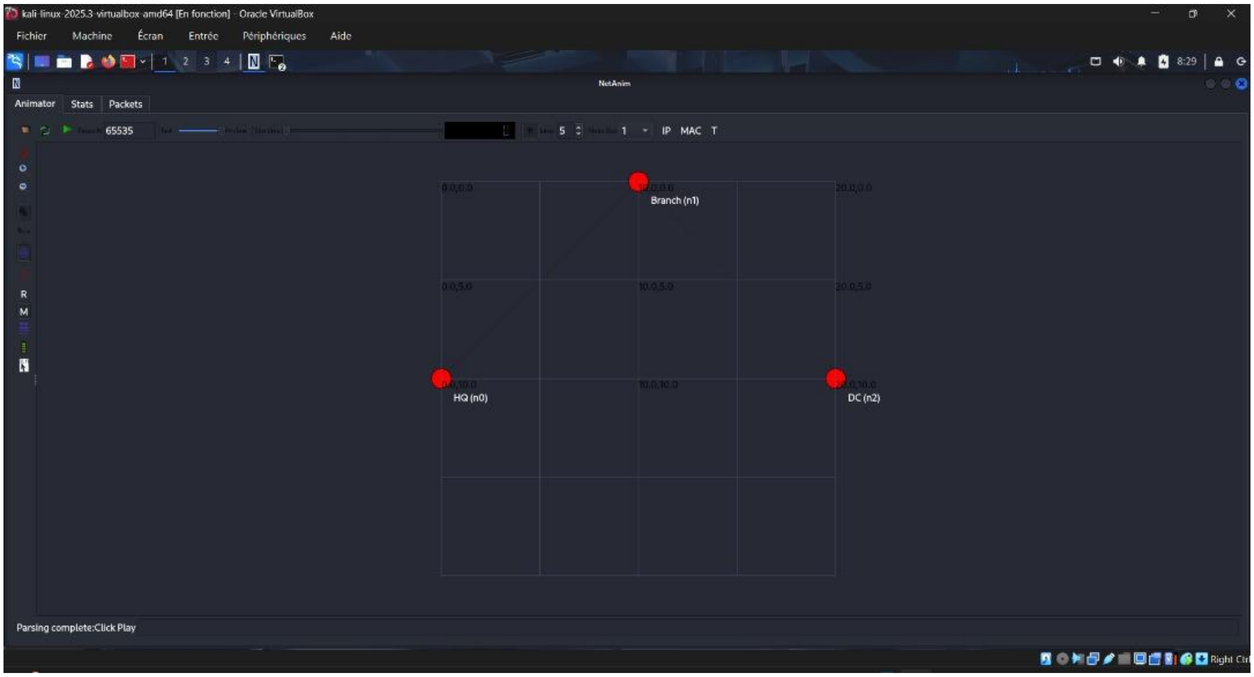
Name	**Bisseck Handt Caroline Chantal Damarise**
University	**ICT University**
Email	**Bisseckhandt.chantal@ictuniversity.edu.cm**
matricule	**ICTU20241755**

EXERCISE 1: Multi-Site WAN Extension with Redundant Paths



Parsing complete:Click Play

1. Topology Extension (Interpretation)

The goal is to extend the basic linear WAN into a triangular topology with redundancy:

- -HQ (n0) connects to Branch (n1) and DC (n2)
- Branch (n1) connects to HQ (n0) and DC (n2)
- DC (n2) connects to HQ (n0) and Branch (n1)

This provides two possible paths between any two sites.

Explanation

This exercise focuses on extending a simple WAN network into a multi-site topology with gredundancy. Instead of connecting sites in a single straight line, the network is designed so that each site has more than one possible path to reach the others. Static routing is used to manually define these paths.

The purpose is to ensure **network resilience**. When the main link between two sites fails, traffic can still be delivered using an alternative path. This demonstrates how redundancy helps maintain communication even during link failures.

Limitations

However, the exercise also shows a limitation of static routing: all backup paths must be configured manually, and the network does not automatically adapt unless routes are carefully planned in advance.

