Experimental Networking (ECSE 4963)

Lab 8, Linux TCP

Yong Xia
xiay@rpi.edu

Oct 17, 2002

Term Project

- Submit your initial proposal for term project by Nov. 1 to webct.

What you will learn?

- Network topology setup
- Linux TCP configuration
- Wide area network emulation
- Data collection, analysis and visulization
- Fairness measurement
- Random Early Detection

TCP Traffic Lab: Objectives

- To learn basic skills to setup network topology and TCP configuration, and collect / analyze / visualize experimental data;

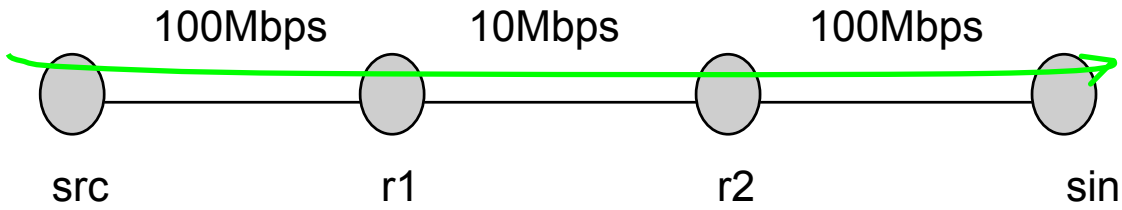
- To understand TCP congestion control algorithms with a single flow running under different conditions;

- To understand TCP dynamics / fairness when multiple flows contending for shared resources;

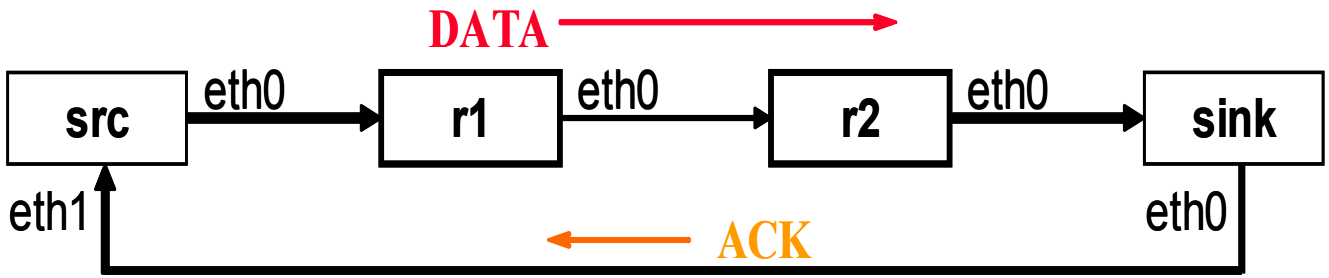
- To implement an Active Queue Mgmt mechanism.

Topology

Logical:



Physical:



- Why use different ACK path?
- How to ensure Data/ACK paths are corretly setup?
  - to setup: physical connections, routing tables
  - to test: traceroute, route, netperf

Linux TCP Settings

- TCP Version
  - TCP Reno vs. SACK
    - src#> echo 0/1 > /proc/sys/net/ipv4/tcp_sack
  - MTU and Path MTU
    - Ethernet MTU is 1500B
    - Default MTU is 576B
    - src#> echo 0/1 > /proc/sys/net/ipv4/ip_no_pmtu_disc

Simulation vs. Emulation

- Simulation
  - A synthetic test environment
  - Easy to repeat
  - Simplified than reality
  - Example: ns-2

- Emulation
  - A "live" implementation environment
  - Also repeatable
  - Closed to reality
  - Example: NIST NeT

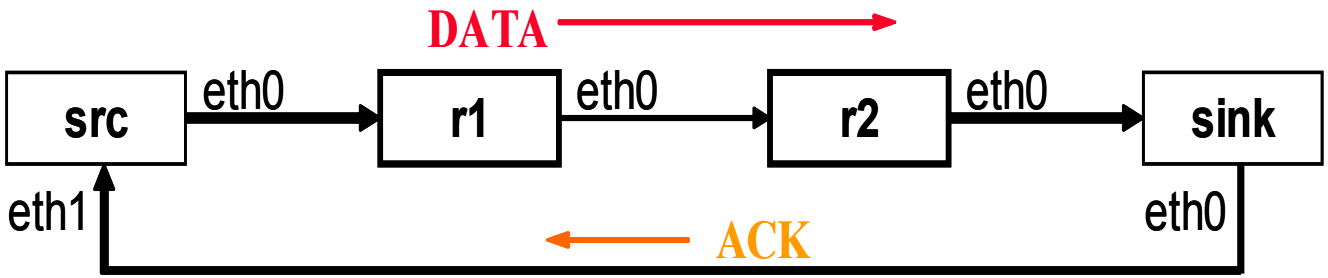
NIST Net Emulation Tool

- NIST Net emulation tool
  - http://snad.ncsl.nist.gov/itg/nistnet
  - The NIST Net network emulator is a general-purpose tool for emulating performance dynamics in IP networks. The tool is designed to allow controlled, reproducible experiments with network performance sensitive/adaptive applications and control protocols in a simple laboratory setting. By operating at the IP level, NIST Net can emulate the critical end-to-end performance characteristics imposed by various wide area network situations (e.g., congestion loss) or by various underlying subnetwork technologies (e.g., asymmetric bandwidth situations of xDSL and cable modems).

