Network Simulator – Test Cases

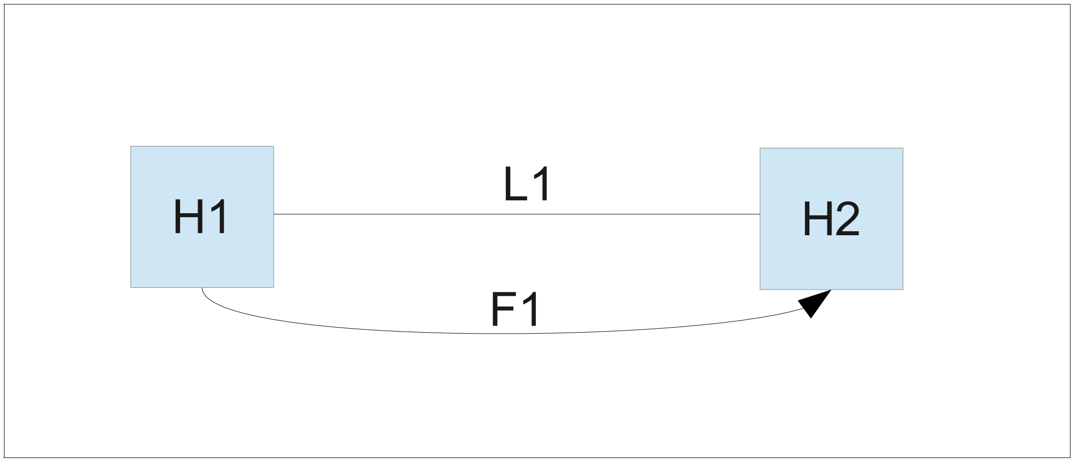
Overview

To help guide you in your project implementation, we provide a collection of test cases for you to help evaluate your simulator. Successful implementation of each test case is required for full project credit.

Observe that part of the project specifications include a means by which test cases can be inputted into the simulator. To test this capability, we will also evaluate your simulator against one or more test cases not published in this document.

In all cases, we want time traces for all flows as well as for the links we identify as 'links of interest'. Sample time traces are provided for the non-trivial cases, and a list of all of the metrics we want you to output can be found in the guidelines document.

Test Case 0



This is the simplest possible test case: two directly connected hosts with one flow running from one to the other.

This test case is designed to validate your host, link, packet, and basic flow code, as well as your basic network initialization code. It is advisable to build something the most basic implementation that will pass this test case before going on to try and implement the more complicated algorithms required later on.

