ECE 6110 – Computer Aided Design of Computer Networks

Final Project

Measuring and Explaining Differences in Wireless Simulation Models

Submitted by:

Justin Eng
Muhammad Zohaib

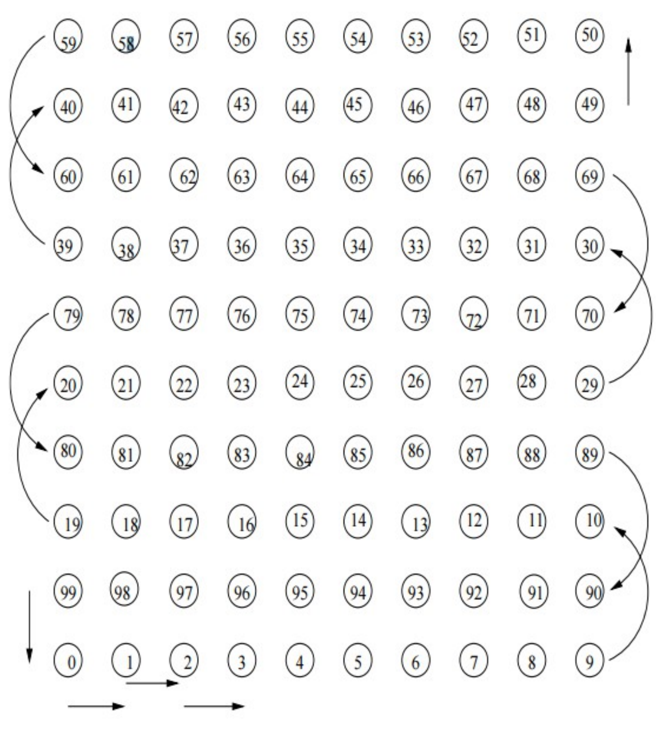
Introduction:

In this report we aim to replicate the experiments performed by Reddy and Riley in comparing various network simulation tools. To briefly summarize the intent of said paper, comparisons of differing implementations involving MAC and PHY level network simulation was carried out in order to identify (and further analyze) discrepancies between the ns-2, GTNetS, and GloMoSim wireless network simulators. Each simulator's implementation of the IEEE 802.11 protocol for wireless communication was scrutinized, and the paper identifies how even minor abstractions taken during the simulation of lower levels of wireless communication can incur significant deviations in simulation results. The paper aims to investigate these differences in order to facilitate improvements to the fidelity of wireless network simulation.

In our report we conduct similar experimentation in the same vein as Reddy and Riley, this time assessing any inherent variability in the ns-3 network simulator. We present our results and any procedures which must be carried out in order to bring the ns-3 simulation results to the deterministic single-packet baseline established by Reddy and Riley for a 100-node network. We continue on to present our results for a network which exhibits randomness as a result of congestion avoidance at the MAC layer.

Experiment:

Similar to Reddy and Riley, we conducted two separate experiments to assess the behavior of ns-3 when compared to both the reference IEEE 802.11 specification and additionally the results put forward by Reddy and Riley's paper. Both experiments make use of an identical network topology involving a total of 100 wireless nodes spaced uniformly across a 1000 by 1000 meter grid. Each node is separated by 100 meter spacing.



We employ static routing between arbitrarily assigned IPs on each node to ensure that packets propagate sequentially through the network in the "next-neighbor" fashion depicted in Figure 1. Note that node #0 represents the "source" node while node #99 is referred to as the "destination" node.

We do not employ any mobility for any nodes, and due to the static routing there is no formal routing protocol in use at any time. We use a maximum radio range of 250 meters for each node; given the layout of our network topology this ensures that each node is capable of transmitting to the next node in sequence without issue.

Nodes are assigned IP addresses in the range from 10.0.0.1 to 10.0.0.100. Effectively, for nodes 0 (IP address 10.0.0.1) to 98 (IP address 10.0.0.99), routing table entries are:

For Node i (IP address 10.0.0.i+1):

Destination	Next Hop
10.0.0.100	10.0.0.i+2

For Node 99 (IP address 10.0.0.100) routing table Entry is:

Destination	Next Hop
10.0.0.1	10.0.0.1

Before sending actual data, each node sends Request-To-Send (RTS) frame, which is received by all its neighbors in the range. Intended destination node replies with Clear-To-Send (CTS) frame. Data frame is sent by source node followed by Acknowledgement from the destination. Time to send RTS includes one DIFS time, sending preamble of 192 bits at 1 Mbps and sending 20 Bytes of RTS at “Basic” rate. Similarly, time to send CTS includes one SIFS time, sending preamble of 192 bits at 1 Mbps and sending 14 Bytes of RTS at “Basic” rate. For data frames, time to send includes one DIFS time, sending preamble of 192 bits at 1 Mbps and sending data (546B in our case) at “Basic” rate.

