Traffic Measurement and Analysis (2)

JAIST Lecture 2003/11/28

Kenjiro Cho
JAIST/Sony Computer Science Labs, Inc.
kjc@csl.sony.co.jp

measurement metrics

- connectivity
- throughput
- delay
- path
- routing

measurement techniques

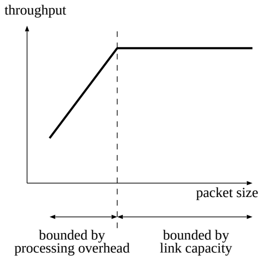
- data reduction techniques
  - filtering: e.g., record only TCP SYN packets
  - aggregation: e.g., flow-based accounting
  - sampling: e.g., record 1 in n packets
- active and passive measurement
  - active: injects measurement packets
  - passive: monitors network without interfering in traffic
- related research area (not covered in the class)
  - visualization
    - how to understand massive information
  - intrusion detection/DDoS detection

throughput measurement

- throughput
  - bits/sec (bps)
  - packets/sec (pps)
  - throughput is average by definition
- benchmarking
  - test with actual load
  - mainly for lab environment
- bandwidth estimation
  - estimates bandwidth without overloading network

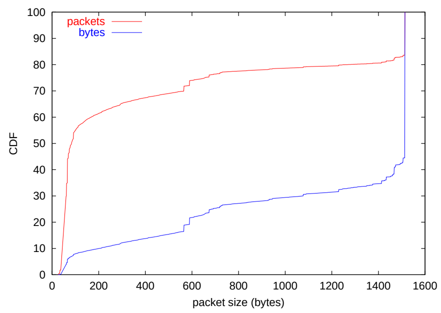
packet processing overhead

- overhead constant for a packet
  - e.g., header processing time
- overhead proportional to packet size
  - e.g., data transfer time



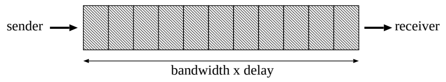
packet size in real traffic

- majority of packets are small
- majority of bandwidth consumed by large packets

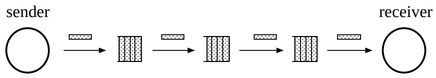


benchmarking

- benchmarking tools for UNIX
  - netperf, ttcp, treno, sting, etc
- TCP is often better than UDP
  - UDP overflows all buffers but TCP adapts to available bandwidth
  - set TCP buffer size larger than delay bandwidth product

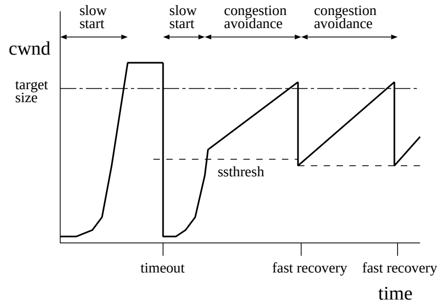


  - delay becomes larger under congestion by buffers at routers



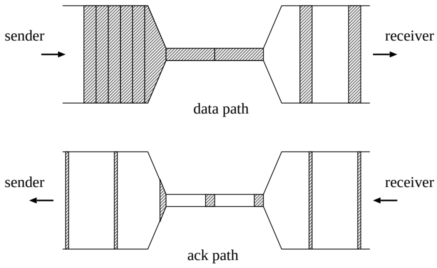
TCP congestion control

- congestion window controls volume of packets in flight
  - slow start/congestion avoidance
  - retransmit timeout
  - fast retransmit/fast recovery



TCP self-clocking

- reception of ack triggers next packet transmission
- adapts to bottleneck bandwidth



bandwidth estimation

- pathchar
  - bandwidth estimation by round-trip time
  - algorithm
    - per hop measurement by TTL (as traceroute does)
    - repeated measurement with varying packet size
    - pick up minimum round-trip time for each packet size
    - linear regression to obtain propagation delay and bandwidth
- limitations
  - many measurements required
  - accumulation of errors
    - especially narrow link behind fat link
  - only one-way
- open-source pathchar clone: pchar, clink

