



Storskaliga Distribuerade System och Nätverk (Large-scale Distributed Systems and Networks)

Slides by Niklas Carlsson (including slides
based on slides by P. Gill and Y. Shavitt)

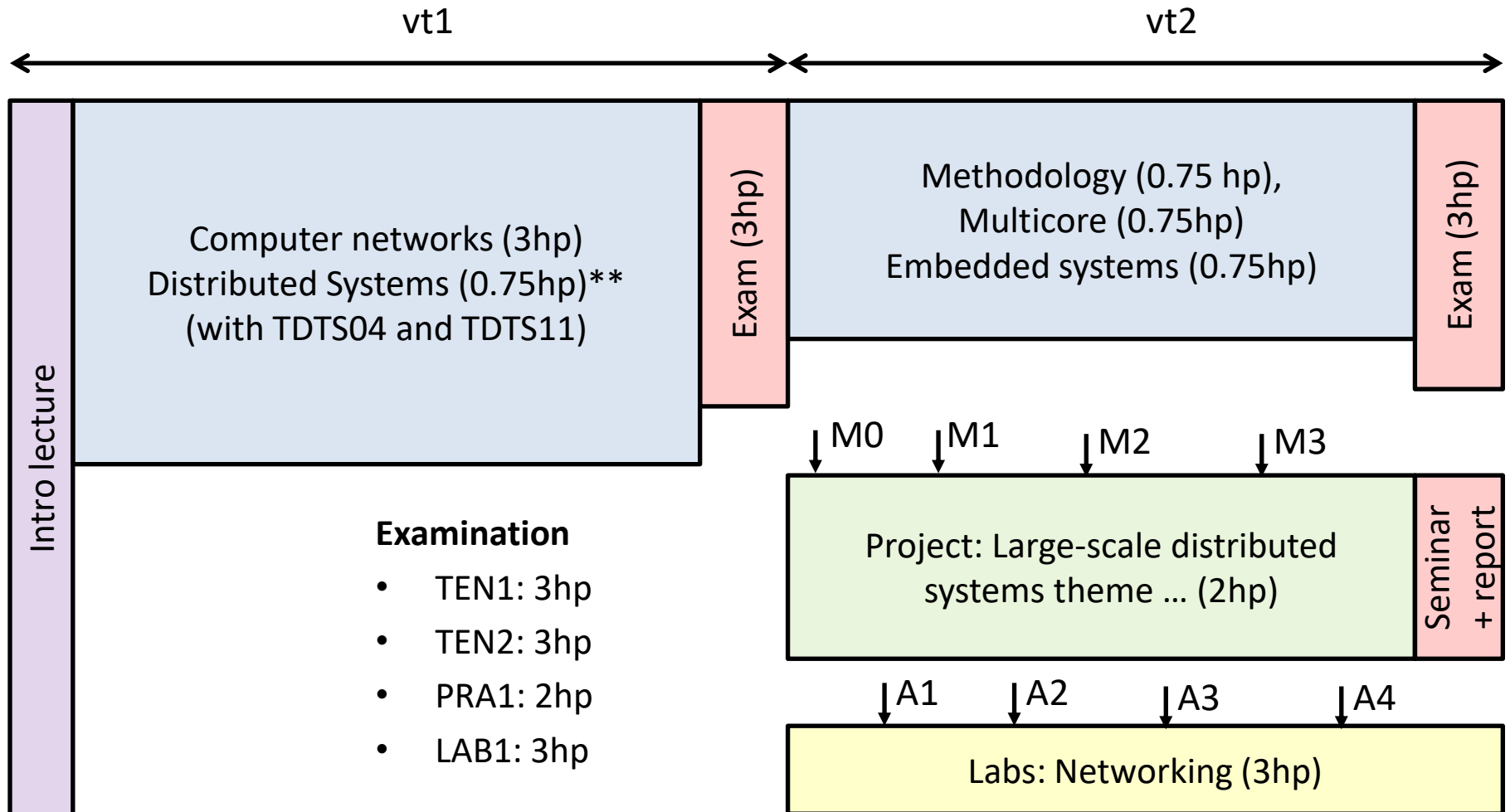
Skalbarhet och systemtänk

- System tänkande med fokus på skalbarhet
 - Helhetsperspektiv (lager, komponenter, etc.)
 - Large distributed systems and services
 - Nätverk och distribuerade systems "hand-i-hand"
 - Single to multicore; single to million machines/users
 - Skalbara metoder och arkitekturer
 - Modellering och abstraction av stora system (inklusive enkel matematisk modellering)
- Mix av teori och praktik
 - "Kunskapen är inte din förrän du använder den"
 - Använda experiment och mätdata för att öka förståelsen för hur "verkligheten" ser ut, men också diskutera framtiden

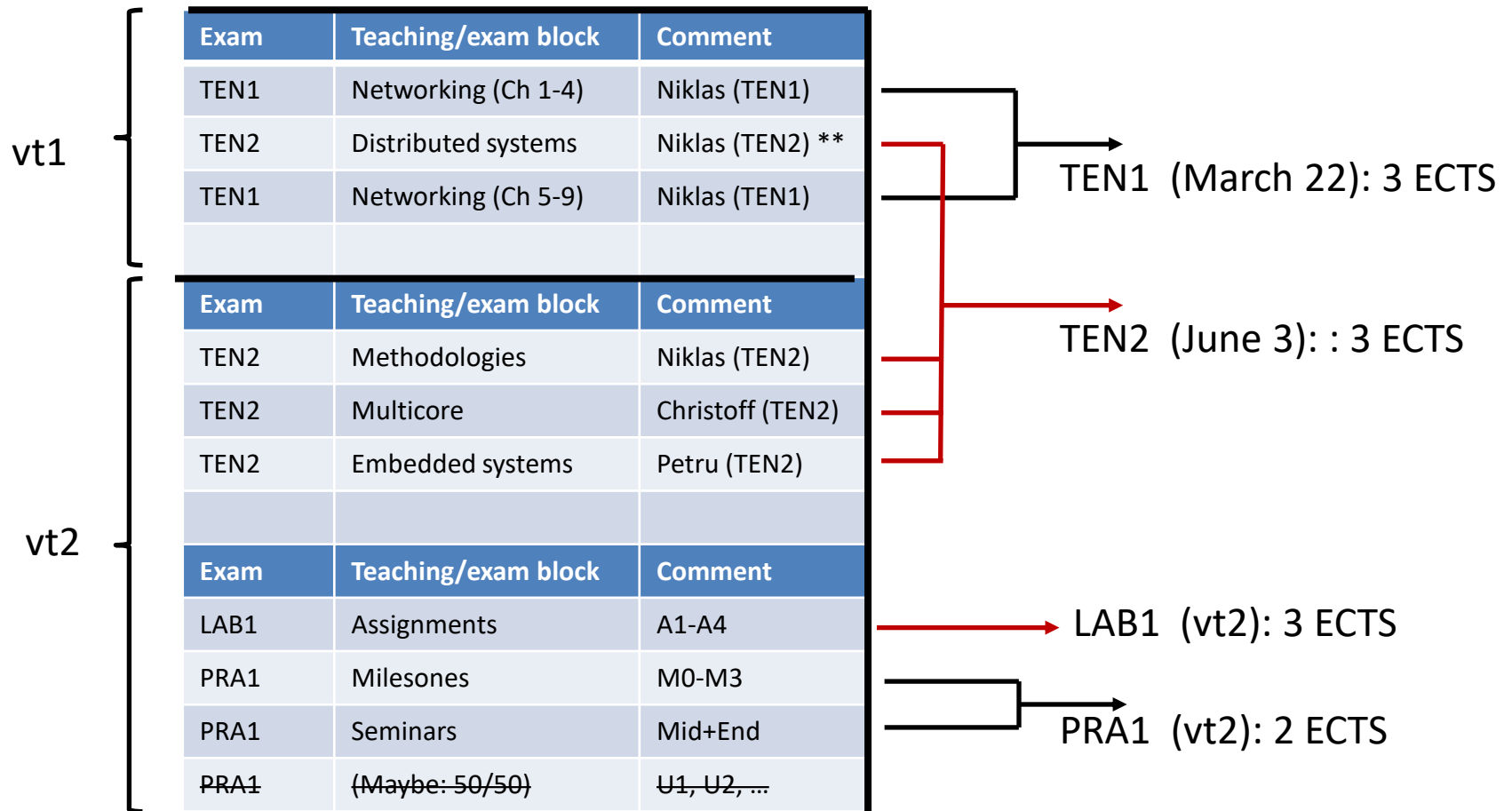
Ämneskunskaper

- Nätverk (vt1)
 - Grundkunskaper, liknande som TDTS06, TDTS11, och TDTS04 (ca. 12-14 lektioner). Detta ger behörighet till kursen TDTS21 (avancerade nätverk).
 - Labbar (åtminstone en per lager 3, 4, och 5)
- Distribuerade system (vt1)
 - Några introduktions lektioner (ca. 4 föreläsningar)
 - Projekt i vilket får erfarenhet av storskaliga distribuerade system och tjänster
- Multicore (vt2)
 - Kristoffer Kessler (ca. 4 föreläsningar)
- Inbyggda system (vt2)
 - Petru Eles (ca. 3 föreläsningar)
- Metoder och användande av riktiga system och data (vt2)
 - Några introduktions lektioner (ca. 4 föreläsningar)
 - Modellering/abstraktion, hypotesformulering och datadriven analys metoder för storskaliga system och tjänster

Overview (2021)



Overview ...