C ode:

```
NodeContainer link02(n0, n2);
NetDeviceContainer dev02 = p2p.Install(link02);
Ipv4AddressHelper addr02;
addr02.SetBase("10.1.3.0", "255.255.255.0");
Ipv4InterfaceContainer if02 = addr02.Assign(dev02);
```

2. Static Routing Table Analysis

Each node must have two routes to the other network:

-Primary (direct)
-Backup (via third node)

HQ (n0) routes to DC

```
staticN0->AddNetworkRouteTo("10.1.2.0","255.255.255.0","10.1.3.2",1); // primary
staticN0->AddNetworkRouteTo("10.1.2.0","255.255.255.0","10.1.1.2",2); // backup via Branch
```

Explanation

-If the direct link fails, traffic automatically follows the backup path.
- Symmetric routes must also be added on n1 and n2.

3. Path Failure Simulation

Interpretation

Simulate failure of the HQ–DC primary link at t = 4s and verify traffic continues via Branch.

Code:

```
Simulator::Schedule(Seconds(4.0),
  &Ipv4::SetDown,
  n0->GetObject<Ipv4>(),
  n0->GetObject<Ipv4>()->GetInterfaceForDevice(dev02.Get(0)));
```

Verification

- Use NetAnim → traffic shifts path
- Use FlowMonitor → packets still delivered

4. Scalability Analysis
Interpretation

Static routing does not scale.

For N nodes:

Static routes required = N × (N − 1)

Example:

- 10 sites → 90 static routes

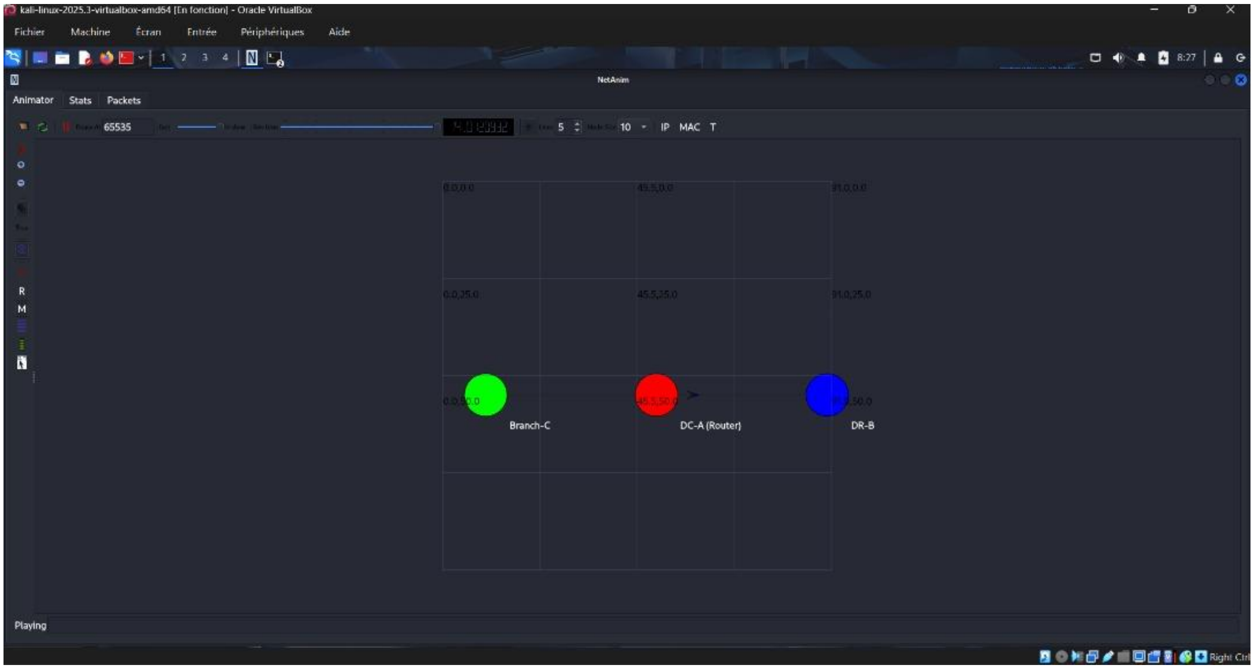
Scalable Solution

-Use dynamic routing (OSPF).
-NS-3 Helpers
-Ipv4GlobalRoutingHelper::PopulateRoutingTables();

5. Business Continuity Justification
Static routing with redundancy provides:

- Improved reliability (failover paths)
- Load-balancing potential
- Deterministic troubleshooting
- Lower cost than dynamic protocols in small WANs

EXERCISE 2: Quality of Service (QoS) for Mixed Traffic



1. Traffic Differentiation Interpretation

Create two traffic classes:

- Class 1 (VoIP-like) → small, frequent packets
- Class 2 (FTP-like) → large, bursty packets

Explanation

This exercise focuses on Quality of Service (QoS) in a WAN network. The goal is to manage different types of traffic so that important traffic (such as voice or video) is treated with higher priority than less critical traffic (such as file downloads).