Link Specifications

Link ID	Link Rate (Mbps)	Link Delay (ms)	Link Buffer (KB)
L1	10	10	64

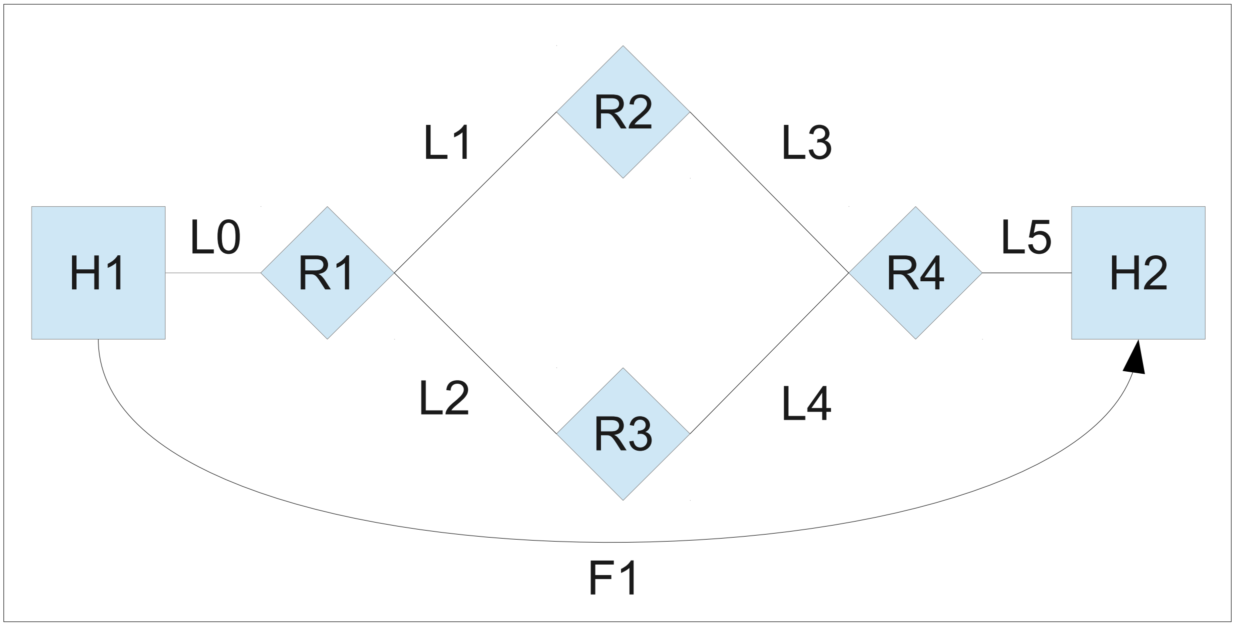
Flow Specifications

Flow ID	Flow Src	Flow Dest	Data Amt (MB)	Flow Start (s)
F1	H1	H2	20	1.0

Sample Results

Sample time traces are not provided for this test case.

Test Case 1



This test case is designed to test your router code, particularly the dynamic routing capability. If your dynamic routing is working correctly, you should see the single flow alternate between the top branch and the bottom branch as the link costs change in each iteration of the routing table calculations.

Link Specifications

Link ID	Link Rate (Mbps)	Link Delay (ms)	Link Buffer (KB)
L0	12.5	10	64
L1	10	10	64
L2	10	10	64
L3	10	10	64
L4	10	10	64
L5	12.5	10	64