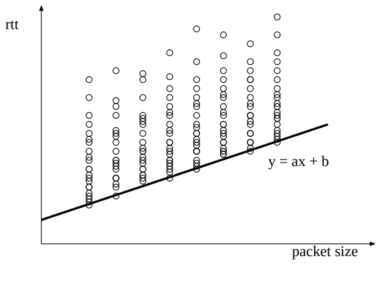
pathchar algorithm

- linear regression

  y = ax + b

  a: RTT-delta/packetsize-delta (bandwidth)

  b: delay for packetsize 0 (propagation delay)



delay measurement

- delay components
  - delay = propagation delay + queueing delay + other overhead
  - if not congested, delay is close to propagation deley
- methods
  - round-trip delay
  - one-way delay requires clock synchronization

  - average delay
  - max delay: e.g., voice communication requires < 400ms
  - jitter: variations in delay

some delay numbers

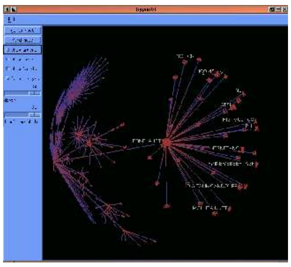
- packet transmission time (so called wire-speed)
  - 1500 bytes at 10Mbps: 1.2msec
  - 1500 bytes at 100Mbps: 120usec
- speed of light in fiber: 180,000 km/s
  - 100km round-trip: 1.1 msec
  - 20,000km round-trip: 220msec
- satellite round-trip delay
  - LEO (Low-Earth Orbit): 200 msec
  - GEO (Geostationary Orbit): 600msec

path measurement

- observe a path to a certain host
  - hop count measurement
    - how many routers between 2 hosts
- topology measurement
  - figure out network connections
    - by path measurement
      - from 1 location to many locations
      - from multiple locations (router has multiple IP addresses)
    - by routing information
      - route flapping

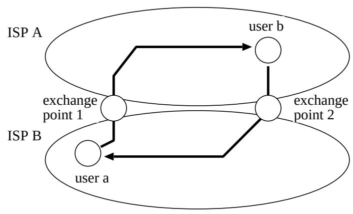
CAIDA skitter

- visualization of measured topology



asymmetric routes

- asymmetric route between 2 ISPs are common
  - packets forwarded via the nearest exchange point to other ISP



packet capturing

- tcpdump
  - can write captured packets to a file for post-analysis
  - many analysis tools read tcpdump output file
- tapping
  - half-duplex hub: tapping machine can see all traffic
  - full-duplex switch: splitter or port-mirroring needed
- tips to avoid packet drops
  - capture length
  - run on lightly-loaded machine
  - use large BPF buffer to avoid packet loss by buffer overflow
  - use filters cleverly

privacy issues

- user private data in packets
  - only packet headers are of interest
  - remove protocol payload
    - just removing payload makes traces much safer
- anonimity of users
  - IP address can identify a user
  - need to scramble IP addresses in open traces
- trade-off:
  - how much privacy vs. how much details
- tcpdpriv: a tool for tcpdump file
  - to remove payload
  - to scramble addresses

filtering techniques

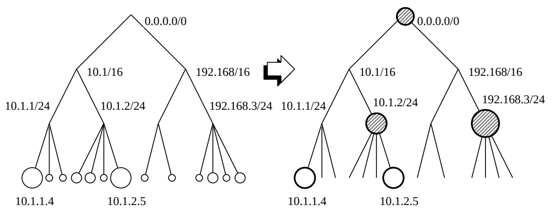
- extract packets of interest
  - 5-tuple: src address, dst address, src port, dst port, protocol
  - unique src address and dst address pair
  - single host, single protocol
  - TCP packets with specific flags
- extract part of packets needed for analysis
  - IP or TCP header
- if possible, capture all packets, and then, filter
  - to make further analysis possible

sampling techniques

- periodic sampling (every Nth packet)
  - easy to implement
  - could synchronize with traffic pattern
- random sampling (with probability 1/N)
  - to avoid synchronization
- hashing
  - e.g., for flow sampling, use hash of 5-tuple

aggregation techniques

- use flows at coarser level
  - e.g., src & dst address prefix, or AS number
  - example: aggregation based traffic profiler by WIDE



