Measuring Application's network behaviour

EuroNGI PhD measurement workshop

University of Linz, Austria

May, 12th 2006

Sven Hessler http://dps.uibk.ac.at/~sven

Institute of Computer Science
University of Innsbruck, Austria

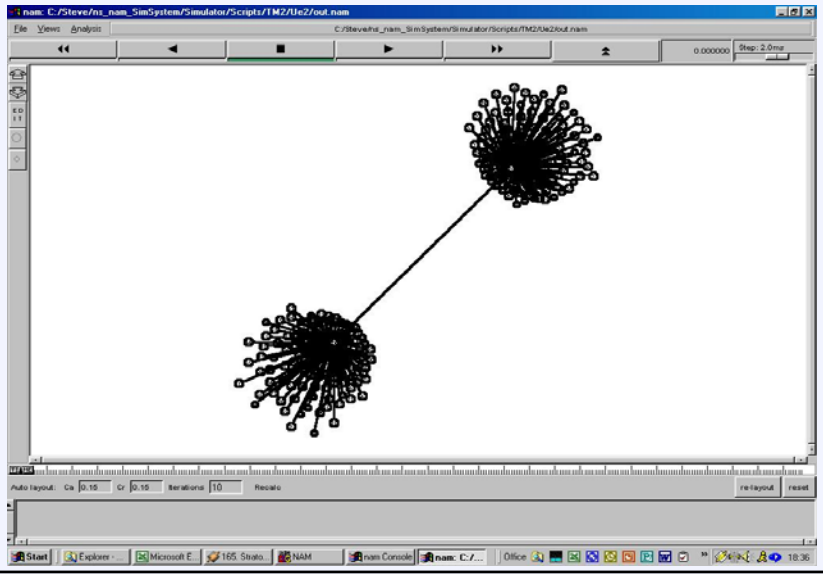
Outline

- Motivaton
- Our testbed and its limitations
- Measurement results
- Valueable tools for conducting network measurements
  - traffic control (tc)
  - traffic generators
  - network emulators
- From theory to RL: measurement demo

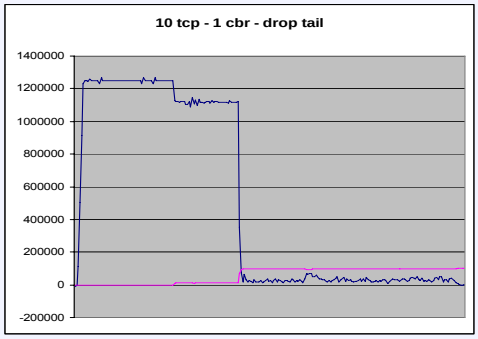
Today's Internet transport protocols

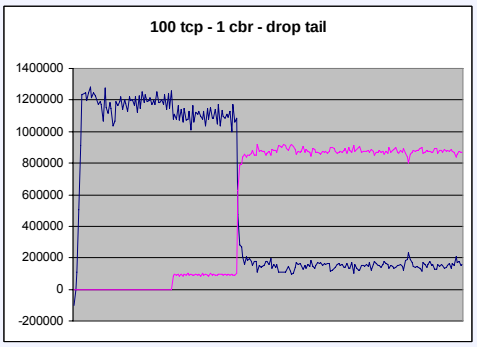
- UDP used for sporadic messages (DNS) and some special apps
- TCP used for everything else
  - in 2003, approximately 83 % according to:
    Marina Fomenkov, Ken Keys, David Moore and k claffy, "Longitudinal study of Internet traffic in 1998-2003", CAIDA technical report, available from http://www.caida.org/outreach/papers/2003/nlanr/
  - backbone measurement from 2000 said 98% ⇒ UDP usage growing
- Original Internet proposition:
  IP over everything, everything over IP
- Today's reality:
  IP over everything, almost everything over TCP, and the rest over UDP