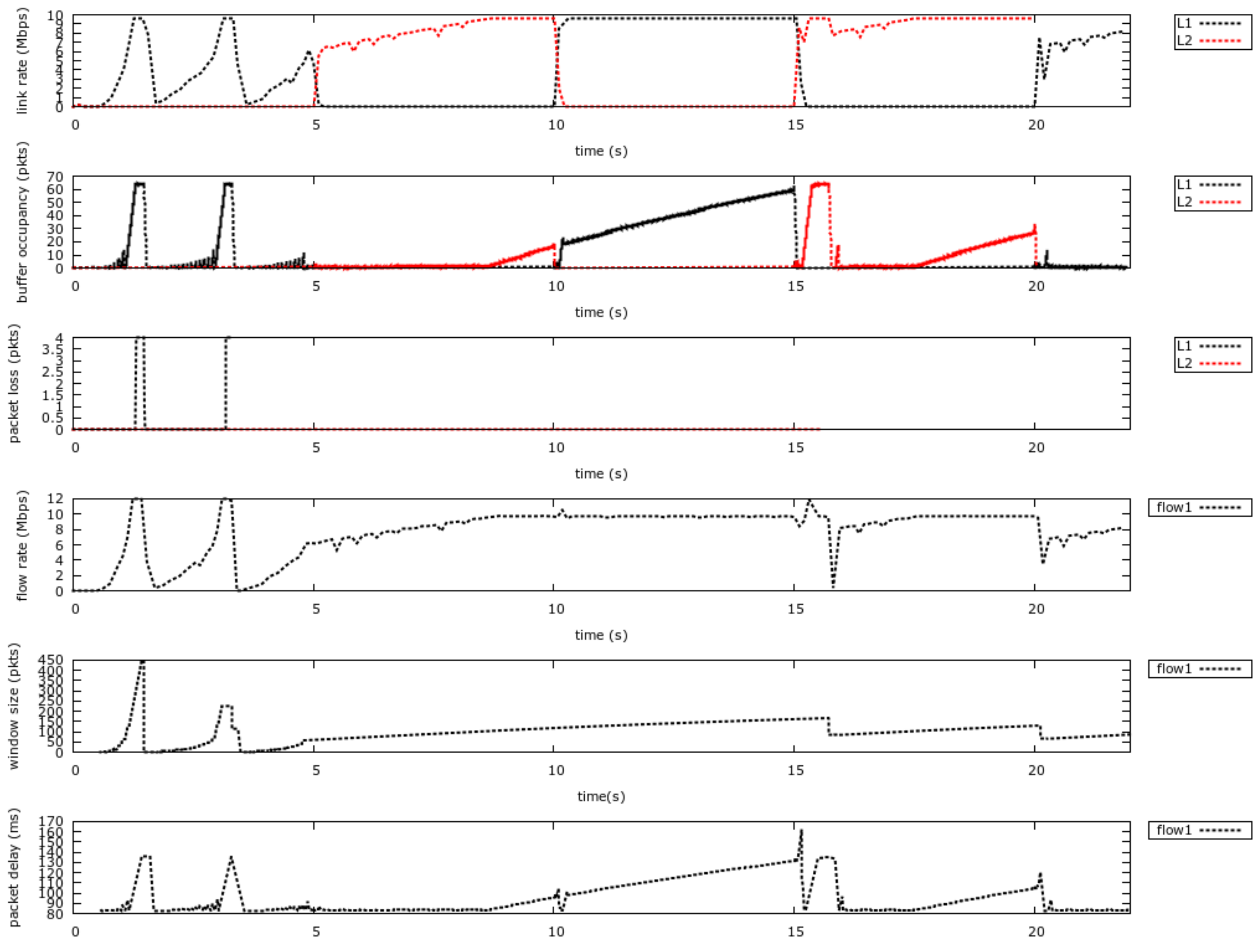
Flow Specifications

Flow ID	Flow Src	Flow Dest	Data Amt (MB)	Flow Start (s)
F1	H1	H2	20	0.5

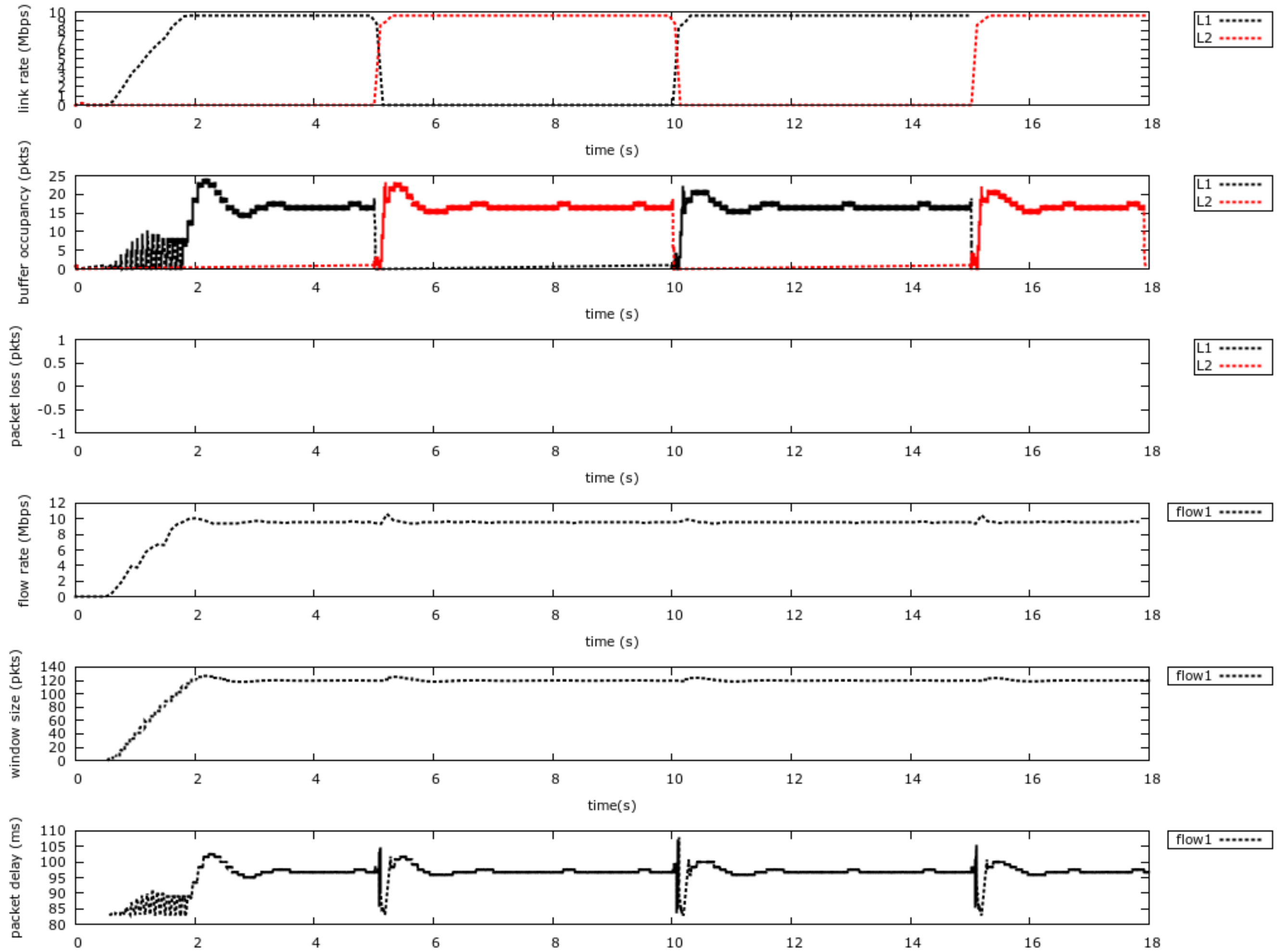
Sample Results

Note that your results may differ from the time traces presented here.

TCP Reno



FAST-TCP

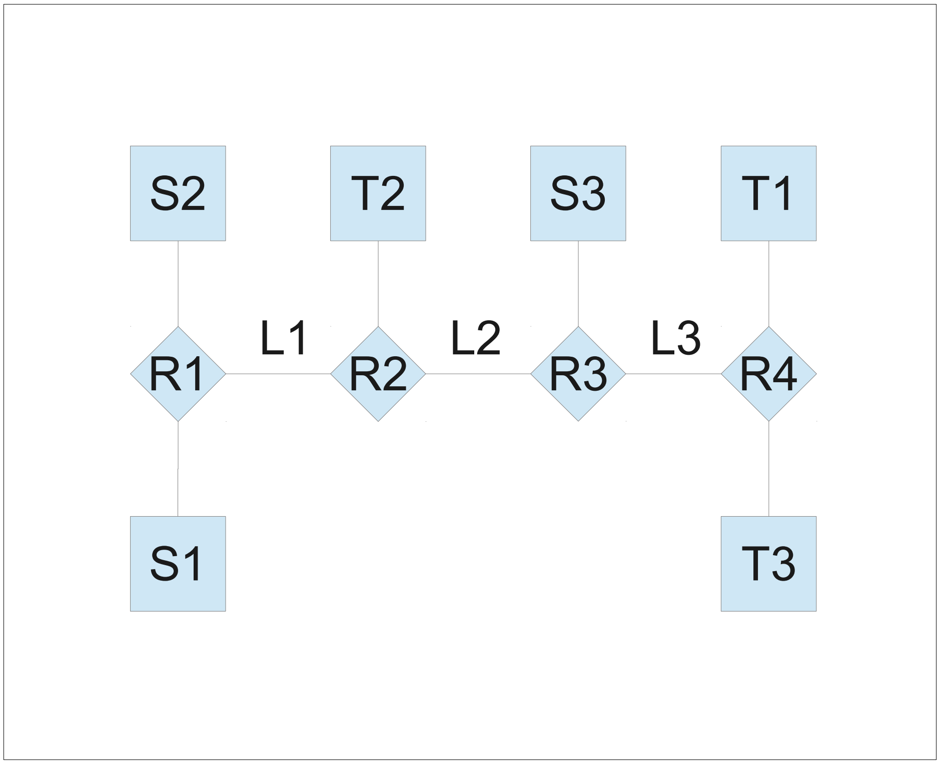


Additional Observations

Your time traces will differ depending on how frequently your routers rebroadcast their routing information, and how your implementation responds when doing so causes a large number of packets to arrive out of order.

If you discover that your Reno window size is crashing to 1 upon a timeout event, try to modify your equation so that it responds only to one timeout event per RTT. You will need to make many such tweaks to all of the algorithms you implement in order to get your simulation working as desired.

Test Case 2



This test case is slightly more complicated than the previous one, and involves multiple flows interacting with each other at different times. This test case demonstrates how different flows affect each other as each one tries to implement congestion control.

