Introduction to Networking and Systems Measurements

Lecture 2: Basic Network Measurements

Dr Noa Zilberman
noa.zilberman@cl.cam.ac.uk

Terminology Matters!

… in greater depth in following weeks

Bandwidth, Throughput and Goodput

- Bandwidth – how much data can pass through a channel.
- Throughput – how much data actually travels through a channel.
- Goodput is often referred to as application level throughput.

But bandwidth can be limited below link’s capacity and vary over time, throughput can be measured differently from bandwidth etc…..

Speed and Bandwidth

- Higher bandwidth does not necessarily mean higher speed
- E.g. can mean the aggregation of links
  - 100G = 2x50G or 4x25G or 10x10G
  - A very common practice in interconnects

RTT, Latency and FCT

Measures of time:

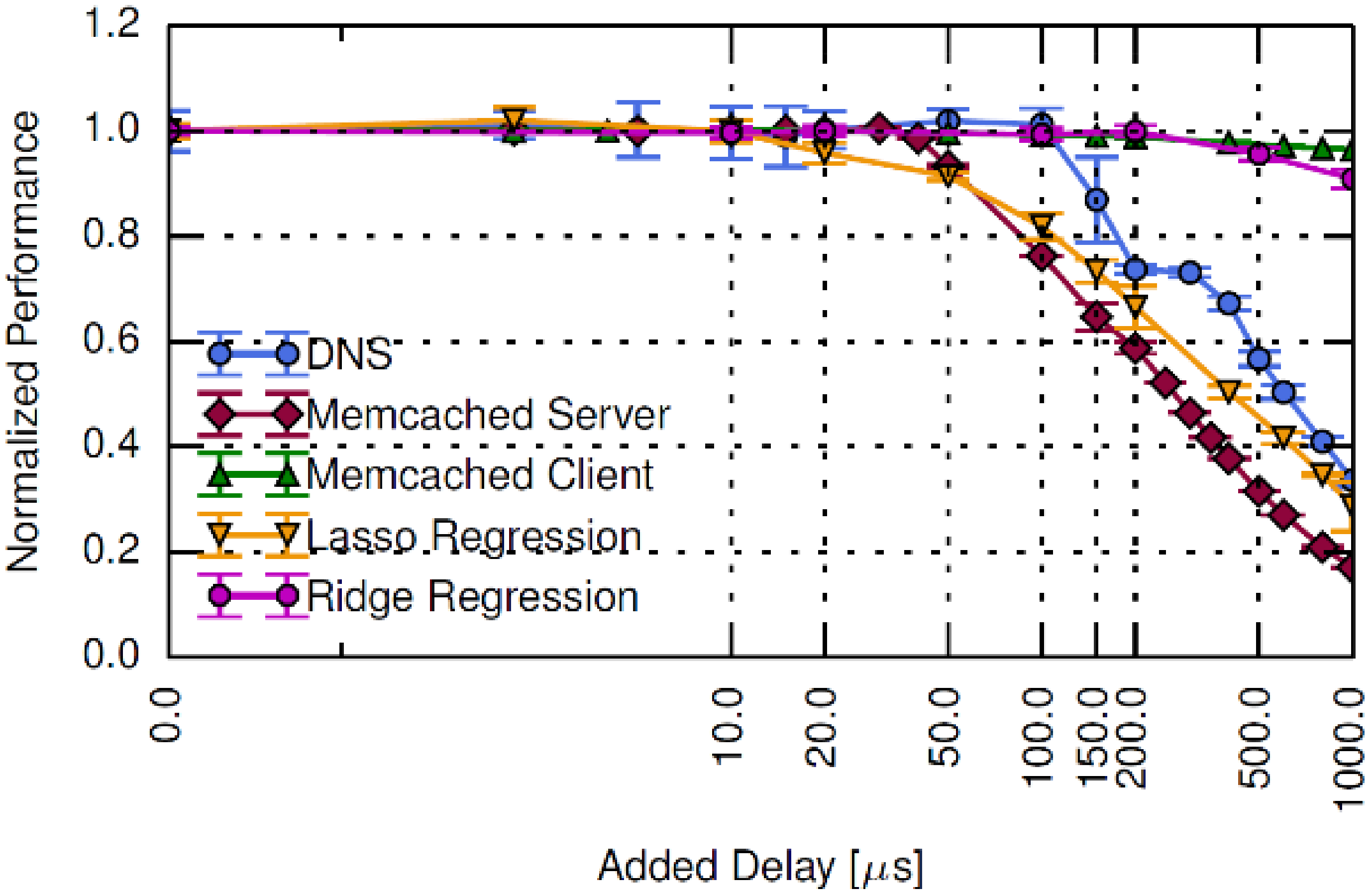
- Latency – The time interval between two events.
- Round Trip Time (RTT) – The time interval between a signal being transmitted and a reply is being received.
- Flow Completion Time (FCT) – The lifetime of a flow.

