPDC.*

6.1.3 Component Description

System 1: Real Source (PMU)

- **Software:** Python PMU Simulator (Generation Tool)
- **Tool Description:** A custom Python-based PMU simulator implementing IEEE C37.118 uses the synchrophasor library to stream data. It waits for a connection request from the PDC and, once the handshake is complete, continuously streams synchrophasor data samples.
- **Configuration:** Generates IEEE C37.118-compliant synchrophasor data over its physical Network Interface Card (NIC).
- **Example IP:** 192.168.0.83

System 2: NetSim Cyber Network

- **Role:** Network Simulation and Attack Injection
- **Real Node Mapping:** The "Real Node" devices in the NetSim topology are mapped to the physical network interfaces of System 2. NetSim Cyber bridges live traffic between the PMU (System 1) and the PDC (System 3).
- **Attack Agent:** The Agent node inside virtual network is positioned logically between the source and destination nodes to intercept real-world synchrophasor traffic and perform payload modification or timing manipulation attacks.
- **Example IP:** 192.168.0.81

System 3: Real Destination (PDC)

- **Role:** Phasor Data Concentrator (PDC)
- **Software:** OpenPDC
- **Tool Description:** OpenPDC functions as the data destination and master station in the IEEE C37.118 communication architecture. It acts as the initiator, establishing a connections to the PMU, collecting synchrophasor measurements, and performing data visualization, time alignment, and archival.
- **Configuration:** Connects to configured PMU sources to receive, visualize, and store IEEE C37.118 data streams.
- **Example IP:** 192.168.0.82

6.1.4 Implemented Attack Vectors

The following seven attack modules were available for selection each targeting specific fields of the IEEE C37.118 data frame:

1. Data Decrement / Increment Attack
2. Random Noise Injection
3. Pulse Surge Attack
4. Ramp Attack
5. Frequency/ROCOF Attack
6. Time Synchronization Attack

6.1.5 Experimental Procedure

To validate the impact of cyber-attacks in a real-network environment, the following procedure is executed:

Step 1: Network Connectivity & Initialization

- Ensure all three systems are interconnected via the Local Area Network (LAN).
- Verify connectivity using ping commands between System 1, System 2, and System 3.

Step 2: Launch NetSim UI (System 2)

- Start NetSim: Launch the NetSim Cyber Suite tile on System 2 (Gateway).
- Wait: Ensure the UI is running. It will begin broadcasting on port 19001 to find clients.

Step 3: Connect Clients (Systems 1 & 3)

- Execute NetSimCyberClient.exe as Administrator on both System 1 (Source) and System 3 (Destination).

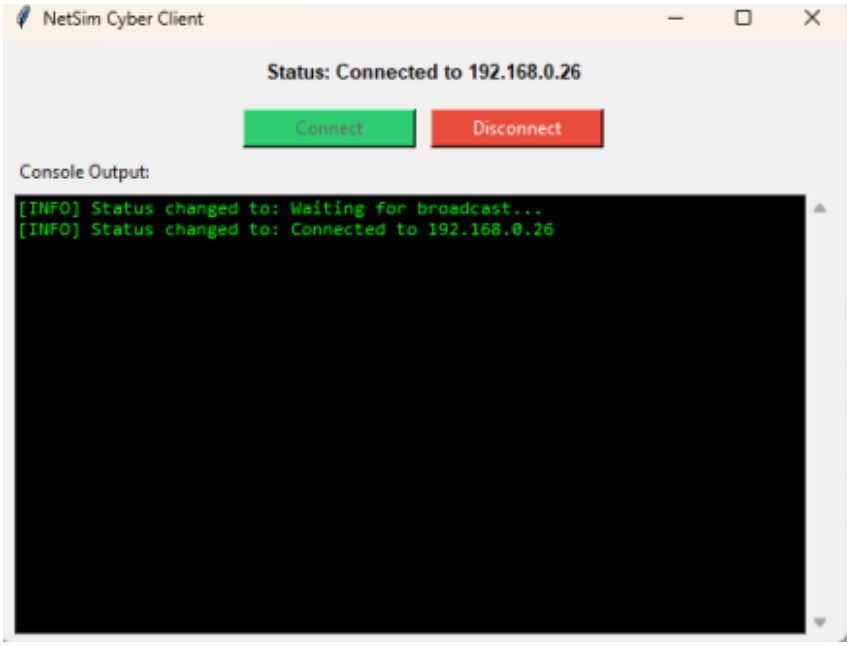

Figure 6-3: *NetSimCyberClient.exe.*

- Check that the status on both clients changes to "Connected to [NetSim IP]"
- Result: The devices (PMU and PDC nodes) will automatically appear in the NetSim design environment on System 2, and their routing tables are updated to force traffic through NetSim.

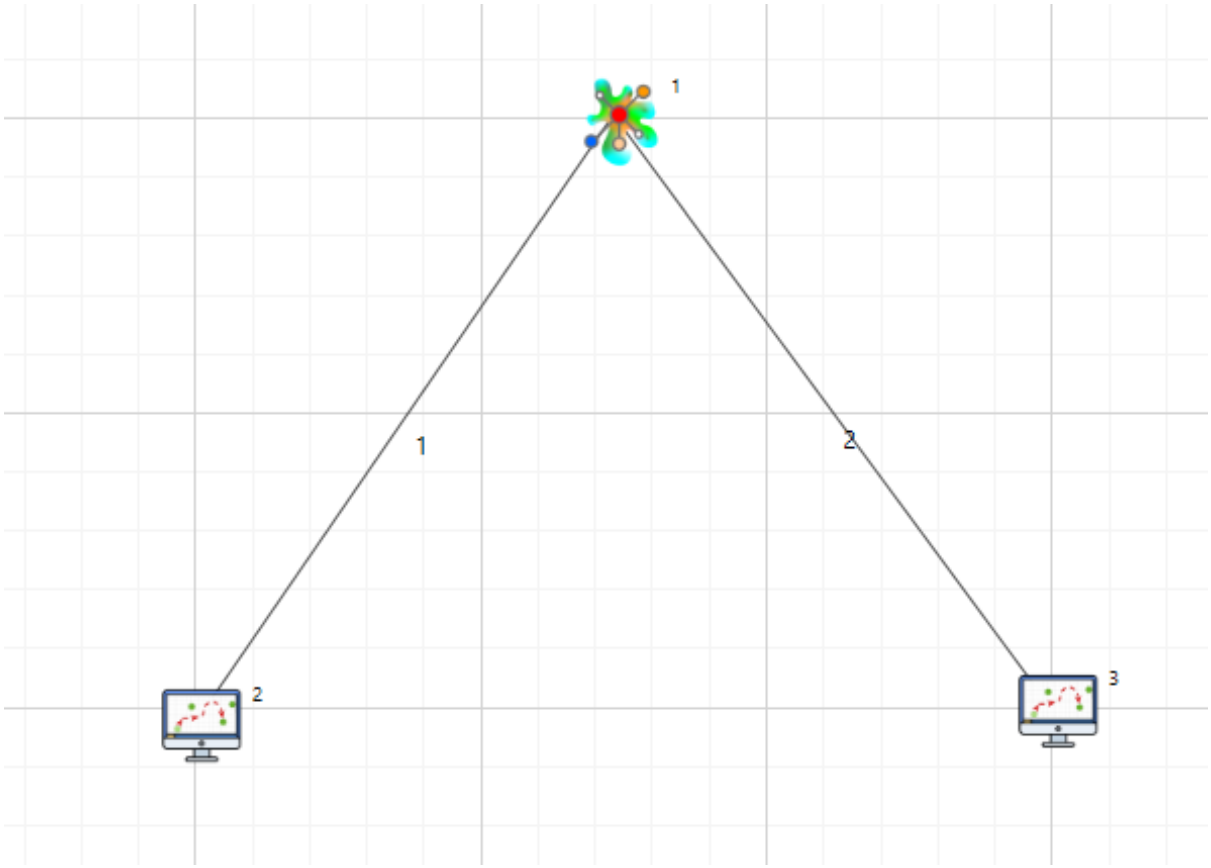

Figure 6-4: *Automatically dropped devices in NetSim.*

Once baseline communication is established between the Source and Destination, setting up NetSim Cyber is the next critical step.

Step 3: Attack Injection

- In the NetSim Design window, locate the Agent node positioned between your newly added Real Nodes.

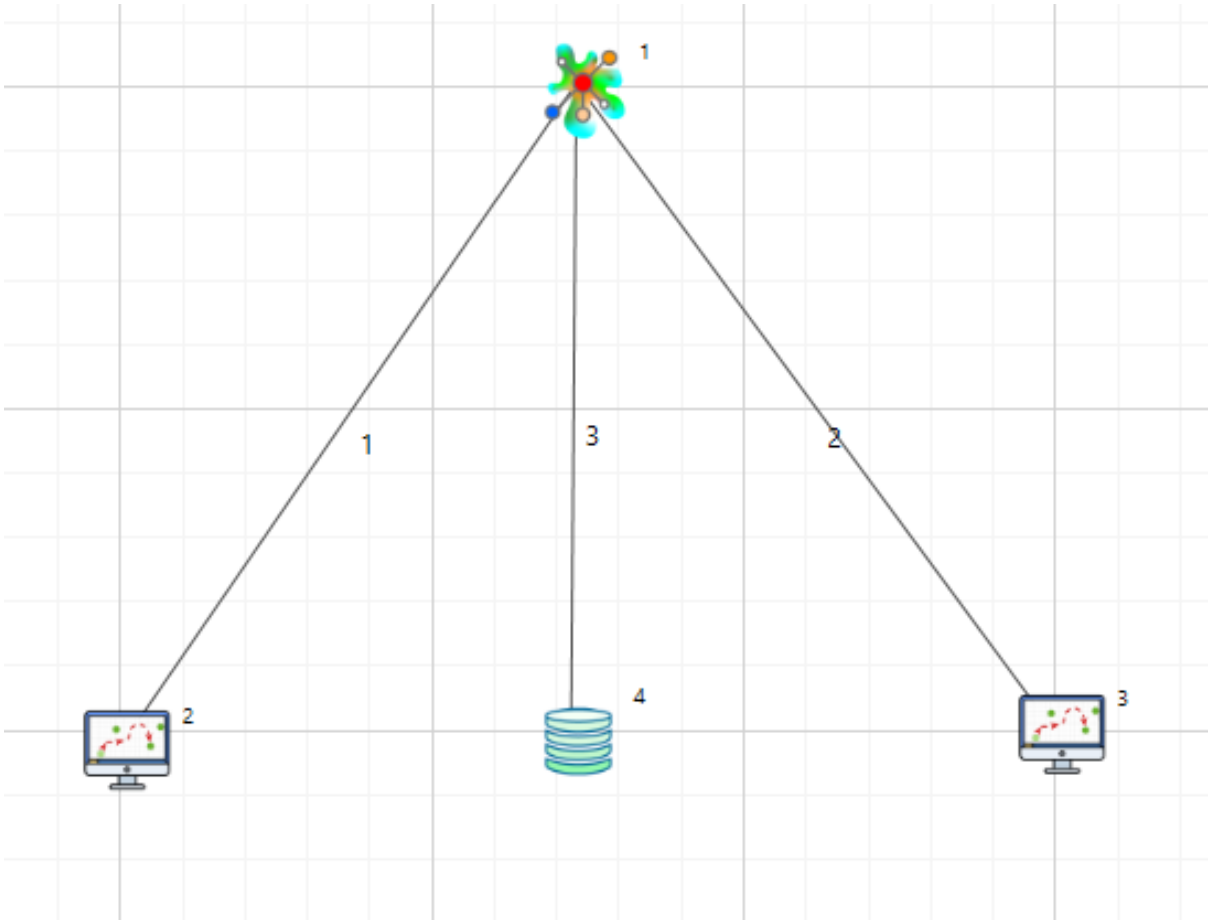

Figure 6-5: *NetSim Agent device.*

- The Agent Properties panel was accessed.

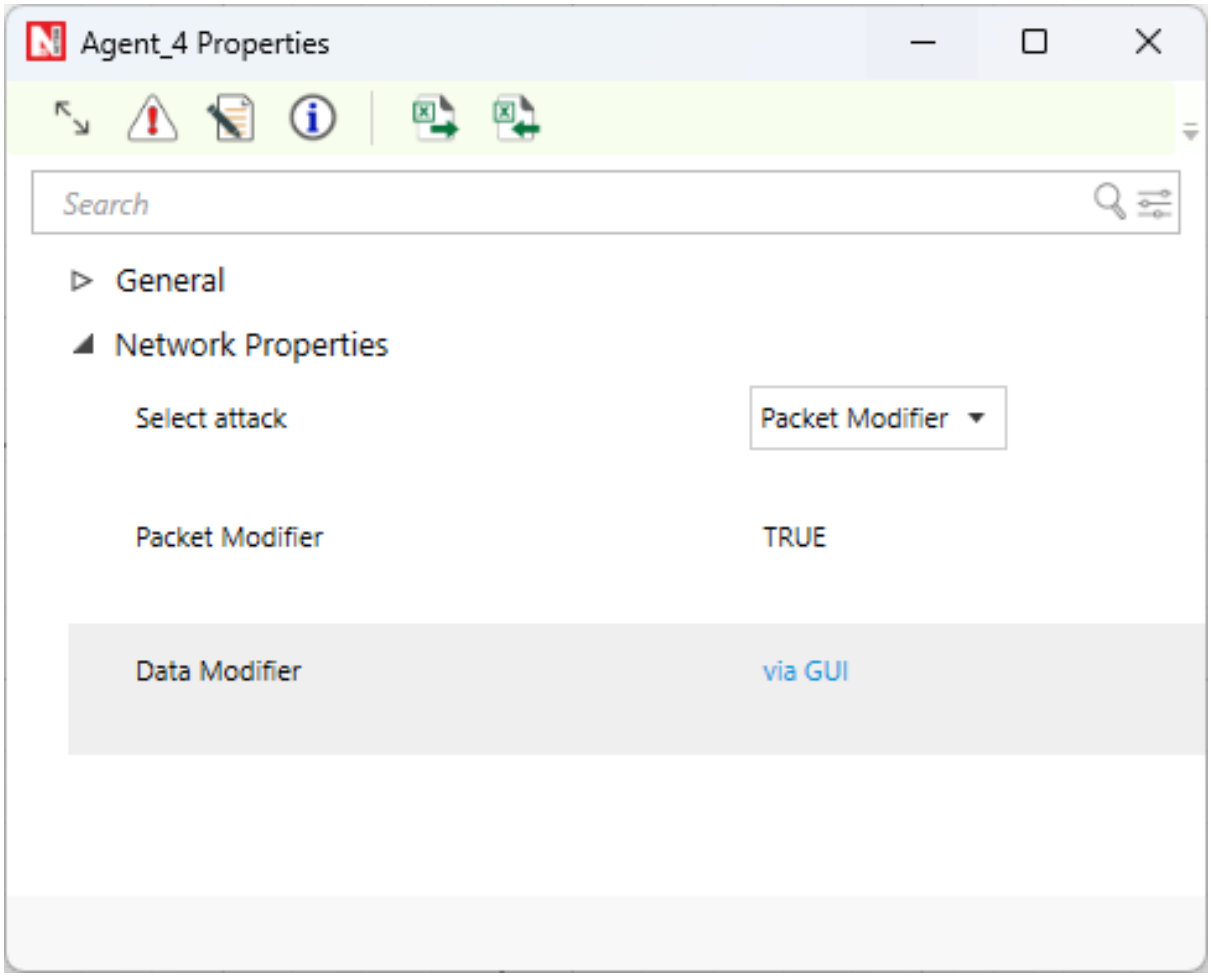

Figure 6-6: *NetSim Agent Properties.*

- In the "Data Modifier" window via gui that opens, locate the Payload row and check the box to enable it.

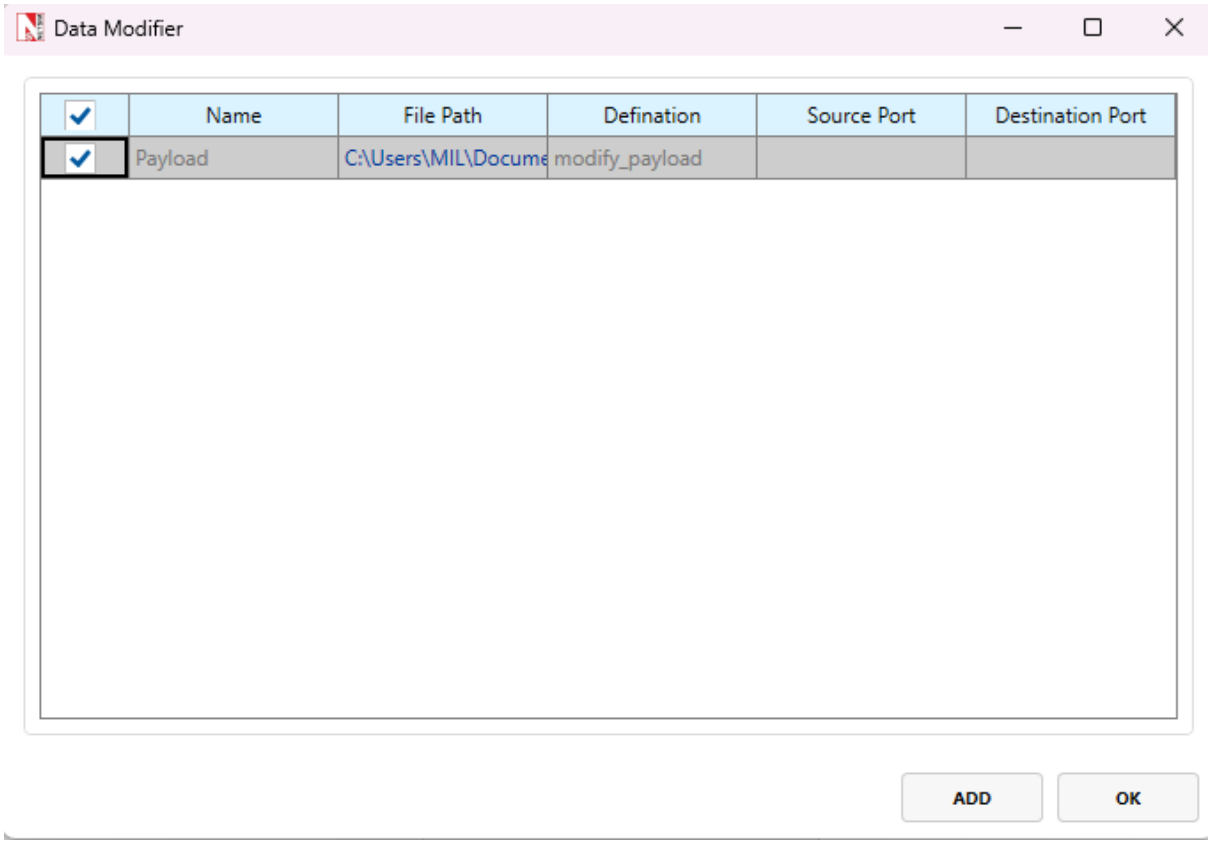

Figure 6-7: *Data modifier window.*

- By default, you can simply select the checkbox and click OK. If you want to use a different file, click the Add icon and browse to the required Python file.
- Select the master control script: `payload_modifier_master.py` (located in your workspace's PythonUserCode folder).
- Click OK to save the configuration.