Projects and assignments

- **2021: Practice working in teams (of 2-4 students) remotely.**
- Assignments (and lab sessions)
 - Groups of 2 students
 - 4 assignments:
 - Split across multiple network layers
 - 2 x wireshark (HTTP + TCP), proxy, and DV
- Project
 - Groups of 3-4 students (larger groups than past courses you have seen)
 - Clear "milestones" introducing both incremental and iterative report writing, as well as oral presentation
 - Multiple "milestones" with "peer-reviewing"
 - Register for webreg by Thursday (Apr. 1, 2021)
 - Projects released by Tuesday (Apr. 6, 2021)
 - Request projects by Thursday (priority if on Wednesday)

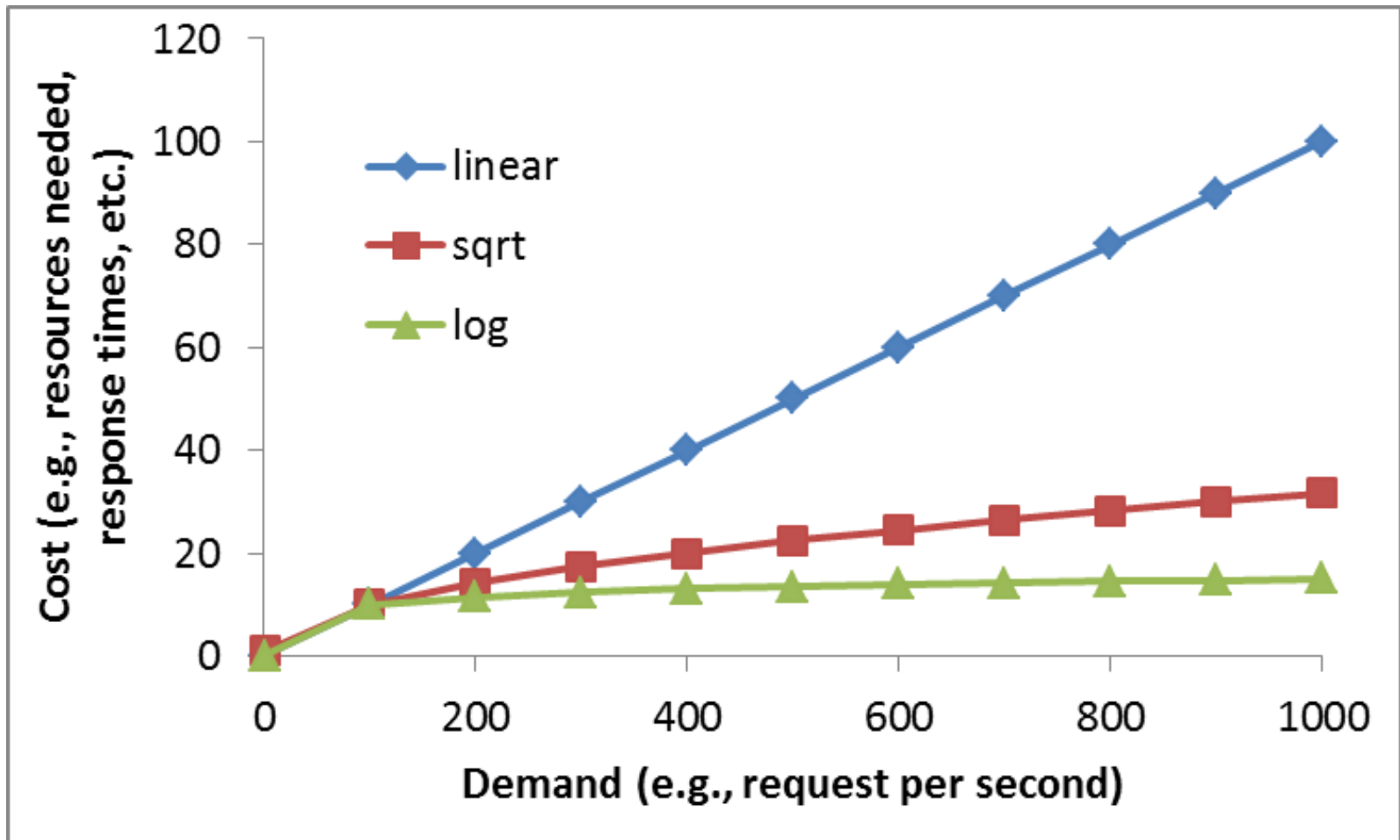
Reminder: Attend lectures ...

- In class: Examples to help build an understanding and intuition for “scalability” and “system thinking”
 - These abilities are hard-to-impossible to learn only from notes!!
- Projects and expectations around the projects (and report/article writing in general) will be discussed in class
- Please attend the lectures (and obtain such information ...)

Scalability

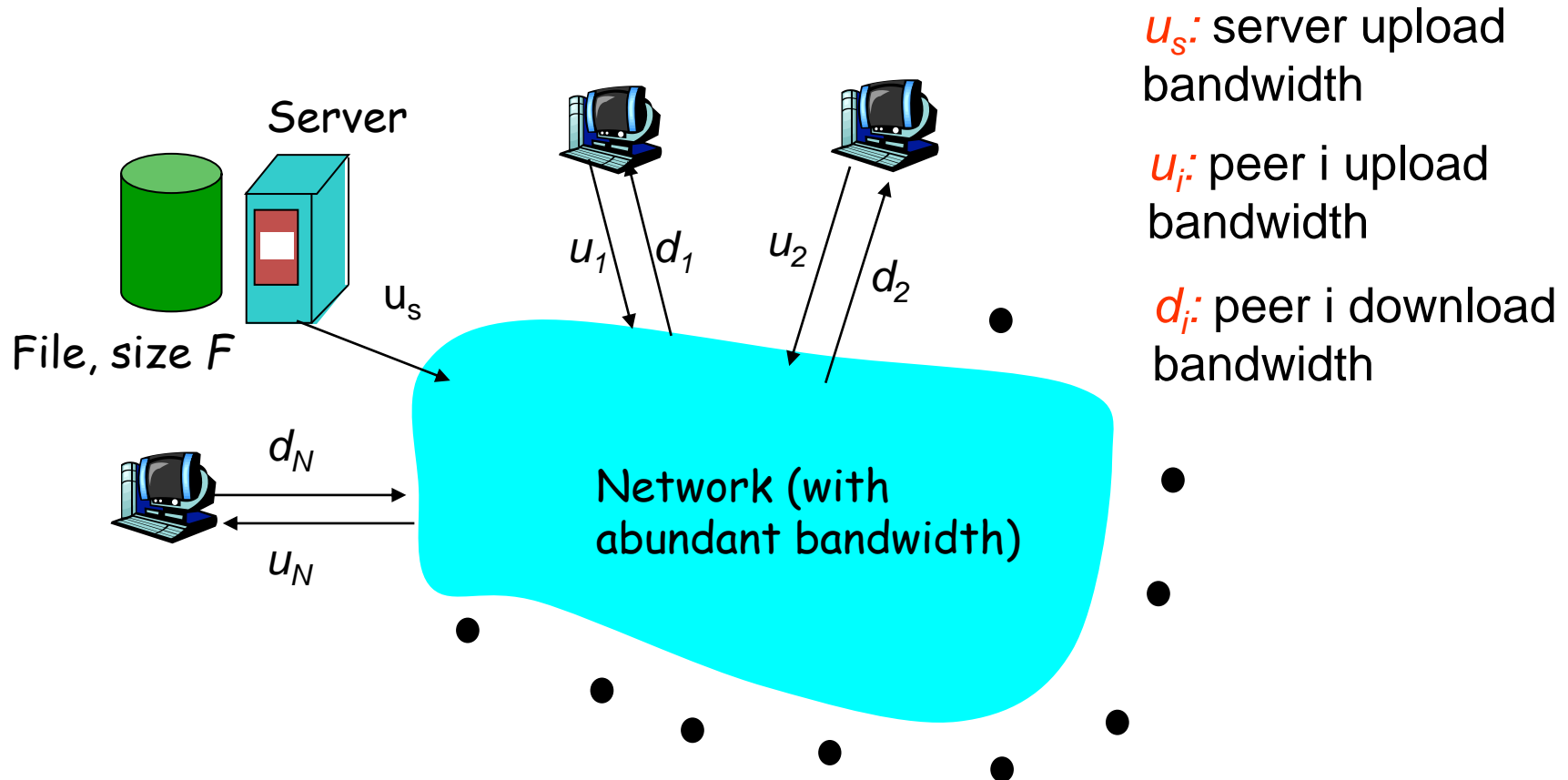
- Typically want solutions that “scales”
 - Ability of a system, network, or process to handle a growing amount of work effectively
 - Capability to increase its total output under an increased load when resources are added
- Typically want:
 - the costs or resource capacity needed to scale sub-linearly with demand, or
 - the performance to improve at least proportionally to the capacity added

