

Lab Assignments using ns-3

From Nsnam

This page serves as a placeholder for ns-3 based lab assignments that can be used for Computer Networks coursework in universities. Feel free to add experiments to this list.

Contents

- 1 Simple client-server communication (NS-3 Warmup)
- 2 TCP variants
- 3 TCP and router queues
- 4 Routing (Optimised Link State Routing)
- 5 Wifi RTS/CTS
- 6 Wifi Channels

Simple client-server communication (NS-3 Warmup)

- **Level:** Introductory
- **Expected learning outcome:** NS-3 simulation basics. Basic client server paradigm. Reading pcap traces.
- **Experiment:**
 1. Create a simple topology of two nodes (Node1, Node2) separated by a point-to-point link.
 2. Setup a UdpClient on one Node1 and a UdpServer on Node2. Let it be of a fixed data rate Rate1.
 3. Start the client application, and measure end to end throughput whilst varying the latency of the link.
 4. Now add another client application to Node1 and a server instance to Node2. What do you need to configure to ensure that there is no conflict?
 5. Repeat step 3 with the extra client and server application instances. Show screenshots of pcap traces which indicate that delivery is made to the appropriate server instance.

TCP variants

- **Level:** Introductory
- **Expected learning outcome:** TCP internals and the difference between each of the variants. NS-3 tracing mechanism.

▪ Experiment:

1. Create a simple dumbbell topology, two client Node1 and Node2 on the left side of the dumbbell and server nodes Node3 and Node4 on the right side of the dumbbell. Let Node5 and Node6 form the bridge of the dumbbell. Use point to point links.
2. Install a TCP socket instance on Node1 that will connect to Node3.
3. Install a UDP socket instance on Node2 that will connect to Node4.
4. Start the TCP application at time 1s.
5. Start the UDP application at time 20s at rate Rate1 such that it clogs half the dumbbell bridge's link capacity.
6. Increase the UDP application's rate at time 30s to rate Rate2 such that it clogs the whole of the dumbbell bridge's capacity.
7. Use the ns-3 tracing mechanism to record changes in congestion window size of the TCP instance over time. Use gnuplot/matplotlib to visualise plots of cwnd vs time.
8. Mark points of fast recovery and slow start in the graphs.
9. Perform the above experiment for TCP variants Tahoe, Reno and New Reno, all of which are available with ns-3.

TCP and router queues

▪ Level: Introductory

- **Expected learning outcome:** Queues, packet drops and their effect on congestion window size.

▪ Experiment:

1. As in previous exercise, Create a simple dumbbell topology, two client Node1 and Node2 on the left side of the dumbbell and server nodes Node3 and Node4 on the right side of the dumbbell. Let Node5 and Node6 form the bridge of the dumbbell. Use point to point links.
2. Add drop tail queues of size QueueSize5 and QueueSize6 to Node5 and Node6, respectively.
3. Install a TCP socket instance on Node1 that will connect to Node3.
4. Install a TCP socket instance on Node2 that will connect to Node3.
5. Install a TCP socket instance on Node2 that will connect to Node4.
6. Start Node1--Node3 flow at time 1s, then measure it's throughput. How long does it take to fill link's entire capacity?
7. Start Node2--Node3 and Node2--Node4 flows at time 15s, measure their throughput.
8. Measure packet loss and cwnd size, and plot graphs throughput/time, cwnd/time and packet loss/time for each of the flows.
9. Plot graph throughput/cwnd and packet loss/cwnd for the first flow. Is there an optimal value for cwnd?
10. Vary QueueSize5 and QueueSize6. Which one has immediate effect on cwnd size of the first flow? Explain why.

Routing (Optimised Link State Routing)

- **Level:** Introductory
- **Expected learning outcome:** What are MANETs and how they work. OLSR basics. Routing issues associated with MANETs.
- **Experiment:**
 1. Create a wireless mobile ad-hoc network with three nodes Node1, Node2 and Node3. Install the OLSR routing protocol on these nodes.
 2. Place them such that Node1 and Node3 are just out of reach of each other.
 3. Create a UDP client on Node1 and the corresponding server on Node3.
 4. Schedule Node1 to begin sending packets to Node3 at time 1s.
 5. Verify whether Node1 is able to send packets to Node3.
 6. Make Node2 move between Node1 and Node3 such that Node2 is visible to both A and C. This should happen at time 20s. Ensure that Node2 stays in that position for another 15s.
 7. Verify whether Node1 is able to send packets to Node3.
 8. At time 35s, move Node2 out of the region between Node1 and Node3 such that it is out of each other's transmission ranges again.
 9. Verify whether Node1 is able to send packets to Node3.
 10. To verify whether data transmissions occur in the above scenarios, use either the tracing mechanism or a RecvCallback() for Node3's socket.
 11. Plot the number of bytes received versus time at Node3.
 12. Show the pcap traces at Node 2's Wifi interface, and indicate the correlation between Node2's packet reception timeline and Node2's mobility.

Wifi RTS/CTS

- **Level:** Introductory
- **Expected learning outcome:** How 802.11 works with and without RTS/CTS. An insight into why its hard to setup efficient wireless networks.
- **Experiment:**
 1. Setup a 5x5 wireless adhoc network with a grid. You may use `examples/wireless/wifi-simple-adhoc-grid.cc` as a base.
 2. Install the OLSR routing protocol.
 3. Setup three UDP traffic flows, one along each diagonal and one along the middle (at high rates of transmission).
 4. Setup the ns-3 flow monitor for each of these flows.
 5. Now schedule each of the flows at times 1s, 1.5s, and 2s.
 6. Now using the flow monitor, observe the throughput of each of the UDP flows. Furthermore, use the tracing mechanism to monitor the number of packet collisions/drops at intermediary nodes. Around which nodes are most of the collisions/drops happening?
 7. Now repeat the experiment with RTS/CTS enabled on the wifi devices.
 8. Show the difference in throughput and packet drops if any.

Wifi Channels

- **Level:** IntermediateInstall
- **Expected learning outcome:** How Radio channel models affect transmission. An insight into why its important to correctly model the channel.
- **Experiment:**
 1. Setup a 2-nodes wireless adhoc network. Place the nodes at a fixed distance in a 3d scenario.
 2. Install all the relevant network stacks, up to and including UDP.
 3. Setup a CBR transmission between the nodes, one acting as a server and one as a client. Take the iperf[1] (<http://sourceforge.net/projects/iperf>) behaviour as an example.
 4. Setup counters and outputs for packets sent and received.
 5. Schedule the simulation to run for enough time to obtain statistically relevant results (suggestion: analyze some test results and reduce the simulation time accordingly).
 6. Repeat the simulation varying the distance between the nodes from a minimum of 1meter to the point where the nodes can't transmit/receive anymore.
 7. Repeat the above varying the channel models and the transmission/receive parameters like node's position above the ground, transmission power, etc.
 8. Show the differences between the various channel models, and comment them. Identify the channel model that is more appropriate for each case (indoor, outdoor, LoS, NLoS, etc.).

Retrieved from "https://www.nsnam.org/mediawiki/index.php?title=Lab_Assignments_using_ns-3&oldid=7264"

-
- This page was last modified on 10 February 2013, at 16:06.
 - This page has been accessed 39,056 times.
 - Content is available under Creative Commons Attribution-ShareAlike unless otherwise noted.