TCP vs. UDP: a simple simulation example



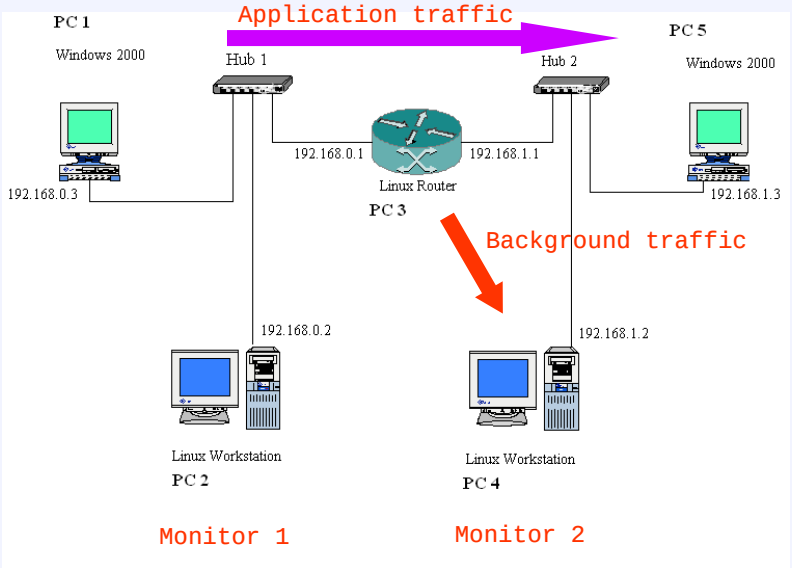
It doesn‘t look good





- For more details, see:
  Promoting the Use of End-to-End Congestion Control in the Internet. Floyd, S., and Fall, K..
  IEEE/ACM Transactions on Networking, August 1999.

Our measurement testbed



What we can measure (and what not)

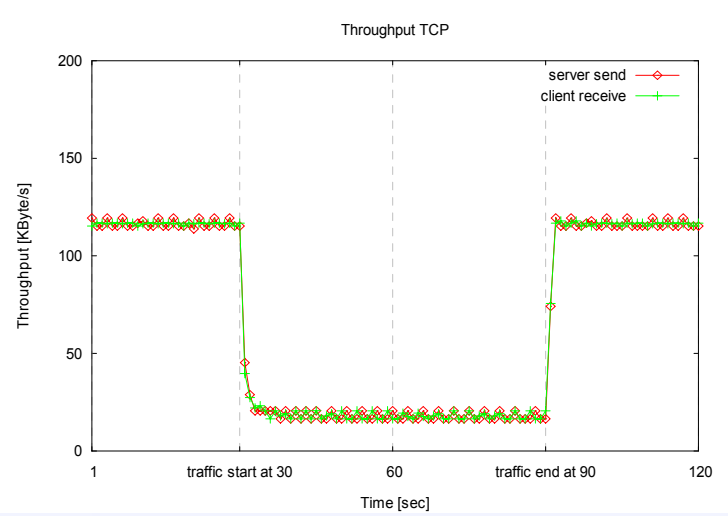
Yes:

- throughput
  - per flow
  - total (if more than one flow)
- delay
- congestion (send ICMP echo request, receive ICMP response)
- loss
  - defined as difference between the sent and received bytes

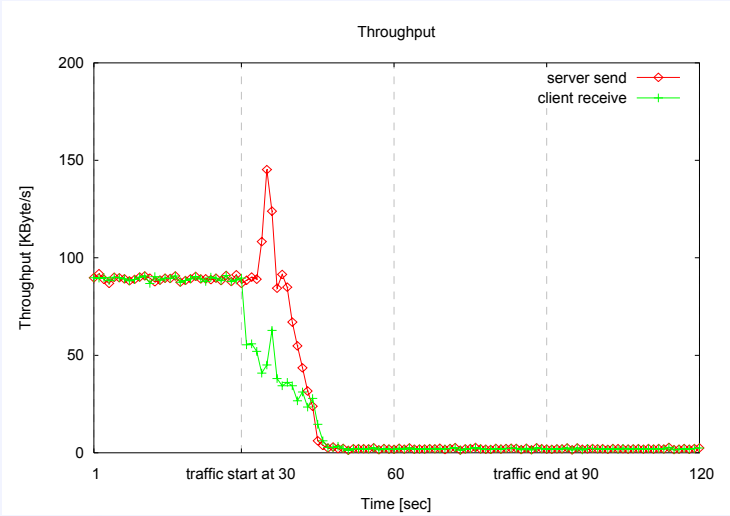
Limitations:

- very limited setup (but useful anyways)
- no realistic (live) cross traffic, just traffic generators
- we do not regard interaction/effects between applications