Scalability examples

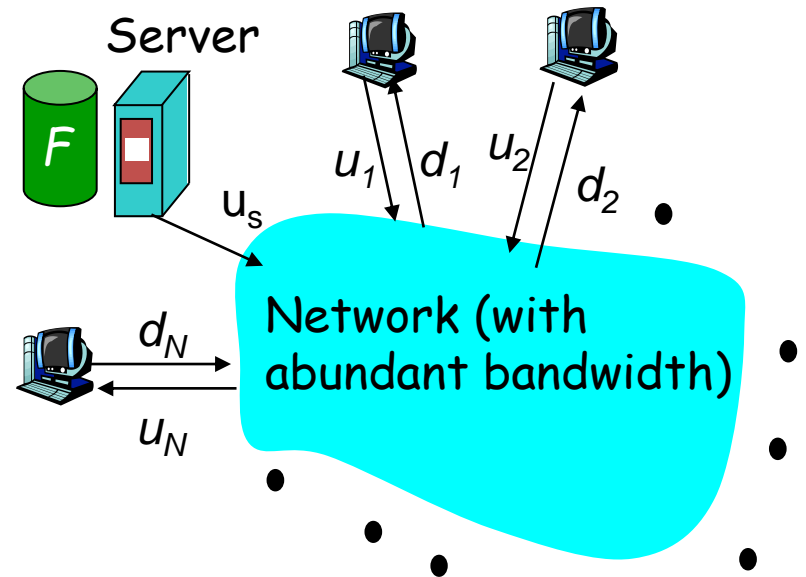


Examples from earlier in the course ...

Question : How much time to distribute file from one server to N peers?



File distribution time: server-client



Time to distribute F to N clients using client/server approach

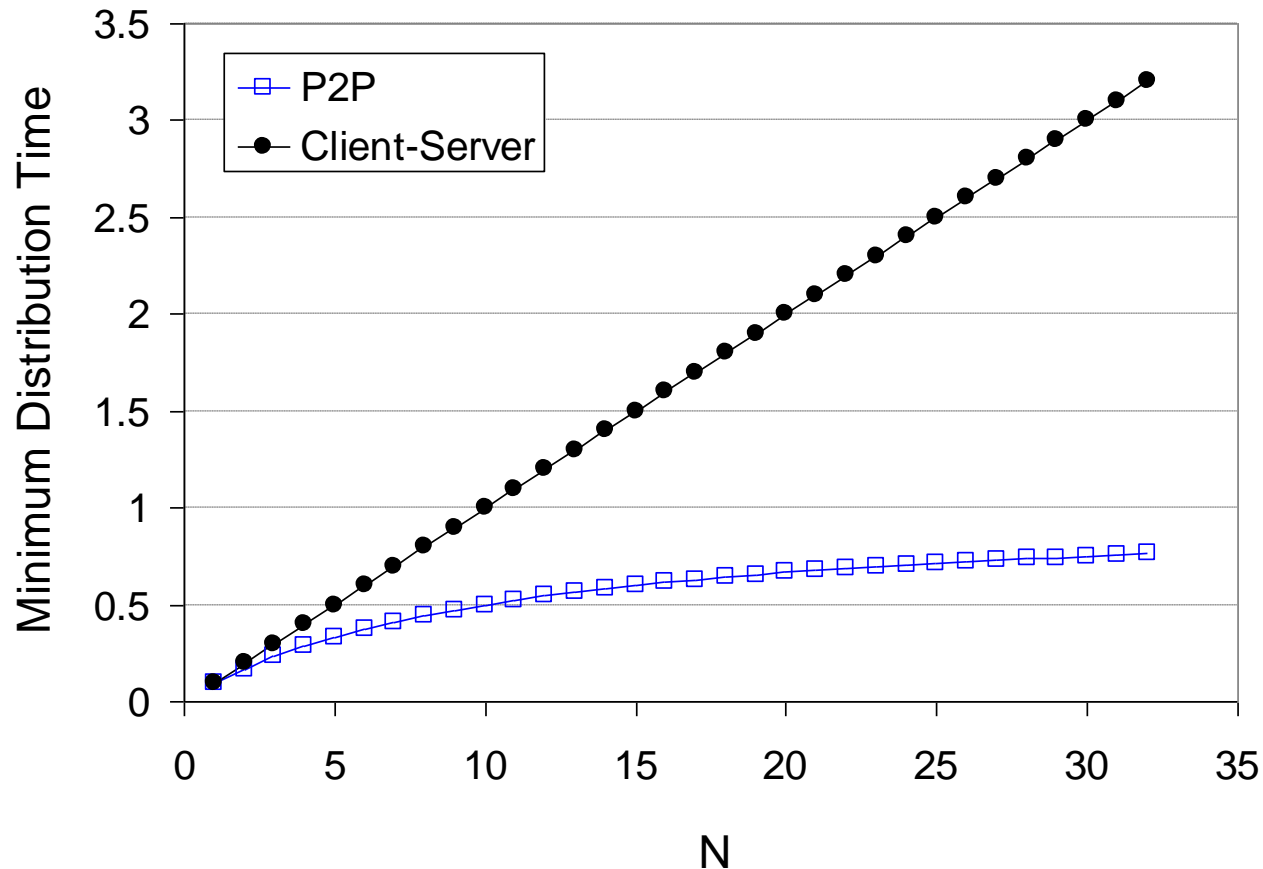
$$= d_{cs} = \max \left\{ NF/u_s, F/\min(d_i) \right\}$$

... and using a P2P approach

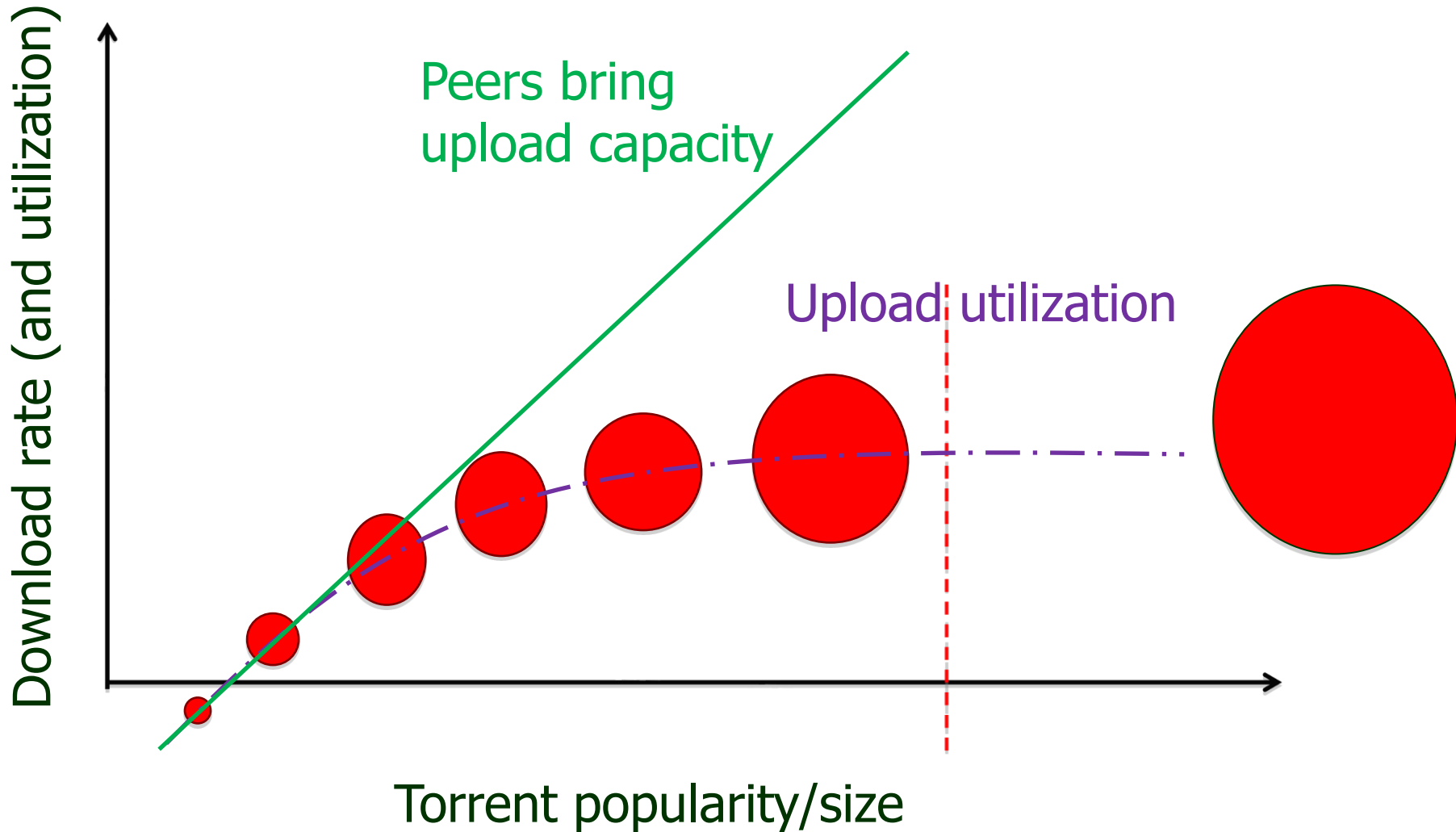
$$d_{p2p} = \max \left\{ F/u_s, F/\min(d_i), NF/(u_s + \sum u_i) \right\}$$