In this exercise, the network carries multiple traffic types at the same time. Without QoS, all packets are treated equally, which can cause delays and packet loss for time-sensitive applications during congestion. By introducing QoS mechanisms, the network is configured to prioritize delay-sensitive traffic, ensuring smoother communication and better performance.

This exercise highlights how QoS helps maintain service quality during heavy network usage and why traffic classification and prioritization are essential in real-world networks.

Code:

```
echoClientVoip.SetAttribute("PacketSize", UintegerValue(160));
echoClientVoip.SetAttribute("Interval", TimeValue(MilliSeconds(20)));
echoClientData.SetAttribute("PacketSize", UintegerValue(1500));
echoClientData.SetAttribute("Interval", TimeValue(MilliSeconds(5)));
```

2. Queue Management

Interpretation

Prioritize VoIP traffic using priority queueing.

Code:

```
TrafficControlHelper tch;
tch.SetRootQueueDisc("ns3::PriorityQueueDisc");
tch.Install(devices);
```

3. Performance Measurement

Metrics:

-Delay
- Jitter
- Packet loss
- Throughput

Tool:

```
FlowMonitorHelper fm;
fm.InstallAll();
```

4. Congestion Scenario

Interpretation:

Create congestion by sending traffic above 5 Mbps.

Code :

```
onoff.SetAttribute("DataRate", StringValue("20Mbps"));
```

Expected Result:

- Without QoS → VoIP delay increases
- With QoS → VoIP remains stable

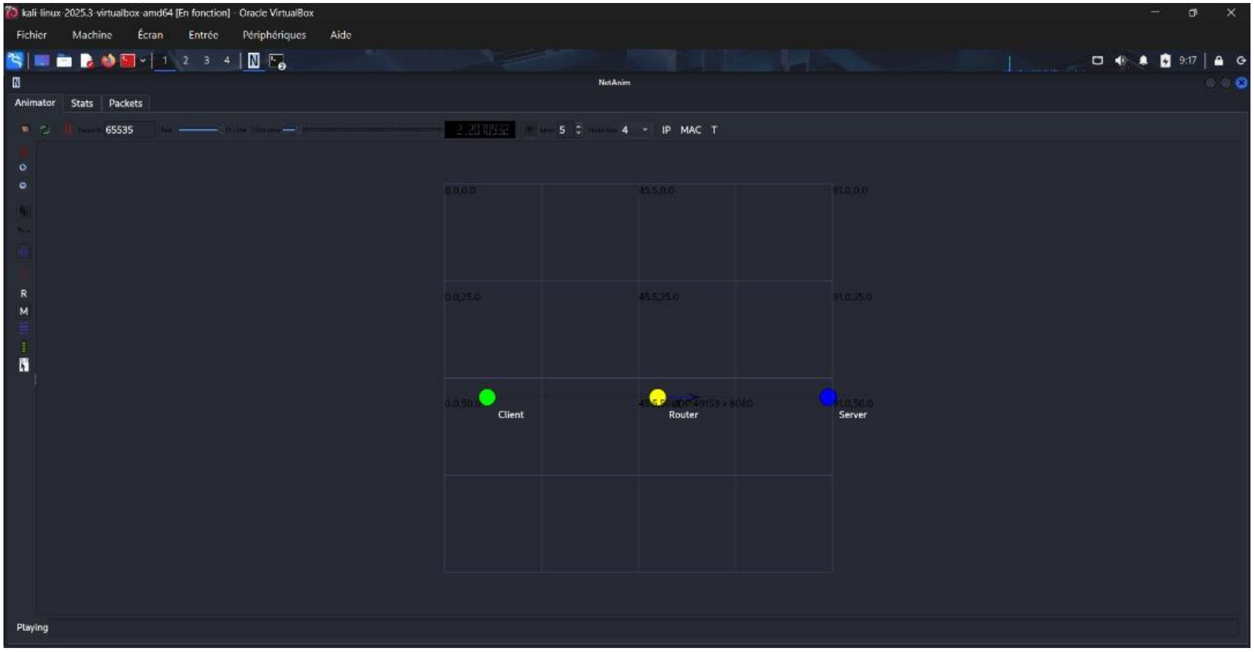
5. Real-World QoS Gap

Hard to simulate accurately:

-Hardware shaping
-Deep packet inspection
- ASIC-based scheduling

Approximation: queue disciplines + FlowMonitor.