Performance Metrics

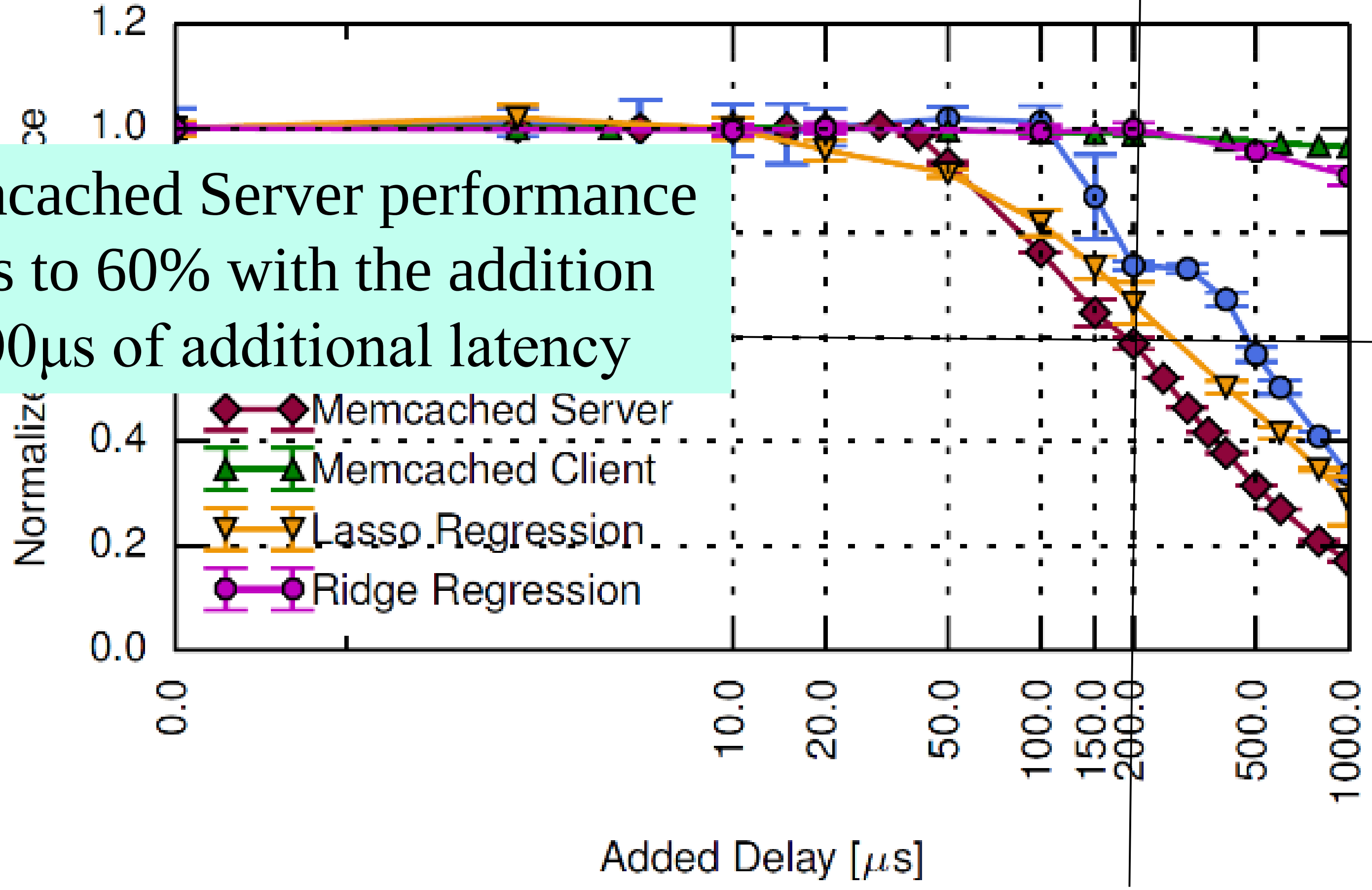
- Throughput, FCT etc. are measures of *Performance.*

- Bandwidth, RTT, packet loss etc. don’t indicate (directly) how good or bad the application / system / network perform.