Following are parameters used for our experiments based on default 802.11b.

Basic Rate	2Mbps / 11Mbps
Data Rate	11 Mbps
Preamble Rate	1 Mbps
RTS Size	20 Bytes
CTS Size	14 Bytes
ACK Size	14 Bytes
Preamble Size	24 Bytes
DIFS	50 µs
SIFS	10 µs
Slot Time	20 µs
UDP Header	8 Bytes
IP Header	20 Bytes
Data Header	34 Bytes
Payload	512 Bytes

We set the default Threshold packet size of 0 Bytes for sending RTS and CTS frames, so that we send RTS and CTS for every data frame.

Just like the paper, initially we conducted first experiment such that a node would send the packet to next hop neighbor immediately after receiving it. Since node has to send acknowledgement to previous node and send packet to next node at the same time, we would get random delay at each hop. Node will choose a random number between (0, Contention Window) and will wait that many slots. To get deterministic results, we introduce 500 µs delay at each node between receiving a packet and forwarding it, which is enough for node to send acknowledgement to the previous node to remove randomness. Tables below show theoretical results for sending data at Control Rate of 11 Mbps and 2 Mbps respectively when either the packet is forwarded to next node immediately or after waiting for 500 µs.

For 11 Mbps Basic Rate:

Deterministic delay:

Request To Send	DIFS (50) + Preamble (192) + send 20 bytes (20*8/11) = 257 µs
Clear To Send	SIFS (10) + Preamble (192) + send 14 bytes (14*8/11) = 212 µs
Data	SIFS (10) + Preamble (192) + send 574 bytes (574*8/11) = 620 µs
Extra Deterministic	500 µs
Total	1589 µs

No delay:

Request To Send	DIFS (50) + Preamble (192) + send 20 bytes (20*8/11) = 257 µs
Clear To Send	SIFS (10) + Preamble (192) + send 14 bytes (14*8/11) = 212 µs
Data	SIFS (10) + Preamble (192) + send 574 bytes (574*8/11) = 620 µs
Expected Random Backoff	300 µs
Total	1389 µs

For 2 Mbps Basic Rate:

Deterministic delay:

Request To Send	DIFS (50) + Preamble (192) + send 20 bytes (20*8/2) = 322 µs
Clear To Send	SIFS (10) + Preamble (192) + send 14 bytes (14*8/2) = 258 µs
Data	SIFS (10) + Preamble (192) + send 574 bytes (574*8/11) = 620 µs
Extra Deterministic	500 µs
Total	1700 µs

No Delay:

Request To Send	DIFS (50) + Preamble (192) + send 20 bytes (20*8/2) = 322 µs
Clear To Send	SIFS (10) + Preamble (192) + send 14 bytes (14*8/2) = 258 µs
Data	SIFS (10) + Preamble (192) + send 574 bytes (574*8/11) = 620 µs
Expected Random Backoff	300 µs
Total	1500 µs

The first experiment served as a baseline to establish a deterministic behavior for the simulation of a single packet traversing the network topology. At the start of simulation, the source node creates a packet which is then forwarded node by node until it reaches the destination node. At this point we say that a "round" has been successfully completed. We run the experiment for 100 such rounds and calculate the total time. Similar to NS2, we ignore first round because of ARP exchange between the nodes.

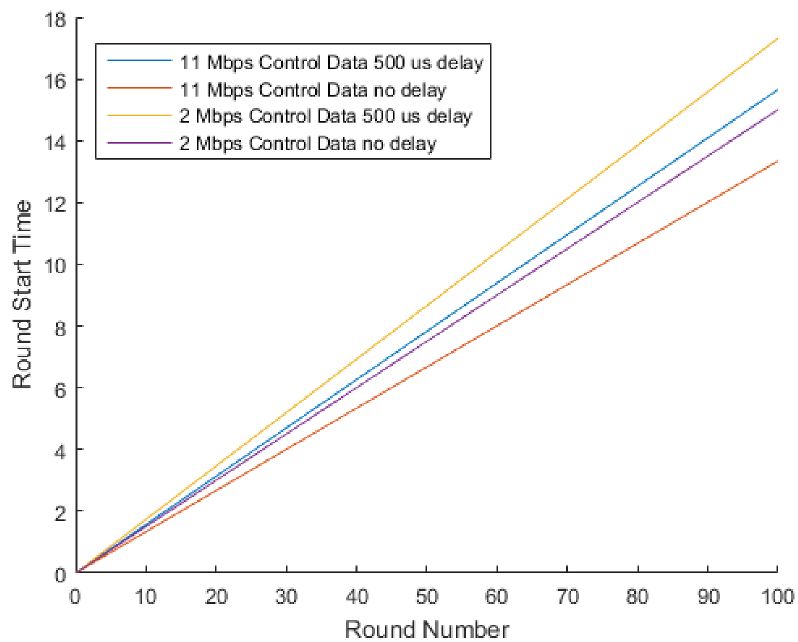
Initially we put one UDP socket on node 0 and one UDP socket on node 99. Routing on each node would take place according to routing tables shown above. We had to change the Time-To-Live (TTL) for IP protocol from default value of 64 as our packet travels maximum of 99 hops before reaching destination node. Using two sockets works fine for experiments where nodes forwarded packet immediately after receiving. But for introducing deterministic delay of 500 µs at each node, we were required to modify kernel code or layers below transport layer in NS3. We came up with an efficient solution such that we installed one socket at each node thus giving us total of 100 sockets. Routing would again take place according to routing tables as before but in this case a new UDP packet is generated at each node after receiving it instead of forwarding the same packet. Node would schedule the packet to be sent 500 µs after receiving it at the transport layer.

