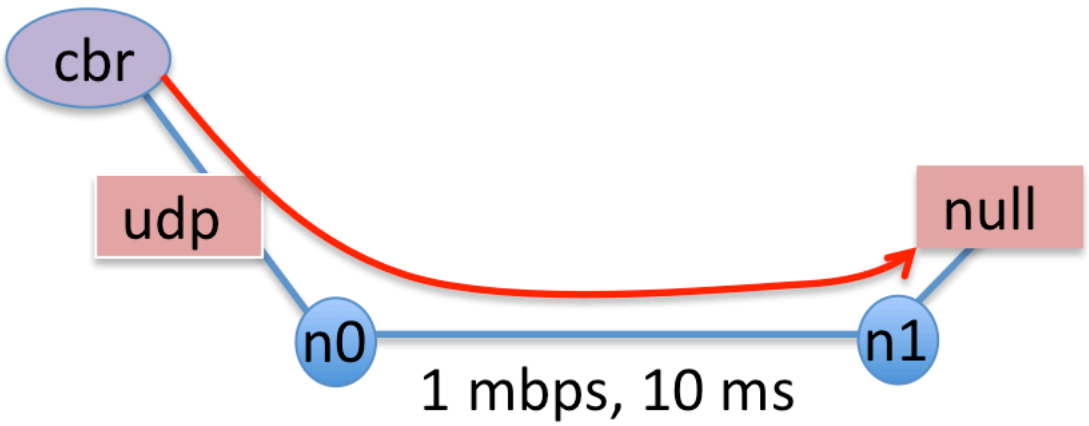
Learning ns2 network simulator with simple examples

Step by step tutorial to create and run ns2 simulations and analyze the results.
http://www.isi.edu/nsnam/ns/tutorial/index.html

Example1: Create and run ns2 simulations and plot avg throughput at each 0.05 sec.

1. Login to the server **cs436.cs.csusm.edu** with fileZilla using your campus account. Download example1.tcl from Cougar Courses and upload it into your folder on cs436 server. Login to the server with PuTTY and run the tcl file. It generates a trace file example1.tr
$ns example1.tcl

2. Open example1.tcl and read the file.
This simulation creates two nodes n0 and n1.
It first creates a duplex link between them with 1Mb bandwidth and 10ms delay.
Then creates a UDP agent (transport layer) and attach it to n0.
Then creates a CBR traffic source (application layer) and attach it to n0's UDP agent.
Then creates a Null agent (transport layer) and attach it to n1.
Now it connects the UDP agent at n0 to Null agent at n1.
When the simulation starts, CBR at n0 starts to generate traffic and UDP transfers it to Null agent at n1.
Simulation starts at second 0.5 and stops at 4.5.



The trace format is shown in Fig.13 at http://nile.wpi.edu/NS/analysis.html
A copy of that figure is given next page in this file. The trace file keeps track of all packets arriving at each node.

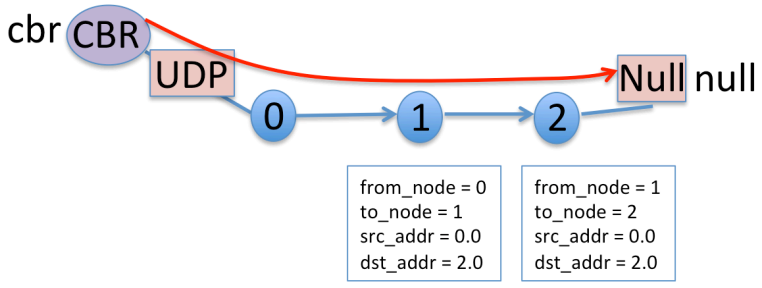
1	2	3	4	5	6	7	8	9	10	11	12
event	time	from node	to node	pkt type	pkt size	flags	fid	src addr	dst addr	seq num	pkt id

```
r : receive (at to_node)
+ : enqueue (at queue)                src_addr : node.port (3.0)
- : dequeue (at queue)                dst_addr : node.port (0.0)
d : drop    (at queue)
```

```
r 1.3556 3 2 ack 40 ------- 1 3.0 0.0 15 201
+ 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
- 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
r 1.35576 0 2 tcp 1000 ------- 1 0.0 3.0 29 199
+ 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
d 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
+ 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
- 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
```

Notes:

- While a packet is travelling from source to destination, at each hop including destination, an event is recorded in .tr file. If the packet gets dropped at a node, a drop event is recorded and then ns2 stops tracking that packet.
- There are 4 types of events: (r) receive, (+) enqueue, (-) dequeue, (d) drop.
- When a packet arrives at a node, the node drops (d) it if its input buffer is full. Otherwise, the node enqueues (+) the packet into its input buffer, dequeues (-) the packet from its input buffer, records a receive (r) event and finally transmit it to the next hop or stop transmission if the current node is the destination. If it's required, the node may enqueue/dequeue the packet into/from output buffer before transmission as well.
- ns2 numbers the nodes from 0, disregarding how the nodes are numbered in .tcl file. Thus, the first node defined in .tcl file is 0, and next one are 1, 2, …
- Field 9 (src_addr) indicates the source of packet and field 10 (dst_addr) indicates the destination. However field 3 (from_node) indicates last hop and field 4 (to_node) indicates this hop, which are the source and destination of current link.
- Src_addr and dst_addr are in form "node.port", e.g. "3.0" means node 3 port 0.
- Example: Assume a scenario contains 3 nodes 0, 1, and 2, where 0 is the source, 2 is the destination, and 1 is a node that connects source and destination. When a packet arrives at node 1, from_node = 0 and to_node = 1. When the packet arrives at node 2, from_node = 1 and to_node = 2. However in both cases, src = 0 and dst = 2.



4. Now we can analyze example1.tr using awk text processing language. A tutorial to awk is available at: http://www.grymoire.com/Unix/Awk.html

5. Copy example1.awk to the server and run it using awk and the input file example1.tr
$awk -f example1.awk example1.tr
The results will be at example1.xls file.

6. Download example1.xls file from cs436 server to your system.

7. Open the file with Microsoft Excel and plot a Scatter chart with data items.
This is a throughput plot of your network for each 0.05 seconds during the 5 seconds simulation.

8. Study example1.awk
It reads the file example1.tr row by row. At each row, it refers to each field by its number. For example, it refers to the 2nd field (time) by $2.
For each 0.05 seconds, it calculates how many packets have arrived at destination (n1) and what is the sum of those packets sizes. Throughput is the amount of packets arrived at destination at each time slice (which is 0.05 seconds here).
In this example, the awk file finds out the total number of packets received at destination.

9. In your project, you should find out throughput and also the number of packets received at destination at each 0.5 seconds separately. Use this file and adjust it to get the required results and plot the charts.

Example2: Create and run ns2 simulations and plot periodic throughput.

In this example, we use LossMonitor objects as traffic sinks (instead of Null). This object stores the amount of received bytes at a destination from different traffic sources. This example creates one trace file for each traffic source and stores the periodic throughput for these traffic sources in files at each 0.5 time slice.

1. Login to the server cs436.cs.csusm.edu

2. Copy example2.tcl to the server and run it. It generates three trace files example2out0.tr, example2out1.tr and example2out2.tr.
$ns example2.tcl

3. Download the .tr files from cs436 server to your system. Note that these .tr files don't follow the general format of .tr files generated by ns2 (similar to the one given in last example). In this example, the programmer defines the format of output file as: "current time slice, amount of bytes received by the traffic sink".

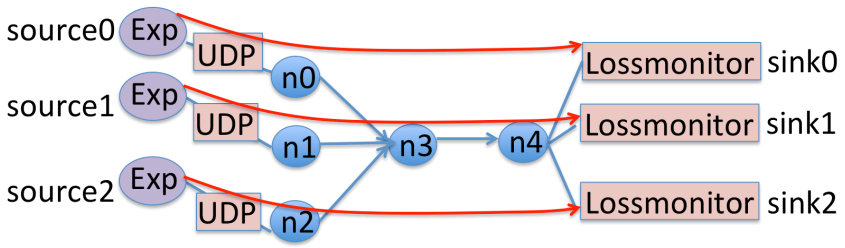
4. Open a new Excel file. Open trace files, and copy the files to the Excel file one by one. After each copy, click on the Paste Options icon and select
Use Text Import Wizard …
In the opened wizard, select Delimited option, and choose Space as Delimiter.
Excel separates the two columns.

5. Now you have three pairs of columns. The first column is the same for all these pairs. Plot three **scatter charts** with the first and second column of each pair.