Example: The Effect of Latency on Application's Performance



Example: The Effect of Latency on Application's Performance



Memcached Server performance drops to 60% with the addition of 200μs of additional latency

Types of Measurements

Measurement Techniques

- Active
  - Issue probe, Analyse response
- Passive
  - Observe events

Example: Active vs. Passive RTT Measurement

- Active measurement – Ping
  - Sends ICMP Echo Request message
  - Waits for Echo Reply message
  - RTT is the time gap between the request and the reply.
- Passive measurement – tcptrace
  - Uses TCP dump files
  - Calculates RTT according to timestamps logged in the dump.

Comparison

Passive	Active
Can only measure in the presence of activity / traffic	Measures even when tapping activity / traffic is not possible
Measures user experience, *behaviour* Measures protocol exchanges	Measures system, network, application *performance*
Raise privacy concerns	Adds probing load: - Overload system/network - May bias inferences

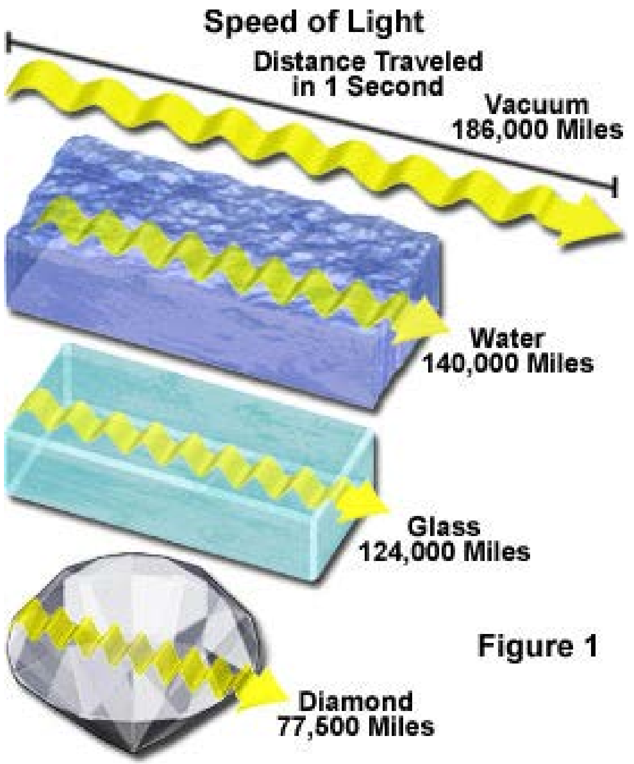
Measurement Vantage Point

- Point where measurement host connects to system / network
- Observations often depend on vantage point
  - Do you have enough vantage points?
  - How are the vantage points distributed?
- Can affect, e.g.:
  - Topology discovery
  - Bandwidth analysis

Possible Vantage Points

- End-hosts
  - Active measurements of end-to-end paths
  - Passive measurements of host’s traffic
- Routers/Measurement hosts in network
  - Active measurements of network paths
  - Passive measurements of traffic, protocol exchanges, configuration

Time flies



- 1ns = 20cm in fibre
- 10Gb/s is about 10 bits per nanosecond
- so a 512byte packet is ~ 8meters long

Ping

- Ping is basically a “are you still there” test
- “connectivity” test
- “how long does it take to get there” test
- “loss approximation” test

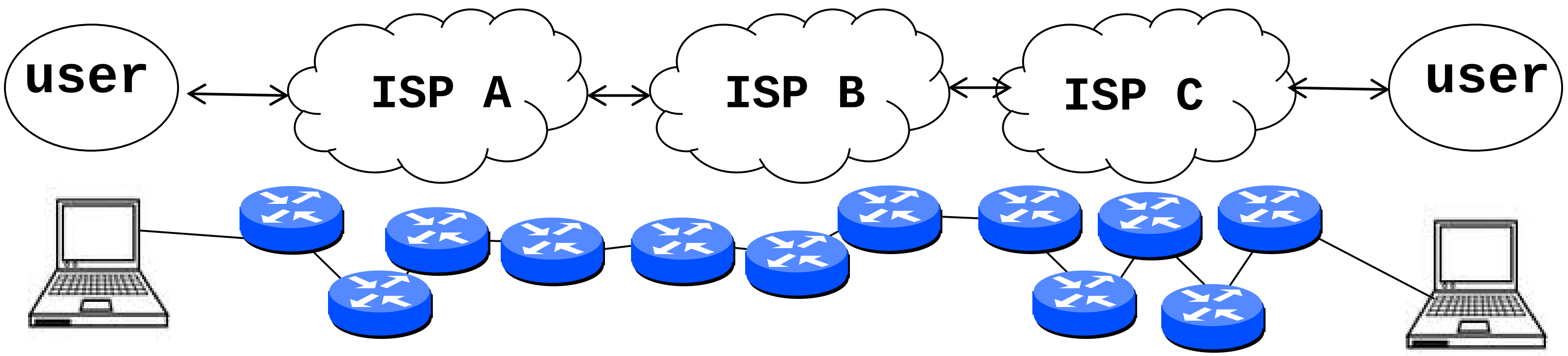
```
$ ping www.stanford.edu
PING www.stanford.edu (54.192.2.121): 56 data bytes
64 bytes from 54.192.2.121: icmp_seq=0 ttl=242 time=3.730 ms
64 bytes from 54.192.2.121: icmp_seq=1 ttl=242 time=3.845 ms
…
^C
--- www.stanford.edu ping statistics ---
8 packets transmitted, 8 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 3.730/3.808/3.849/0.047 ms
```