time in measurement

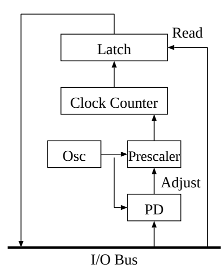
- absolute time
  - difference from UTC (Universal Coordinated Time)
- relative time
  - difference between events
- clock adjustment
  - clock could jump forward or backward!
  - ntp slews clock if difference is less than 128ms

clock uncertainty

- clock uncertainty
  - synchronization
    - difference of 2 clocks
  - accuracy
    - a given clock agrees with UTC
  - resolution
    - precision of a given clock
  - skew
    - change of accuracy or of synchronization with time
- time precision
  - local clock skew/drift: 0.1-1sec/day
  - NTP: synchronizes clock within 10-100ms
  - tcpdump timestamp: 100usec-100msec (usually < 1msec)

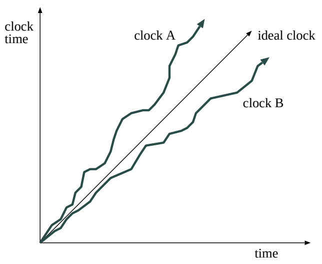
PC clock

- i8254 programmable interval timer
  - free-running 16-bit down-counter
    - driven by 1,193,182 Hz oscillator
    - when counter becomes zero
      - generates interrupt, and reloads the counter register



clock drift

- oscillator drift
  - hardware error margin: 10^(-5)
    - 0.86 sec/day within the spec
  - drift heavily affected by temperature



alternative clocks

- Pentium TSC (Time Stamp Counter)
  - high-resolution
    - a 64bit counter driven by CPU clock
    - 1nsec resolution for 1GHz CPU
  - light-weight
    - one instruction to read counter value
    - even in user space
  - drawbacks
    - frequency varies
    - power management changes CPU clock
- external clock source
  - GPS, shortwave radio, CDMA
    - serial interface overhead

OS time management

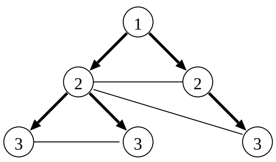
- OS manages software clock
  - initialized at boottime from time-of-day chip
  - updated by hardware clock interrupts
- standard UNIX sets the clock counter (and divider) to interrupt every 10ms (configurable)

UNIX gettimeofday

- older OS has only clock-interrupt resolution
- modern OS has much better resolution
  - interpolate software clock by reading the remaining counter value
    - resolution: 838ns (1 / 1193182)
  - inside kernel
    - access to the i8254 register: ~1-10usec
    - conversion to struct timeval: ~10-100usec
  - user space - kernel
    - system call overhead: ~100-500usec
    - process might be scheduled: ~1-100msec or more
- timer events (e.g., setitimer):
  - triggered only by timer tick (10msec by default)
  - effects of process scheduling

NTP (Network Time Protocol)

- multiple time servers across the Internet
  - primary servers: directly connected to UTC receivers
  - secondary servers: synchronize with primaries
  - tertiary servers: synchronize with secondary, etc
- scalability
  - 20-30 primaries, 2000 secondaries can synchronize to < 30ms
- many features
  - cope with server failures, authentication support, etc



NTP synchronization modes

- multicast (for LAN)
  - one or more servers periodically multicast
- remote procedure call
  - client requests time to a set of servers
- symmetric protocol
  - pairwise synchronization with peers

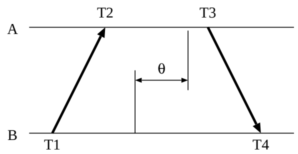
NTP symmetric protocol

- measuring offset and delay

  a = T2 - T1 and b = T3 -T4

  clock offset θ = (a + b) / 2

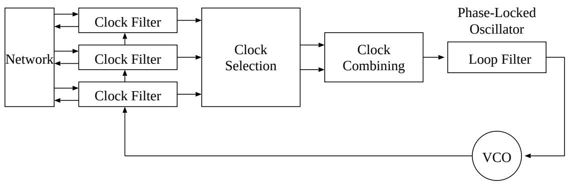
  roundtrip delay δ = a - b



- every message contains
  - T3: send time (current time)
  - T2: receive time
  - T1: send time in received message