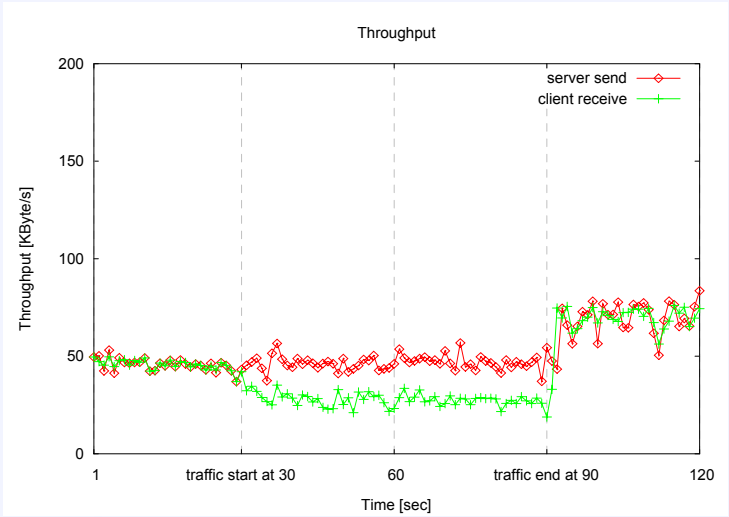
Results: TCP (the way it should be)



Results: Streaming Video: RealPlayer



Results: Streaming Video: Windows Media Player



VoIP: MSN

VoIP: Skype

Valuable tools for conducting network measurements

- Limit bandwidth (using tc)
- Generate cross traffic (traffic generators)
- emulate an intermediate network (network emulators)

Traffic control (tc) in the Linux kernel

- Shaping (egress)
  - controls sending rate
  - e.g. limit available bandwith, smooth out bursts
- Scheduling (egress)
  - prioritizing traffic
  - e.g. improve interactivity for one traffic class while guaranteeing bandwith for another
- Policing (ingress)
  - allows/denies incoming traffic
- Dropping (ingress & egress)
  - traffic exceeding a set bandwith may be dropped

tc explained

#> tc qdisc add dev eth0 root tbf rate 220kbit latency 50ms burst 1540

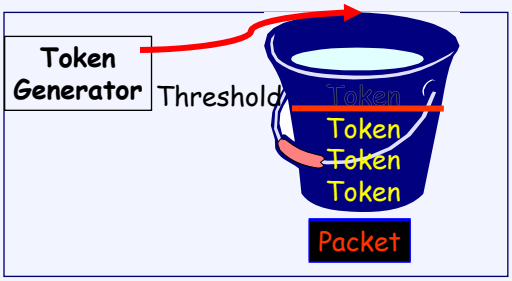
queueing discipline

add to root of device 'eth0'

Tocken Bucket Filter:

- send at rate 220kbit (if tokens are available)
- packets can sit max. 50ms in the bucket
- size of the bucket (1540 bytes)

A Token-Bucket-Filter



tc classless queueing disciplines

- tbf – Tocken Bucket Filter
- sfq – Stochastic Fair Queueing, i.e. send data in a 'round-robin-like' manner
- pfifo_fast – TOS-aware queue with 3 bands, i.e. within each band FIFO rules apply
- red – Random Early Detection (stochastic dropping of packets)

tc classfull queueing disciplines:

- PRIO – non-shaping container for a configurable number of classes dequeued in order, TOS-aware
- CBQ – Class Based Queueing, implements a hierarchy of classes for shaping and prioritizing
- HTB – Hierarchy Token Bucket
- Theory of classes:
  - form a tree
  - each client has a single parent, a parent may have multiple children
  - when a packet arrives --> classification into one of the classes:
    - tc filters – tc filters are attached to classes and consulted first
    - TOS – Type of Service (TOS) field (QoS extension)
    - skb->priority – userspace programs encode priority (SO_PRIORITY)

Traffic generators

- generate traffic (mostly) according to a mathematical model (e.g. poisson, exponential, self-similar)
  - mgen – multi message generator (UDP/IP traffic)
  - ttcp – test tcp throughput
  - Realtime voice traffic generator
  - ...