PING traps

- Uses ICMP (control messages of the Internet)
- Might not follow the same path as *normal packets*
- Might be filtered
- A ping test is not the actual round trip time for an application
- One way delay is not simply twice round trip time
- Learn by doing (run tcpdump at the same time)

Recall the Internet *federation*

- The Internet ties together different networks
  - >18,000 ISP networks



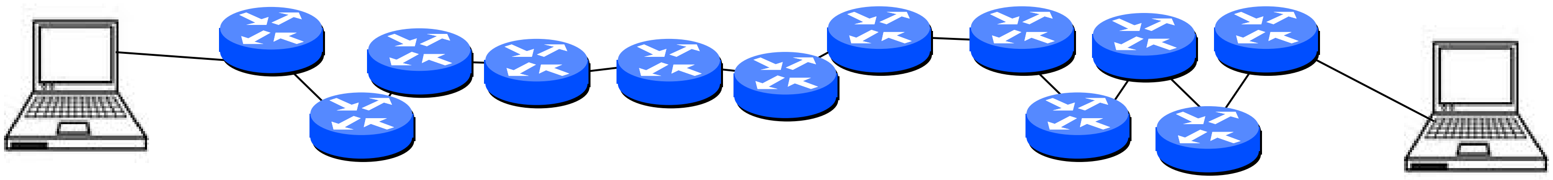
We can see (hints) of the nodes and links using traceroute…

Traceroute: Internet debug thy self

- Recall the Internet ***Zombie plan*** – *Time-To-Live (TTL)*
- *Each router decrements TTL; when TTL =0 send error*

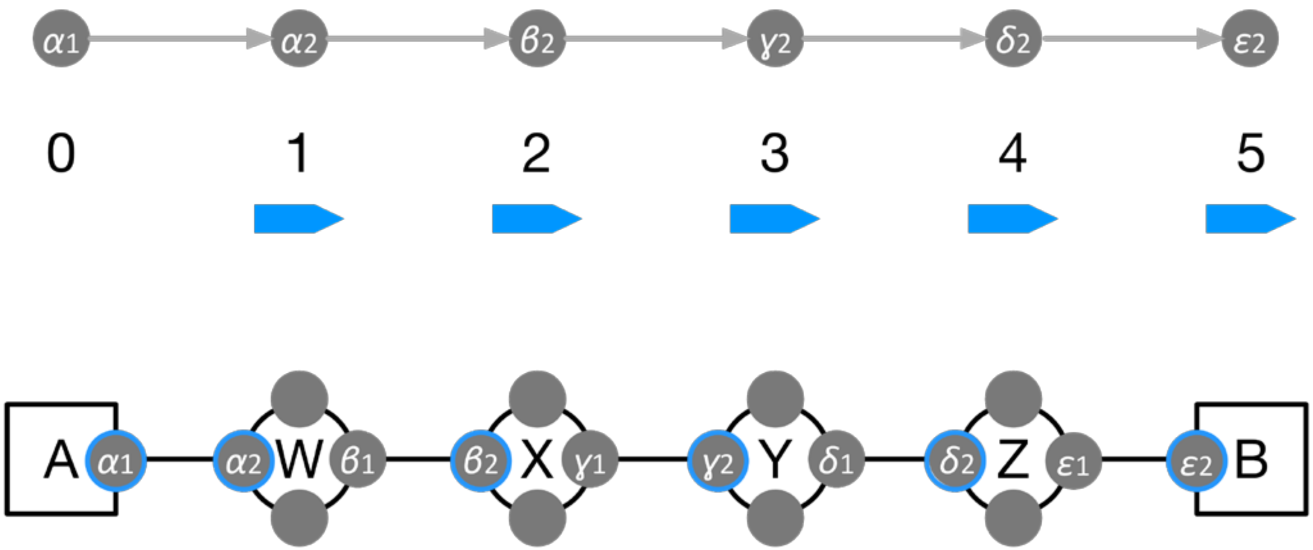
`Traceroute` artificially sets low TTL and receives the error

Each step of the path is iteratively discovered



...