Lab Data Processing

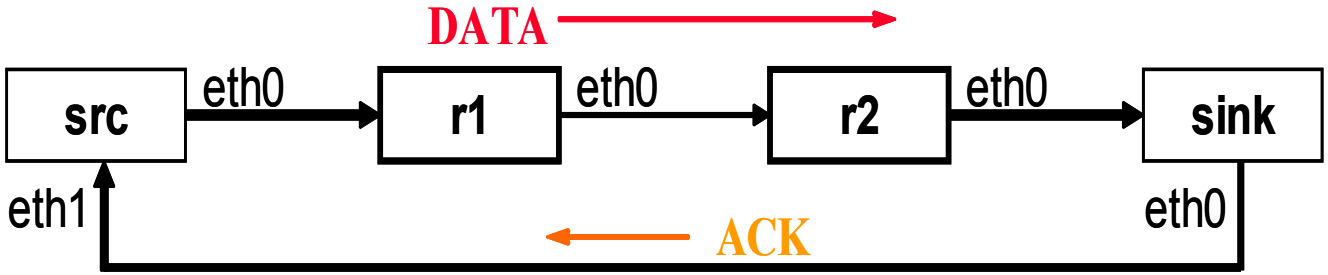
- Data collection
  - Tcpdump, see man page
    - src#> tcpdump –N –v –w trace.dmp host sink
- Data analysis
  - Tcptrace at http://www.tcptrace.org/manual.html
    - src> tcptrace –l –r –s –w –zxy –G –C trace.dmp
- Data visualization
  - Xplot at http://www.tcptrace.org/manual.html
    - src> xplot filename.xpl

TCP Traffic Lab: A Single Flow



- Repeat lab by changing
  - -Maximum Segment Size (MSS)
  - -Round Trip Time (RTT)
  - -Bottleneck Buffer (router r1)
  - -Asymmetric Channel (data vs. ack)
  - -Bursty Ack

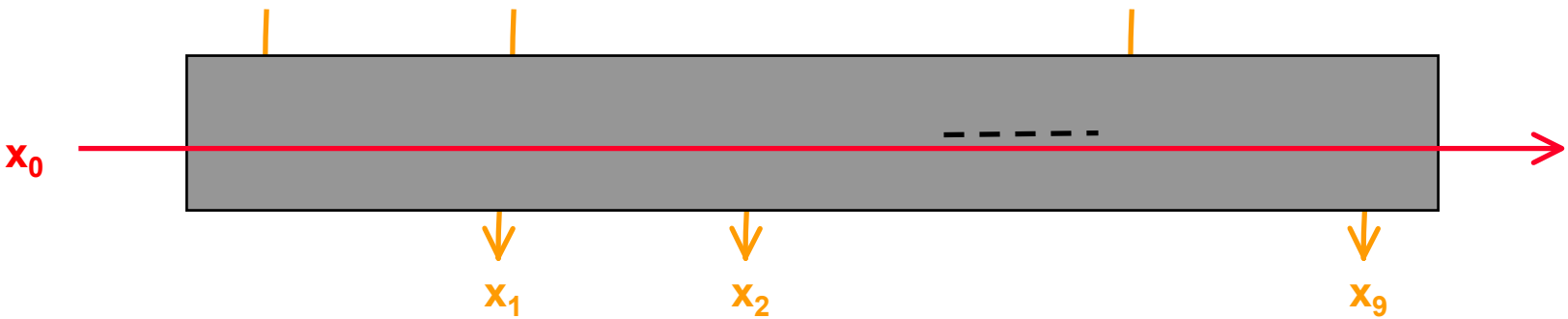
TCP Traffic Lab: Multiple Flows



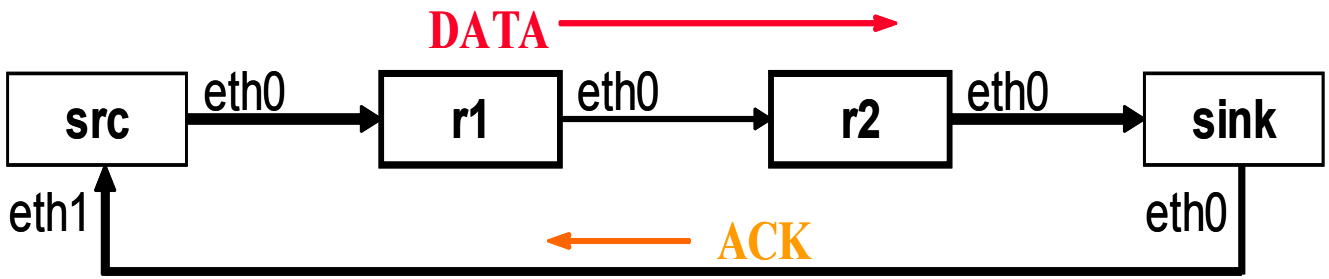
- To understand resources allocation fairness
  - Jain's fairness index
  - Max-min fairness
  - Proportional fairness
  - Coefficient of Variance (CoV)
- We use CoV
- Just change parameters in *tg.conf*
  - e.g. set the number of server/client to 50

what is fairness ?

- proportional fairness
  - the more a flow consumes critical network resources, the less allocation
  - network visible inside
  - network operators' view
  - $x_0^* = 0.1,\ x_{1\sim9}^* = 0.9$
- max-min fairness
  - every flow has the same right to all network resources
  - network as a black box
  - network users' view
  - $x_0^* = x_{1\sim9}^* = 0.5$



TCP Traffic Lab: AQM



- To understand the design rationales behind Random Early Detection
  - Control average queue size
  - Avoidance of global synchronization
  - Avoidance of bias against bursty traffic
- RED algorithm
- Implementation into Linux OS Kernel at r1

Assg #7

- Answers to all questions given in course material
- Write TCP socket program to emulate multiple users sending from src to dst, collect statistics at dst.
- Experiment results, including data and graphs
- Suggestions for improving the design of this lab
- Due Sunday Oct 27, 11:55pm