Server-client vs. P2P: example

Client upload rate = u , $F/u = 1$ hour, $u_s = 10u$, $d_{\min} \geq u_s$



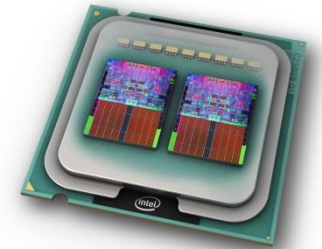
Similarly, BitTorrent upload utilization ...



... more examples later ...

Systems thinking

- We want to understand the full system and the ecosystem it operates within; e.g.,
 - Understanding the full system
 - Looking at the parts and how they interact
- This course provide many examples ...



Measurements

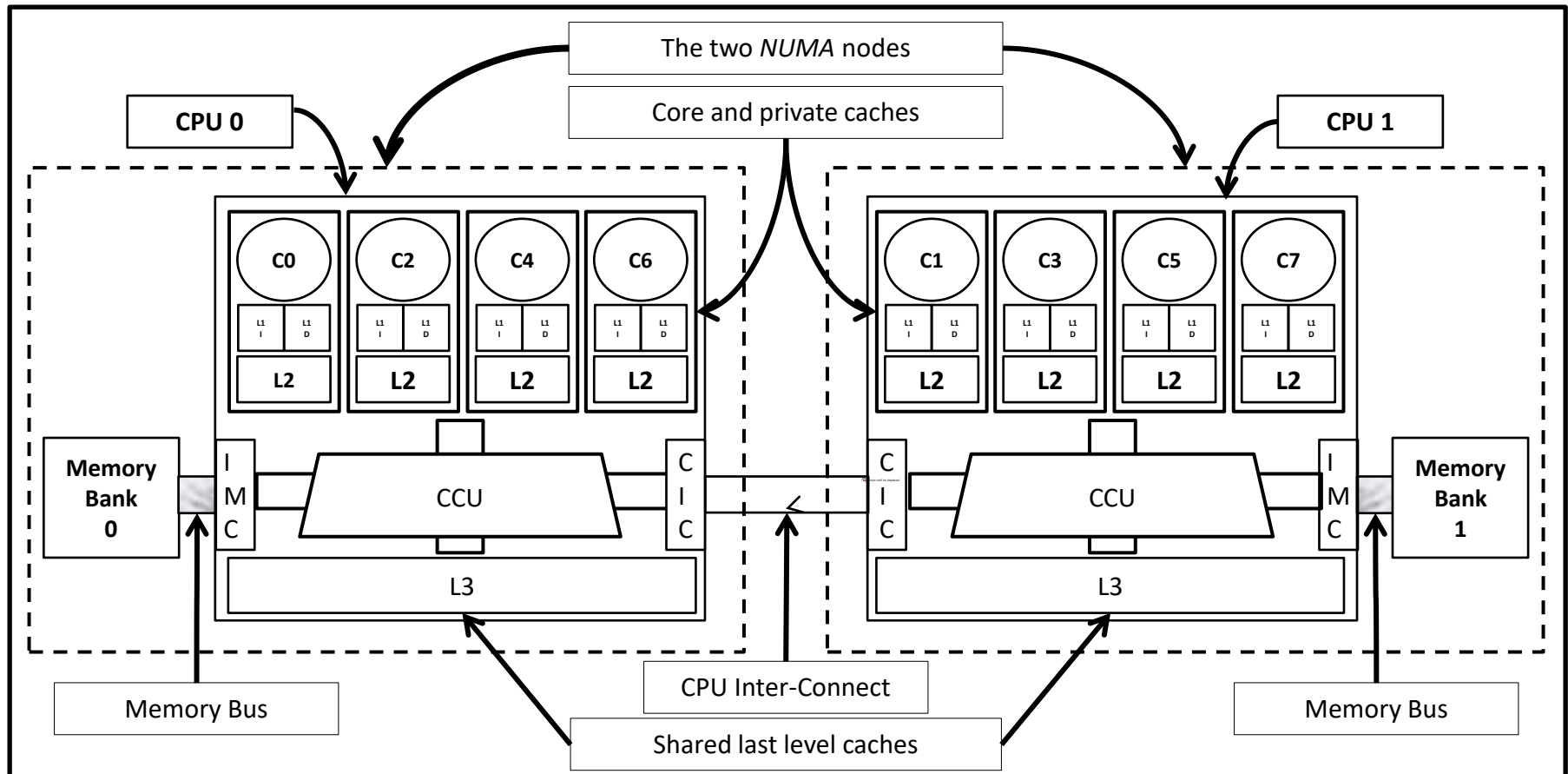
- It has often been stated that
 - “you can’t manage what you can’t measure” ...
- Effective tool to understand, model, test, and improve existing systems ...
 - E.g., often want to identify (and fix) system bottlenecks

Multicore systems



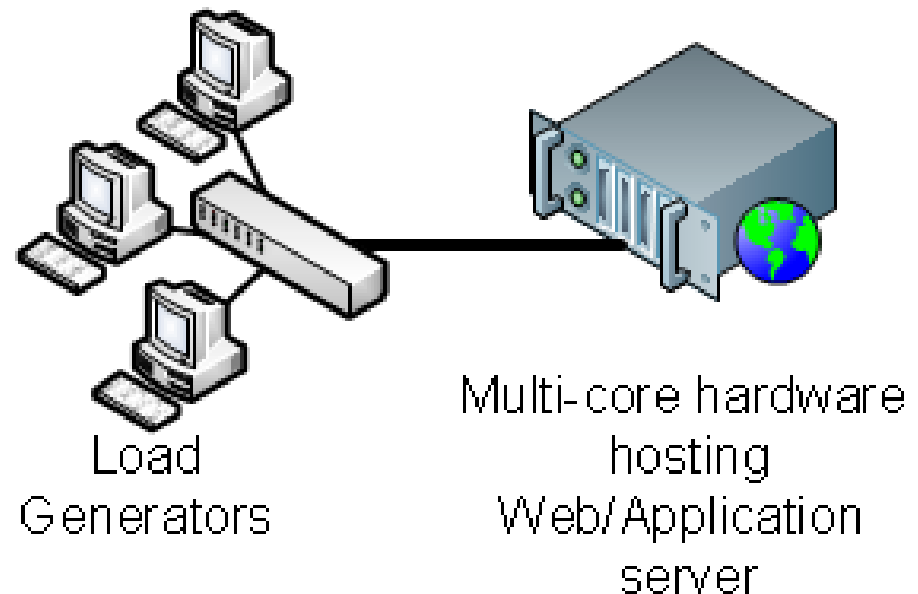
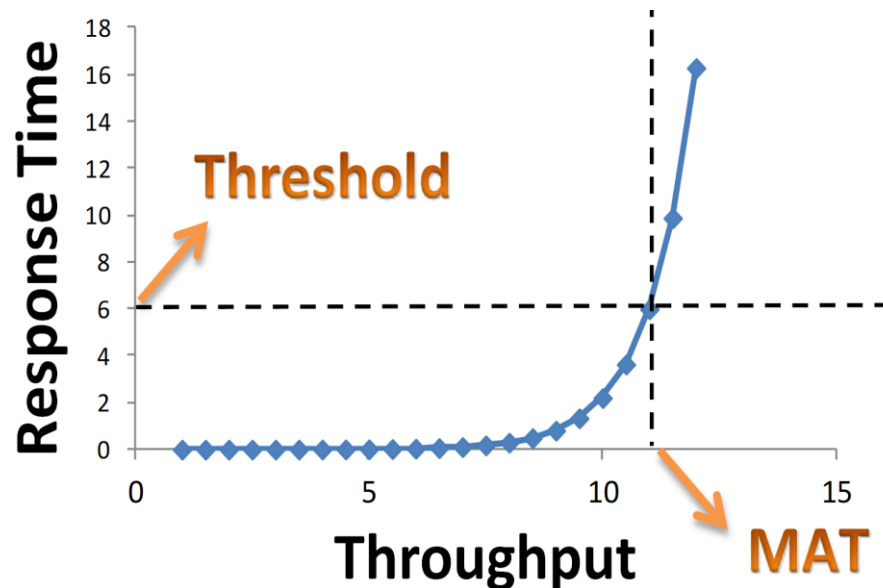
NUMA Architecture