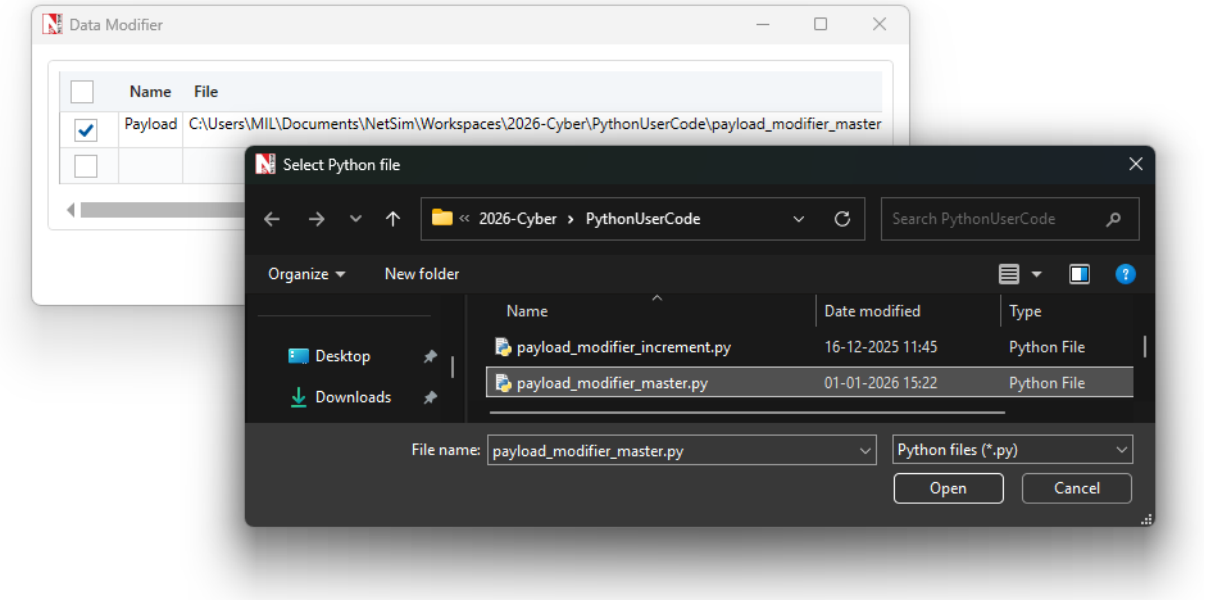

Figure 6-8: *Data Modifier configuration window showing the selection and attachment of a custom Python payload modification script for user-defined packet manipulation within the NetSim attack framework.*

Note: To configure a data modification rule, you must specify a Name (unique identifier), a File Path (location of the script), and a Definition (the specific logic within the file to execute). Example: In our case, the Name is set to `payload` and the Definition is set to `modify-payload`, as defined in our Python script.

Step 4: Setting up a traffic filter

Click on the Application icon from rapid configurator from the top toolbar.

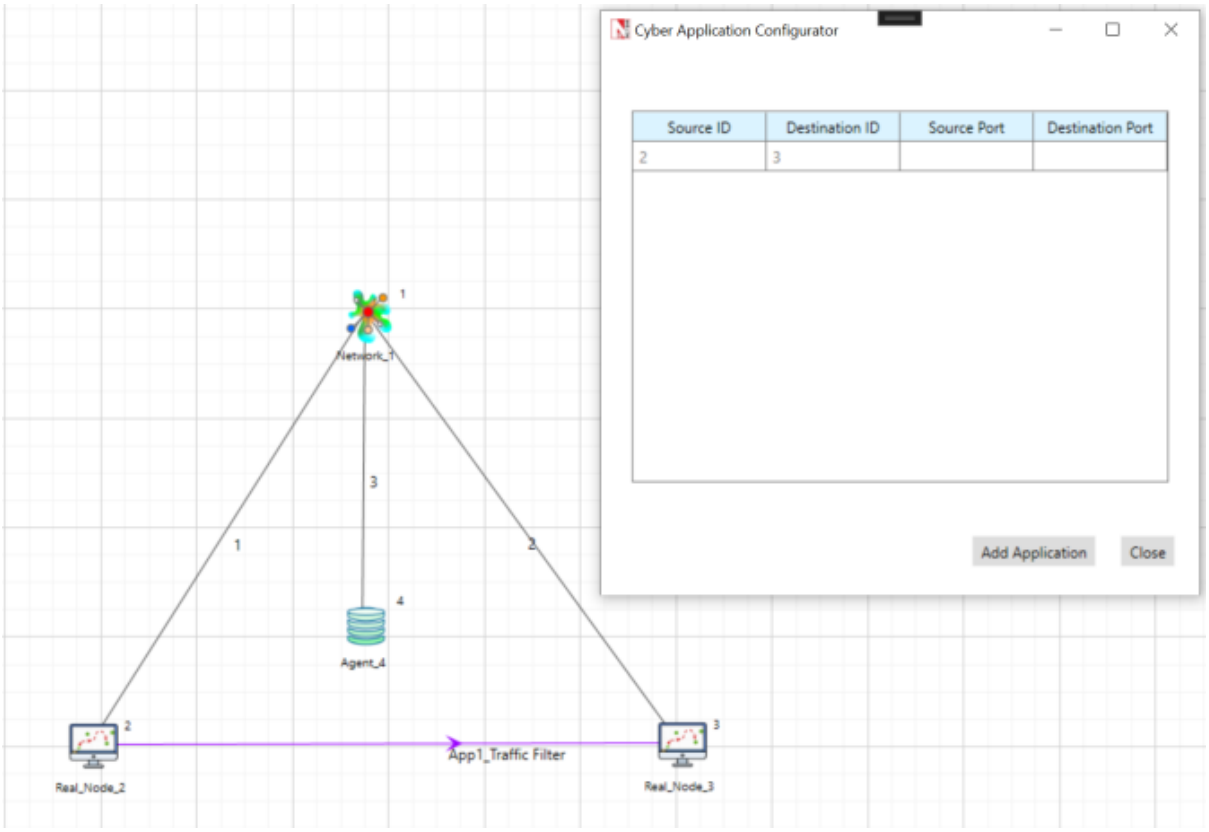

Figure 6-9: *Cyber Application Configurator window.*

- Application needs to be configured from source and destination and click on add application.

Note: If you are utilizing the Method 1: Manual Route Configuration described in the Lab Setup 7.1.1.

Step 5: Run Simulation

- Click Run: Press the Run Simulation button in the NetSim toolbar.

Two windows will launch on System 2:

NetSim Cyber Console: A black command-line window for logs.

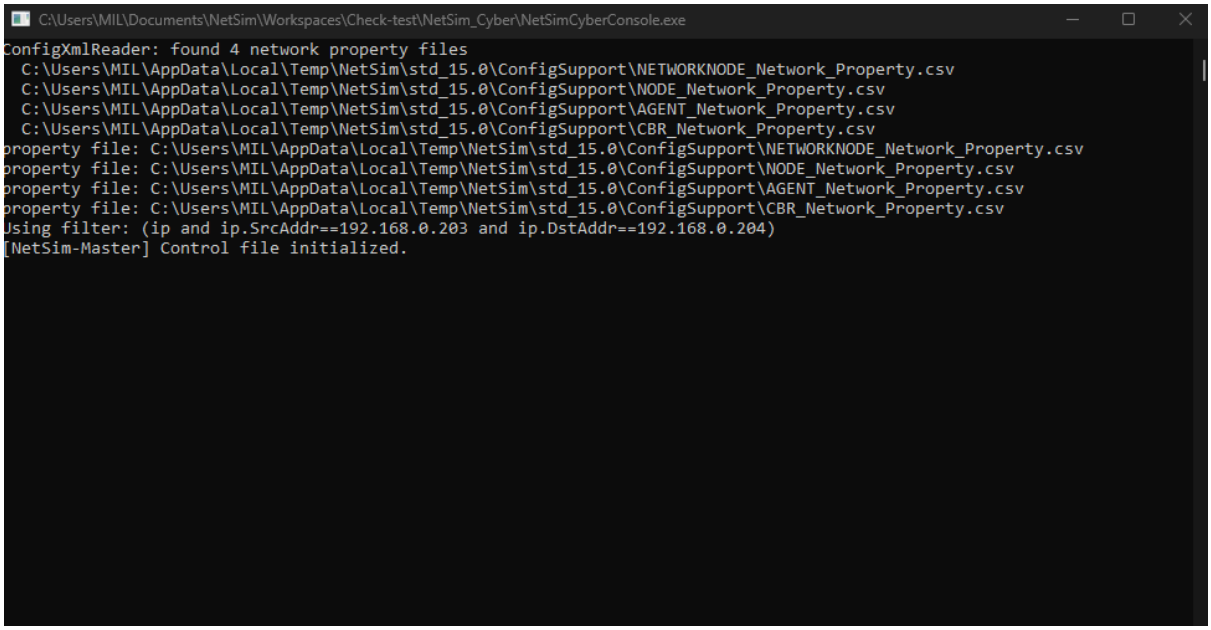

Figure 6-10: *NetSim Cyber Console.*

- **Attack Dashboard:** The GUI with Dropdown option for different attack types (e.g., Pulse, Ramp). Which also has Automatic and Manual mode.

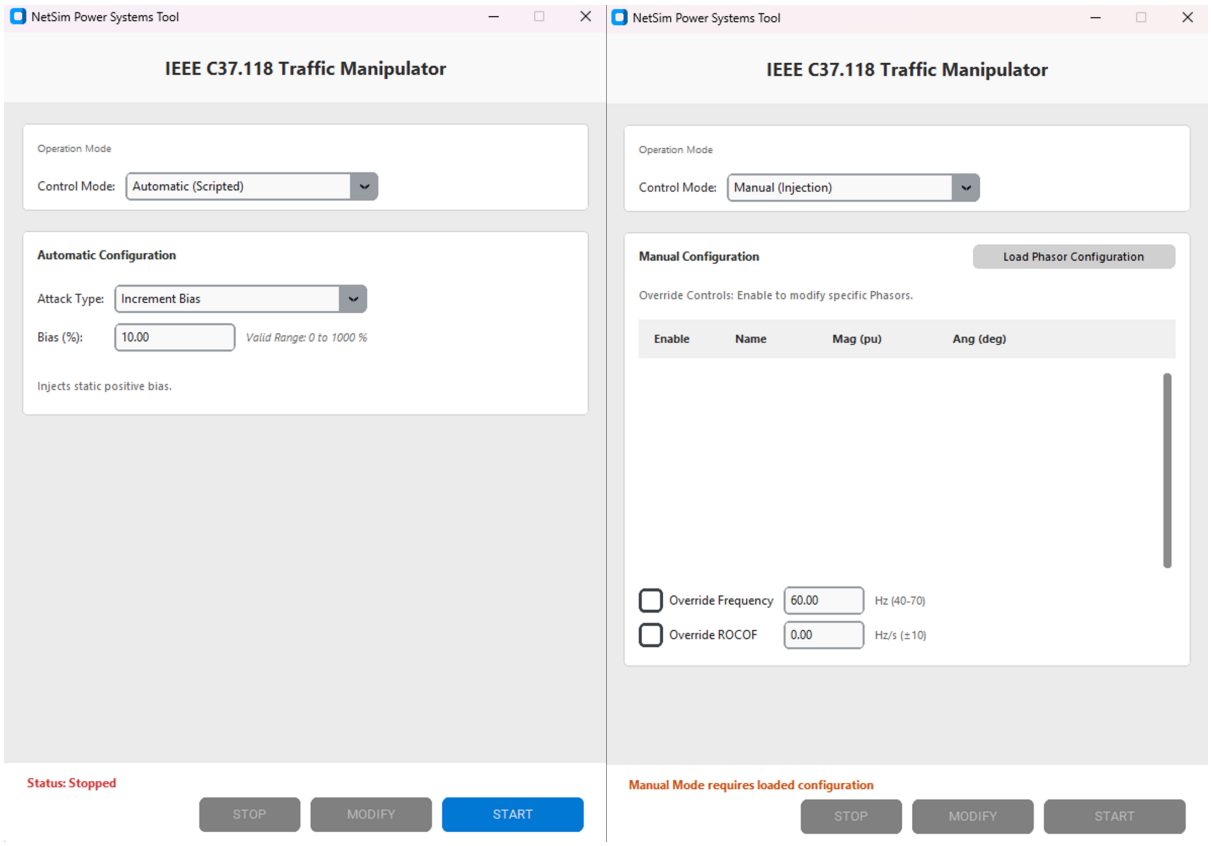

Figure 6-11: *Attack Dashboard with Automatic and Manual options.*

- The NetSim network is now live and waiting for traffic; we can setup PMU and PDC.

Note: In manual mode, the phasor configuration information is only loaded once the PMU and PDC

start communicating. The simulation must intercept the live Configuration Frame (CFG) from the handshake to populate these settings.

Step 6: PMU Initialization (System 1)

- Navigate to the Pypmu master folder on System 1.
- Double-click the `run_gui.bat` file to launch the PMU Launcher interface.
- By default, the IP address is set to the loopback address. For multi-system testing, configure the IP address to match the actual IP of the system running the PMU (e.g., if the PMU runs on 192.168.0.26, use 192.168.0.26).
- Click the Start PMU Simulator button.

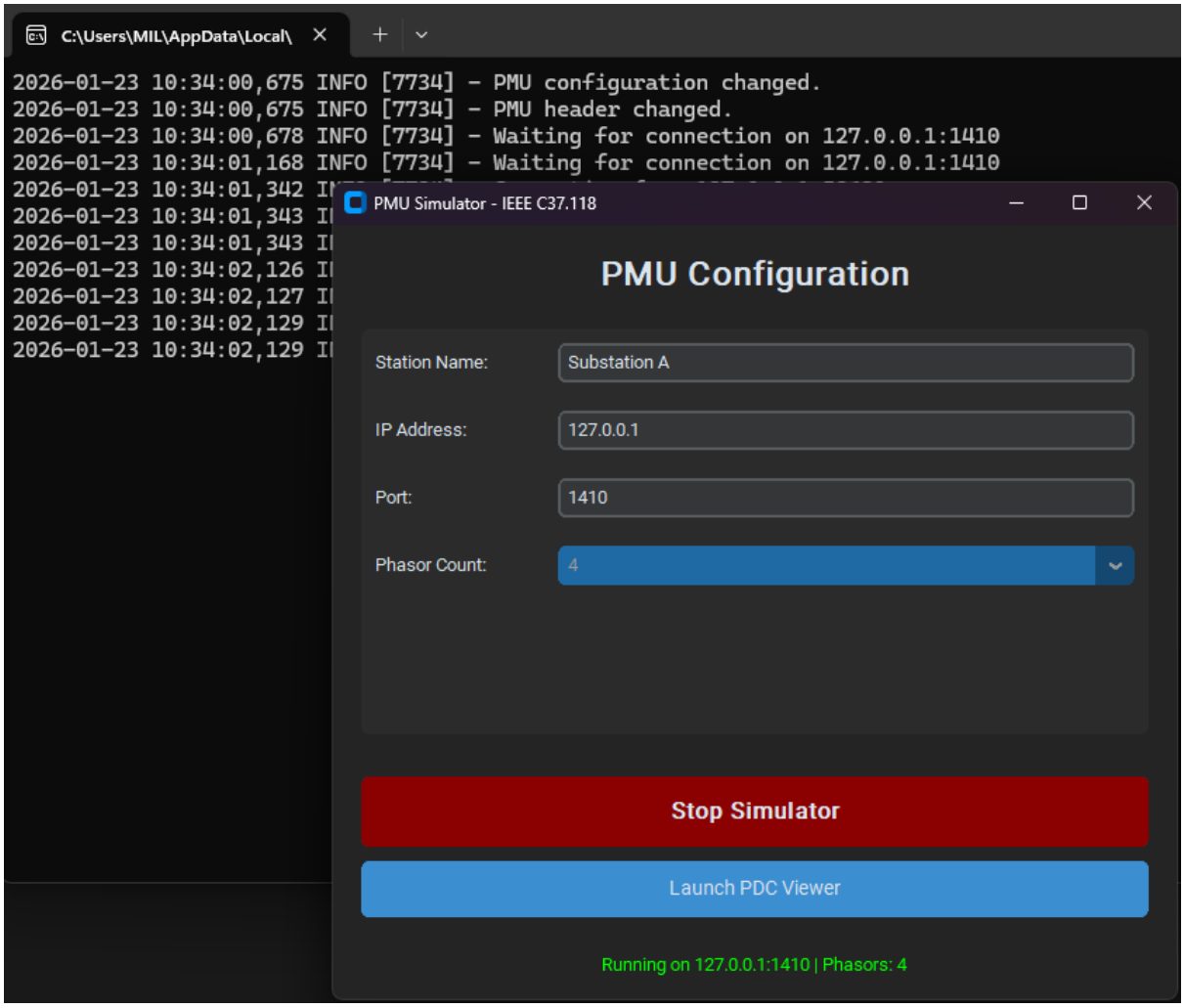

Figure 6-12: *PMU Simulator interface and console output showing successful configuration and the simulator waiting for a PDC connection on port 1410.*

- A terminal window will appear. Verify it displays: `Waiting for connection on....`

Step 7: Connect PDC & Verify NetSim (System 3 & 2)

Import Configuration (OpenPDC Manager)

- Ensure the provided `PMU_Connection.xml` file is accessible on System 3.
- Launch the OpenPDC Manager on System 3.

- On the Home screen, click the Input Device Wizard button in the “Quick Links” panel.

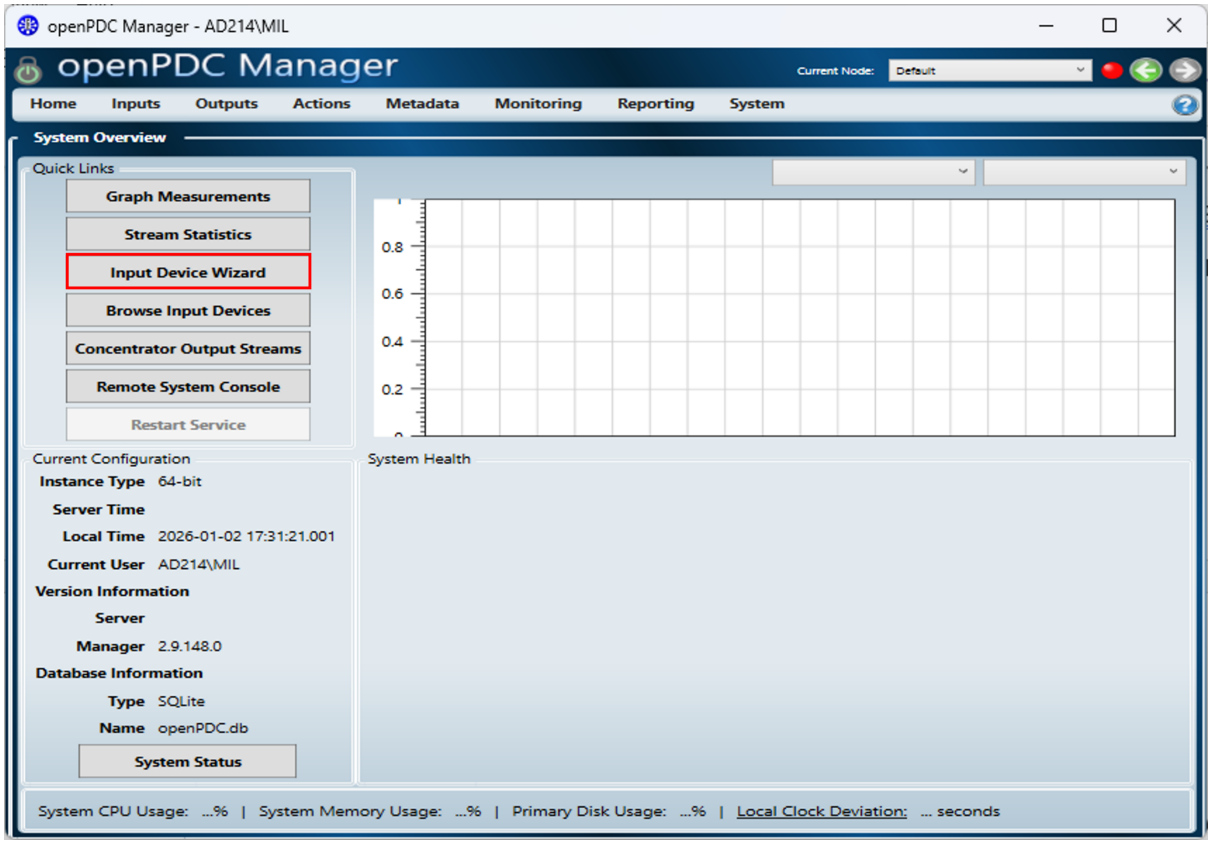

Figure 6-13: *openPDC Manager interface highlighting the Input Device Wizard used to configure and add PMU data sources for synchrophasor data acquisition.*

- A popup titled “Welcome to the Wizard Walkthrough” may appear. Close this popup to proceed manually.