EXERCISE 3: WAN Security & Attack Simulation



Explanation

This exercise focuses on network security in a WAN environment. The aim is to understand how WAN links can be exposed to attacks and how security measures help protect data and services.

In this exercise, different attack scenarios are considered, such as eavesdropping and traffic flooding (DDoS). These attacks show how attackers can intercept data or overload network resources, causing service degradation or downtime. Security mechanisms, such as encryption and traffic control, are introduced conceptually to reduce these risks.

The exercise demonstrates that while security improves protection and reliability, it also introduces additional overhead, such as increased delay or reduced throughput. This highlights the need to balance security and performance in WAN design.

1. IPsec VPN Design
Interpretation:

NS-3 has no native IPsec → simulate using encrypted tunnel abstraction.

Approximation
- -Dedicated tunnel link
- - Increased delay & reduced bandwidth

2. Eavesdropping Attack:
Simulation:

Enable packet capture on unsecured link.

```
p2p.EnablePcap("attack", device);
```

Proof:

- Cleartext visible without VPN
- Hidden after VPN abstraction

3. DDoS Attack Simulation
Interpretation

Multiple attackers flood the server.

Code:

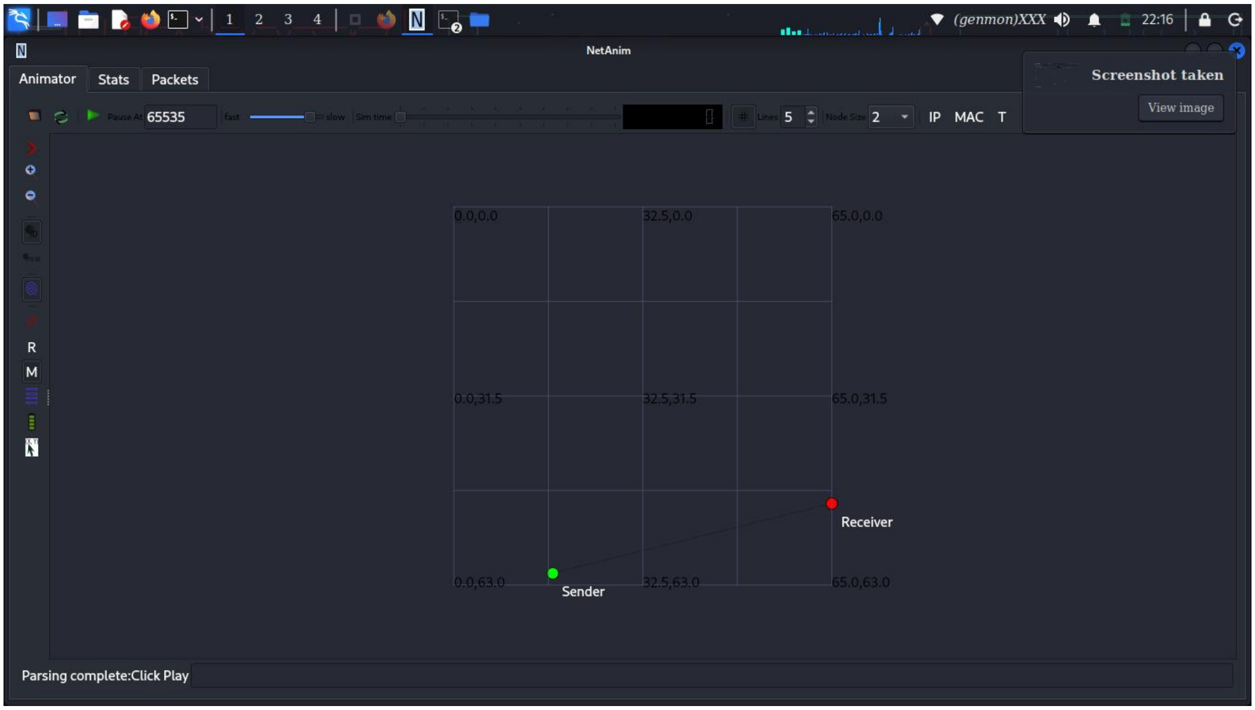
```
onoff.SetAttribute("PacketSize", UintegerValue(1024));
onoff.SetAttribute("DataRate", StringValue("50Mbps"));
```