An example of a two processor eight core NUMA system



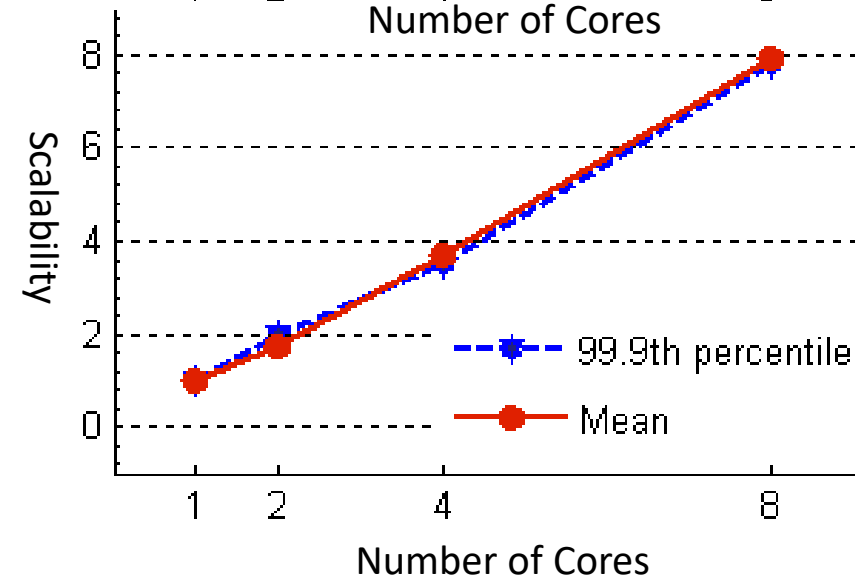
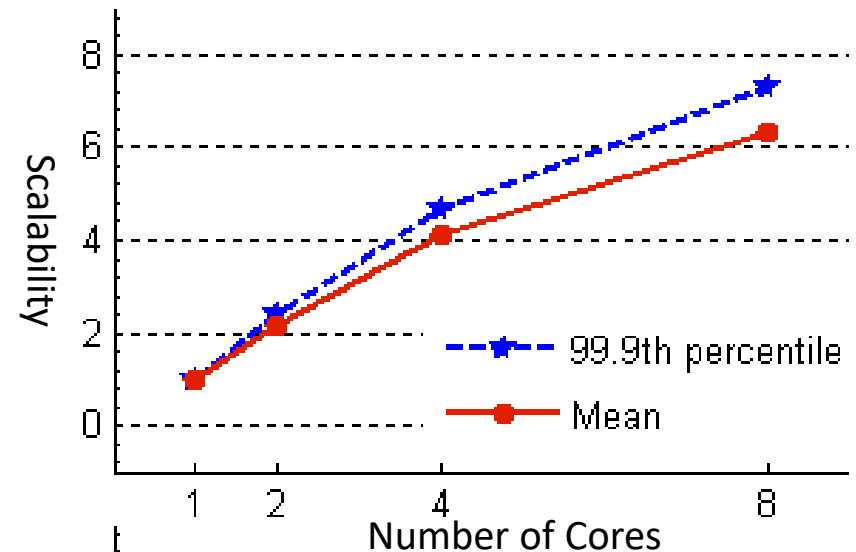
Scalability Evaluation Measurements

- E.g., Measure Web server scalability for workloads [ICPE '13]
 - Typically want to provide some 99% response time
 - Example scalability measure: Maximum Achievable Throughput (MAT)



RESULTS

- TCP/IP Intensive workload
 - Sub-linear
 - Maximum Achievable Throughput
 - 146,000 req/sec
- SPECweb Support workload
 - Almost linear
 - Maximum Achievable Throughput
 - 23,000 req/sec

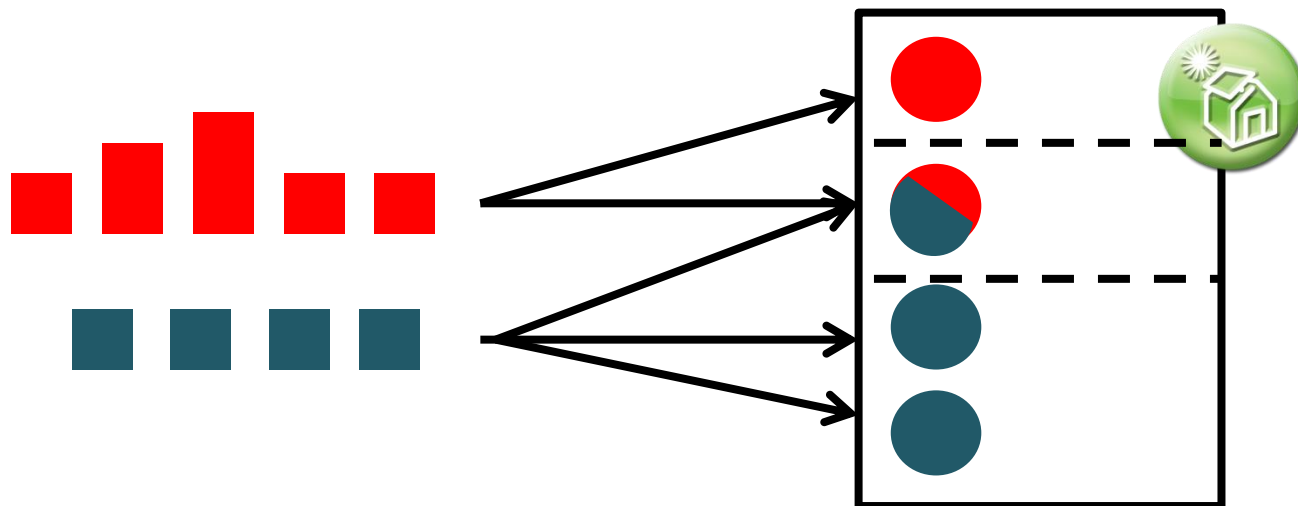


Identification of bottlenecks

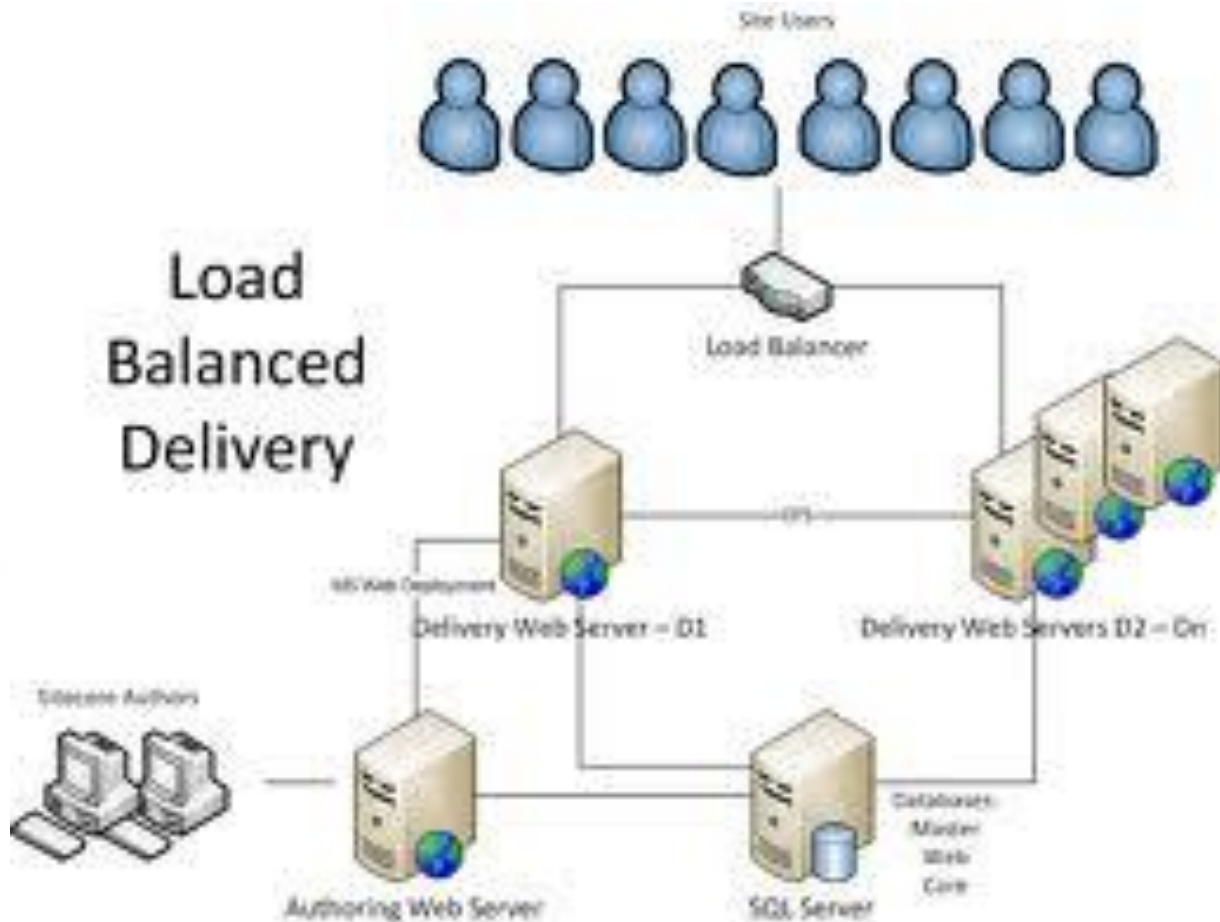
- E.g., memory, CPU, network, cache hierarchy, interconnect bus, scheduler, ...
 - Black-box testing
 - Low-level instrumentation

Identification of bottlenecks

- E.g., memory, CPU, network, cache hierarchy, interconnect bus, scheduler, ...
 - Black-box testing
 - Low-level instrumentation
- Multiple workloads ...