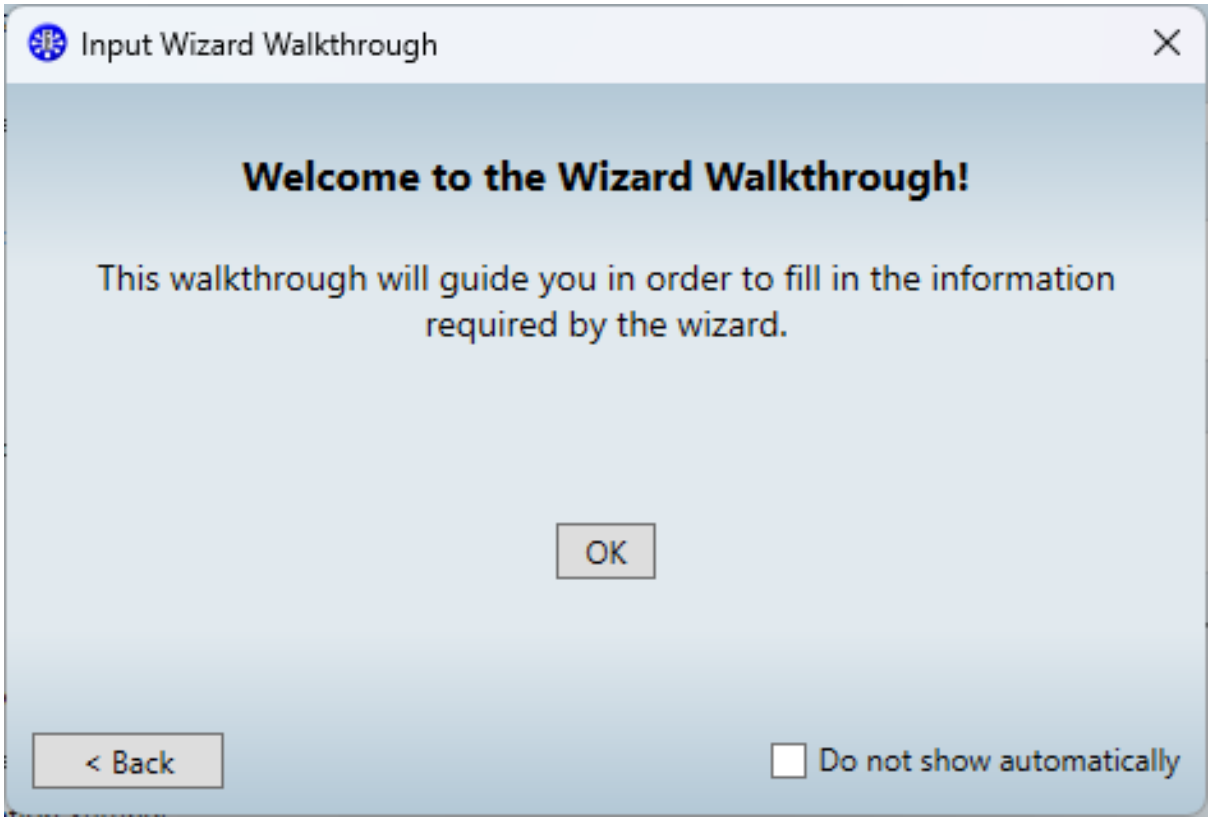

Figure 6-14: *Input Device Wizard walkthrough welcome screen in openPDC.*

- In the Wizard, locate the section Step 2: Select Device Configuration Settings.

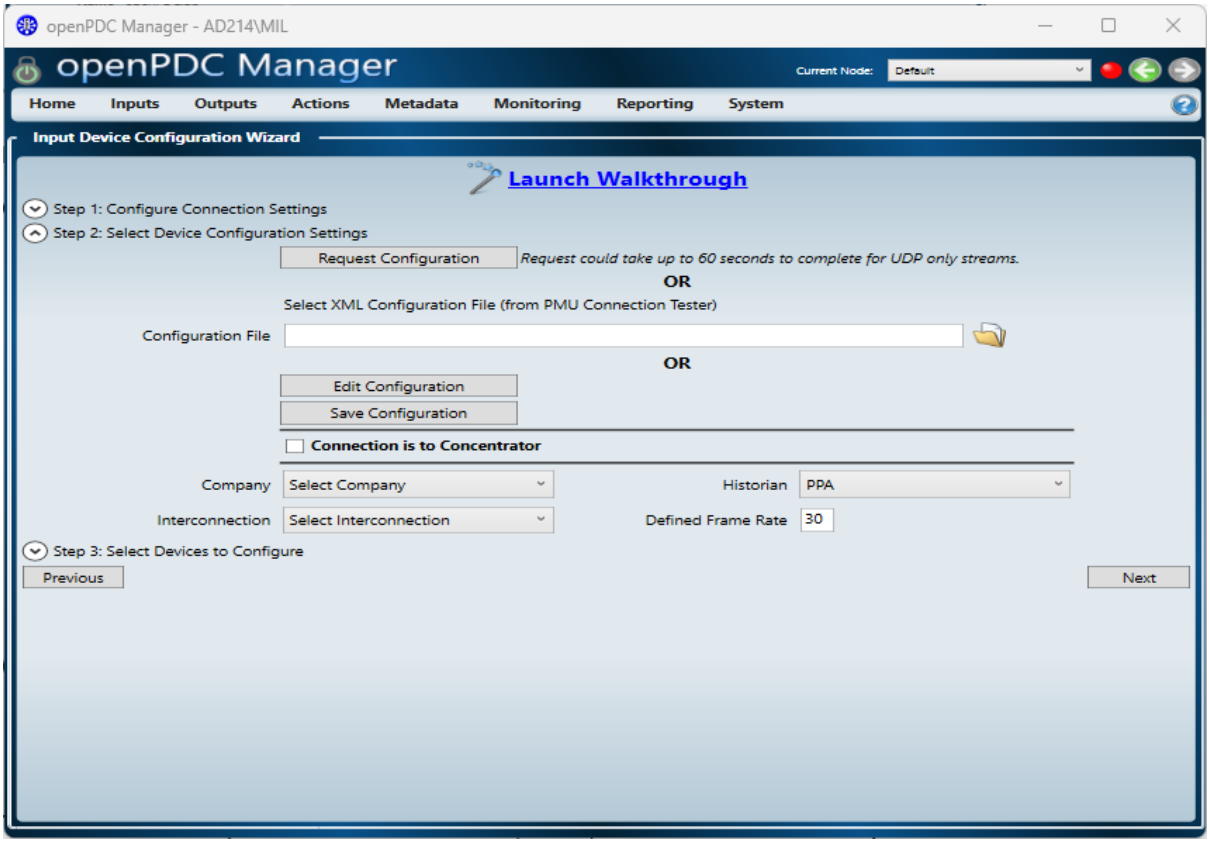

Figure 6-15: *openPDC Input Device Configuration Wizard showing the connection and device configuration settings used to register and configure a PMU data source.*

- Find the field labeled "Configuration File". Click the folder icon to browse and select the pre-configured `PMU_Connection.xml` file provided with the lab files.

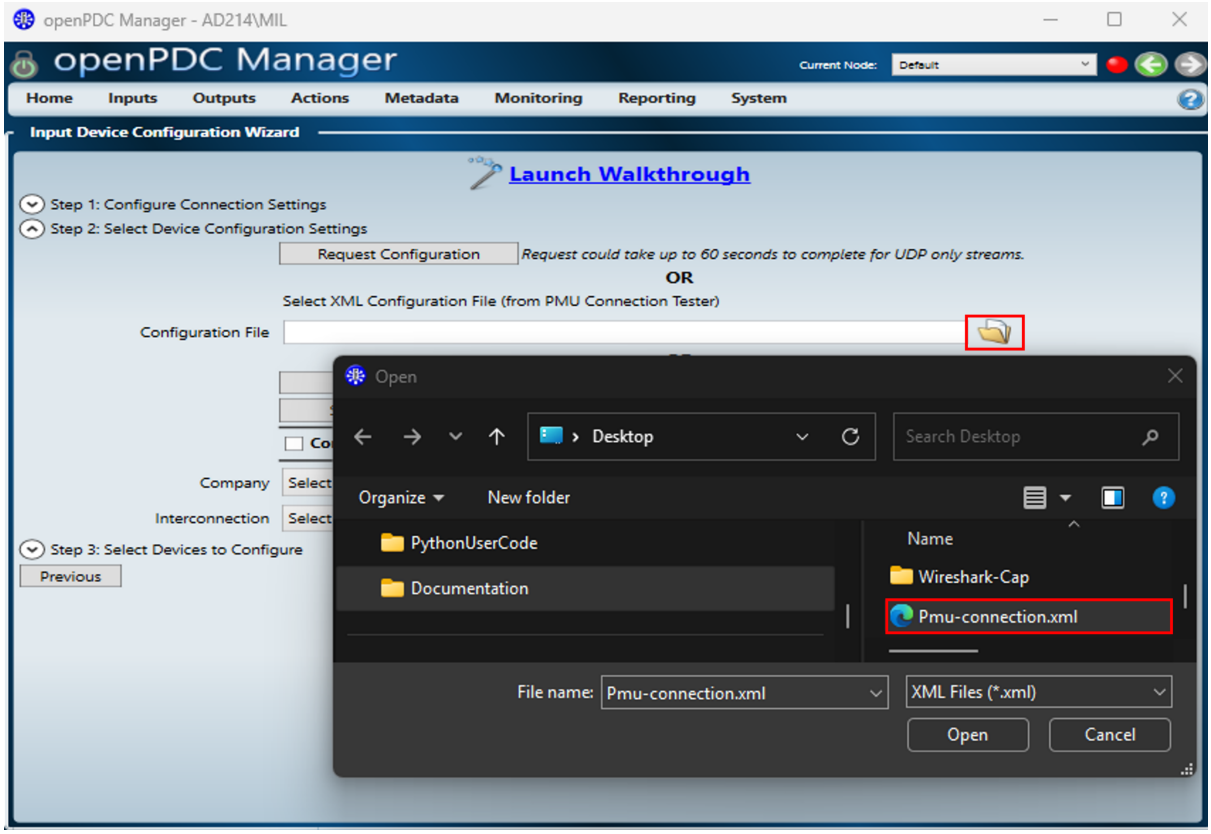

Figure 6-16: *Selecting the PMU XML configuration file (pmu-connection.xml) in the openPDC Input Device Configuration Wizard to import connection parameters for the PMU.*

- Click Next.
- The wizard will advance to Step 3: Select Devices to Configure. Verify that your device (e.g., "STATION_A") is listed and checked.

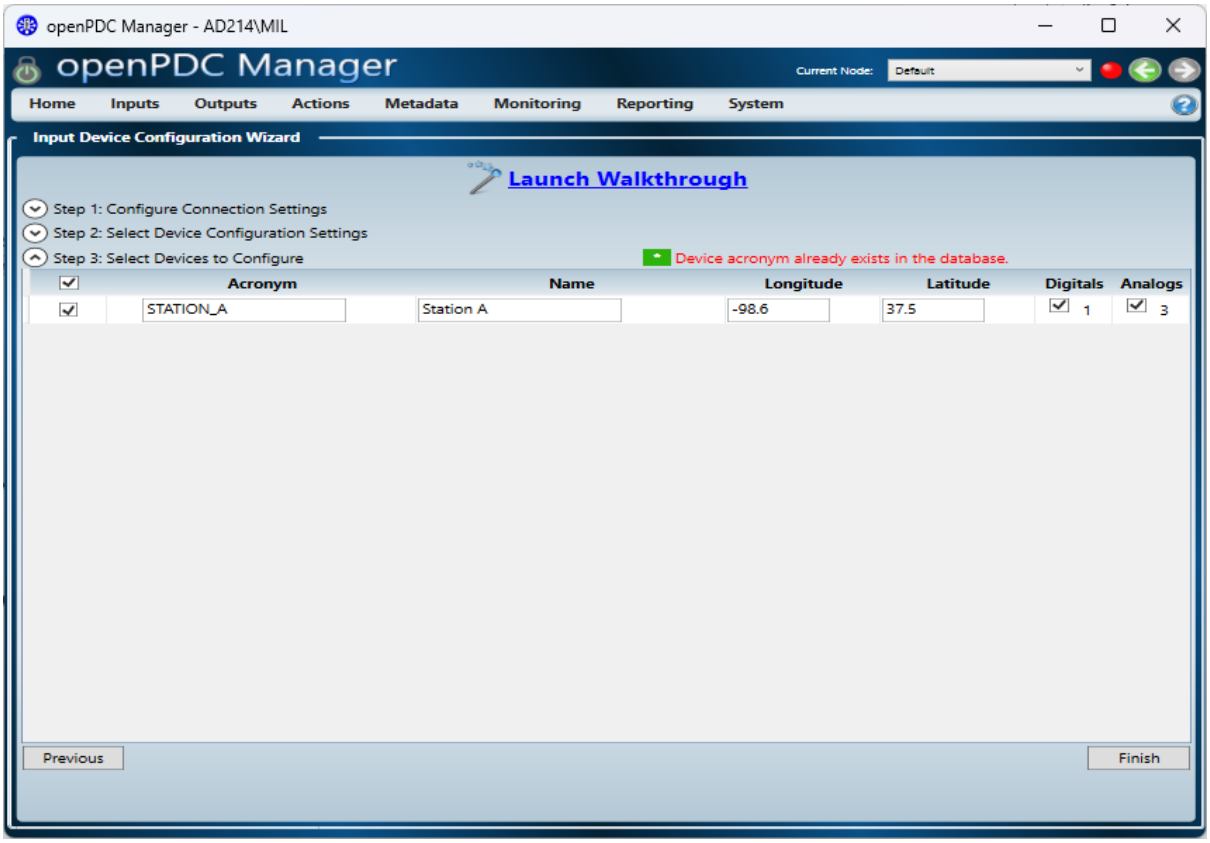

Figure 6-17: *openPDC Input Device Configuration Wizard displaying the detected PMU (Station A) and confirming that the device is already registered in the database.*

- Click Finish.

Configure Phasor Units

- After the wizard closes, navigate to **Inputs** → **Browse Input Devices** in the top menu.
- Click on phasors the newly added device (e.g., STATION_A) to open its settings.
- In the Manage phasors settings window, locate the rows for VA, VB, and VC.
- Manually change the Type drop-down for all three phases from "Current" to **Voltage** one by one and save those settings.

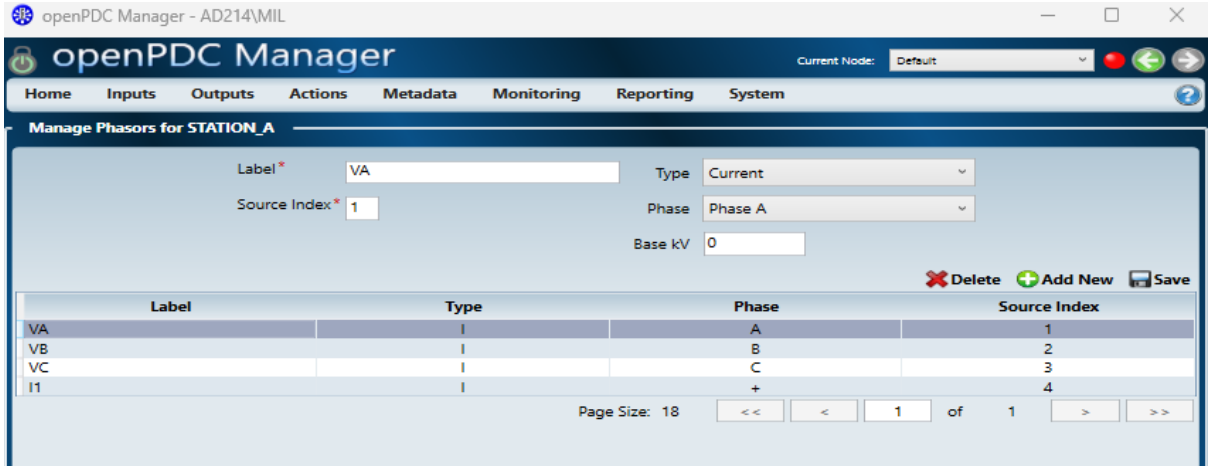

Figure 6-18: *Managing phasor settings in openPDC to update the VA, VB, and VC phasors by changing their type from Current to Voltage and saving the configuration.*

Verify Live Data

- Navigate to **Home** → **Graph Measurements**.
- Expand Directed connected, and select Station_A and Edit.
- Locate the Connection String input field (often found at the top of the graph window or under a "Settings" tab).
- Enter the following string to explicitly point the graph to your PMU source:
  `transport=Tcp; server=192.168.0.83; port=1410`
  (Note: Replace 192.168.0.83 with the actual IP address of your System 1 if different).

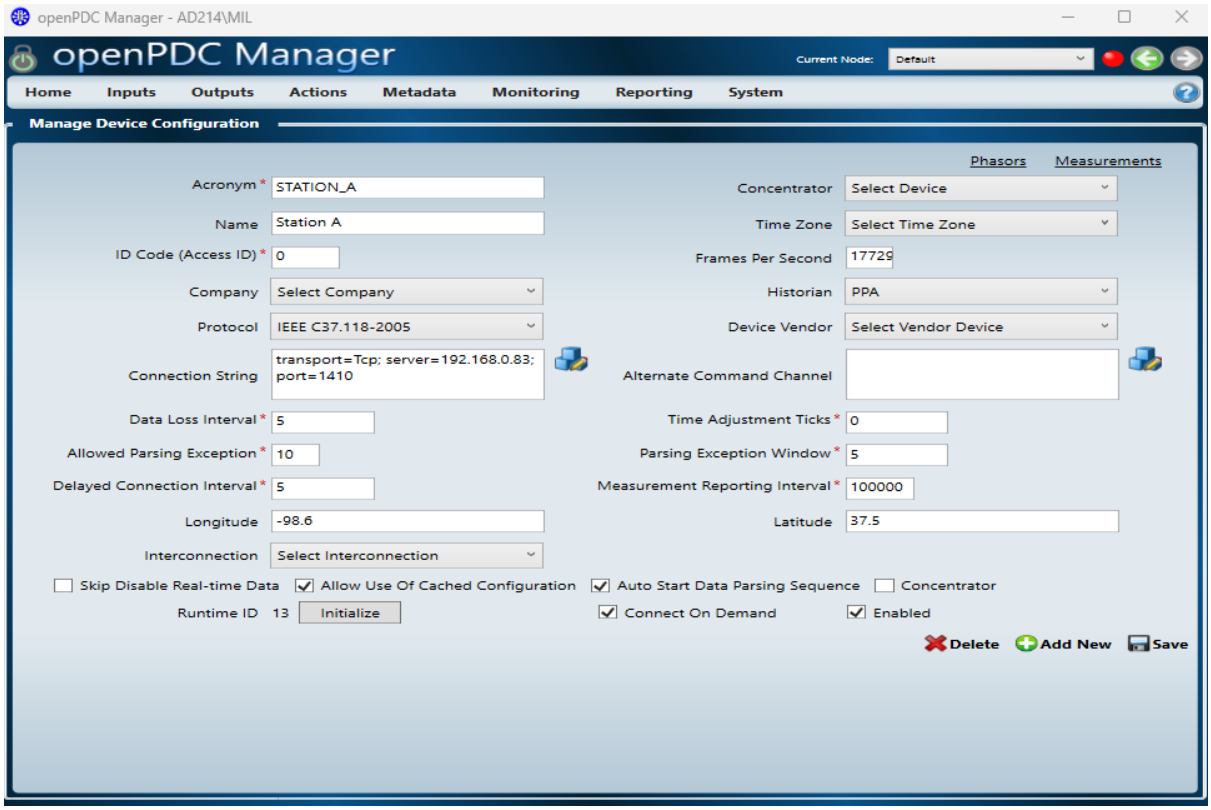

Figure 6-19: *Configuring the PMU connection string in openPDC to explicitly bind the input device to the PMU source using TCP (server IP and port 1410).*

- Click Save.
- Confirm that the voltage and frequency graphs display a stable, flat line (e.g., 62.5 Hz / 14635 Volts).