The second experiment was conducted using the same methodology employed by Reddy and Riley, and aims to demonstrate features which may incur non-deterministic behavior during our network simulation. Unlike experiment one in which a single packet traversed the network and guaranteed determinism in the results via a 500 microsecond forwarding delay, experiment two aimed to purposefully introduce randomness as a result of channel contention and the resulting effects from congestion avoidance measures such as random backoff.

The second experiment was conducted using an identical topology to the first experiment. At the start of simulation a packet is generated and is propagated through the network in sequential node order. Once this "source packet" has been received at the destination node it is forwarded back to the source node in order to continue propagation through the network. This will continue for up to 100 rounds in total, with the time taken for the source packet to traverse from the source node to the destination node being recorded as the "round time". At the start of simulation a random "congestion burst" time is selected in the range of [0 to 10ms). At this time one node in each row will generate a "congestion packet" addressed to its next hop neighbor, in effect introducing 10 additional packets to propagate throughout the network in addition to the source packet already in flight. The choice to limit the number of congestion packets to 10 in total marks a minor deviation from Reddy and Riley's methodology, though this was chosen in order to keep simulation time tractable. With fewer congestion packets being introduced at the congestion burst time, we expect far less volatility and variability in our simulation results. We conducted the 100-round experiment 30 times in total, and report the average round time in addition to confidence intervals. Similar to the behavior experienced by Reddy and Riley, we note that despite the limited number of congestion packets employed, rounds frequently fail to complete due to the source packet being dropped as a result of MAC layer congestion.

In implementing the network functionality we identify all nodes other than the source and destination nodes as "intermediate" nodes. Callback handlers are installed on each intermediate node to implement the forwarding behavior of the network, while still ensuring a 500 microsecond processing delay on all received packets. Both the source and congestion packets are identical in size, however all source packets generated are "tagged" with a unique payload in order to differentiate them from other packets introduced at the congestion burst time. In some simulations, the source packet is dropped from the network. A counter on the source node is used to detect this scenario and stop the simulation; if the source node detects that it has forwarded a large number of congestion packets without encountering a tagged source packet it will halt all simulator activity and end the simulation.

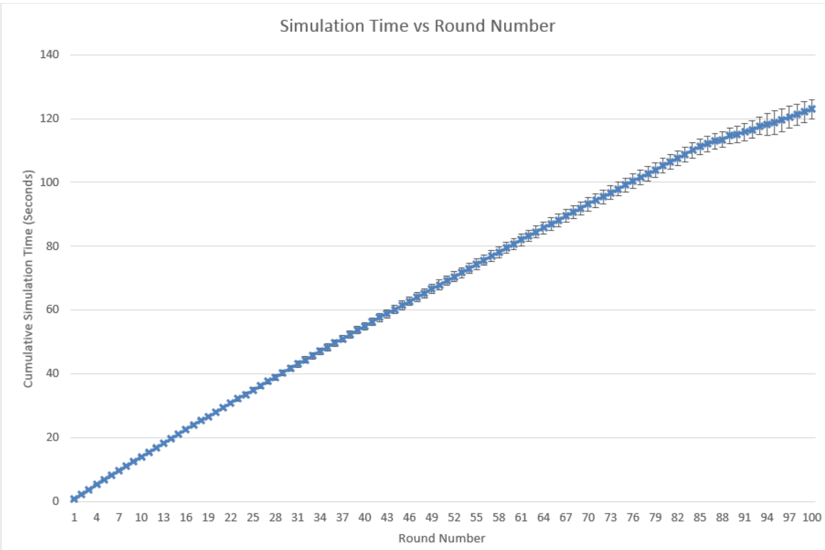
Results:



For Experiment 1, we conducted four sets of experiments choosing from Basic rate of 11 Mbps and 2 Mbps and with the option of with or without forwarding delay. We plot starting time of each round against the number of corresponding round. Each experiment was run 20 times and the results are averaged. Our results are in agreement with theoretical results presented in the tables above and with results of the other three simulators as presented in the paper. This proves that deterministic results of NS3 are in accordance with any other simulator. This forms the basis for experiment 2 where we make comparison of NS3 with other simulators in scenarios with randomness.