Often many servers (and racks)



... and data centers ...



... cost-efficient delivery ...

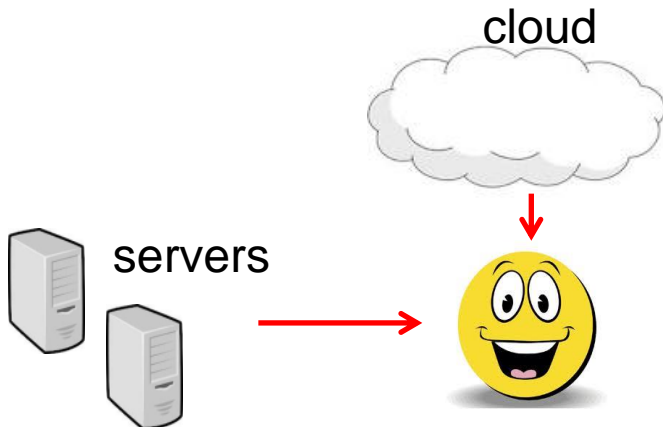


... and different flexibility ...

- Minimize content delivery costs

	Bandwidth	Cost
Cloud-based	Elastic/flexible	\$\$\$
Dedicated servers	Capped	\$

How to get the best of two worlds?



... and from who?



Measurements of Distributed Systems and Networks



Let's consider the Internet itself

- We are very reliant on the Internet
 - Today, it is hard to imagine a world without the internet
 - Yet it is growing increasingly complex ...
- Today: Wide area network that is too complex to fully grasp
 - Many protocols at various levels interact and effect behavior
- Many applications have performance requirements
 - End-to-end delay, loss, reliability, ...
- It is an interesting complex system with emergent characteristics like many living systems
 - Biological systems
 - Social networks

Internet Measurement Challenges

- Network size [quick “guestimates” ...]
 - $\sim O(1B)$ hosts in DNS, billions of users (and routers), $\sim O(100K)$ ASes, 20-30 billion connected devices ...
- Network Complexity
 - Interaction between components, protocols, applications, users
- All change over time
 - New applications are added
 - New protocol versions (TCP, QUIC, ...)
 - New router design (AQM)

Why do we measure the Internet?

- Already mentioned
 - Because it is there!
 - Operational reasons
- We cannot improve the Internet if we don't understand it
 - We cannot understand it if we don't measure
 - We cannot build effective models or simulators if we don't measure

What can we measure in the Internet?

- Structure
 - Topology (router/network) connectivity, link capacities, link loss, available bandwidth, routing, ...
- Traffic
 - End-to-end performance, packet arrival process (congestion built-up), ...
- Users and applications
 - WWW, peer-to-peer, streaming, gaming, ...
- Malicious behavior
 - Attack patterns, port scans, ...

Where can we measure the Internet?

How to choose representative measurement points?

Example: traffic samples

- LAN traffic vs. WAN traffic
- Inside an ISP vs. between continents
- Country biases
- Commercial location vs. educational
- More locations is better, BUT most of all, one point is better than no point

How can we measure the Internet?

- Active measurements
 - Probes: Traceroute, ping, packet trains
 - Application simulation
- Passive measurement
 - Logs (WWW)
 - Monitors, sniffers

When should we measure the Internet?

- Diurnal and weekly traffic cycles
- Time scales depend on “what” and “how”
- Passive measurement are typically continuous
 - Can generate **huge** datasets
 - Log access problems
 - Privacy concerns
- Active measurements are typically discrete
 - Important characteristics can be missed
 - Probes can be filtered and/or detected

Who is measuring the Internet?

- Businesses do a great deal of measurement
 - Mostly do not share with the research community
 - examples:
 - Akamai: http delay from server side
 - Google: everything
- Academia and Research institutes
 - Publish papers, but data may not always be available
 - Inform public and make recommendations
- Governments and their affiliates (e.g., MSB)

Publishing Internet Measurement Studies

- All major networking conferences & journals accept measurement papers
 - ACM SIGCOMM, IEEE INFOCOM, ACM SIGMETRICS
 - IEEE/ACM ToN, IEEE TPDS
- Dedicated meetings
 - ACM Internet Measurement Conf. (IMC)
 - Passive & Active Measurements Conf. (PAM)

E.g., PAM 2021 (live on YouTube Mon-Wed this week ...)

Active Measurement Techniques

Active Probes

- Active probes send stimulus (packets) into the network and then measure the response
 - Done on network, transport and application layers
- Active probes are useful to measure various things:
 - Delay, delay jitter, and loss
 - Topology and routing behavior
 - Capacity, bandwidth, and throughput

Example: RTT

Simple delay/loss probing with ping

```
C:\>ping www.fer.hr
```

Pinging www.fer.hr [161.53.72.111] with 32 bytes of data:

Reply from 161.53.72.111: bytes=32 time=113ms TTL=49

Reply from 161.53.72.111: bytes=32 time=111ms TTL=49

Reply from 161.53.72.111: bytes=32 time=113ms TTL=49

Reply from 161.53.72.111: bytes=32 time=118ms TTL=49

Ping statistics for 161.53.72.111:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

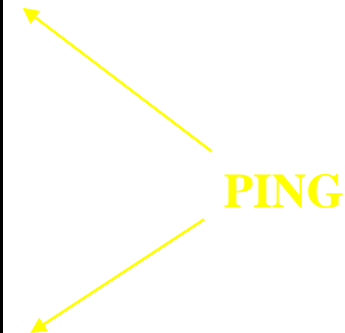
Minimum = 111ms, Maximum = 118ms, Average = 113ms

ICMP

ICMP is the IP error diagnosis protocol.

IP header	
Type	Code
Checksum	
Sequence number	
Any ICMP data	

ICMP Message Types	
Type No.	Meaning
0	Echo reply
3	Destination unreachable
4	Source quench
5	Redirect
8	Echo
9	Router advertisement
10	Router solicitation
11	Time exceeded
12	Parameter problem
13	Timestamp
14	Timestamp reply
15	Information requeste
16	Information reply



Application layer “ping”