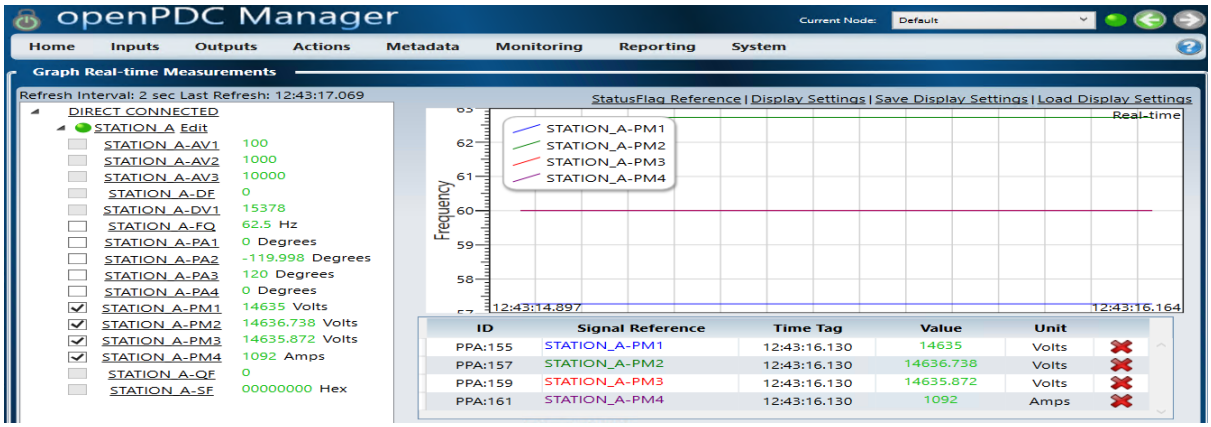

Figure 6-20: *openPDC real-time measurement view displaying live PMU phasor data (voltage and current magnitudes) from STATION_A after successful PMU–PDC connectivity and configuration.*

- Immediately check the NetSim Cyber Console (black window) on System 2.

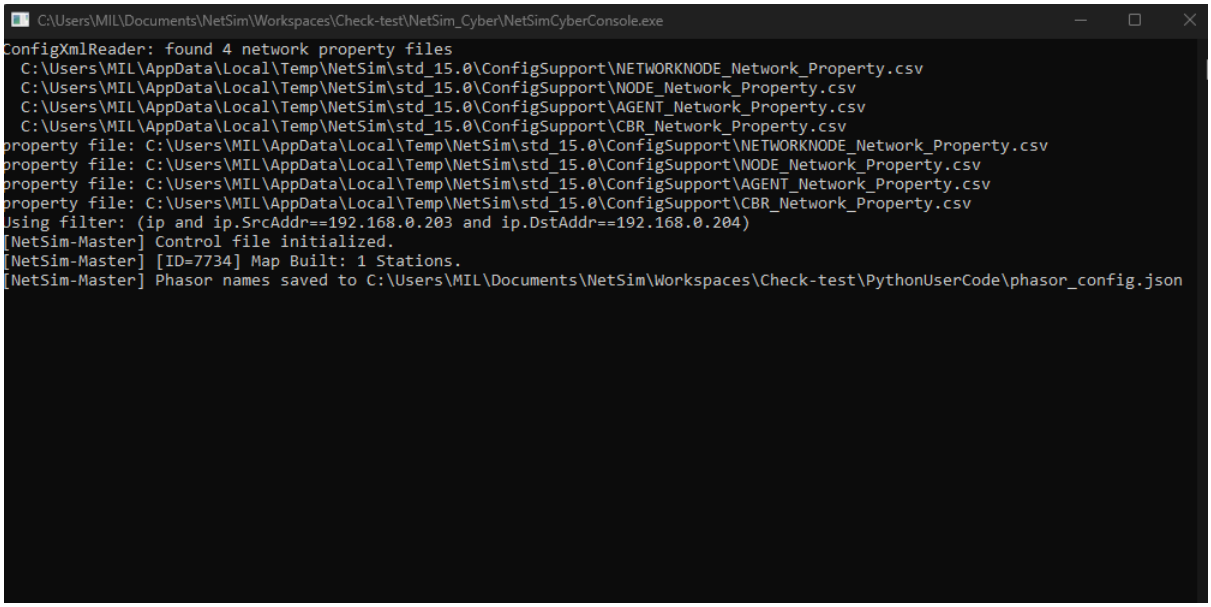

Figure 6-21: *NetSim Cyber Console once connection established between PMU and PDC.*

- The console will display a MapBuilt with number of stations present; also phasor names will be saved to JSON file which will later be loaded in manual mode UI.

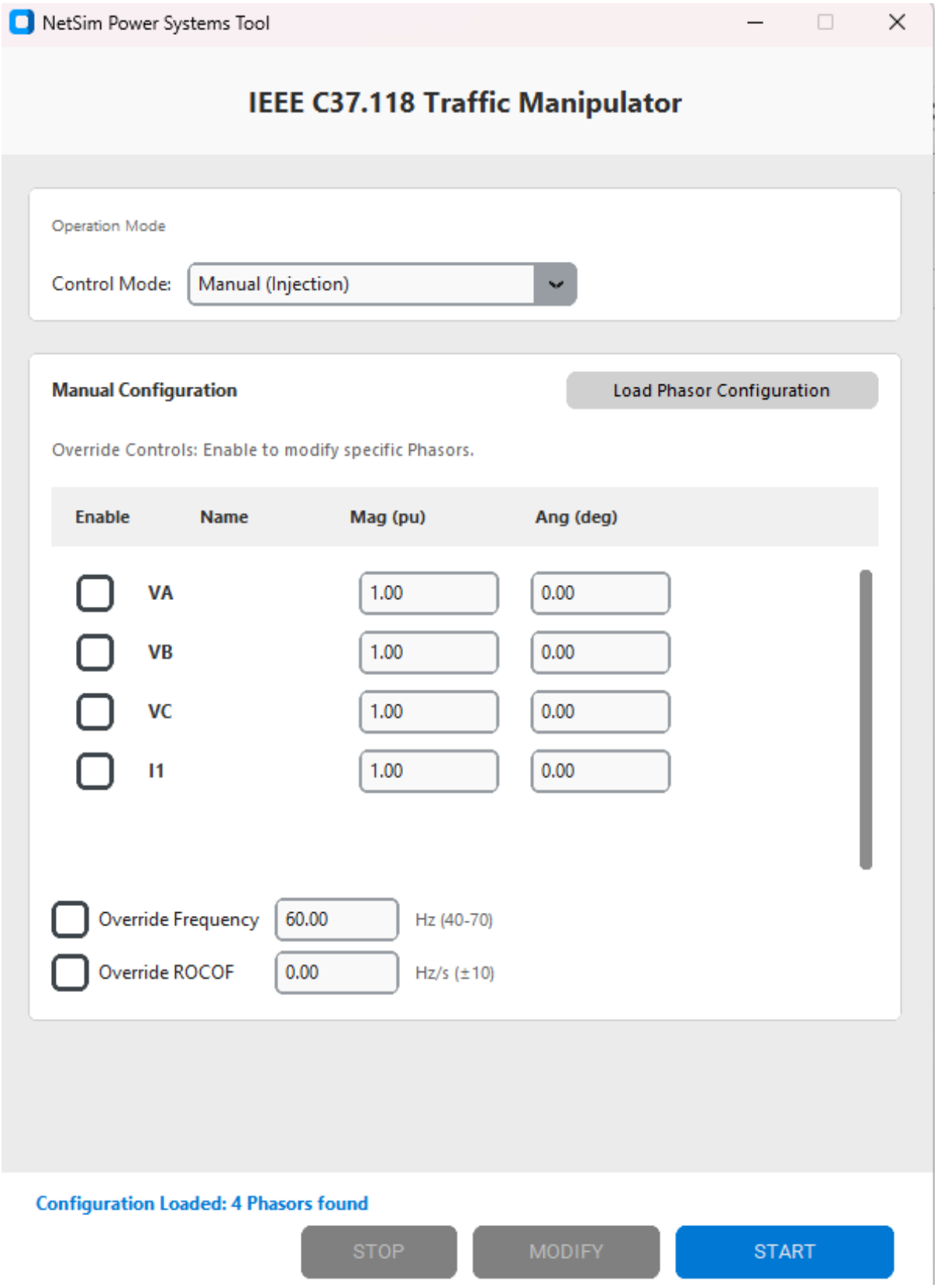

Figure 6-22: *Manual Mode window once after phasor information received.*

- Only after seeing this confirmation can you proceed to attack injection in manual or automatic mode.

Step 8: Data Visualization (System 3)

- The OpenPDC dashboard was observed for immediate deviations. In the below example, we have performed a random noise attack.

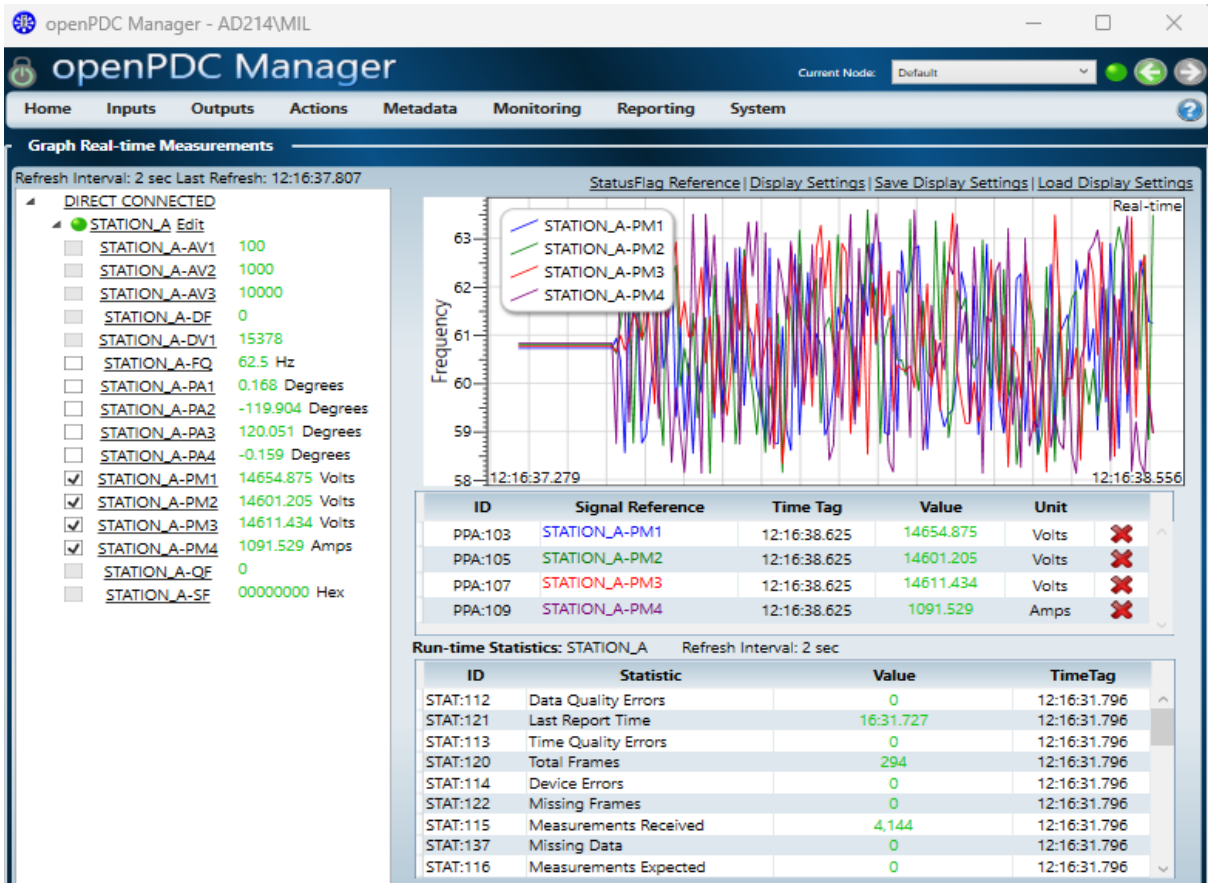

Figure 6-23: *openPDC Manager real-time monitoring view displaying IEEE C37.118 synchrophasor.*

- Screenshots of the real-time graph were captured as evidence of the attack's successful execution.

6.1.6 Results and Analysis

To check the impact of the cyber-threats, the system was monitored under two distinct conditions: baseline state and a multi-vector attack scenario.

Normal Operation

The system was first operated without any active attack agents. The OpenPDC "Real-Time Measurements" dashboard confirmed the successful reception of the IEEE C37.118 data stream from the Python simulator.

6.1.7 Without Attack

We observe that the data is of constant value from pypmu simulator:

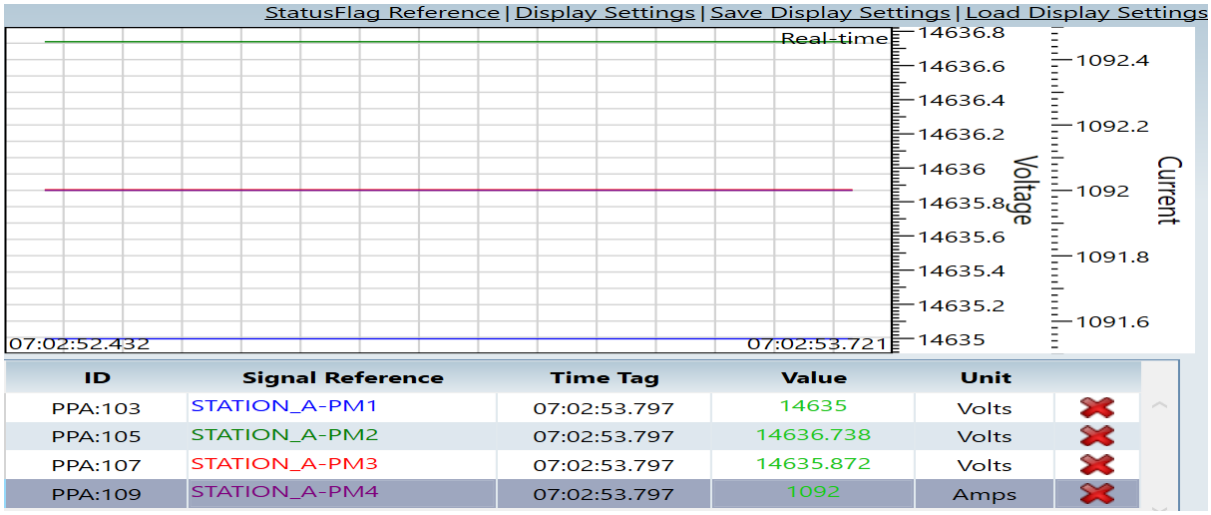

Figure 6-24: *Plots without attack.*

- **Voltage Stability:** The voltage magnitude across the monitored phases (PM1, PM2, PM3) held steady at approximately 14.6 kV (ranging from 14,635 V to 14,636.7 V).
- **Current Load:** The current measurement (PM4) remained constant at 1092 Amps, indicating a stable load condition.

6.1.8 With Attack

Attack Impact Analysis

- The NetSim Agent was configured to inject five distinct payload modification attacks.
- This set of attacks targets the integrity of synchrophasor measurements by deliberately altering voltage and current values in different temporal patterns. Such manipulations can mimic realistic power system disturbances, making them difficult to distinguish from genuine events using simple threshold-based monitoring.

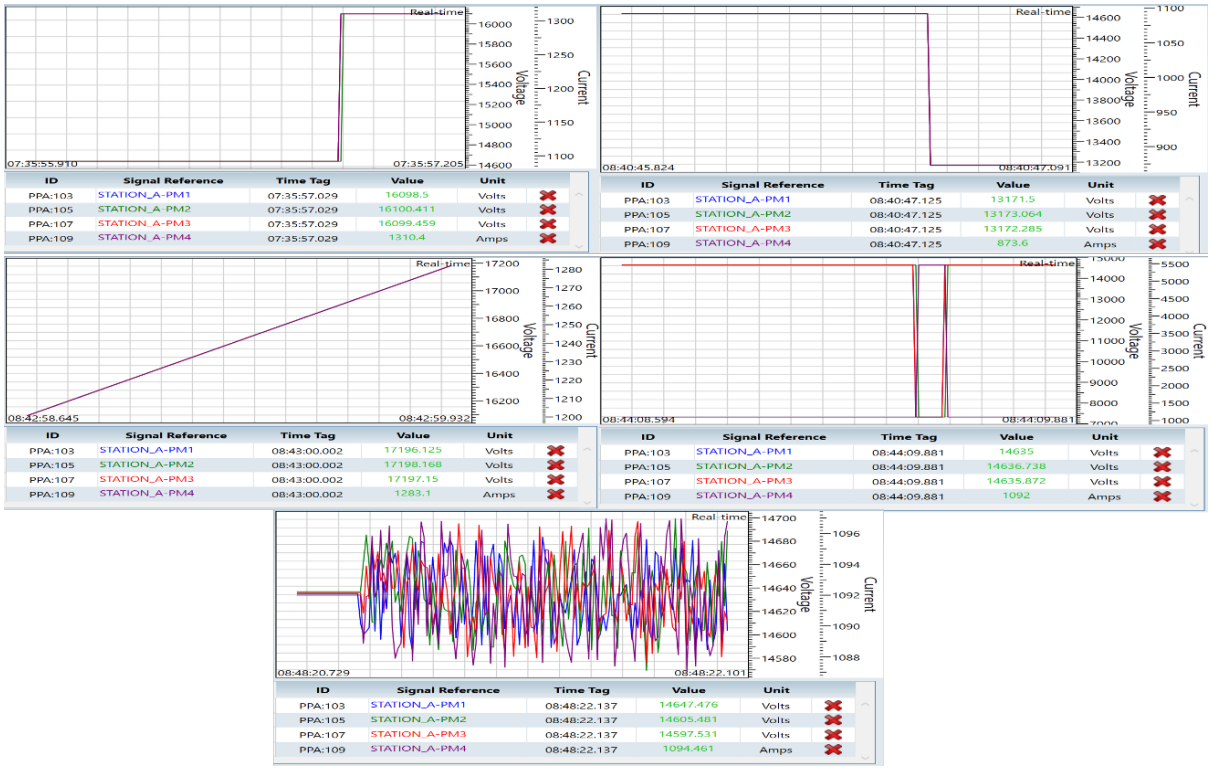

Figure 6-25: *Observed effects of data manipulation attacks on synchrophasor measurements in openPDC. Increment, decrement, ramp, pulse/surge, and random-value modifications applied to voltage and current phasors in real time.*

Increment / Decrement Manipulation

Sudden step changes in measurement values that appear as legitimate operating shifts.

Ramp Manipulation

Gradual bias injection that closely resembles slow-moving system drift or load variation.

Random Noise Injection

Low-amplitude fluctuations that degrade measurement quality while remaining protocol-compliant.

Pulse / Surge Injection

Short-duration spikes that mimic transient disturbances or fault-like events.

Frequency/ROCOF Attack:

- In this attack scenario, the adversary manipulates the frequency (and implicitly the Rate of Change of Frequency, ROCOF) values in the synchrophasor data stream without disrupting communication. Such attacks can mislead grid monitoring and protection systems by creating false indications of frequency instability or generation–load imbalance.