Traceroute as hoped…



But ***ONLY*** one *direction*

"Real" Internet traceroute

traceroute: rio.cl.cam.ac.uk to munnari.oz.au

(tracepath on windows is similar)

Three delay measurements from rio.cl.cam.ac.uk to gatwick.net.cl.cam.ac.uk

```
traceroute munnari.oz.au
traceroute to munnari.oz.au (202.29.151.3), 30 hops max, 60 byte packets
 1  gatwick.net.cl.cam.ac.uk (128.232.32.2)  0.416 ms  0.384 ms  0.427 ms
 2  cl-sby.route-nwest.net.cam.ac.uk (193.60.89.9)  0.393 ms  0.440 ms  0.494 ms
 3  route-nwest.route-mill.net.cam.ac.uk (192.84.5.137)  0.407 ms  0.448 ms  0.501 ms
 4  route-mill.route-enet.net.cam.ac.uk (192.84.5.94)  1.006 ms  1.091 ms  1.163 ms
 5  xe-11-3-0.camb-rbr1.eastern.ja.net (146.97.130.1)  0.300 ms  0.313 ms  0.350 ms
 6  ae24.lowdss-sbr1.ja.net (146.97.37.185)  2.679 ms  2.664 ms  2.712 ms
 7  ae28.londhx-sbr1.ja.net (146.97.33.17)  5.955 ms  5.953 ms  5.901 ms
 8  janet.mx1.lon.uk.geant.net (62.40.124.197)  6.059 ms  6.066 ms  6.052 ms
 9  ae0.mx1.par.fr.geant.net (62.40.98.77)  11.742 ms  11.779 ms  11.724 ms
10  ae1.mx1.mad.es.geant.net (62.40.98.64)  27.751 ms  27.734 ms  27.704 ms
11  mb-so-02-v4.bb.tein3.net (202.179.249.117)  138.296 ms  138.314 ms  138.282 ms
12  sg-so-04-v4.bb.tein3.net (202.179.249.53)  196.303 ms  196.293 ms  196.264 ms
13  th-pr-v4.bb.tein3.net (202.179.249.66)  225.153 ms  225.178 ms  225.196 ms
14  pyt-thairen-to-02-bdr-pyt.uni.net.th (202.29.12.10)  225.163 ms  223.343 ms  223.363 ms
15  202.28.227.126 (202.28.227.126)  241.038 ms  240.941 ms  240.834 ms
16  202.28.221.46 (202.28.221.46)  287.252 ms  287.306 ms  287.282 ms
17  * * *
18  * * *
19  * * *
20  coe-gw.psu.ac.th (202.29.149.70)  241.681 ms  241.715 ms  241.680 ms
21  munnari.OZ.AU (202.29.151.3)  241.610 ms  241.636 ms  241.537 ms
```

trans-continent link

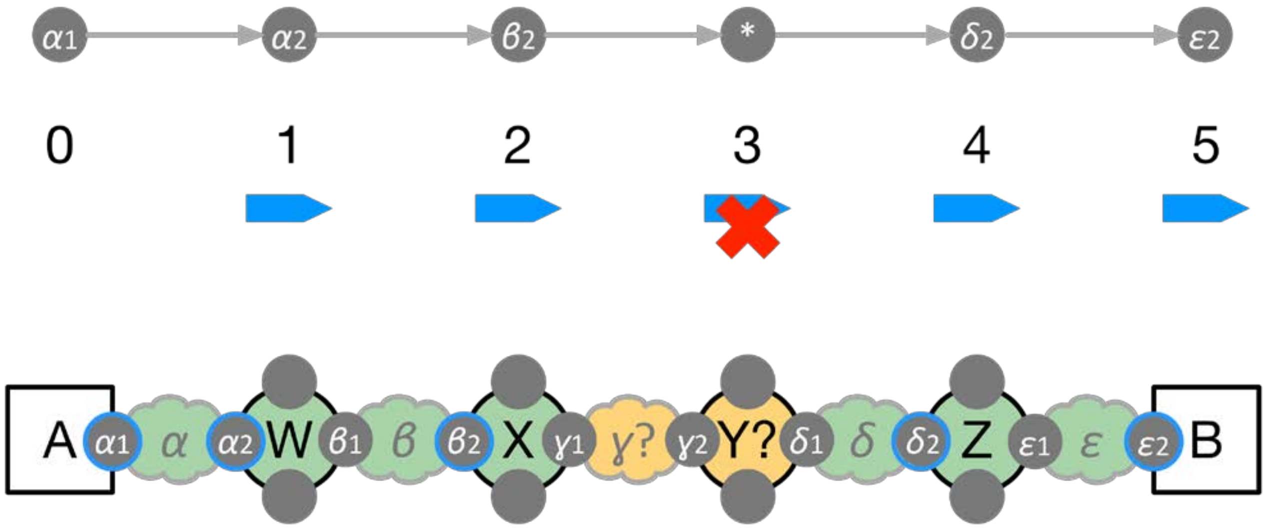
* means no response (probe lost, router not replying)