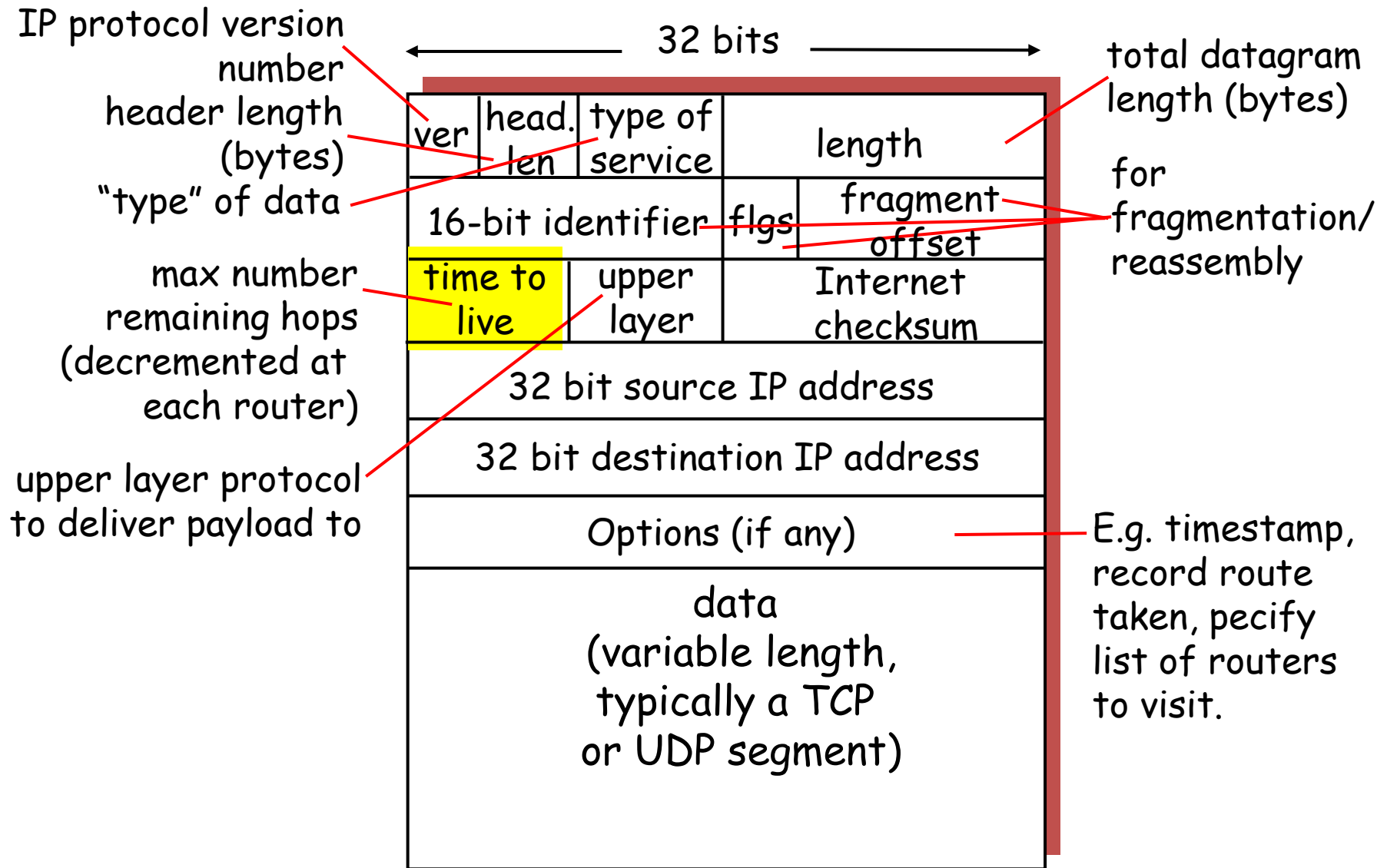
- One can generate application layer messages to test application reaction time
- Most common:
 - TCP SYN message to port 80

Example: Path

traceroute

- Useful to learn the route characteristics between two hosts.
- Sends a series of probes to successive nodes along a route to an intended destination and records the source address and time delay of the message returned by each.
- Based on ICMP “TTL expired” message

IP datagram format



ICMP Message Types

Type No.	Meaning
0	Echo reply
3	Destination unreachable
4	Source quench
5	Redirect
8	Echo
9	Router advertisement
10	Router solicitation
11	Time exceeded
12	Parameter problem
13	Timestamp
14	Timestamp reply
15	Information requeste
16	Information reply

<u>Type</u>	<u>Code</u>	<u>description</u>
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown

traceroute

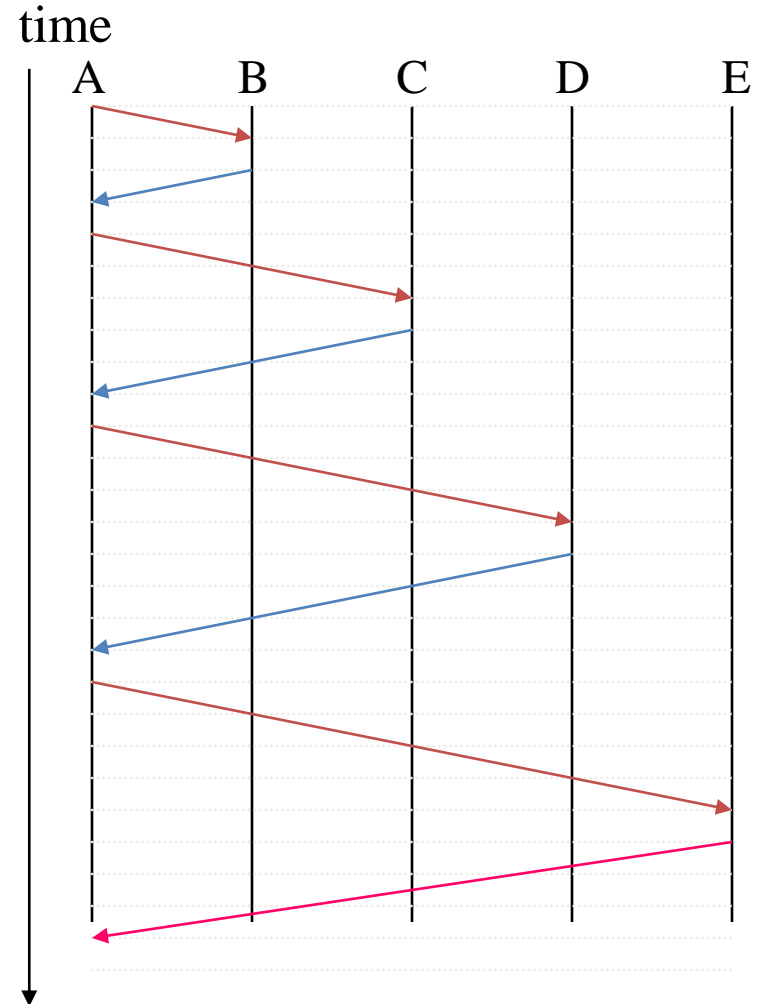
traceroute

Regular UDP packets

- successive TTLs

ICMP “TTL expired”
message

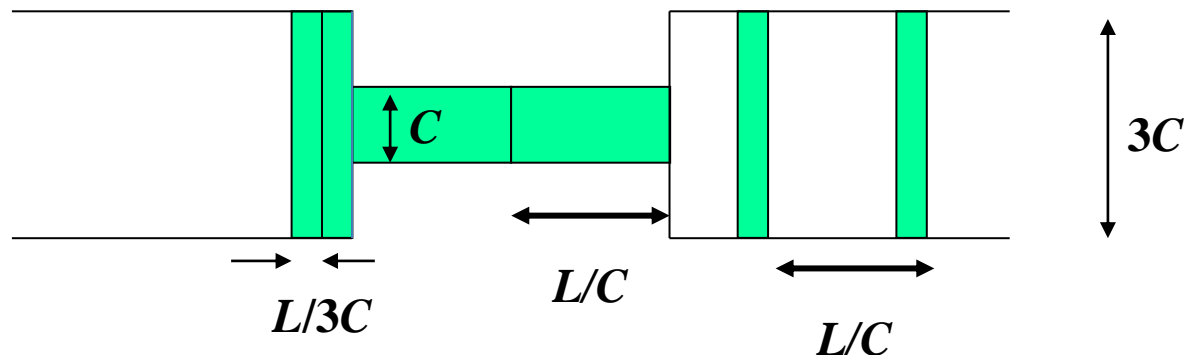
ICMP “port unreachable”
message



Example: Bottleneck capacity

Packet Dispersion to Estimate Capacity

- Packet transmission time: $\tau = L/C$
- Send two packets back-to-back
- Measure dispersion Δ at the receiver
- Estimate C as L/Δ

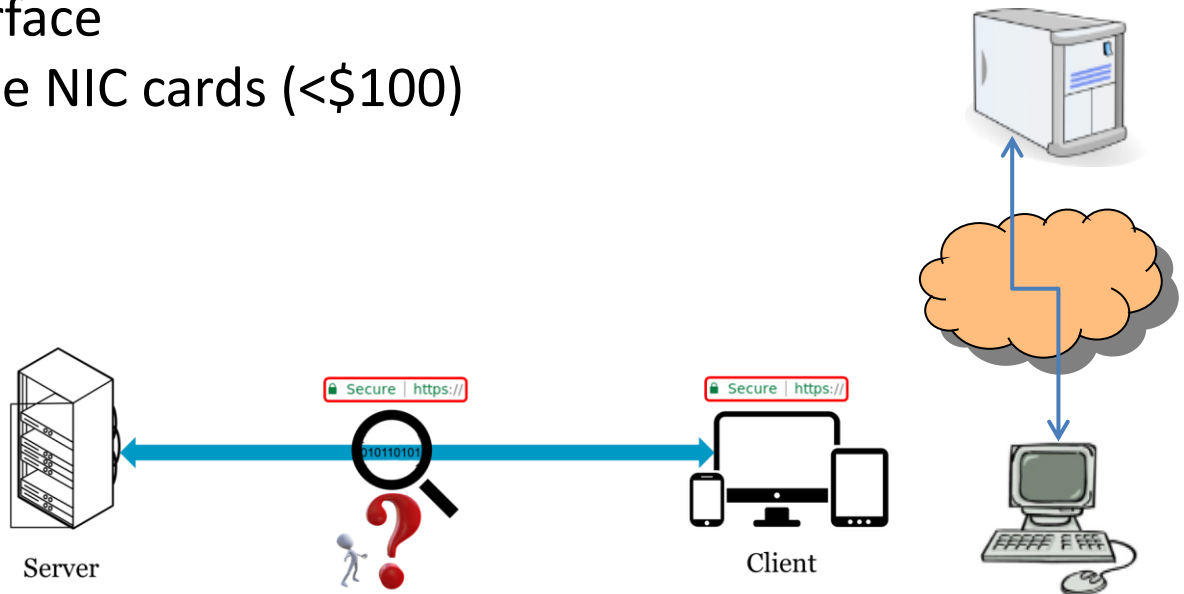


- But cross-traffic ‘noise’ can effect Δ .
- E.g., patchar “allows any user to find (estimate) the bandwidth, delay, average queue and loss rate of every hop between any source & destination on the Internet”