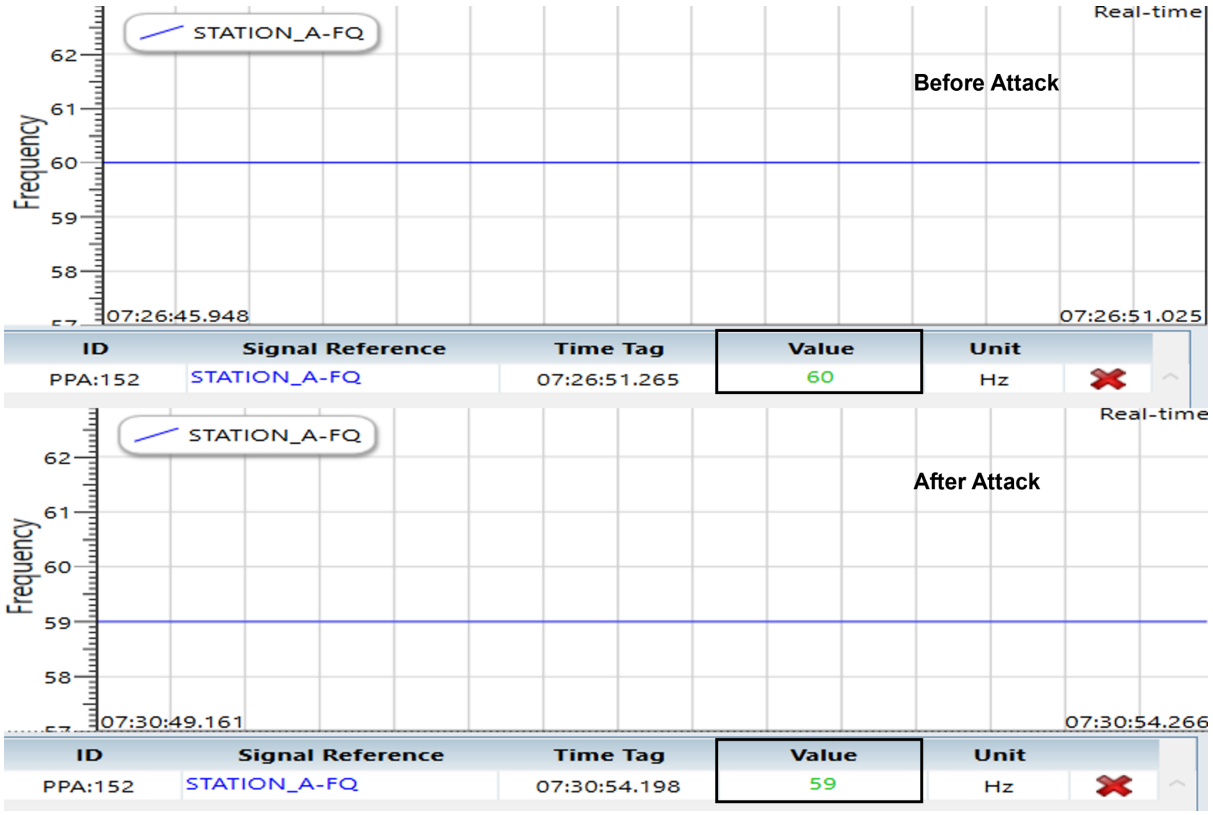

Figure 6-26: *Impact of a frequency manipulation attack on synchrophasor data: real-time frequency measurement at Station_A showing nominal operation before the attack (62.5 Hz) and a sudden deviation after the attack (59 Hz), as observed in openPDC.*

Time Synchronization Attack:

In this attack, the adversary tampers with the timestamp information (time tag) of the synchrophasor data while keeping the actual measurement values unchanged. Such manipulation disrupts temporal alignment across PMUs, which is critical for wide-area monitoring, state estimation, and event correlation in power systems.

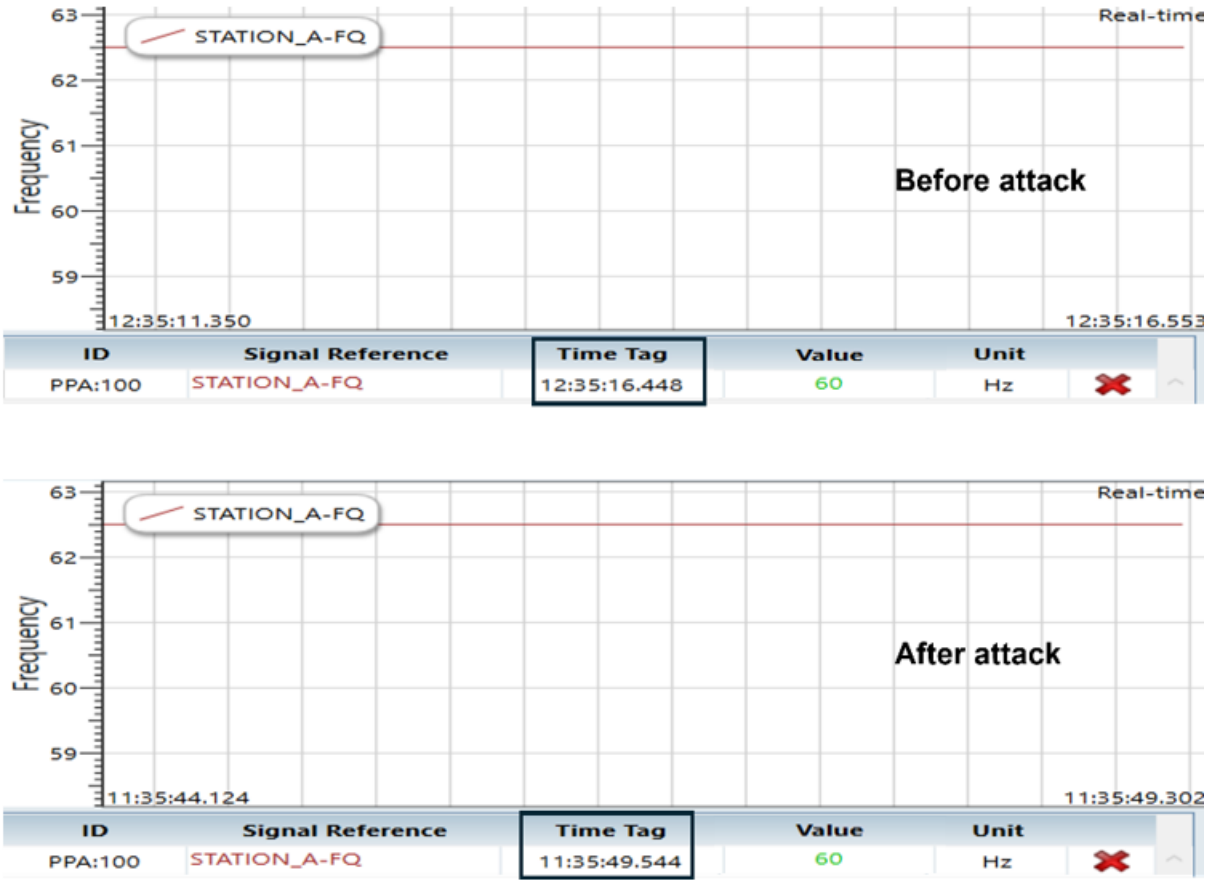

Figure 6-27: *Time tag is shifted backward by 3600 seconds, demonstrating a stealthy timestamp manipulation without altering measurement magnitude, as observed in openPDC.*

7 Common Network Attack Scenarios

NetSim Experiment Setup

For the common network attacks discussed in the manual, all will have a fixed setup in NetSim as shown in the figure below.

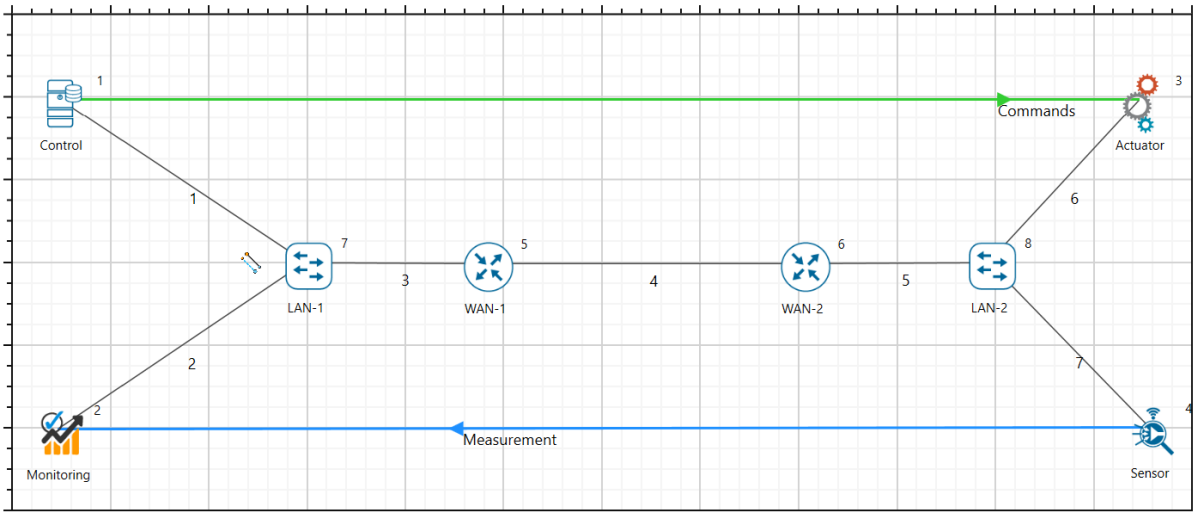

Figure 7-1: *Experimental scenario in NetSim for performing common attacks on a CPS scenario.*

Network Topology:

- **Control Centre:** Central node responsible for managing and monitoring the CPS.
- **Field Devices (Sensors/Actuators):** Nodes that provide data and execute control commands.
- **Communication Links:** Network connections between the control center and field devices, configured to simulate packet drops.
- **Attacker Node:** This will be added in each attack based on the attack type.

7.1 Delay Attack

7.1.1 Deterministic Delay Attack in Communication Link

Abstract In NetSim, an attacker exploits the deterministic delay in the communication links used for time synchronization. By introducing a subtle but consistent delay, the attacker can disrupt the time synchronization process, causing desynchronization among devices in the CPS. This can lead to inaccurate data reporting, improper load balancing, and even failures in automated control mechanisms.

Note: To perform deterministic delay attack in NetSim, there are no changes to code or scripts to be run. This can be performed within NetSim UI.

Objective In this experiment there are mainly 2 parts where an attacker exploits the deterministic nature of this delay by performing a timing attack in the (a) Links and (b) Devices:

- By performing simulation in NetSim. Here the data transferred between legitimate source and destination is virtual data generated by NetSim.
- By performing emulation in NetSim. Here the data between legitimate source and destination are generated by real devices by running python scripts/real application to generate real-data and transmitted over electrical protocol via NetSim.

Parameters and Settings

- In the network setup, all the node parameters remain unchanged except the name of the node.
- Set the traffic to a desired rate since we are not focusing here on the data rate.
- Configure Deterministic Delay attack: Introduce Malicious Router:
- Add an attacker node to the network such that the CPS devices communicate over the attacker node as shown in the figure below.

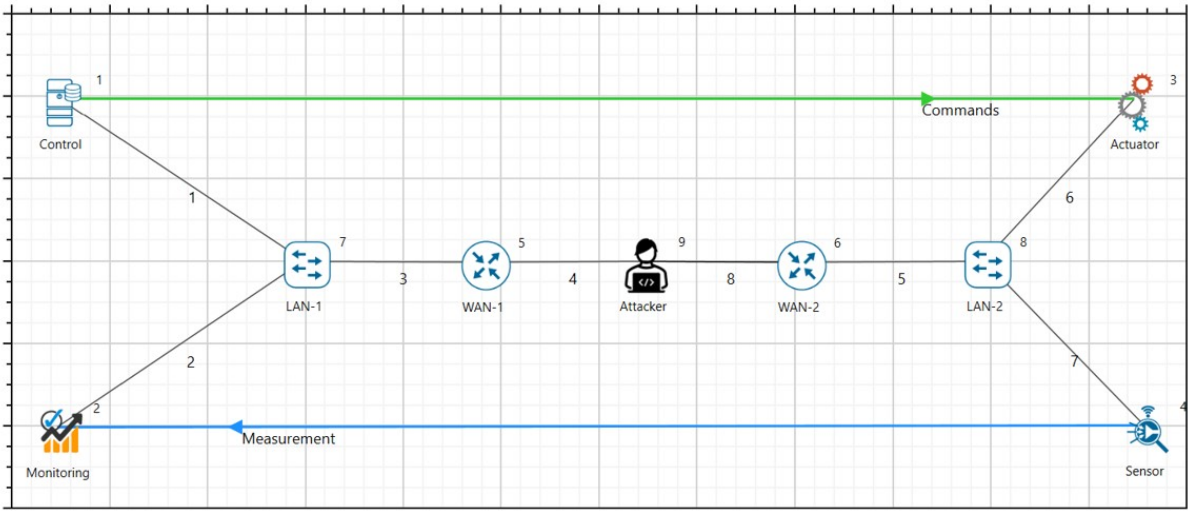

Figure 7-2: *Adding an attacker node to perform delay attack.*

- Add an additional Processing delay in the node (e.g., an extra 1s).

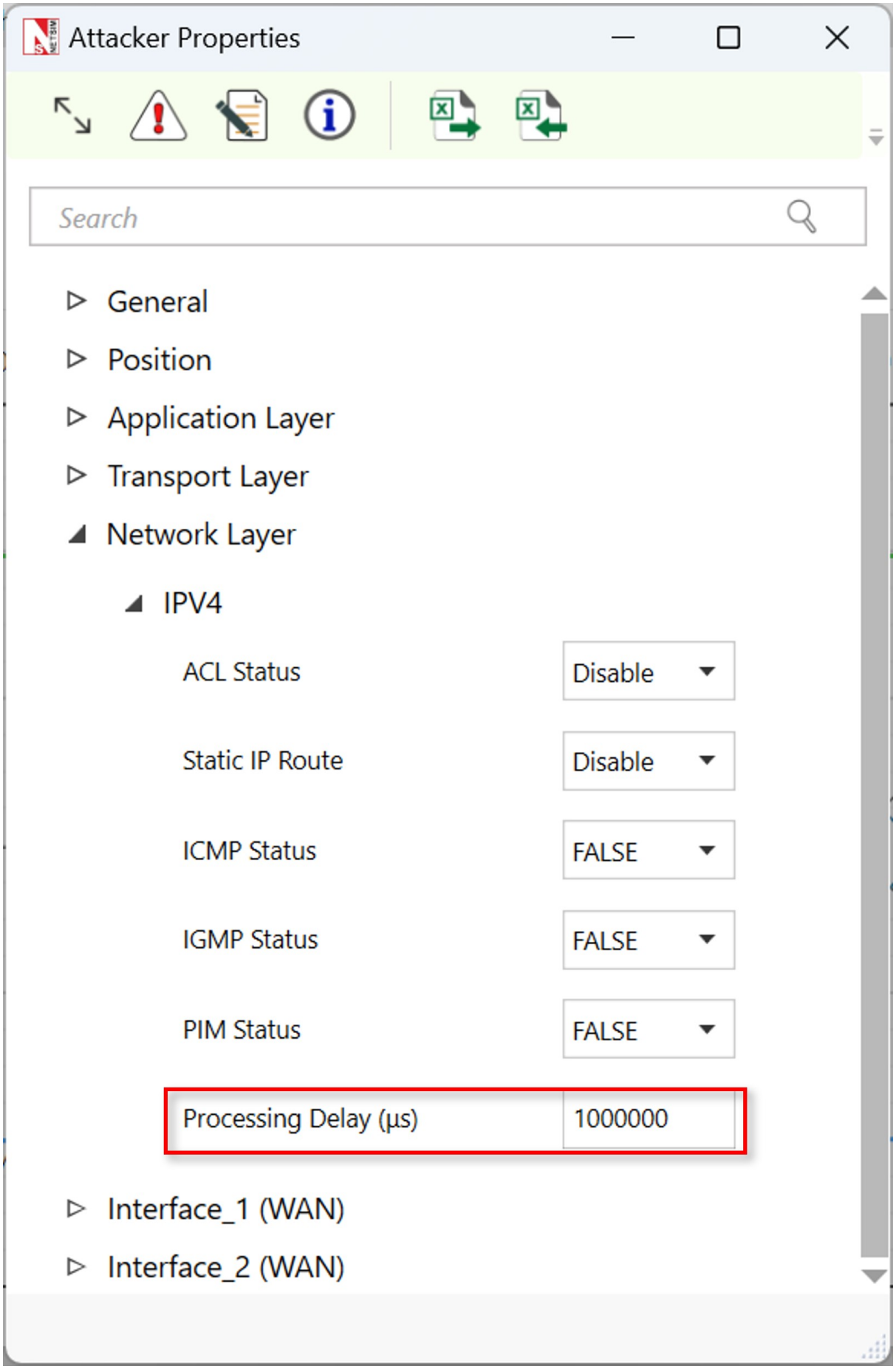

Figure 7-3: *Setting Deterministic Processing Delay in the attacker node.*

- Add an additional propagation delay in the links where router is connected to the attacker node.

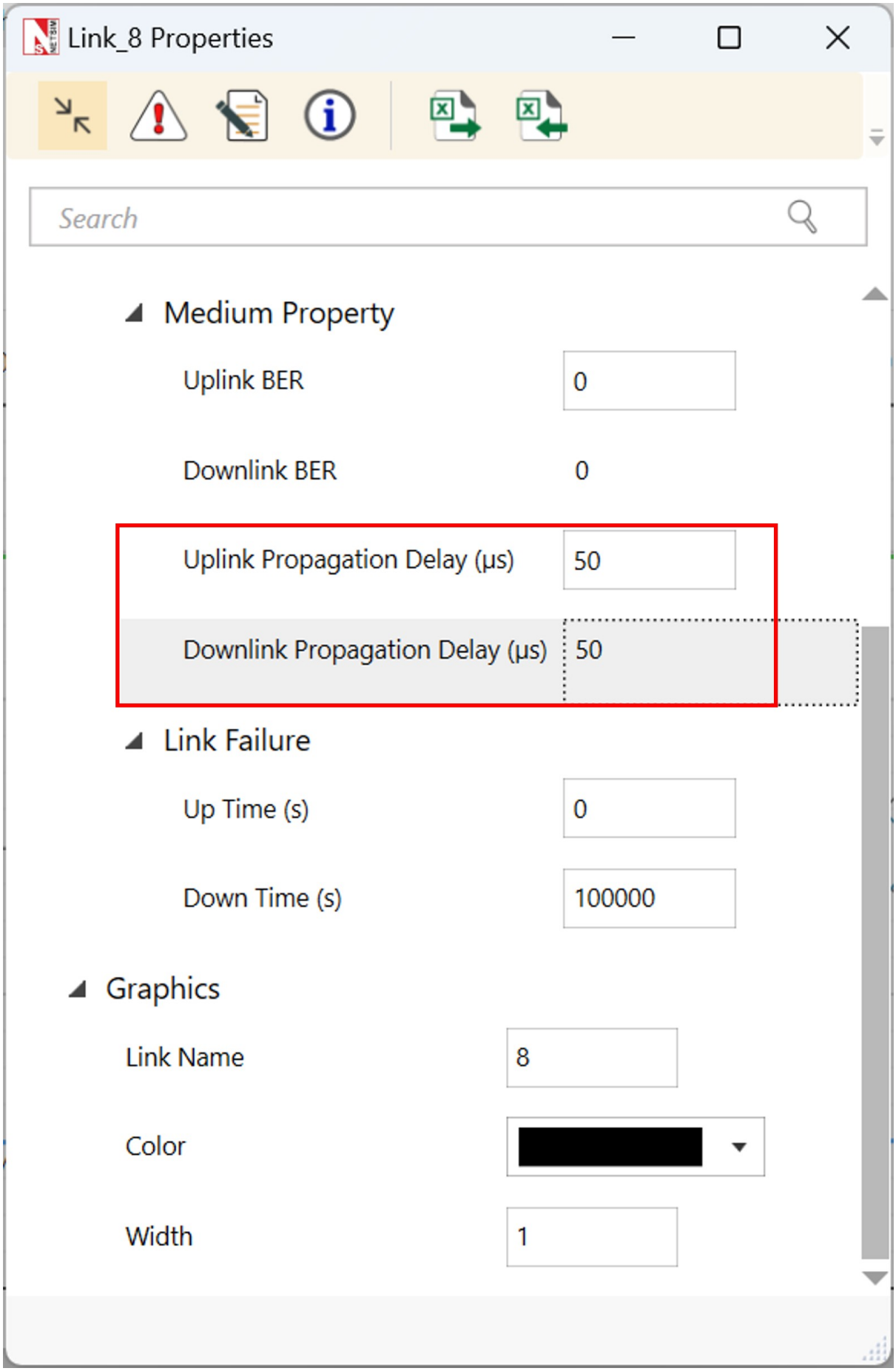

Figure 7-4: *Setting propagation delay in the links connected to the attacker node.*

Step-by-step Procedure

- We have shared the sample file, where users can download and import where NetSim is installed.
- The procedure to download and import to NetSim is given in the last section of this manual.

