

Simulating Packet Drops in NS2

This tutorial demonstrates how to create a network simulation experiment in NS2 to simulate packet drops by creating a mismatch between link capacities and transmission rates.

Setting up the Network

The tutorial begins by instructing users to create a new wired scenario in NSG, similar to a previous video. → This involves setting up three nodes (N0, N1, and N2) and connecting them. Unlike the previous example, this simulation uses UDP at the transport layer because it allows for sending packets at any desired rate, making it suitable for simulating packet drops. → N1 is configured with a UDP transmitter, while N2 uses a null receiver. →

Configuring Application and Link Parameters

At the application layer, CBR (Constant Bit Rate) is selected, enabling control over the packet transmission rate. → The CBR parameters are set with a start time of 0, an end time of 10, a packet size of 1000, and a rate slightly exceeding the link capacity (100 Mbps). → The instructor notes that the link capacity between N1 and N0 might need adjustment in the TCL script. → After attaching the CBR agent and setting the simulation time, the TCL script is generated and saved. →

Modifying the TCL Script and Simulating Packet Drops

The instructor opens the generated TCL script (packet-draft.tcl) and introduces a mismatch in link capacities to induce packet drops. → The link between N0 and N1 remains at 100 Mbps, while the link between N0 and N2 is reduced to 99.9 Mbps. → With N1 transmitting at 100 Mbps and N0-N2 link having a lower capacity, packet drops are expected. → The script is saved, and the simulation is run using ns packet-draft.tcl. →

The network animation visually confirms packet drops due to the capacity mismatch. → The instructor then explains that analyzing the trace file (out.tr) directly is impractical due to the large number of events. →

Analyzing Packet Drops with Grep

Instead of manually inspecting the trace file, the instructor uses the grep command to count the dropped packets. → The command `cat out.tr | grep drop | wc -l` counts the lines containing "drop" in the trace file, indicating the number of dropped packets (initially 64). →

To demonstrate increasing packet drops, the TCL script is modified again, reducing the N0-N2 link capacity further to 99.0 or 99.5 Mbps. → Rerunning the simulation and the grep command shows a significant increase in dropped packets (to 514). → This highlights the impact of even small capacity mismatches on packet loss.

Conclusion

This tutorial provides a practical demonstration of simulating packet drops in NS2 by creating a controlled mismatch between link capacities and transmission rates. It also introduces the use of command-line tools like grep for efficiently analyzing simulation results, specifically for counting dropped packets. The instructor concludes by suggesting further exploration of NS2 functionalities in subsequent videos, focusing on simulating and plotting congestion windows. →