

Advanced Image Information: Self-Study (1)

Dept of Computer Science and Engineering, Jiro Katto

Try the questions below and submit a report (PDF) to CourseN@vi 「画像情報特論(2012)」[4/20]「理解度確認 / Self-Study」.

※If you cannot login to CourseN@vi, send your report by e-mail to katto 'at' waseda.jp.

Deadline: 2012/5/10(Thu)

[Preparation]

For Windows/Mac users, VirtualBox (<https://www.virtualbox.org/>) is convenient and any kinds of Linux distributions (CentOS, Ubuntu and so on) can be installed.

[Questions]

Download and install ns-2 from the ns-2 site, and download scripts.tgz from CourseN@vi (or class home page). Then, answer to the next questions.

- (1) Read “one-flow.tcl” and “two-flow.tcl” included in scripts.tgz and explain their simulation scenarios (e.g. network topology and flows) in conjunction with figures.
- (2) Read “run-linux-oneflow.csh” and “run-linux-twoflow.csh” and explain what are the purposes of the loop parameters in ns-2 simulations.
e.g. “flownum” specifies the number of TCP flows.
- (3) Let's focus on TCP-Reno, TCP-Vegas, CUBIC-TCP and Compound-TCP. For the cases that the router buffer size is 220 and 1066, execute single flow simulations by using “run-linux-oneflow.csh”, and report results and discussions.
- (4) Similar to (3), for the cases that the router buffer size is 220 and 1066, execute two flow simulations by using “run-linux-twoflow.csh”, and report results and discussions.
(note 1) For TCP evaluations, efficiency and friendliness are popular metrics.
(note 2) Simulation results are different when the order of flows is changed.
- (5) TCP implementation codes written by C-language are located at tcp/linux/src of the ns-2 installation directory. Read C-codes below and summarize behaviors of the three functions, XXX_cong_avoid, XXX_ssthresh, and XXX_min_cwnd, commonly included in the codes.
 - ① tcp_cong.c : TCP-Reno (loss-based)
 - ② tcp_vegas.c : TCP-Vegas (delay-based)
 - ③ tcp_cubic.c : CUBIC-TCP (loss-based)
 - ④ tcp_compound.c : Compound-TCP (hybrid)

【Appendix】

○ Note :

1. ns-2 has many versions. ns-2.33 and later version include TCP-Linux automatically, but older versions need a patch (see <http://netlab.caltech.edu/projects/ns2tcp/linux/ns2linux/index.html>).

○ BDP (Bandwidth-Delay Product)

In the attached script case, since bottleneck link bandwidth is 100Mb/s, one-way propagation delay is 64ms, and router buffer size is 1500byte, the number of packets corresponding to BDP becomes $100 \times 10^6 \times 2 \times 64 \times 10^{-3} / (1500 \times 8) \doteq 1066$ packets.

○ Files overview

Files included in scripts.tgz are based on the script files published at

<http://netlab.caltech.edu/projects/ns2tcp/linux/ns2linux/>

You can also modify them for the simulations.

[TCP-Linux original script]

- test-linux.tcl: simulation scenario (tcl)
- run-linux.csh: parameter update and simulation execution (shell script)
- script-gnuplot: gnuplot command to show cwnd temporal behavior

[modified script]

- one-flow.tcl: single flow simulation scenario (based on test-linux.tcl)
- run-linux-oneflow.csh: parameter update and simulation execution (based on run-linux.csh)
- script-gnuplot-oneflow: gnuplot command to show cwnd temporal behavior (based on script-gnuplot)

- two-flow.tcl: two flow simulation scenario (based on test-linux.tcl)
- run-linux-twoflow.csh: parameter update and simulation execution (based on run-linux.csh)
- script-gnuplot-twoflow: gnuplot command to show cwnd temporal behavior (based on script-gnuplot)

○ URLs

Network Simulator - ns-2: <http://www.isi.edu/nsnam/ns/>

nsnam (sourceforge.net): <http://sourceforge.net/projects/nsnam/>

NS-2 TCP-Linux: <http://netlab.caltech.edu/projects/ns2tcp/linux/ns2linux/index.html>