Results and Analysis

- Once you import the experiment, open and run the sample.
- You will observe the additional deterministic delay in the results dashboard.

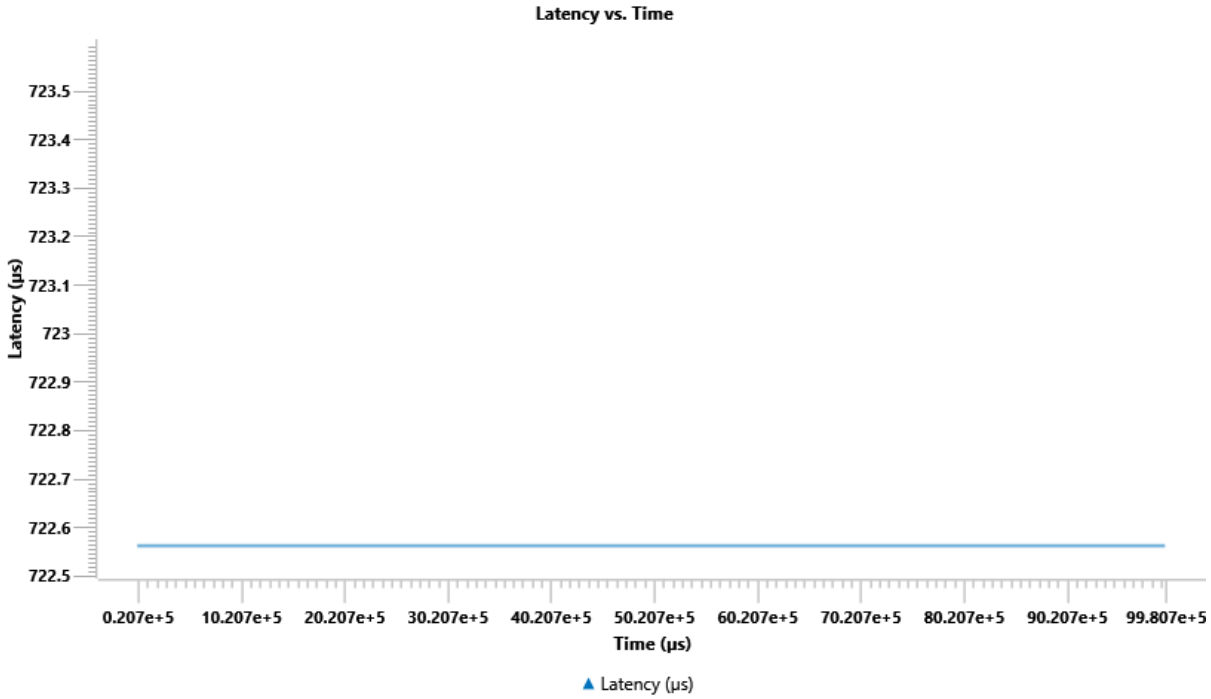

Figure 7-5: *This plot is without attack, where we see a minimum delay throughout the simulation.*

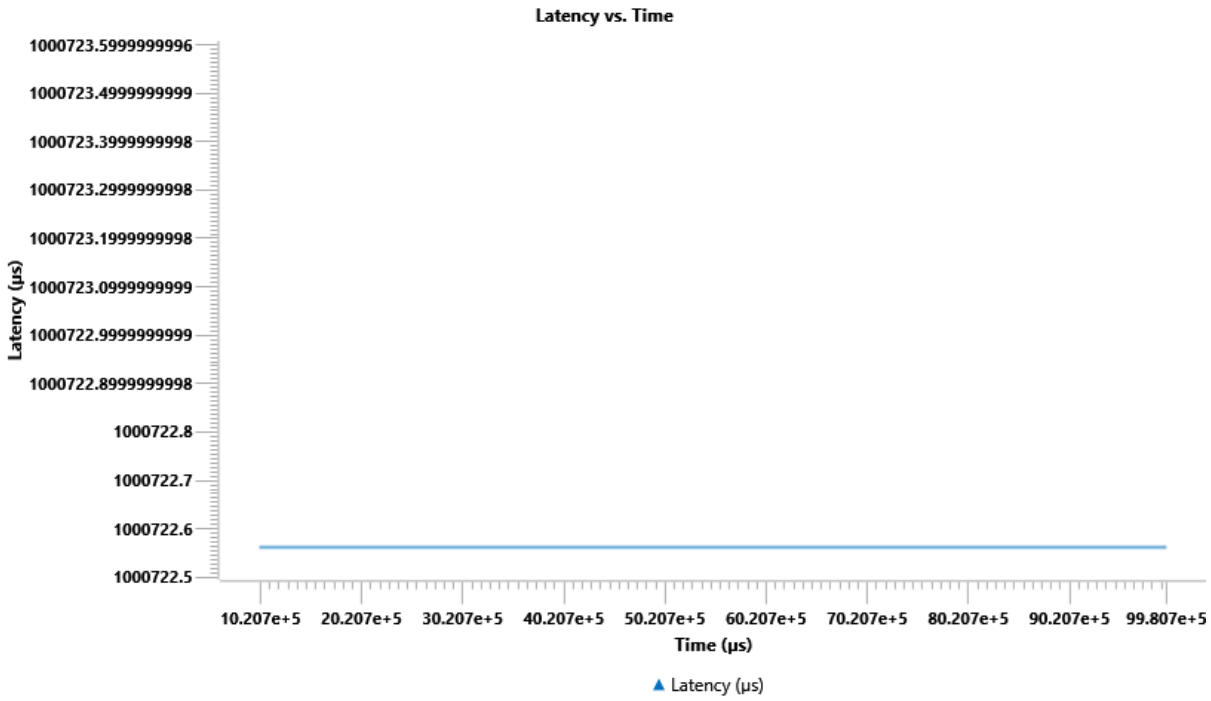

Figure 7-6: *This plot is after adding an additional processing delay of 1s.*

7.1.2 Stochastic Delay Attack in Communication Link

Abstract In NetSim, an attacker exploits the stochastic nature of delays in communication links used for time synchronization. By introducing subtle, random delays, the attacker can disrupt the time synchronization process, causing desynchronization among devices in the CPS. This can lead to inaccurate data reporting, improper load balancing, and failures in automated control mechanisms.

Note: To perform stochastic delay attack in NetSim, we have performed changes to code to run this attack to randomize the delay. Users will require Standard version of NetSim.

Parameters and Settings In the network setup, all the node parameters remain unchanged except the name of the node. And all the settings are per Deterministic delay attack scenario.

Code Changes The given function below will be used for adding random delay in the attacker node as a processing delay.

```
long int stochastic_delay(unsigned int seed) {
    int delay = rand() % 1000 + 1; // Generate a random delay
                                   // between 1 and 1000 microseconds
    return(delay);
}
```

Results and Analysis

- Once you import the experiment, open and run the sample.
- You will observe the additional stochastic delay in the results dashboard.

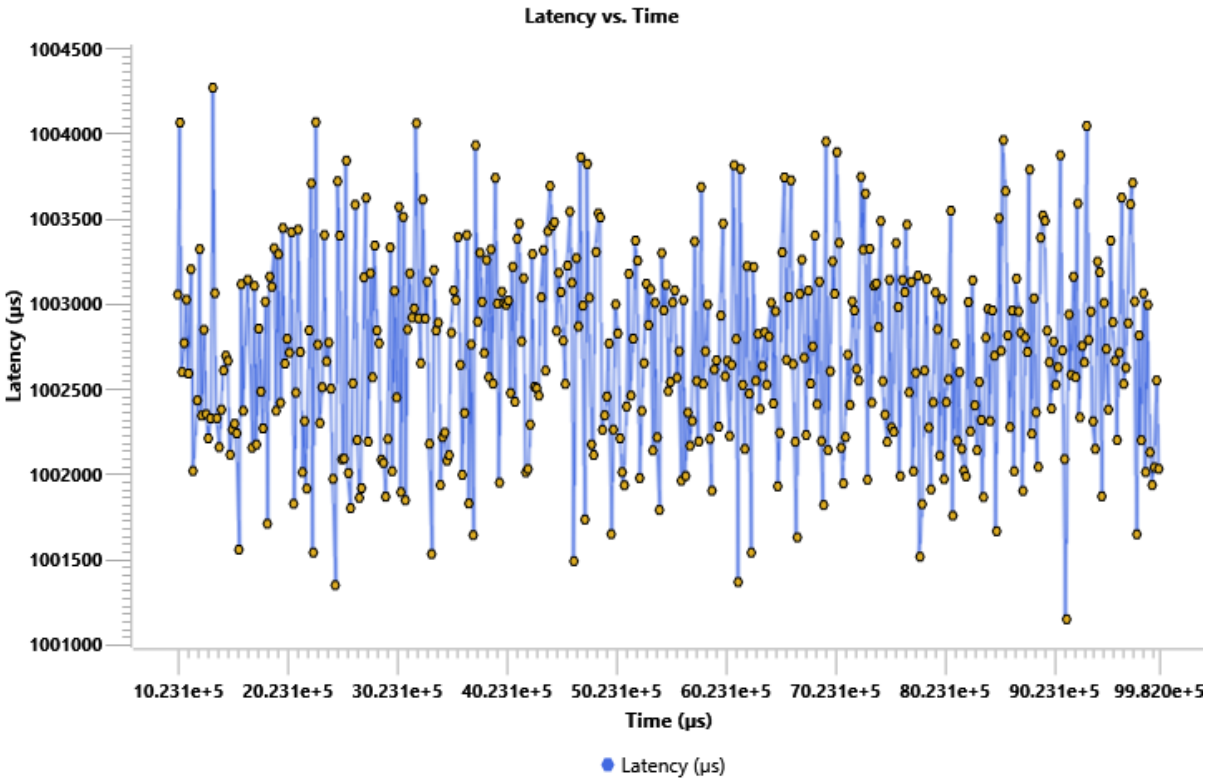

Figure 7-7: *Stochastic Delay attack by the attacker node as processing delay.*

7.2 Communication Link Failure Attack (TCP and UDP)

7.2.1 Objective

The experiment aims to simulate and analyse the effects of communication link failures on both TCP and UDP protocols within a Cyber-Physical System (CPS). The experiment will highlight the differences in how these protocols handle link failures and the resulting impact on system operations.

Note: To perform this attack in NetSim, there are no changes to code or scripts to be run. This can be performed within NetSim UI.

7.2.2 Experiment Setup

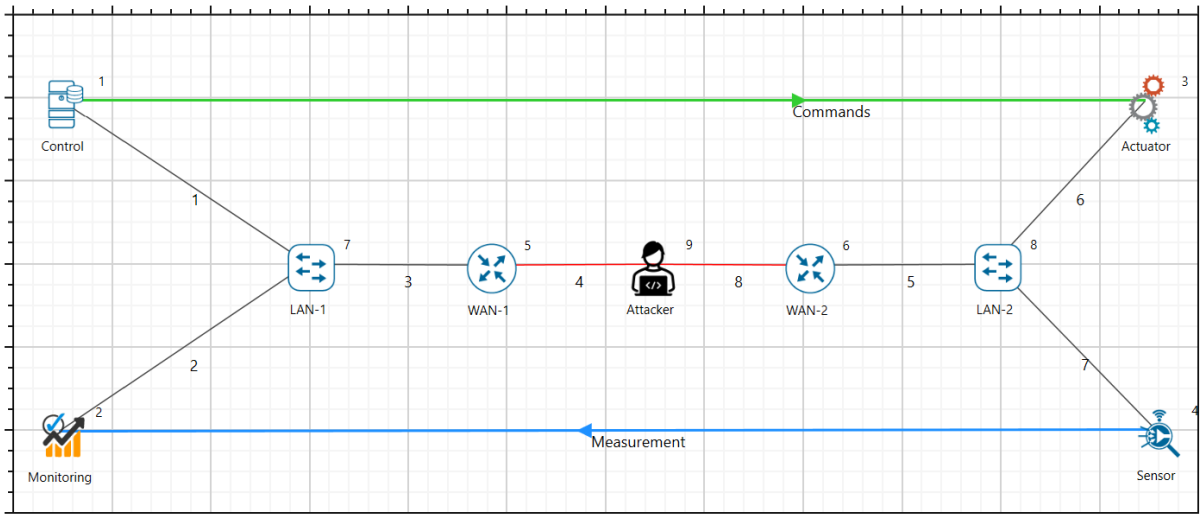

Figure 7-8: *Simulating link failure attack, by introducing an attacker in the communication link between Control Center and Field Devices.*

- **Control Center:** Central node for monitoring and controlling the CPS.
- **Field Devices (Sensors/Actuators):** Nodes that collect data and execute control commands.
- **Communication Links:** WAN connections between the control center and field devices, using both TCP and UDP protocols.
- **Attacker Node:** Node configured to simulate communication link failures.

7.2.3 Procedure

- **Setting Up the Network:**
- Define the CPS network environment.
- **Add and Configure Nodes:**
  - **Control Center:** Add as the central server node.
  - **Field Devices:** Add nodes representing sensors and actuators.
  - **Attacker Node:** Include a node to simulate communication disruptions.
- Establish Communication Links.
- **Protocol Configuration:**
  - **TCP Configuration:** Configure TCP links to prioritize reliability, setting appropriate parameters for window size, acknowledgment, and retransmission.
  - **UDP Configuration:** Configure UDP links for minimal latency, focusing on packet delivery without guarantee of order or reliability.

7.2.4 Simulating Communication Link Failures

Normal Operation Phase

- Run the simulation under normal conditions to establish a baseline for TCP and UDP communication.

Introduce Link Failures: TCP Application

- Set the application Transport Layer protocol to TCP.
- In the links that are connected to the attacker node, configure link failure duration.

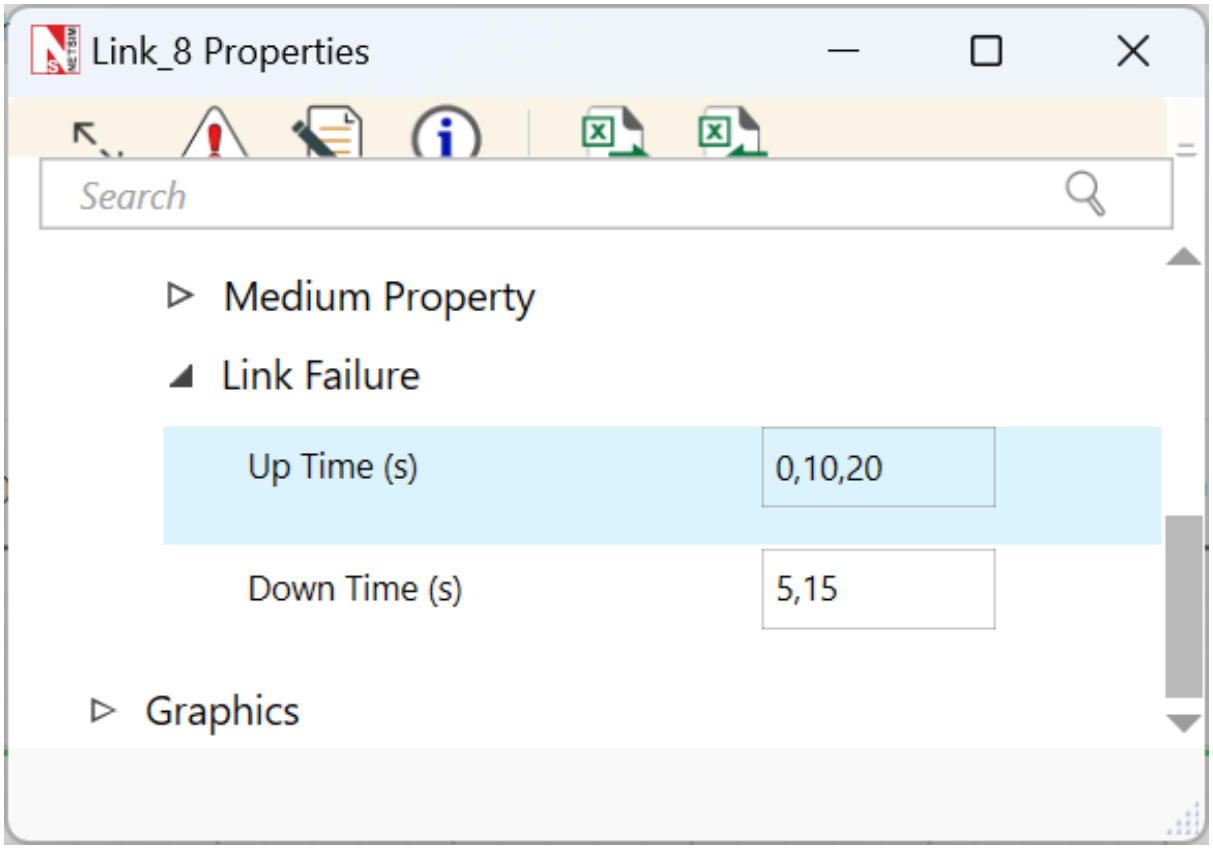

Figure 7-9: *Setting the link failure parameter per the attacker requirement for the frequent intervals of time by setting the 'Up time and Down time' of the links that are connected to the attacker node. Here in this example we have considered 5s interval i.e., every 5s the link will go up and then again down.*

UDP Link Failures

- Similarly, configure the UDP as Transport Layer Protocol in the application properties.
- Similar adjustment for failure parameters to observe the impact on UDP traffic.