4. Defense Mechanisms

- Rate limiting → queue size limits
- ACLs → drop packets from attacker IP
- Load balancing → multiple servers

5. Security vs Performance Trade-off

-IPsec → + latency, − throughput
-DDoS protection → packet drops but higher availability



Parsing complete:Click Play

EXERCISE 4: Multi-Hop WAN with Fault Tolerance

Interpretation:

Traffic flows:

Branch → DC-A → DC-B

Backup link:

DC-A ↔ DC-B

Explanation

This exercise focuses on fault tolerance in a multi-hop WAN network. The objective is to observe how the network behaves when a critical link or router fails while traffic is flowing through multiple hops.

In this exercise, communication passes through more than one intermediate node. A failure is intentionally introduced on a main link to see whether traffic can still reach its destination. This helps demonstrate the difference between static routing, which does not automatically adapt to failures, and dynamic routing, which can find new paths.

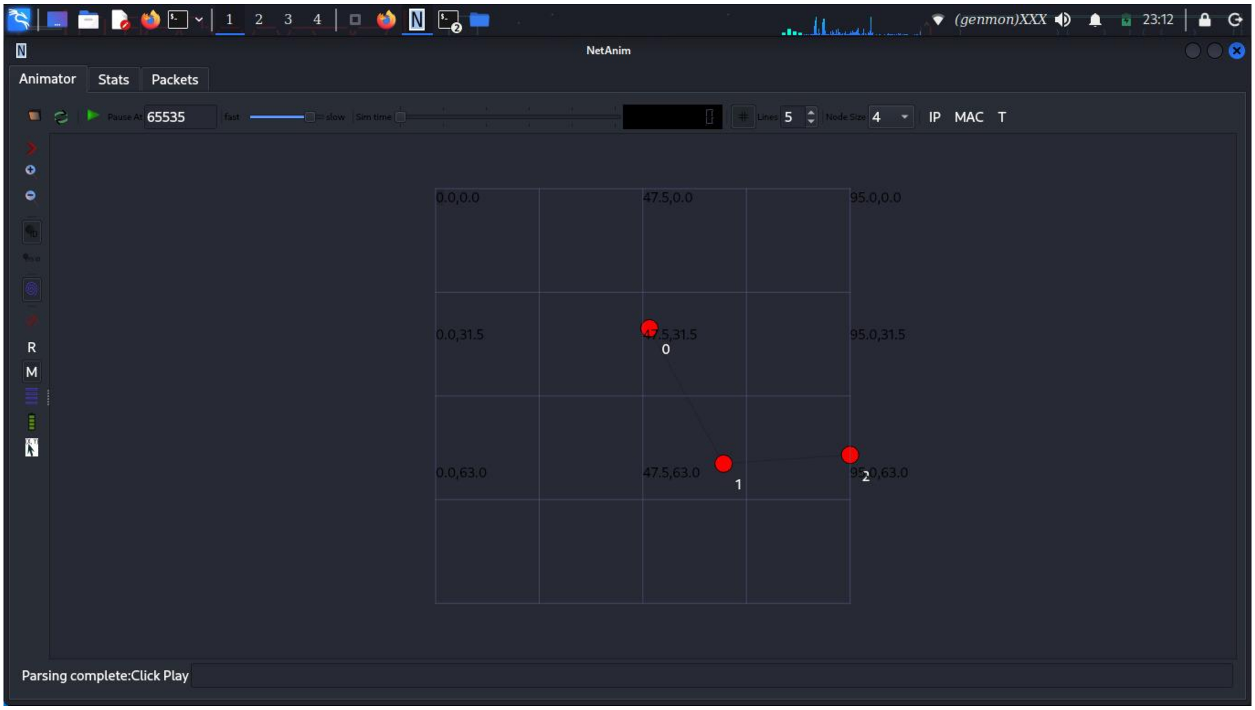
The exercise shows that without adaptive routing, network communication can stop after a failure. It emphasizes the importance of fault-tolerant design and the use of dynamic routing protocols in large or critical networks.