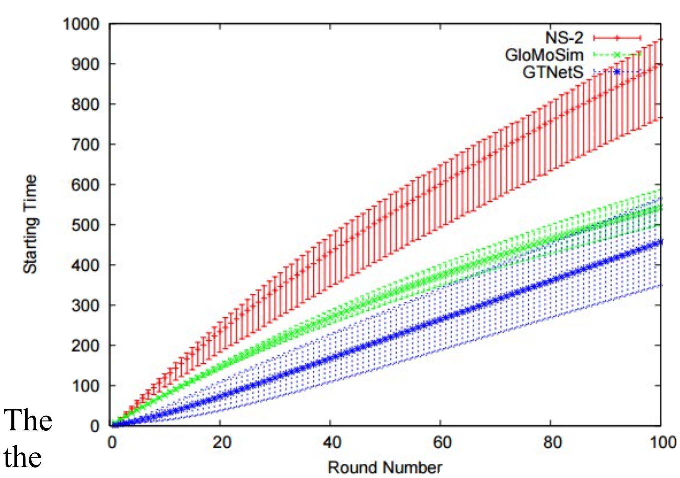
	Simulation	Theoretical
11 Mbps – 500µs delay	15.66s	15.89s
11 Mbps – no delay	13.34s	13.89s
2 Mbps – 500µs delay	17.32s	17.00s
2 Mbps – no delay	15.00s	15.00s

For experiment 2, we set Basic rate as 11 Mbps and use 500 µs forwarding delay. Shown below are the results from our experiment 2 trials. As mentioned earlier, we ran 30 iterations of experiment 2, halting the simulation when the generated source packet completed 100 runs through the network topology. In exactly half of the 30 runs we completed, we noted that the source packet was dropped due to congestion. Specifically, we note that on average across all 30 experiments we are able to complete 95.2 rounds per experiment. This is a much more successful average than any of the reported results for ns-2, GTNetS, and GloMoSim, however this is expected given the reduced number of congestion packets being introduced at the congestion burst time.



Plotted are the average simulation times for our 30 experiment iterations in addition to 90% confidence intervals for each round number. We can identify a few trends from inspection, namely a slow but steady increase in runtime variation stemming from channel congestion. This manifests itself in the form of larger confidence intervals as the round number increases. Additionally, we can see a marginal tapering of the trend line as the experiments continue, reflecting a reduction in average simulation time per round as the simulation runs over time. Comparing these ns-3 results to those put forward by Reddy and Riley, it is easy to see that our experiment produced much less volatility from randomness during experimentation. This can be attributed to the greatly reduced amount of network congestion in our experimentation, and again it bears stating that we did not completely flood the network with congestion packets. Instead we employed the comparatively small amount of 10 congestion packets, seeded evenly throughout the topology in order to decrease runtime.

In analyzing the slow tapering off of simulation time we can draw parallels to the results for NS-2 and GloMoSim, albeit with less pronounced effect. It is likely that in our experiments we can attribute this to the same root cause evaluated by Reddy and Riley, which posits that as the simulation continues to run fewer packets are present in the network over time as the congestion packets are dropped.



Overall we see that our recreation of experiment two is able to mirror expected simulation time data in the face of randomness from network congestion, as compared to the paper by Reddy and Riley. While significant differences exist in terms of the magnitude of our overall simulation time, this can easily be attributed towards differences in the implementation of our experiment (number of congestion packets). The impact of this choice is also reflected in the average number of rounds completed per iteration (95.2 rounds), which demonstrates as a rough measure just how little congestion we experience with 10 congestion packets as compared to the 100 used by Reddy and Riley.

Future Work:

It remains to be seen how ns-3 will behave given a full recreation of the experiment by Reddy and Riley. While we are able to establish similar baseline behavior for single packet and multi-packet experiments, we did not conduct a full 100 packet congestion of our network topology for tractability reasons. We expect that the results would be similar to those put forward already, and do not expect any major departures from established trends (confidence intervals, trend line behavior, run time trajectory), however it bears further experimentation to confirm this.

References:

[1] D. Reddy, G. F. Riley, B. Larish, Y. Chen, "Measuring and Explaining Differences in Wireless Simulation Models," *14th IEEE Symposium on Modeling, Analysis, Simulation*, Sep. 2006.