7.2.5 Results

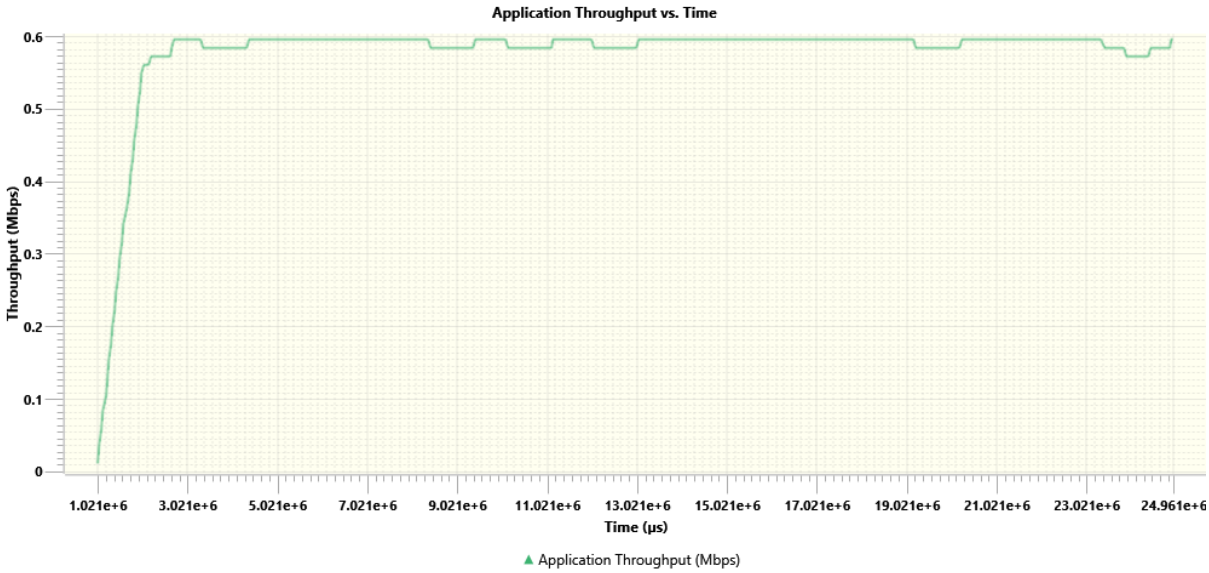

Figure 7-10: *This is the plot with no attackers enabled in the scenario. Where we can see the healthy traffic throughout the simulation.*

Normal Operation Phase

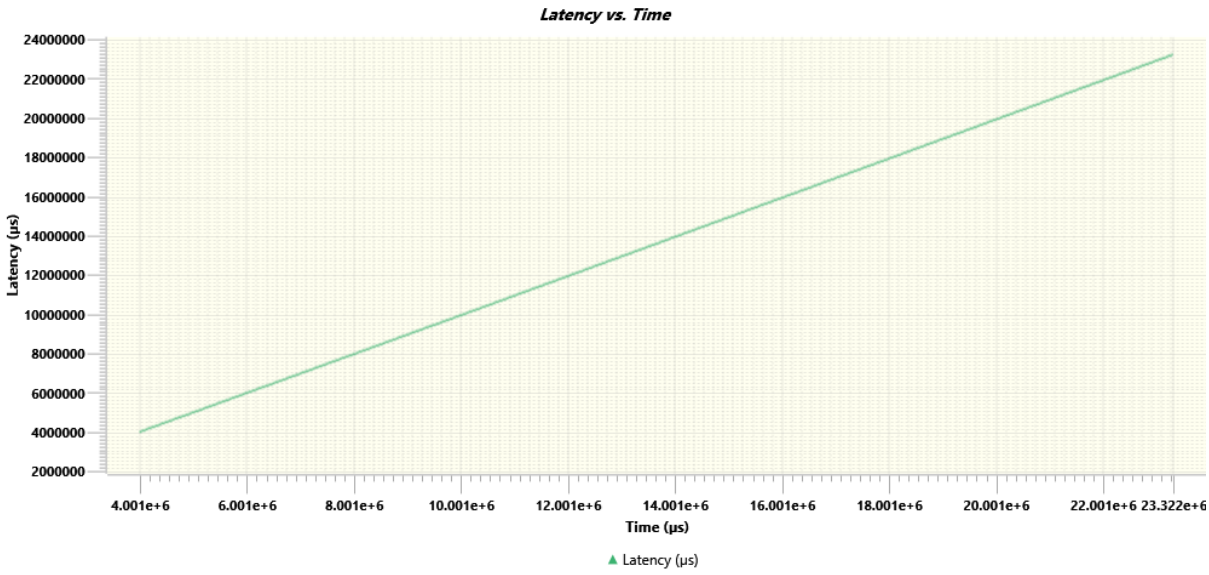

Figure 7-11: *High latency in a TCP application due to link failure: Increased packet drop due to link failure leads to frequent retransmissions and longer recovery times, causing delays in data delivery.*

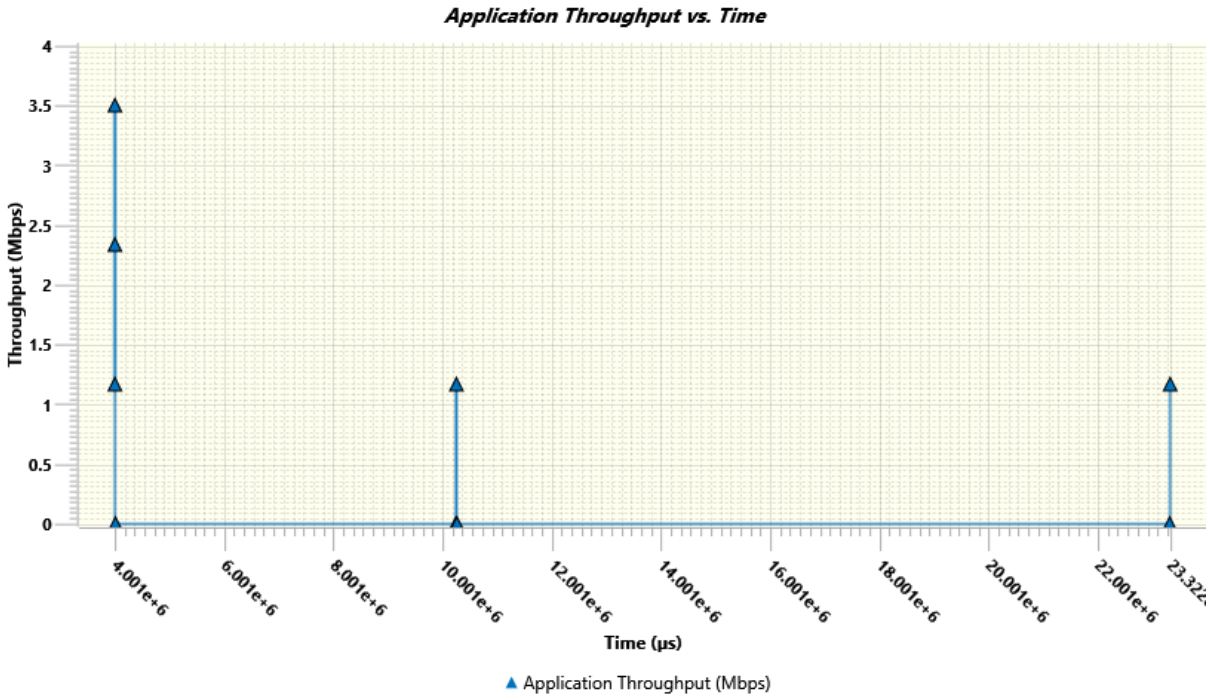

Figure 7-12: *TCP traffic effected by the link failure attack. There were repeated attempts for 3 way handshake but failed due to packet drop.*

Communication Link Failure for TCP Traffic

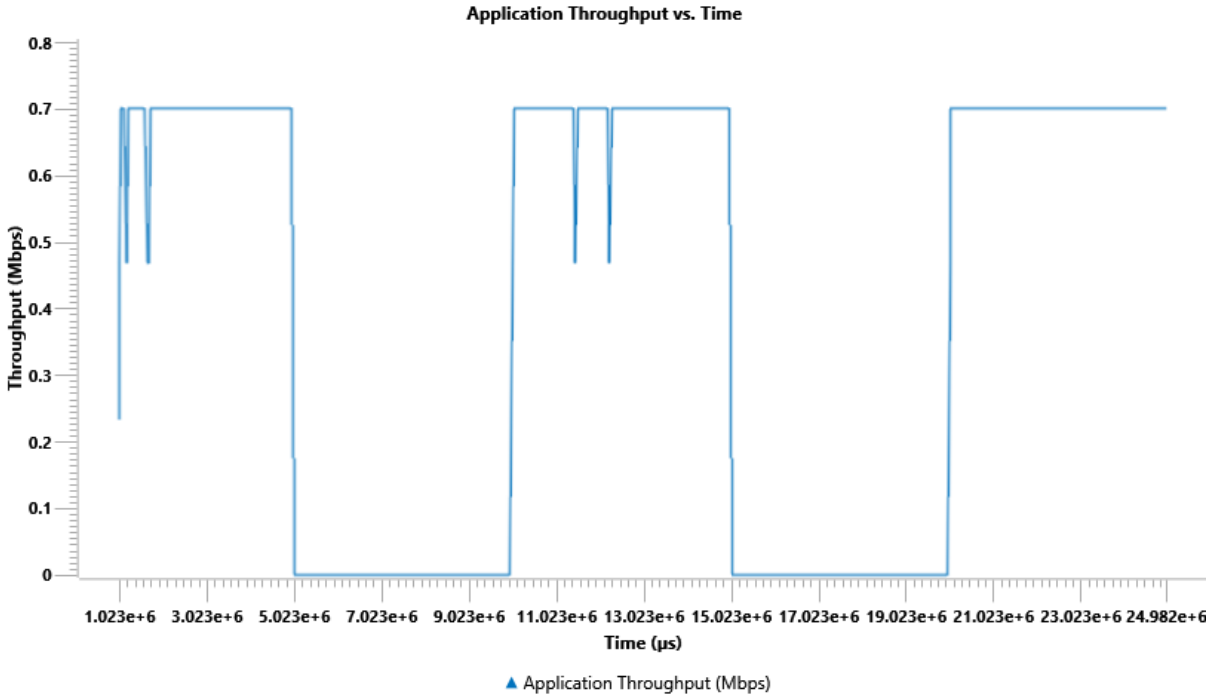

Figure 7-13: *UDP traffic affected by link failure attack.*

Communication Link Failure for UDP Traffic

7.3 Denial-of-Service (DOS) Attacks and DDoS

7.3.1 Data Flooding Attack

Objective The goal of this experiment is to simulate a data flooding attack in a Cyber-Physical System (CPS) using NetSim. This experiment aims to demonstrate the impact of excessive data traffic on the performance and reliability of the CPS, highlighting potential vulnerabilities.

Note: To perform data flooding attack in NetSim, there are no changes to code or scripts to be run. This can be performed within NetSim UI.

Setting Up the Network

- Define the simulation environment to reflect a CPS setup.
- **Add and Configure Nodes:**
  - **Control Center:** Add a node to represent the main server.
  - **Field Devices:** Add nodes representing various sensors and actuators in the CPS.
  - **Attacker Node:** Include a node to simulate the data flooding attack.
- Connect the control center to the field devices using WAN links.
- Configure the communication links with standard parameters (bandwidth, delay, etc.)

Simulating Data Flooding Attack in the CPS Network **Configuring Normal Operation:**

- Normal Traffic Configuration
- Set up regular data traffic patterns between field devices and the control center.
- Define normal packet sizes and inter-packet arrival times for both sensors (sending data) and actuators (receiving commands).
- **Baseline Measurement:**
- Run a simulation to establish baseline performance metrics, including latency, throughput, and error rates under normal traffic conditions.
- Running the Attack Simulation
- Start the simulation with the deactivated attacker node.
- Monitor the network for signs of congestion, such as latency, packet loss, and throughput.

Simulating the Data Flooding Attack:

- **Attack Configuration**
- The attacker node has to generate a high volume of traffic, targeting the communication links. Configure the attacker node to send large packets at a high frequency, overwhelming the network.
- **Flooding Parameters:**
- Set parameters such as packet size and inter-packet arrival time to maximize congestion. Optionally, vary these parameters to observe the effects of different levels of flooding intensity.
- Running the Attack Simulation
- Start the simulation with the attacker node active.
- Monitor the network for signs of congestion, such as increased latency, packet loss, and reduced throughput.

Results **Healthy traffic without data flood attack for both TCP and UDP:**

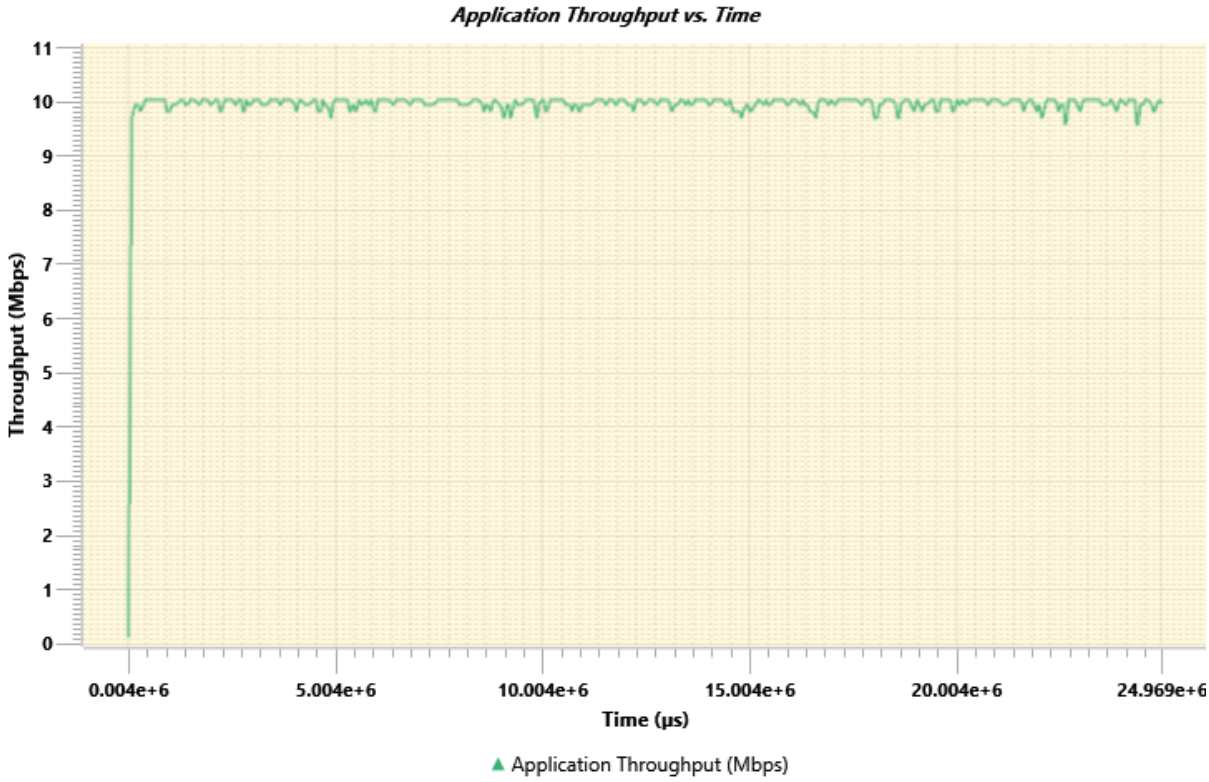

Figure 7-14: *Healthy traffic without data flooding attack. We can see that the throughput of all the commands is generated at the rate of 10 Mbps and approximately the same rate is achieved.*

Data flooding attack for random 5s interval for UDP traffic:

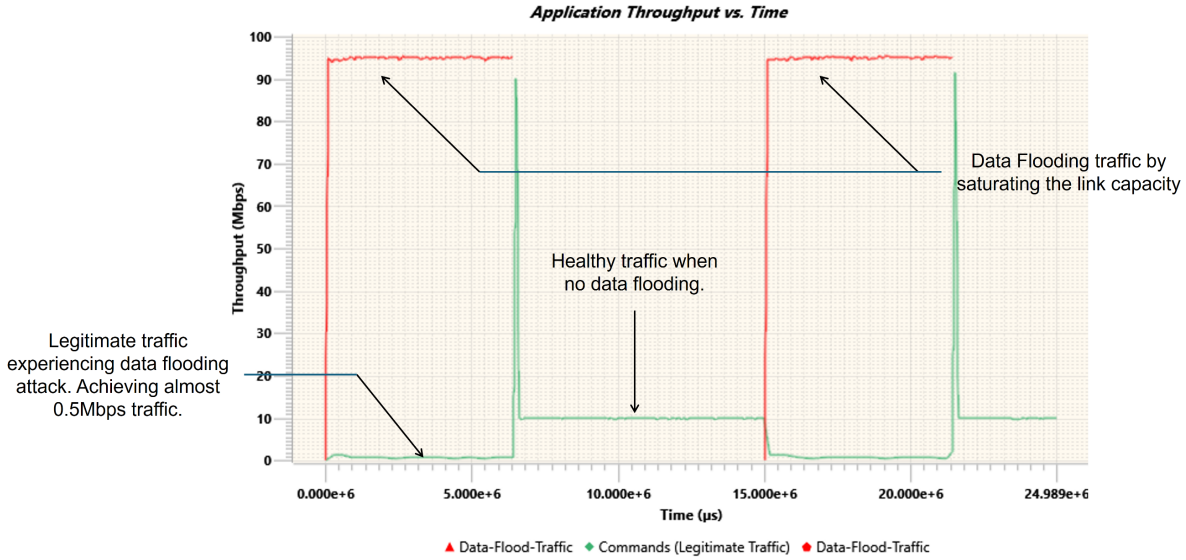

Figure 7-15: *NetSim plot showing how data flooding attack can affect the legitimate traffic flow of a CPS model. In this attack the attacker is attacking for every random interval of time and making the system experience low rate traffic flow. We can see that during the attack the traffic is affected due to saturation of the bandwidth by attacker node.*

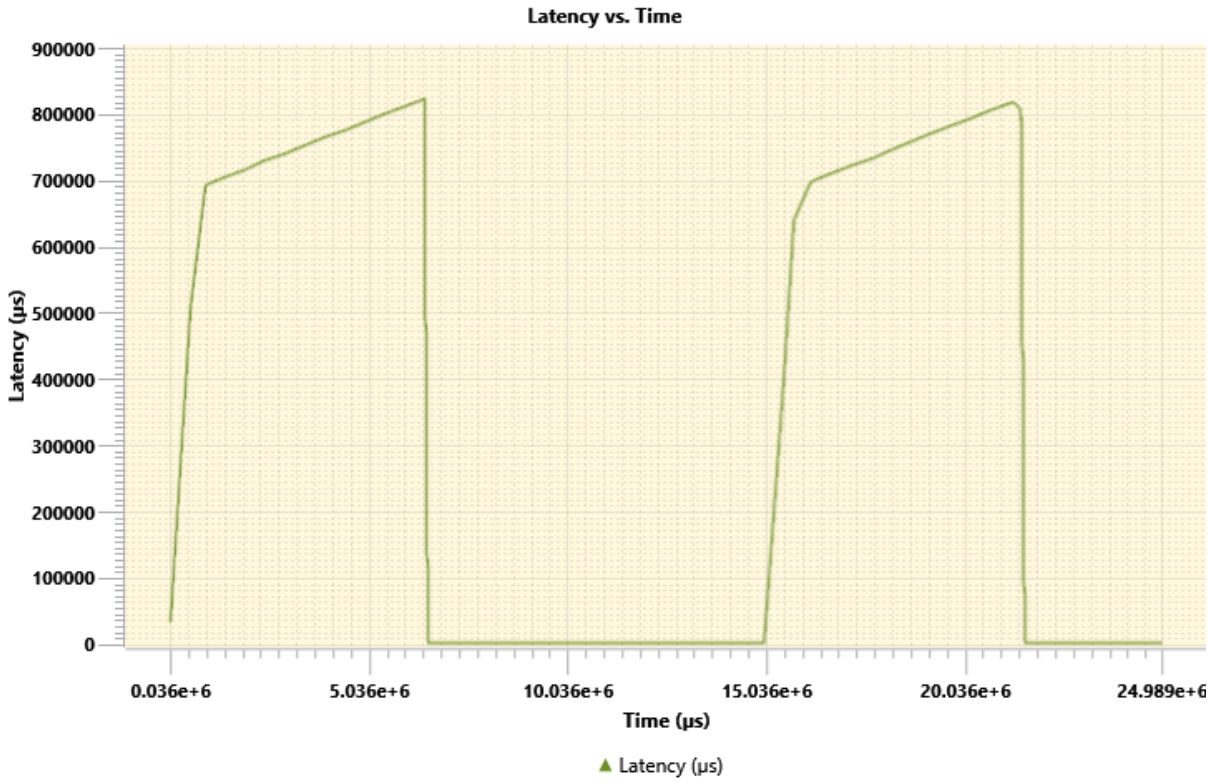

Figure 7-16: *Similarly observing the latency due to data flooding attack. Whenever there is a data flooding attack, the latency in the traffic shoots up until the attack is stopped.*

7.3.2 Probabilistic Packet Drop Attack

Objective The objective of this experiment is to simulate the impact of probabilistic packet drops in the communication links of a Cyber-Physical System (CPS) using NetSim. The experiment aims to demonstrate how random packet losses can affect the reliability and performance of the system, and how different protocols handle these losses.

Note: To perform probabilistic packet drop attack in NetSim, there are no changes to code or scripts to be run. This can be performed within NetSim UI.