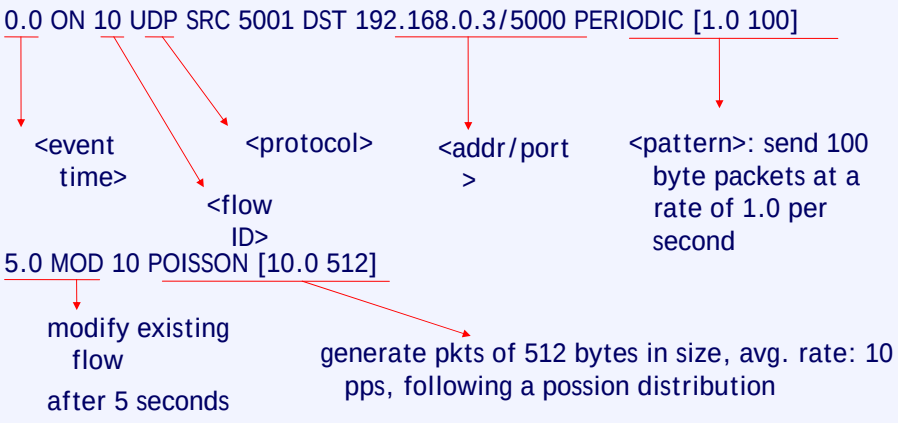
Excellent overview of free and commercial traffic generators:
http://www.grid.unina.it/software/ITG/link.php

MGEN

multi message generator for UDP/IP traffic

- generates real-time traffic patterns so that the network can be loaded in a variety of ways
- available for Unix-based (incl. MacOS X) and Win32 systems
- support unicast and multicast
- uses script files for driving tests

MGEN sender: setup the configuration file



MGEN: pattern types

Syntax:

… <pattern type> [parameters] …

Pattern types:

- PERDIODIC [10 1540]
  - send 10pps with a size of 1540 byte each
- POISSON [10 1024]
  - send in avg. 10pps with a size of 1024 byte each
- BURST [RANDOM 10 PERIODIC [10 100] EXP 5.0]
  - send bursts of 100 byte packets with a periodic rate of 10pps
  - bursts occur in random intervals with avg. from the start of one burst until the start of the next of 10 seconds
  - burst duration is exponential with avg. duration of 5 seconds

Starting MGEN

MGEN is a client-server application:

- @ sender-side: mgen ipv4 input crosstraffic.mgn
- @ receiver-side: mgen ipv4 input receive.mgn

crosstraffic.mgn:
0.0 ON 10 UDP SRC 5001 DST 192.168.1.2/5000 PERIODIC [1 100]
[…]

receive.mgn:
0.0 LISTEN UDP 5000-5001

Network emulation/simulation tools

- emulab (http://www.emulab.net/)
- PlanetLab (http://www.planet-lab.org/)
- dummynet (http://freshmeat.net/projects/dummynet)
- NISTNet (http://www-x.antd.nist.gov/nistnet/)

pretty complete overview: two bachelor thesis @University of Innsbruck:
http://www.welzl.at/research/projects/testbedeval/index.html

Quick overview: emulab

- emulab is a cluster of computers with access for the public (researchers):
  - University of Utah (168 nodes, P3 class)
  - Georgia Tech (40 nodes, P3 class)
  - Kentucky (48 nodes, P3 class)
- emulates complex networks, e.g. emulate edge nodes, traffic shaping nodes, traffic generators
- while experiments are running, it gets exclusive access to assigned machines, incl. root access
- uses a modified network simulator (NS2) script to describe network topologies

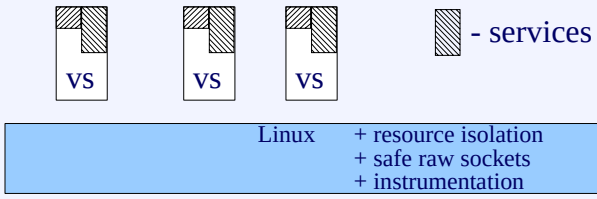
Quick overview: planet lab

- planet lab is a test environment of globally distributed computers:
  - ~ 460 machines (> 1.5GHz, > 0.5GB RAM, > 100GB HDD)

terms and definitions:

- nodes: each site has to provide at least 2
- slices: number of shared resources on on or more nodes
- virtual server: environment of an slices
- sliver: a part of a slice or an application running on virtual server

Planet lab: network measurement services



- ScriptRoute (UWashington)
- PlanetProbe (Cambridge)
- Network Weather Service (UTK)
- Ping (everyone)