Traceroute traps – a bit like ping

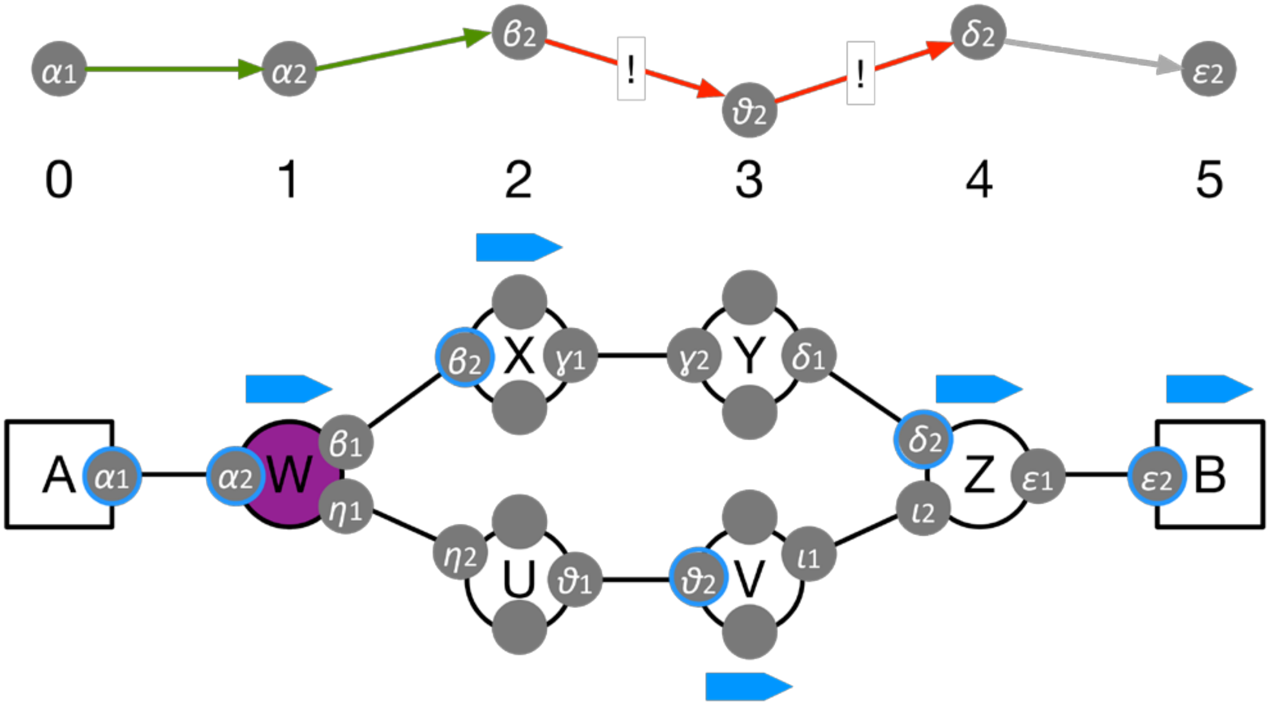
- Uses UDP or ICMP (but traffic is often TCP)
- Might not follow the same path
- Might be filtered
- Only infers one direction of the path
- Replies can be very weird
- One way delay is **not** simply twice round trip time (networks may have many paths)

- Learn by doing (try with and without the –I option)

Traceroute doesn't always know



Traceroute lies



Beyond traceroute

- *Paris traceroute*

  Uses many probes to identify multiple paths

  www.paris-traceroute.net

- *Reverse traceroute*

  Uses a remote server to probe reverse path

Link capacity…..

