//Enable Tracing using flowmonitor

FlowMonitorHelper flowmon;

Ptr<FlowMonitor> monitor = flowmon.InstallAll();

//Set when to stop simulator

Simulator::Stop (Seconds (simulationTime + 5));

//Add visualization using Netanim

/\*AnimationInterface anim ("ex2.xml");

AnimationInterface::SetConstantPosition(nodes.Get(0), 1.0, 1.0);

AnimationInterface::SetConstantPosition(nodes.Get(1), 2.0, 2.0);

AnimationInterface::SetConstantPosition(nodes.Get(2), 3.0, 2.0);

AnimationInterface::SetConstantPosition(nodes.Get(3), 1.0, 3.0);

anim.EnablePacketMetadata ();\*/ // Optional

//Run the simulator

Simulator::Run ();

// Print per flow statistics

monitor->CheckForLostPackets ();

Ptr<Ipv4FlowClassifier> classifier = DynamicCast<Ipv4FlowClassifier> (flowmon.GetClassifier ());

std::map<FlowId, FlowMonitor::FlowStats> stats = monitor->GetFlowStats ();

for (std::map<FlowId, FlowMonitor::FlowStats>::const\_iterator iter = stats.begin (); iter != stats.end (); ++iter)

{

Ipv4FlowClassifier::FiveTuple t = classifier->FindFlow (iter->first);

NS\_LOG\_UNCOND("Flow ID: " << iter->first << " Src Addr " << t.sourceAddress << " Dst Addr " << t.destinationAddress);

NS\_LOG\_UNCOND("Tx Packets = " << iter->second.txPackets);

NS\_LOG\_UNCOND("Rx Packets = " << iter->second.rxPackets);

NS\_LOG\_UNCOND("lostPackets Packets = " << iter->second.lostPackets);

NS\_LOG\_UNCOND("Throughput: " << iter->second.rxBytes \* 8.0 / (iter->second.timeLastRxPacket.GetSeconds()-iter->second.timeFirstTxPacket.GetSeconds()) / 1024 << " Kbps");

}