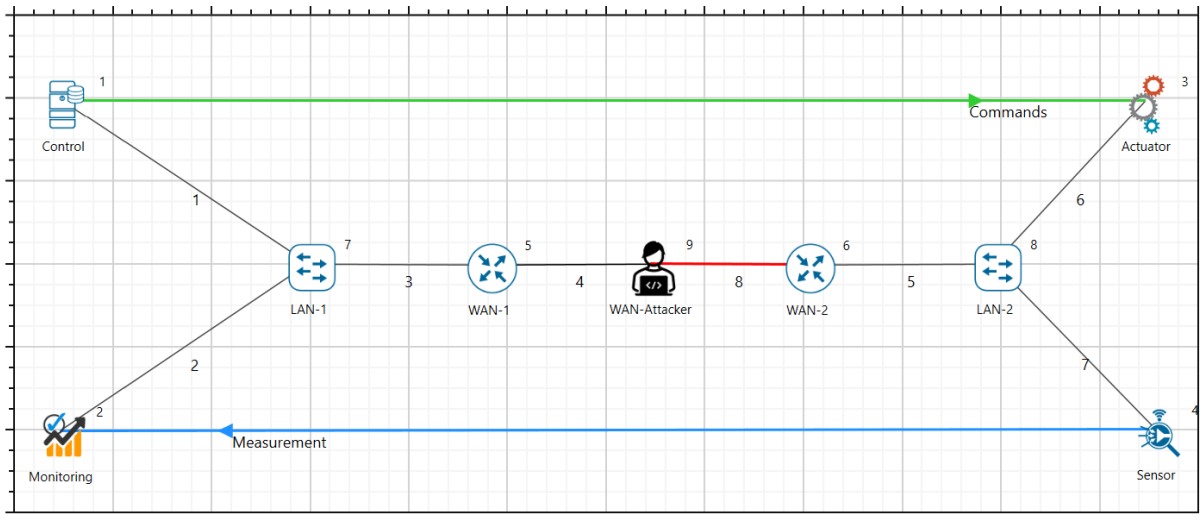

Figure 7-17: *The attacker attacks the links and induces the high error rate in the links.*

Network Topology

Setting Up the Network

- Set up the CPS environment by defining the nodes and their roles.
- **Add and Configure Nodes:**
  - **Control Center:** Add and configure as the central server.
  - **Field Devices:** Add sensors and actuators, configuring them to communicate with the control center.
  - **Attacker Node (optional):** If including an attacker node, configure it to simulate conditions leading to packet drops.
- Connect the control center to the field devices using WAN links.
- Set up the links for both TCP and UDP traffic, as applicable.

Simulating Probabilistic Packet Drop Attack in the CPS Network **Configuring Normal Operation:**

- **Baseline Configuration**
- Set up normal communication conditions, defining regular packet sizes and inter-packet intervals for data transmission.
- Run a preliminary simulation to establish baseline metrics such as latency, throughput, and packet delivery ratio.

Introducing Probabilistic Packet Drops:

- **Setting Up Packet Drop Conditions**
- Configure the communication links to introduce probabilistic packet drops.
- Specify the probability of packet drops (e.g., 5%, 10%) to simulate different levels of network unreliability.
- If using an attacker node, program it to generate conditions that lead to packet drops, mimicking a denial-of-service or interference attack.
- **Protocol-Specific Configurations:**
  - **TCP:** Set parameters that allow TCP to handle packet drops, such as retransmission timeouts and congestion control settings.
  - **UDP:** Since UDP does not handle packet recovery, observe the direct impact of packet loss on data delivery.
- **Running the Simulation**
- Start the simulation with the packet drop conditions in place.
- Monitor the network for performance changes, including increased latency, reduced throughput, and changes in packet delivery ratios.

Result Analysis Normal Operation:

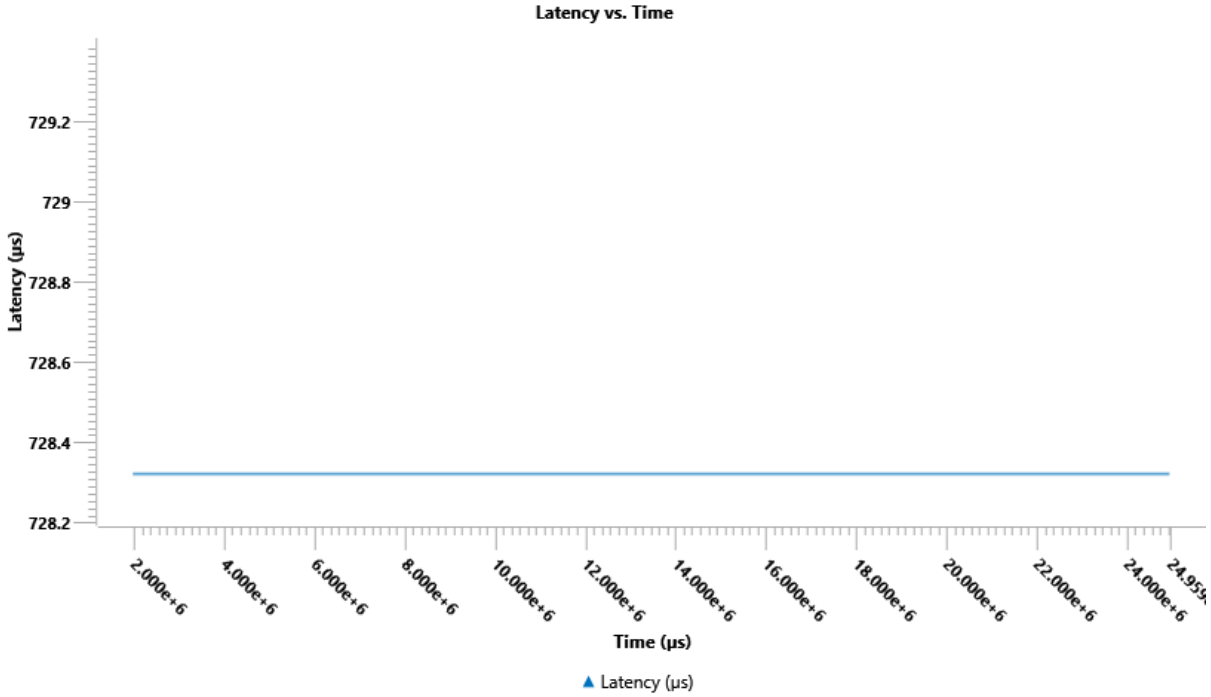

Figure 7-18: *A minimum latency in the network. This plot with no attackers in the scenario.*

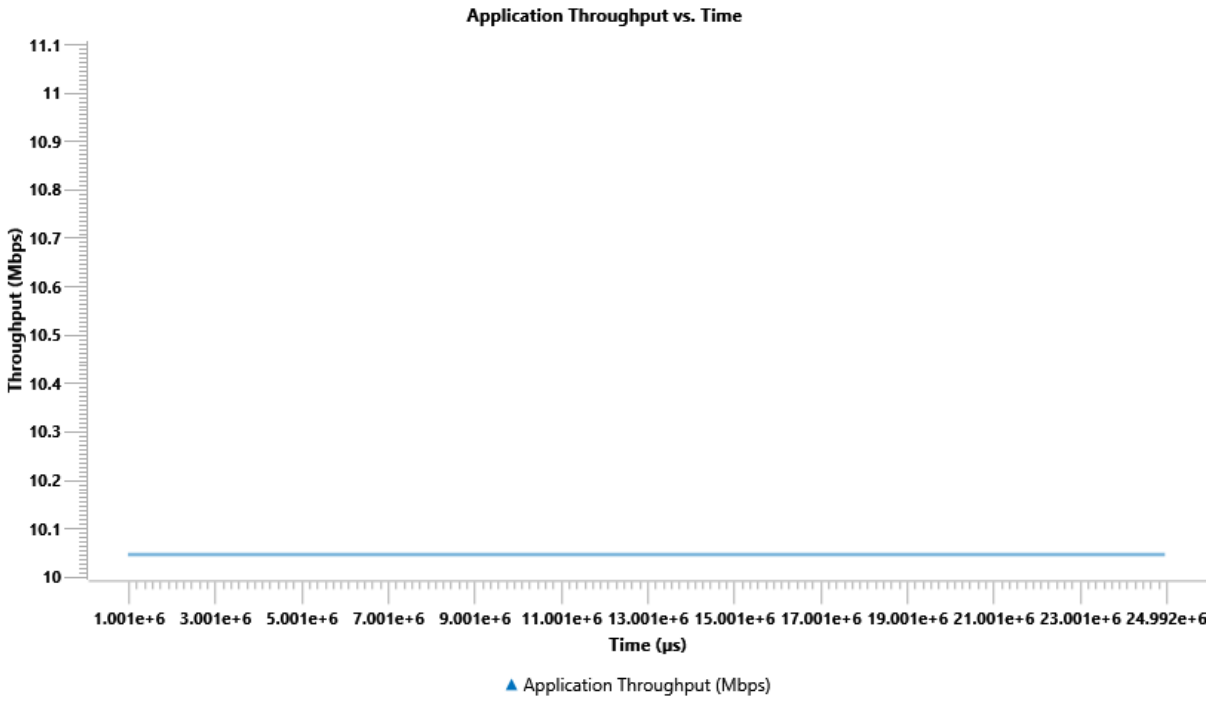

Figure 7-19: *Throughput plot of 10 Mbps which matches the generation rate of commands from the control center.*

Probabilistic packets drop attack on a TCP based command traffic:

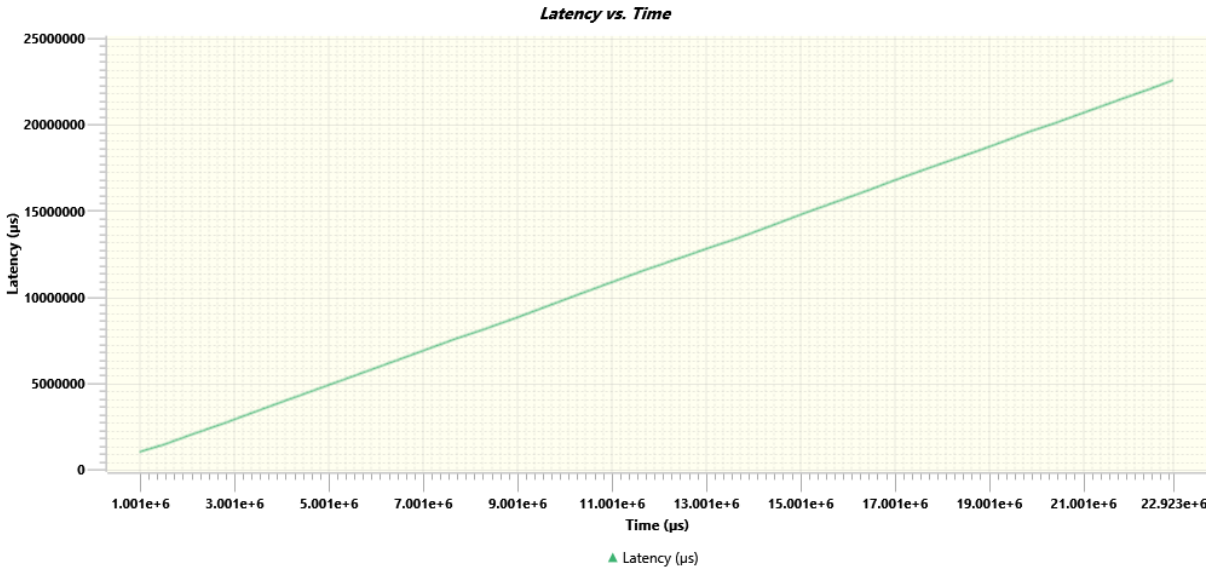

Figure 7-20: *High latency in a TCP application due to high packet drop: Increased packet loss leads to frequent retransmissions and longer recovery times, causing delays in data delivery.*

Probabilistic packets drop attack on a UDP based command traffic:

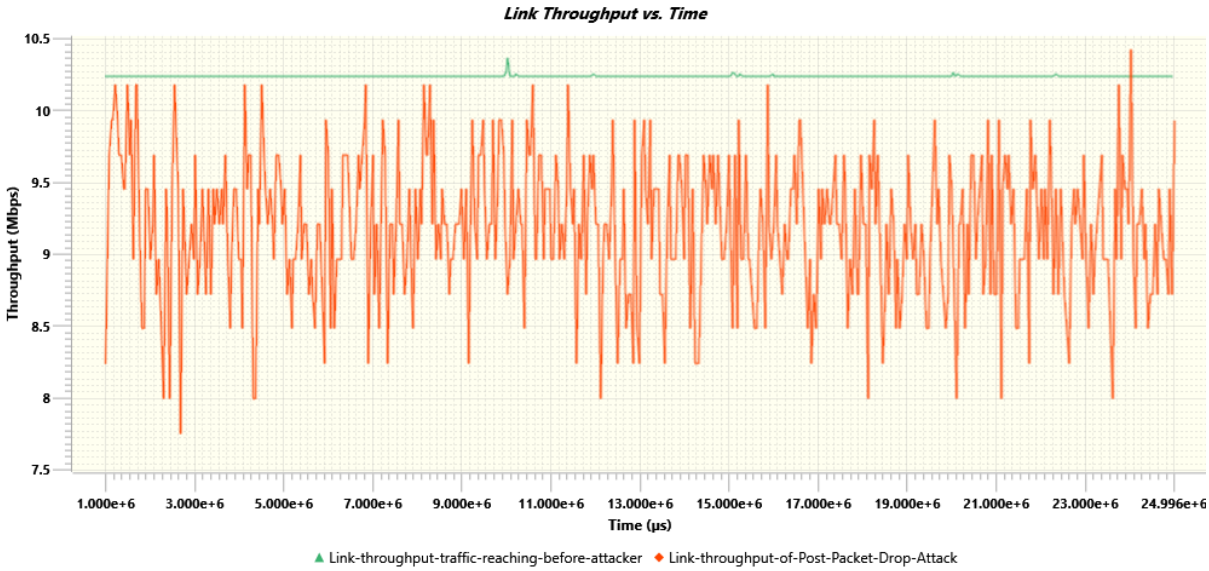

Figure 7-21: *In this plot, we can observe the red line/graph that there was a huge packet drop once after the traffic started flowing through the attacker node. The green line/graph indicates the traffic before attack which is achieving almost 100% throughput without being attacked.*

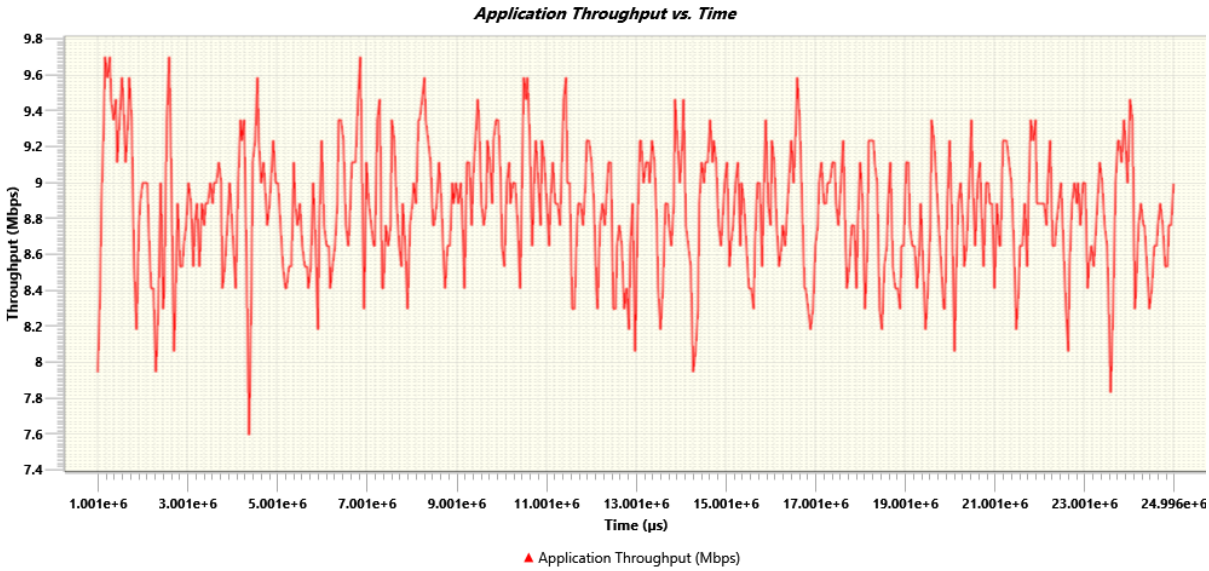

Figure 7-22: *Overall command traffic throughput at the destination which got affected by the attack.*

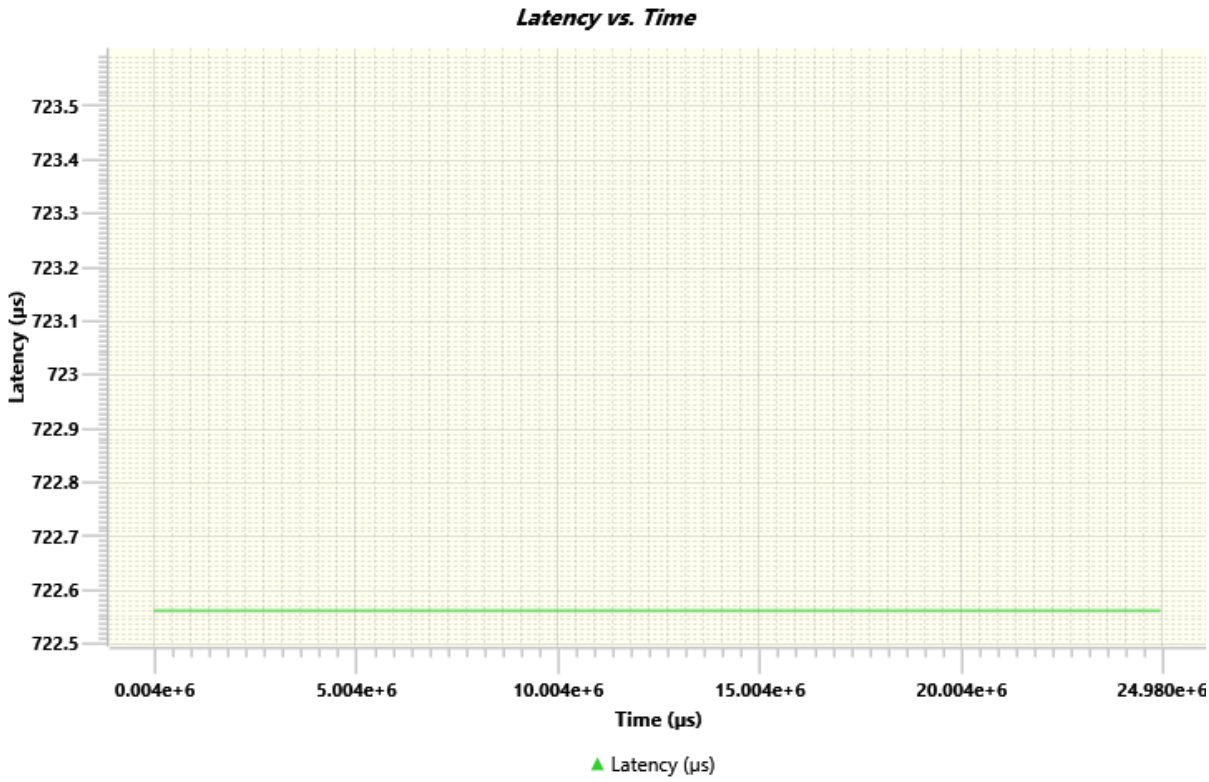

Figure 7-23: *In this plot, we observe that even though there was a packet drop attack on the network the latency of the UDP based command application was not increased. This is because of non-reliable protocol chosen for the transmission. The trace of the packets are not monitored in UDP, hence the dropped packets are simply ignored.*

8 Limitations

- Cyber-attack detection algorithms are currently not supported and are under development.
- Cyber-attack counter measures are currently not supported and are under development.

9 References

1. Schenato, L., Franceschetti, M., & Sastry, S. S. (2007). Foundations of Control and Estimation Over Lossy Networks. *Proceedings of the IEEE*, *95*(1).