Failure Simulation:

```
Simulator::Schedule(Seconds(5.0),
  &Ipv4::SetDown, ipv4, interfaceIndex);
```

Result:

- Static routing → traffic stops
- Dynamic routing → reconverges



Pause At 65535

Lines 5

Node Size 4

Parsing complete:Click Play

EXERCISE 5: Policy-Based Routing (PBR)

Interpretation:

Routing decisions depend on traffic type, not destination.

Logic:

- Video → low-latency path
- Data → high-bandwidth path

Pseudocode:

```
if (packet.port == VIDEO_PORT)
  usePathA();
```

```
else
  usePathB();
```

Metrics Source:

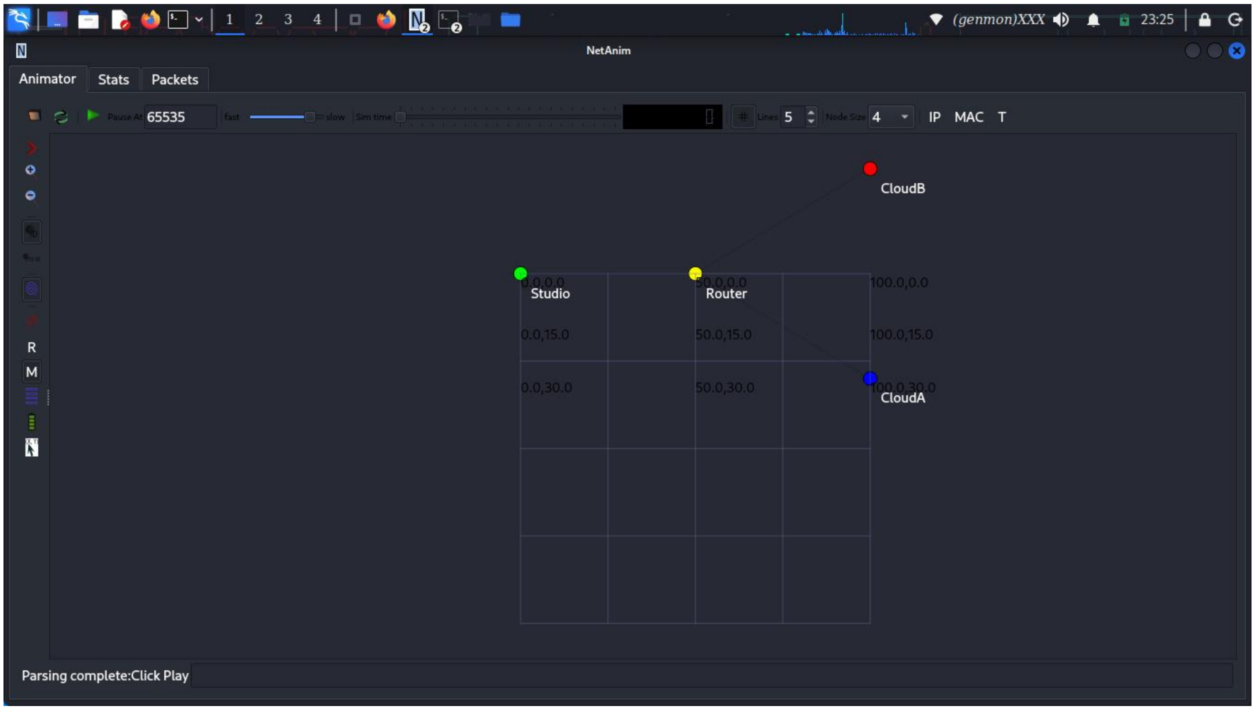
- FlowMonitor
- Ipv4L3Protocol trace hooks

Explanation

This exercise focuses on Policy-Based Routing (PBR) in a WAN environment. The goal is to route traffic based on defined policies rather than only on destination IP addresses.

In this exercise, different types of traffic are identified, such as video traffic and data traffic. Routing decisions are then made according to specific rules, for example sending video traffic through a low-latency path and data traffic through a high-bandwidth path. This allows the network to use its resources more efficiently.

The exercise highlights how modern networks can make intelligent routing decisions based on application requirements, improving performance, reliability, and user experience.