6. You observe that the bursts of the first flow peak at 0.1Mbit/s, the second at 0.2Mbit/s and the third at 0.3Mbit/s.

7. Study example2.tcl

- What are the nodes defined in this example? What are the links? What are the bandwidths and delays?
- Which agents are attached to the nodes? Which traffic generators (applications) are attached to these agents? What are the connections? For each connection, what is the source and what is the destination? This example attaches 3 sink agents to the destination, one for each source.
- When does the simulation start and stop? What about source nodes?
- There is a function named record. It calls itself every 0.5 seconds and records average throughput at the last 0.5 seconds in the output files. Each file belongs to one of the sink agents attached to the destination node n4.



Example3: Analysis of delay, jitter, and number of received or lost packets

This example is based on http://nile.wpi.edu/NS/Example/ns-simple.tcl
The programs that calculate the parameters are based on
https://trungtm.wordpress.com/2007/07/03/how-to-measure-packet-loss-rate-jitter-and-end-to-end-delay-for-udp-based-applications/

1. Login to the server cs436.cs.csusm.edu

2. Copy example3.tcl to the server and run it. It generates a trace file example3.tr
$ns example3.tcl

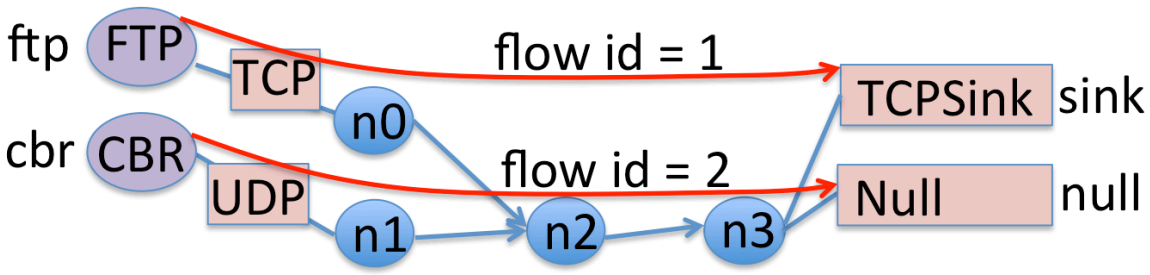
3. Now we can analyze example3.tr using gawk
The trace format is shown in example1, step4 in this tutorial.

4. Copy example3delay.awk, example3jitter.awk and example3pktloss.awk to the server and run them using gawk and the input file example3.tr
$gawk -f example3delay.awk example3.tr
$gawk -f example3jitter.awk example3.tr
$gawk -f example3pktloss.awk example3.tr
The results will be available at example3delay.xls and example3jitter.xls files.

5. Download the xls files from cs436 server to your system.

6. Open the files with Microsoft Excel and plot a Scatter chart with data items.
These are the delay and jitter plots of your network for five seconds simulation.

7. Study example3.tcl and the awk files. Answer the questions of example2 step7 for this scenario. You can use “fid_” to define a flow id for some traffic. This feature helps you to distinguish between traffic flows in your project.



Example4: Analysis of dynamic routing

This example is based on:
http://cs-study.blogspot.com/2013/05/dynamic-nodes-generation-and-traffic.html
Routing protocol is defined as a distance-vector dynamic routing: rtproto DV
At time 0, the simulation starts. Node 0 stars sending packets to node 3 through shortest path (0-1-2-3). At time 1, link 1-2 goes down, and packets get dropped at node 1. Since the routing protocol is dynamic, a new path is found and the traffic dynamically flows through this new path 0-6-5-4-3. At time 2, link 1-2 goes up, and dynamic routing returns to the shorter path 0-1-2-3.

1. Login to the server cs436.cs.csusm.edu.

2. Copy example4.tcl and run it. It generates a trace file example4.tr
$ns example4.tcl

3. Open example4.tr by vi program and search for word “link” in this file.
For a vi manual refer to https://www.cs.colostate.edu/helpdocs/vi.html
$vi routing.tr
/link
Enter n or / to repeat search forward
Enter N to repeat search backward

4. You observe that at time 1 link 1-2 is turned down and at time 2 it is turned up again. Before time 1 the traffic from 0 to 3 flows through path 0-1-2-3. However, after the link is turned down, some packets are dropped and immediately the routing protocol send rtProtoDV messages to find a new path. Then it flows the traffic through path 0-6-5-4-3. At time 2, link 1-2 goes up, and dynamic routing returns to the shorter path 0-1-2-3.

5. Read example4.tcl carefully. Answer the questions of example2 step7 for this scenario.