"Using planet lab for network research: Myth, Realities, and Best Practices"
www.cs.princeton.edu/nsg/papers/myths_worlds_05/myths.pdf

Quick overview: dummynet

- is a feature of the FreeBSD packet filtering firewall 'ipfw'
- dummynet bases on pipes:

Delay ICMP messages:

```
ipfw add pipe 1 icmp from any to any out        # create pipe for icmp-traffic
ipfw pipe 1 config delay 100ms                  # config delay for icmp-traffic
ipfw add allow icmp from any to any             # allow icmp traffic
```

Limit bandwidth webtraffic from 192.168.0.2 to 128Kbit/s:

```
ipfw add pipe 1 tcp from 192.168.0.2 port 80 to any
ipfw pipe 1 config bw 128Kbit/s
ipfw add allow ip from any to any
```

NISTNet

- emulates any degree of "real-network badness" in a small, lab-environment network
- works on the Linux (2.0.x/2.2.x/2.4.x/2.6.x) IP-forwarding rules
- offers a commandline interface and a GUI

- functions:
  - packet delay: fix or variable (jitter)
  - packet reordering (due to very long delay)
  - packet loss (random or congestion dependent)
  - packet duplication
  - bandwidth limits

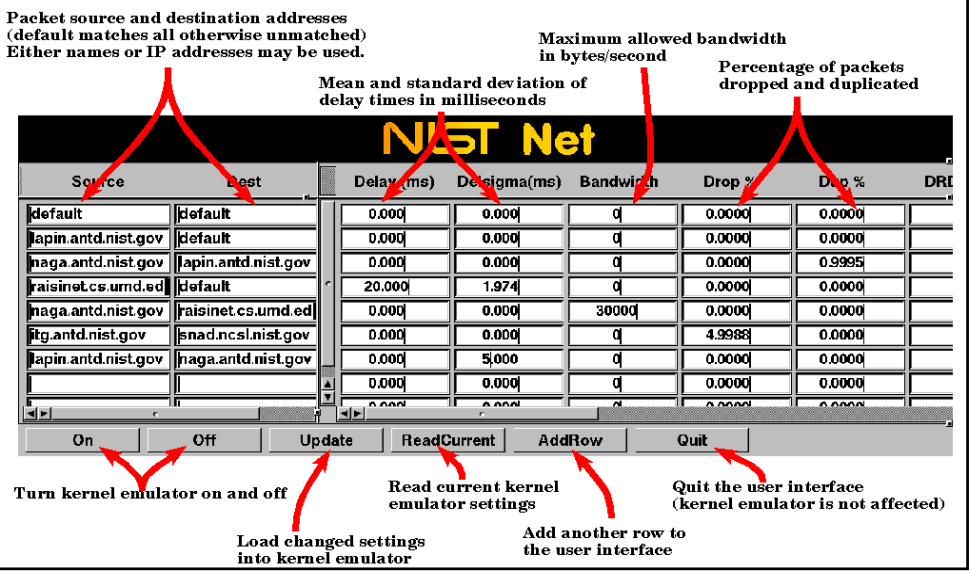
NISTNet: how to use

- get NISTnet
- get the Linux kernel sources
- make & install linux kernel (2.0.xx, 2.2.xx, 2.4.xx, 2.6.xx)
- configure & make NISTnet

- Load Nistnet modules (modprobe nistnet)
- configure emulator (xnistnet, cnistnet)
- start emulator

- start measurements

xnistnet: the graphical user interface



References

Recommended reading

- Recommended URLs:
  - Traffic generators:
    - overview: http://www.grid.unina.it/software/ITG/link.php
    - MGEN: http://pf.itd.nrl.navy.mil/mgen/mgen.html
  - Network emulators:
    - NISTNet: http://snad.ncsl.nist.gov/nistnet/
    - Emulab: http://www.emulab.net/
  - Passive measurement traces:
    - CAIDA: http://www.caida.org
- Neil Spring et al. "Using planet lab for network research: Myth, Realities, and Best Practices": www.cs.princeton.edu/nsg/papers/myths_worlds_05/myths.pdf
- Michael Welzl: "Network Congestion Control: Managing Internet Traffic", John Wiley & Sons, July 2005

Thank you for your attention!
Questions ?

Demonstration…