CS 525M – Mobile and Ubiquitous Computing Seminar

End-to-End TCP Performance Evaluation over Wireless Networks

Devanshu Mehta

Abstract

Traditional end-to-end protocols built for wired networks.

Modifications suggested for wired-cum-wireless networks need to be evaluated.

Using NS I will evaluate end-to-end protocols

I will mainly focus on energy consumption, though my results include confirm previous observations about throughput.

Though there are obvious limitations to my approach, I believe it is a good platform for future research in this direction.

Background

- There has been some work in this direction- most notably Singh, et al. at Portland State
- That paper actually built a network to test their results.
- I will be simulating the a similar environment using NS.
- This may seem counterproductive- moving from a built system to a simulation. Reasons?
    - Power of a good simulation
    - Time limitations of this class!

Phases: The things we do to avoid real work!

Phase 1: Got a grip on NS & Tcl

Phase 2: Got cygwin working on Windows.

Phase 3: Got NS-2 working on cygwin.

Phase 4: Scrapped all that, installed Linux and started again!

…and then I started the REAL WORK!

Phase 1: Created methodology

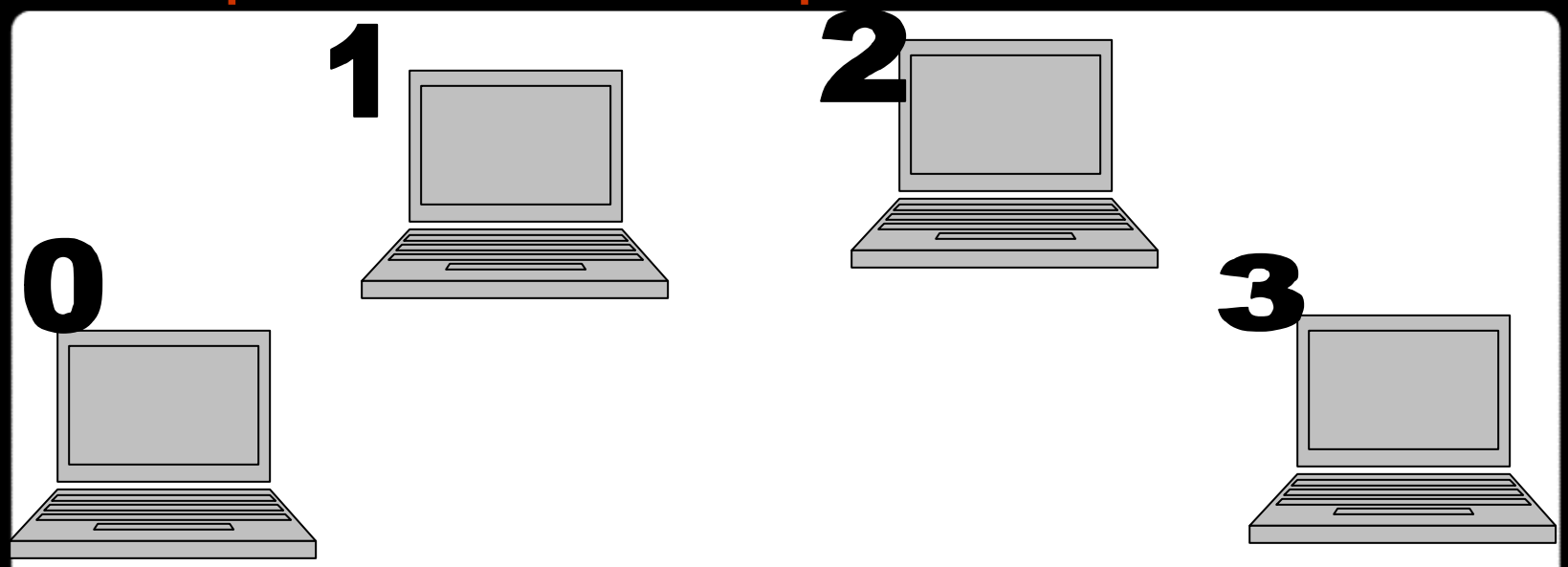
Phase 1: Built topology

Phase 1: Coded, hacked, pieced together

Phase 1: Got Results

Phase 5: Made sense of results!

Experimental Setup



EXPERIMENTAL SETUP

- Four Nodes
- Ad Hoc Routing (DSR)
- MAC 802.11
- Continuous FTP transfer from Node 0 to Node 3 for 40 seconds

METRICS

- GOODPUT: Measured as ratio of unique sent packets to unique received packets
- ENERGY/Mbit: Energy in Joules consumed per Mbit tranferred

Experimental Design

Experimental Parameters for Case I : Uniform Loss

Parameter	Values
Packet Loss	1%, 5%, 10%
MTU Size	512, 1500
RTS/CTS	ON, OFF

Experimental Parameters for Case II : Bursty Loss

Parameter	Values
Packet Loss	85% for 1s every 12s
MTU Size	1500
RTS/CTS	ON, OFF

A Grand Total of 42 Experimental Runs!

Results: Uniform Loss ->Energy Consumed

	RTS ON (J/Mbit)	RTS OFF (J/Mbit)
Reno	0.9303947	0.662709118
Newreno	0.9191457	0.663426958
SACK	0.8667037	0.659412242

	512 byte	1500 byte
Reno	0.914420107	0.678683668
Newreno	0.903888986	0.678683668
SACK	0.919937749	0.60617815

	0.01% loss	0.05% loss	0.1% loss
Reno	0.676684949	0.729337	0.98363361
Newreno	0.676684949	0.732715	0.96445939
SACK	0.583825114	0.709042	0.9963071

Bursty Losses: More Energy Consumption

	Bursty Loss (Joules/Mbit)
Reno	0.49
Newreno	0.49
SACK	0.46

Overall performance

	All Cases (Joules/Mbit)
Reno	0.79655189
Newreno	0.79128633
SACK	0.76305795

Results: Goodput

- I have a similar set of results for goodput; you should thank me for not displaying all of them!
- Goodput results confirm observations of past experiments.
- In most cases, Reno and NewReno perform similarly.
- In case of heavier losses, New Reno outperforms Reno.
- SACK outperforms both of the others.

Making Sense of it All

- Keeping RTS/CTS ‘ON’ involves overhead which utilizes energy.
- Larger packets seem more energy efficient, but this is probably because of NS. However…
- For lower losses, SACK is more energy efficient than the others. But as losses increase, the advantage shifts to NewReno.
- For bursty losses, SACK outperforms the others by a fraction. Longer runs may help verify these results…
- Overall, though, selection of protocol depends on type of traffic expected.

Measuring up to the Singhs

- Like I said before, I was modeling my experiments after a paper by Harkirat Singh and Suresh Singh.
- So how did I measure up?
- Results match!
    - Even a small result, that SACK performs poorly only when the loss is 0.1 is the same! Hurray…
    - Do not match where experimental setups differ: idle energy, packet re-ordering
- Still, I believe simulations will offer more flexibility in conducting experiments which can then be validated in real-world tests.

Experimental Conclusions: The Future

There are many limitations to my experiment design

- Limitations of NS's energy model
- Mobility
- Naïve routing
- Being an NS newbie…

However, I believe that I have a solid framework on which to base future research.

Improvements can be made to NS's energy model and more parameters can be added to the experiment to test for:

- Different routing algorithms
- Mobile nodes
- More TCP protocols
- Longer runs

After this, no more…

Any Questions?

Any Suggestions?

Thinking about the summer already?

Me too…