- Recall capacity is a property of where and what we measure
- Nominal network capacity is physical

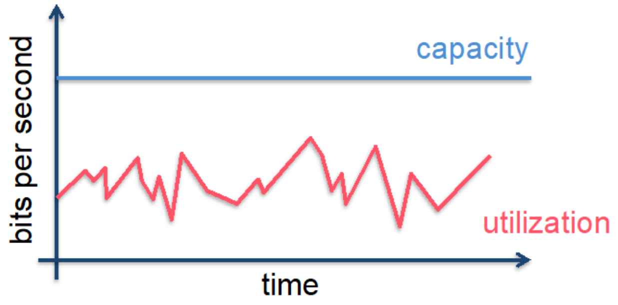
eg 100BaseTX Ethernet: 100 Mbps
WiFi 802.11g: 54 Mbps

- IP-layer capacity < nominal capacity
  - Coding schemes
  - Framing bits, overhead
  - Medium access control

Link capacity & utilization

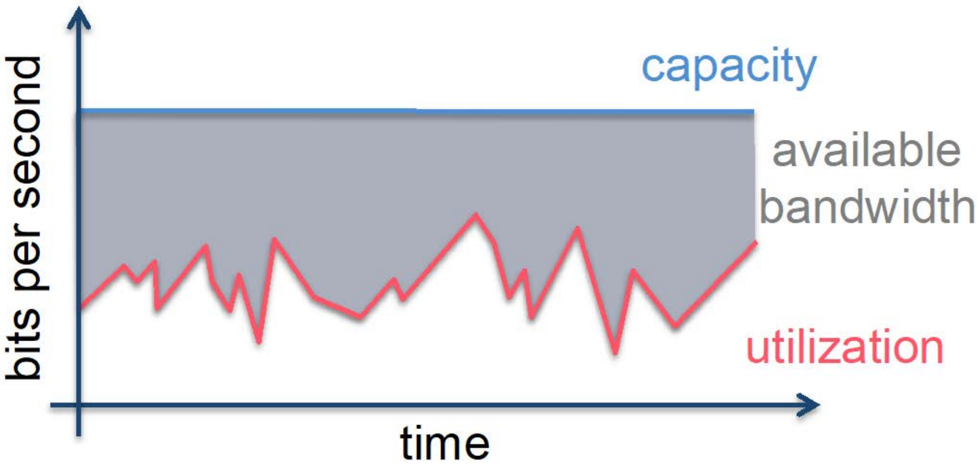
- Link capacity (C(Δt)) ≈ IP-layer capacity
  - Maximum IP-layer rate of maximum-sized packets
  - IP-layer capacity depends on size of packet relative to layer-2 overhead
- Link utilization (u(Δt))
  - u(Δt) = Average bits transmitted on the link during Δt
  - Percent utilization =

% link capacity that is utilized



Available Capacity

- Available bandwidth (A(Δt))
  - Maximum unused bandwidth
  - $A(\Delta t) = C(\Delta t) - u(\Delta t)$



End-to-end capacity and End-to-end effective bandwidth

Router1 -----C1----- Router2 -----C2----- Router

C1: 100 Mbps	C2: 30 Mbps
u1: 80 Mbps	u2: 3 Mbps
A1: 20 Mbps	A2: 27 Mbps

End-to-end capacity: min{C1, C2}=30 Mbps

End-to-end available bandwidth: min{A1, A2}=20 Mbps

Probing method

Flooding

Issue enough probes to “fill” path

- Pro
  - Measure what users can get
- Con
  - Large overhead affect network and users

Advanced methods

A number of methods in literature:

Packet pair, size-delay, self-induced congestion

- Pro
  - Less overhead than flooding
- Con
  - Rely on assumptions that don’t always hold in practice

Effective Bandwidth Measurement

- How much capacity in my network?
  - Is it working at spec.? Am I getting my money's worth?
  - Systems can adapt to change of Effective Bandwidth

Considerations

- TCP versus UDP
  - UDP not biased by congestion/flow control
  - Flooding with UDP may create too much congestion and bias results
  - Multiple TCP connections reduces bias
- Multi-threaded TCP
  - How many threads?
  - Which size transfers?
- UDP
  - How to pick sending rate?

iperf versions and other tools for measuring available bandwidth

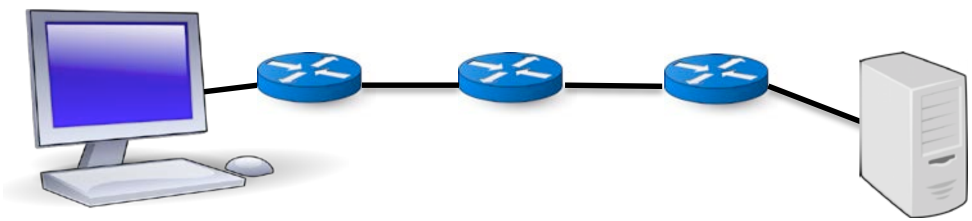
- iperf/iperf3
  - Control of client and server
  - Configurable tests
- iperf2 for UDP
- iperf3 is a rewrite with different/improved TCP

Others: eg.