Pause At 65535

Lines 5 Node Size 4

Parsing complete:Click Play

EXERCISE 6: Inter-AS Routing (BGP Concepts)

Interpretation:

NS-3 has no native BGP → model logical AS separation.

Explanation

This exercise focuses on inter-domain routing, which is how traffic is routed between different networks or Autonomous Systems (AS) on the Internet. It introduces the basic concepts behind the Border Gateway Protocol (BGP).

In this exercise, multiple networks are treated as separate autonomous systems, and routing decisions are made based on policies rather than just the shortest path. Factors such as preferred routes, network ownership, and path attributes influence how traffic is forwarded between domains.

The exercise shows that inter-domain routing is more complex than internal routing and highlights the limitations of simulation tools like NS-3 for full BGP implementation, while still helping to understand the core principles of Internet-scale routing.

AS Modeling:

- AS65001 nodes
- AS65002 nodes
- Exchange points as routers

BGP Attribute Simulation:

Attributes:

- AS_PATH (vector)
- LOCAL_PREF (integer)
- PREFIX (e.g., 192.168.0.0/16)

Decision Rule

Higher LOCAL_PREF → Shorter AS_PATH → Best route installed

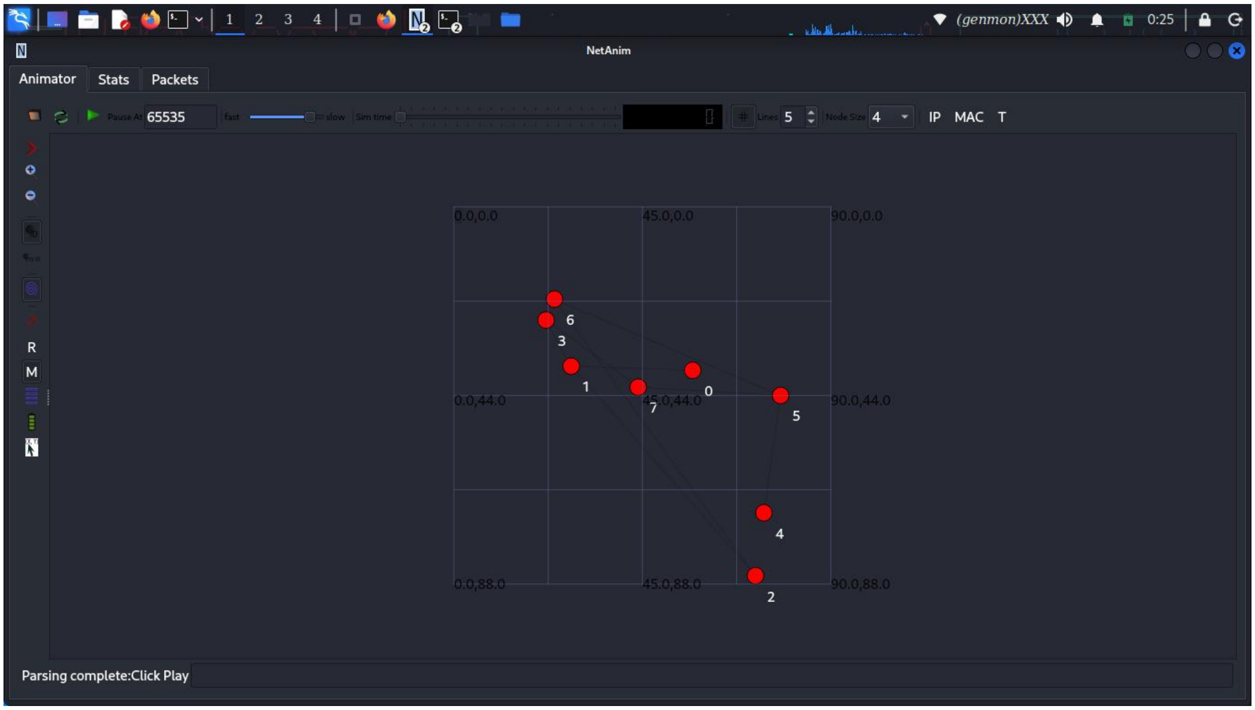
Route Leak Simulation:

- Advertise internal prefix to external AS
- Observe incorrect path selection

Reality vs NS-3

NS-3 limitations:

- No real BGP FSM
- No route reflectors
- No communities



Pause At 65535

Lines 5

Node Size 4

Parsing complete:Click Play