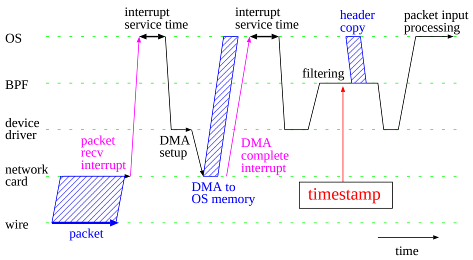
NTP system model

- clock filter
  - temporally smooth estimates from a given peer
- clock selection
  - select subset of mutually agreeing clocks
  - intersection algorithm: eliminate outliers
  - clustering: pick good estimates
- clock combining
  - combine into a single estimate



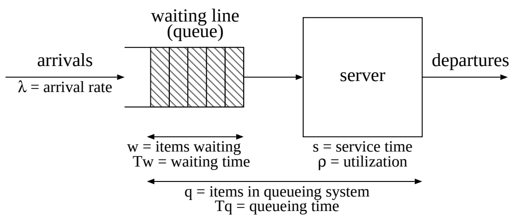
BPF timestamp

- timestamp usually placed after 2 interrupts
  - recv packet, DMA complete



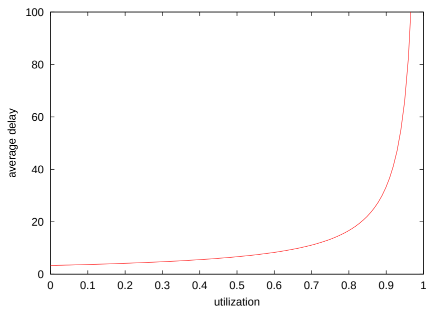
queueing theory

- observations of how queues behave
  - powerful mathematical tool for performance analysis
    - model a system as a queue and a server (black box)
    - input: average rate with some known distribution
      - compute service waiting time, queue length, etc



system performance by queueing theory

- performance degrades rapidly when load approaches to 100%



queueing applications

- applications
  - after-the-fact analysis based on actual values
  - make simple projection by scaling up from existing data
  - develop analytic model based on queueing theory
  - run simulation based on queueing model
- limitations
  - Poisson inputs often assumed
    - under-estimation for bursty inputs
      - e.g., 1st generation ATM switch with small buffer

common mistakes in data analysis

- false assumptions
- errors and bugs in tools
  - precision of calculation: valid digits, rounding errors, overflows
    - integer (32/64bits)
      - 32bit signed integer only up to 2G
    - 32bit floating point (IEEE 754 single precision)
      - sign:1bit, exponent:8bits, mantissa:23bits
        16,000,000 + 1 = 16,000,000 !
    - 64bit floating point (IEEE 754 double precision)
      - sign:1bit, exponent:11bits, mantissa:52bits
  - random numbers
    - pseudo random number generator
      - period, distribution, (predictability)

summary

- measurement metrics
  - throughput, delay, path, etc
- data reduction techniques
  - filtering, sampling, aggregation
- time in measurement
  - clock, OS time management, NTP
- introduction to queueing theory

- final word
  - Internet measurement is still under active reasearch
  - better measurement technologies are needed for better Internet

References

Raj Jain. The art of computer systems performance analysis. Wiley, 1991.

measurement activities in WIDE project: http://mawi.wide.ad.jp/mawi/
ntp: http://www.ntp.org
tcpdump: http://www.tcpdump.org
caida: http://www.caida.org
caida Internet Tools Taxonomy: http://www.caida.org/tools/taxonomy/
skitter: http://www.caida.org/tools/measurement/skitter/
RRDtool: http://www.caida.org/tools/utilities/rrdtool/
MRTG: http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/
RIPE routing information service: http://www.ripe.net/ripencc/pub-services/np/ris/
The Internet Traffic Archive: http://ita.ee.lbl.gov/index.html
Internet Measurement Research Group: http://imrg.grc.nasa.gov/imrg/

assignment

- observe ping response time for 24 hours
  - e.g., ping -i 30 133.138.1.149
    - "-i 30" sends request every 30 seconds
    - you may use 133.138.1.149 as a target host
  - calculate
    - average response time and standard deviation
    - minimum and maximum response time
    - 10th, 50th, 90th-percentile
- optionally, draw graphs (similar to ones shown in class)
  - histogram and CDF
  - daily plots
  - (use gnuplot on UNIX)
- compare the results with the results shown in the class