- NetPerf is yet another TCP and UDP tool
  - NetPerf implicitly codes ideas of confidence, sample size, etc.

iperf Vantage points

- Runs application at both client and server



source:
iperf client
`iperf -c server`

destination:
iperf server
`iperf -s`

An Example *iperf* Output

```
$ iperf3 -u -t 10 -b 100Mbit --get-server-output -c 192.168.1.174
Connecting to host 192.168.1.174, port 5201
[  4] local 192.168.1.231 port 51069 connected to 192.168.1.174 port 5201
[ ID] Interval        Transfer     Bandwidth       Total Datagrams
[  4]   0.00-1.00   sec  10.8 MBytes  90.2 Mbits/sec  1379
              :
[  4]   9.00-10.00  sec  12.0 MBytes   100 Mbits/sec  1532
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval        Transfer     Bandwidth       Jitter    Lost/Total Datagrams
[  4]   0.00-10.00  sec   118 MBytes  99.0 Mbits/sec  0.839 ms  2034/15114 (13%)
[  4] Sent 15114 datagrams

Server output:
Accepted connection from 192.168.1.231, port 58542
[  5] local 192.168.1.174 port 5201 connected to 192.168.1.231 port 51069
[  5]   0.00-1.00   sec  7.05 MBytes  59.2 Mbits/sec  1.190 ms  226/1129 (20%)
              :
[  5]   9.00-10.00  sec  11.4 MBytes  95.9 Mbits/sec  2.670 ms  74/1537 (4.8%)
```

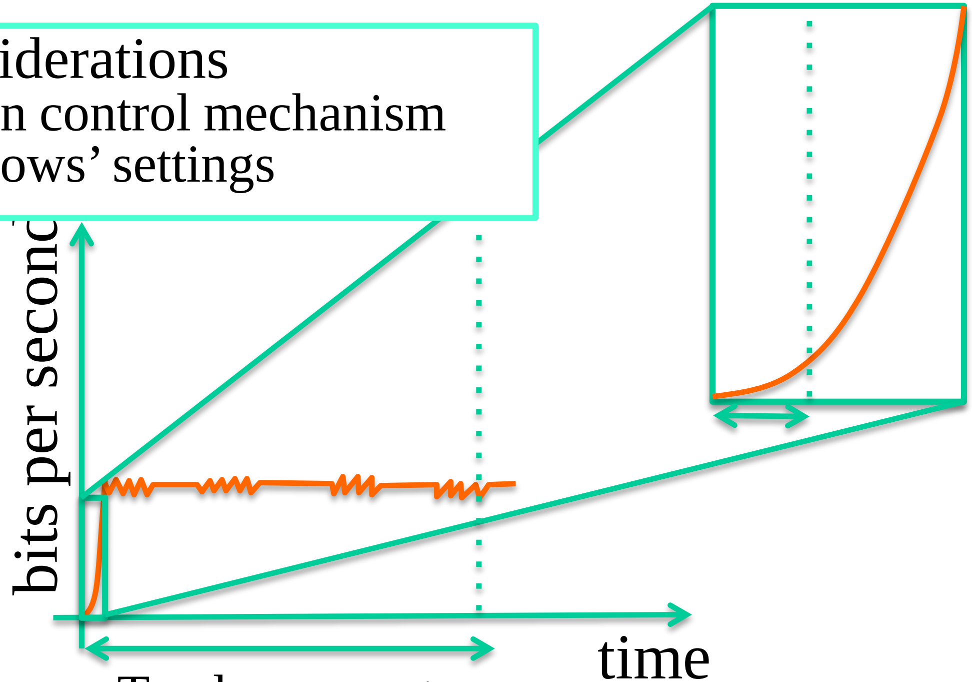
Effective bandwidth traps
or

how to do *Effective* effective-bandwidth measurement

- Bulk transfer capacity depends on many factors
- Transfer size
- TCP variant and configuration
- Cross traffic
- Congestion on reverse (ACK) path

Consideration: Transfer size

Other considerations
- Congestion control mechanism
- TCP windows' settings



Too small mainly measures slow start

Too large creates unnecessary overhead

Thanks to Renata Teixera for inspiring this slide

Consideration

- TCP versus UDP
  - UDP not biased by congestion/flow control
  - Flooding with UDP may create too much congestion and bias results
  - Multiple TCP connections reduces bias
- Multi-threaded TCP
  - How many threads?
  - Which size transfers?
- UDP
  - How to pick sending rate?

Thanks to Renata Teixera
for inspiring this slide