Link Specifications

Link ID	Link Rate (Mbps)	Link Delay (ms)	Link Buffer (KB)
L1, L2, L3	10	10	128
All Other Links	12.5	10	128

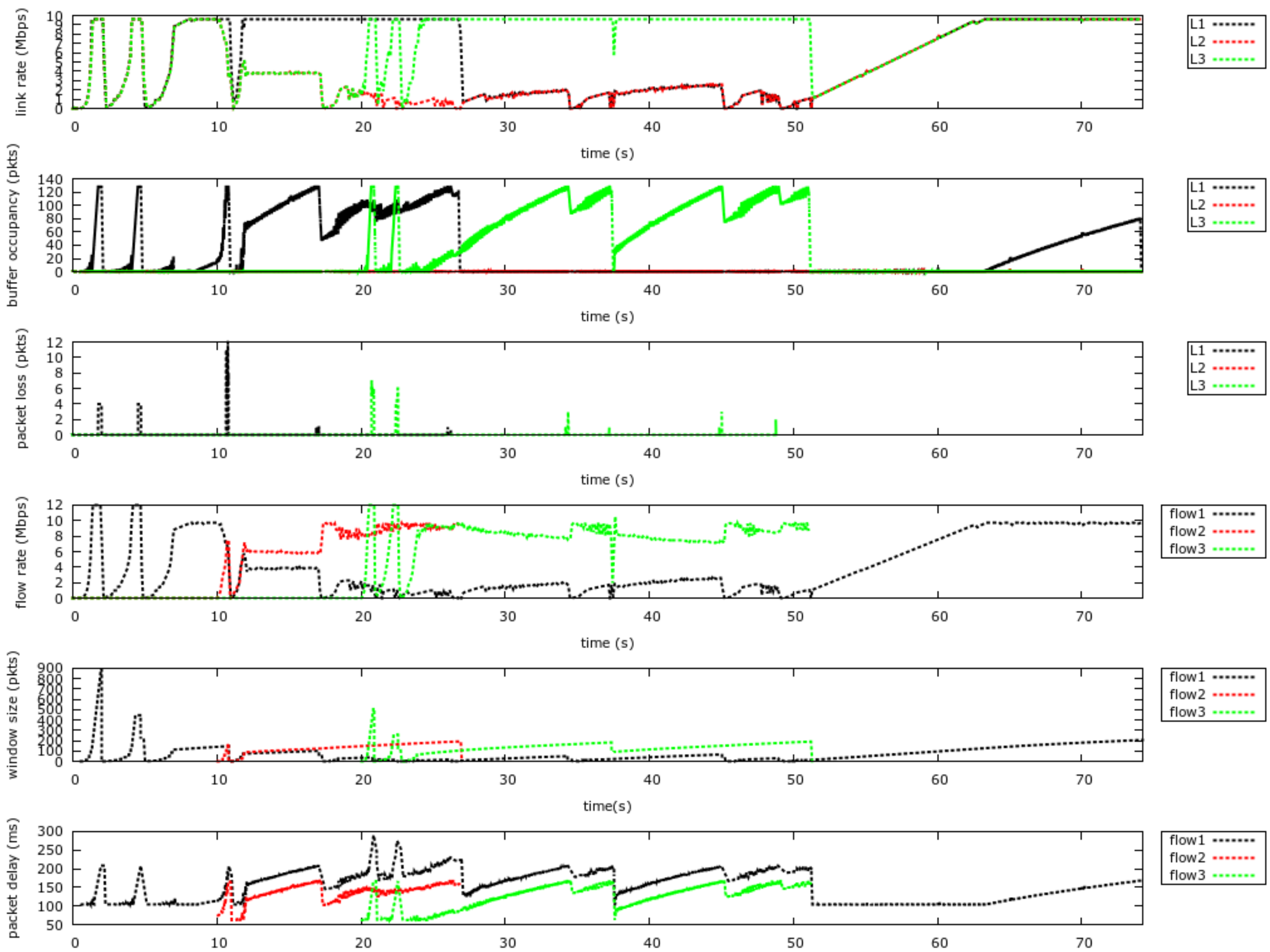
Flow Specifications

Flow ID	Flow Src	Flow Dest	Data Amt (MB)	Flow Start (s)
F1	S1	T1	35	0.5
F2	S2	T2	15	10
F3	S3	T3	30	20

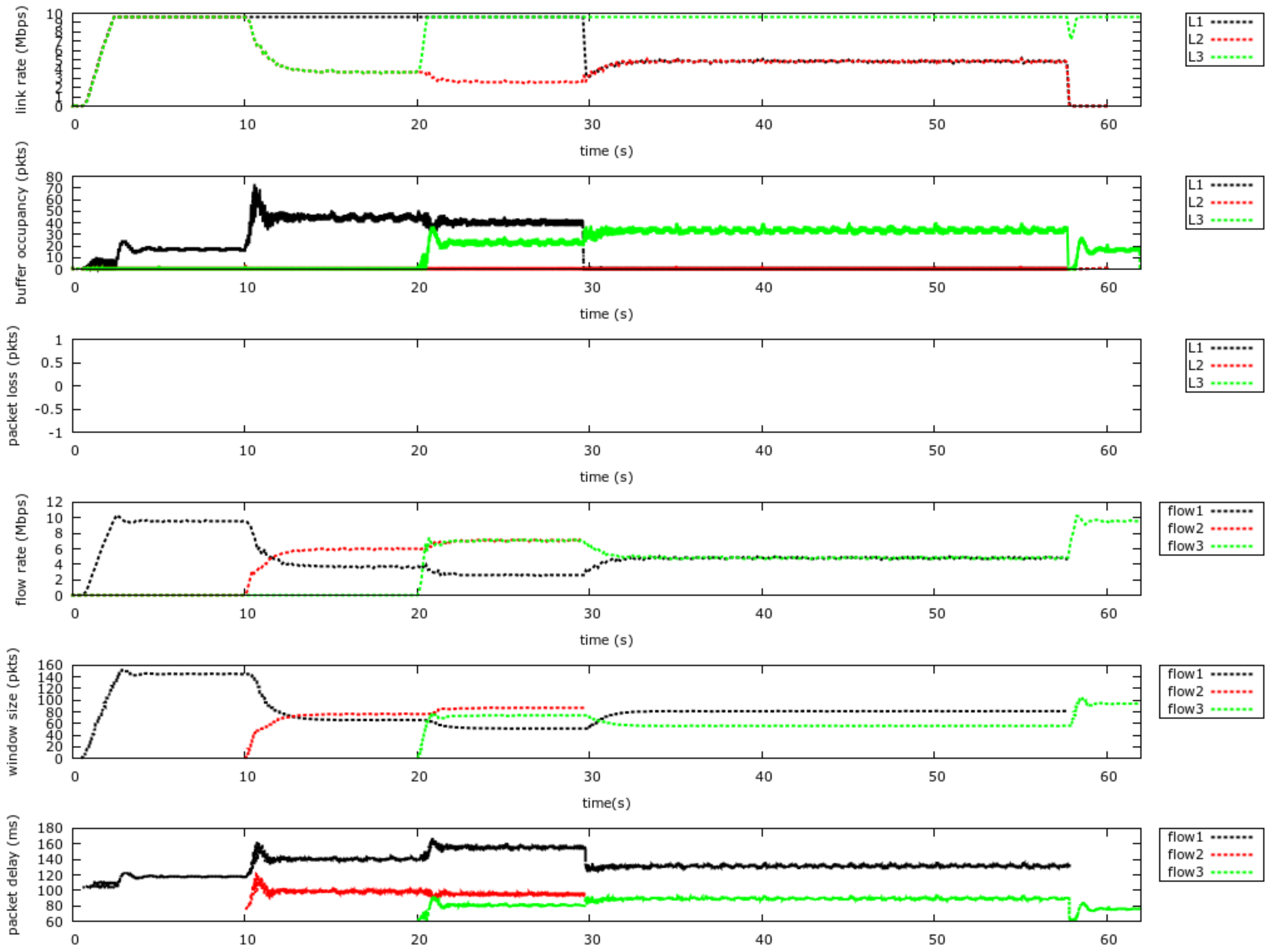
Sample Results

Note that your results may differ from the time traces presented here.

TCP Reno



FAST-TCP



Additional Observations

Notice what happens to the window sizes and flow rates in the FAST case as flows enter and leave the system. There is some behavior, unintuitive at first glance, that can be explained analytically; can you account for these?

(Hint: First set up the window size equation for a set of FAST flows at equilibrium, along with the queueing equations for the intermediate links. Then treat these as the optimality condition of a network utility maximization problem, find the corresponding utility functions, and find the requested maximum. Finally, see what happens if you introduce a perturbation in an important quantity; how does this affect the utility functions, and what does this do to the maximum of the system?)