Passive Measurement Techniques

Passive packet measurement

- Capture packets as they pass by
 - Packet capture applications (e.g., tcpdump) on hosts use packet capture filter
 - Requires access to the wire
 - Promiscuous mode or mirror ports to see other traffic
 - Hardware-based solutions
 - Endace, Inc.'s DAG cards for monitoring almost every type of network interface
 - Programmable NIC cards (<\$100)
- Example issues:
 - Timestamps
 - Data volumes
 - Privacy



Passive IP flow measurement

- An IP flow is defined by the five-tuple:
 - src addr, src port, dst addr, dst port, protocol
- Cisco's NetFlow
 - Provide template-based flow records
- Many tools can manipulate NetFlow data

tcpdump

- Can capture entire packet or n first bytes
- Timestamps each packet
- Can filter based on any combination of header field

HTTP Logs

- Have data about the client IP, transaction time, command (GET/POST), return code, bytes transferred, referrer, metadata (browser type, OS, languages, etc.)
- Tools are available to analyze HTTP logs
 - Webalizer

HTTP Log Example

24.77.192.99 - - [15/May/2005:23:54:59 +0300] "GET /science_down.gif HTTP/1.1" 200 1138 "http://www.netdimes.org/science.html" "Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.7.7) Gecko/20050414 Firefox/1.0.3"

68.231.117.28 - - [15/May/2005:23:52:05 +0300] "GET /ipmap.png HTTP/1.1" 200 4874697 "http://slashdot.org/" "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.7.7) Gecko/20050414 Firefox/1.0.3"

24.236.177.187 - - [15/May/2005:23:55:00 +0300] "GET /home_up.gif HTTP/1.1" 200 1096 "http://www.netdimes.org/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)"

24.236.177.187 - - [15/May/2005:23:55:00 +0300] "GET /AboutUs_up.gif HTTP/1.1" 200 1169 "http://www.netdimes.org/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)"

24.77.192.99 - - [15/May/2005:23:55:00 +0300] "GET /Install_down.gif HTTP/1.1" 200 1219 "http://www.netdimes.org/science.html" "Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.7.7) Gecko/20050414 Firefox/1.0.3"

69.141.103.137 - - [15/May/2005:23:54:50 +0300] "POST /DIMES/server HTTP/1.1" 200 3 "-" "Java/1.4.1_03"

24.236.177.187 - - [15/May/2005:23:55:00 +0300] "GET /news_up.gif HTTP/1.1" 200 1086 "http://www.netdimes.org/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)"

24.236.177.187 - - [15/May/2005:23:55:00 +0300] "GET /community_up.gif HTTP/1.1" 200 1199 "http://www.netdimes.org/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)"

24.236.177.187 - - [15/May/2005:23:55:00 +0300] "GET /datastat_up.gif HTTP/1.1" 200 1233 "http://www.netdimes.org/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)"

24.236.177.187 - - [15/May/2005:23:55:00 +0300] "GET /science_up.gif HTTP/1.1" 200 1126 "http://www.netdimes.org/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)"

71.106.2.53 - - [15/May/2005:23:55:00 +0300] "GET /favicon.ico HTTP/1.1" 200 5694 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.8) Gecko/20050511 Firefox/1.0.4"

62.179.197.156 - - [15/May/2005:23:54:02 +0300] "GET /ipmap.png HTTP/1.1" 200 4874697 "http://slashdot.org/" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.8) Gecko/20050511 Firefox/1.0.4"

24.236.177.187 - - [15/May/2005:23:55:00 +0300] "GET /Install_up.gif HTTP/1.1" 200 1219 "http://www.netdimes.org/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)"

24.236.177.187 - - [15/May/2005:23:55:00 +0300] "GET /EVERGROW40.gif HTTP/1.1" 200 4089 "http://www.netdimes.org/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)"

195.252.52.155 - - [15/May/2005:23:55:00 +0300] "GET /science_down.gif HTTP/1.1" 200 1138 "http://www.netdimes.org/science.html" "Mozilla/5.0 (Windows; U; Windows NT 5.1; sv-SE; rv:1.7.6) Gecko/20050318 Firefox/1.0.2"

```
[root@jupiter httpd]# grep "GET / " access_log | tail -10
68.54.223.47 - - [19/May/2005:12:36:20 +0300] "GET / HTTP/1.1" 200 14067 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322)"
132.76.80.118 - - [19/May/2005:12:49:44 +0300] "GET / HTTP/1.1" 304 - "http://www.eng.tau.ac.il/~shavitt/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322)"
24.169.148.213 - - [19/May/2005:13:06:58 +0300] "GET / HTTP/1.1" 200 14067 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.8) Gecko/20050511 Firefox/1.0.4"
84.170.181.64 - - [19/May/2005:13:07:14 +0300] "GET / HTTP/1.1" 200 14067 "http://www.google.de/search?hl=de&q=dimes&meta=" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)"
130.240.136.220 - - [19/May/2005:13:07:25 +0300] "GET / HTTP/1.1" 304 - "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)"
81.72.13.30 - - [19/May/2005:13:11:00 +0300] "GET / HTTP/1.1" 200 14067 "http://www.miranet.it/php/Articolo.php?id=708" "Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"
194.78.199.123 - - [19/May/2005:13:13:44 +0300] "GET / HTTP/1.1" 200 14067 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; .NET CLR 1.1.4322)"
82.152.182.12 - - [19/May/2005:13:23:10 +0300] "GET / HTTP/1.1" 200 14067 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)"
80.119.126.44 - - [19/May/2005:13:38:08 +0300] "GET / HTTP/1.1" 200 14067 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.8) Gecko/20050511 Firefox/1.0.4"
80.250.186.101 - - [19/May/2005:13:46:14 +0300] "GET / HTTP/1.1" 200 14067 "http://distributed.ru/forum/?a=topic&topic=583" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.8) Gecko/20050511 Firefox/1.0.4"
```

Other examples

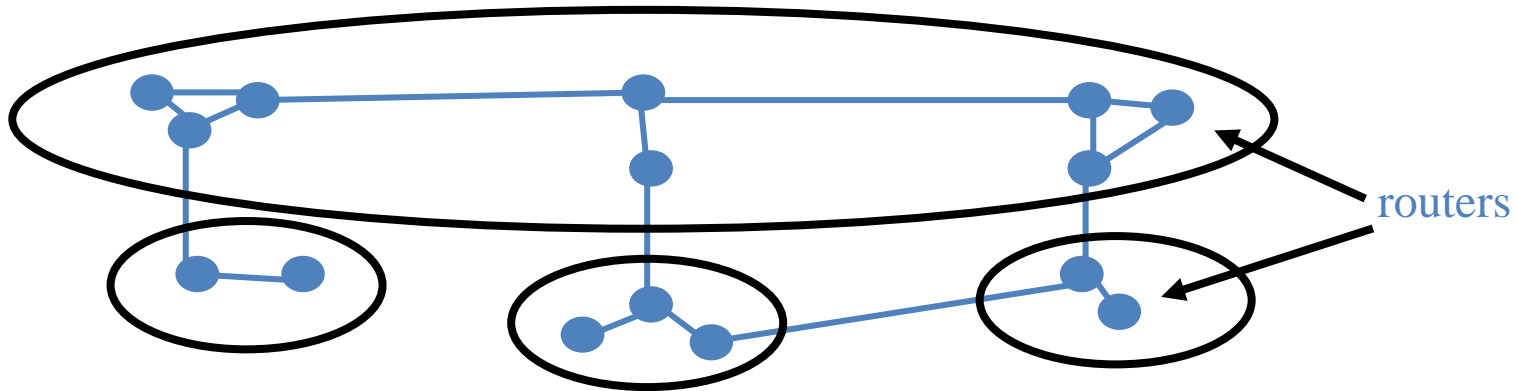
- Zeek (formerly Bro)
 - Open-source network security monitoring tool that allows easy extraction of information from the network traffic
 - Flexible and powerful when wanting to extract information from the various network layers
 - Typically use scripts to create logs
- Wireshark (used in labs)
 - Has “cute” user interface, is more “plug-and-play”, and faster to get up-to-speed

Measuring the Internet's topology

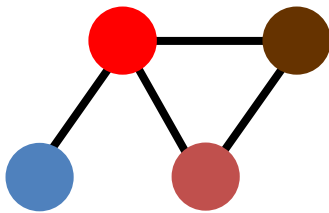
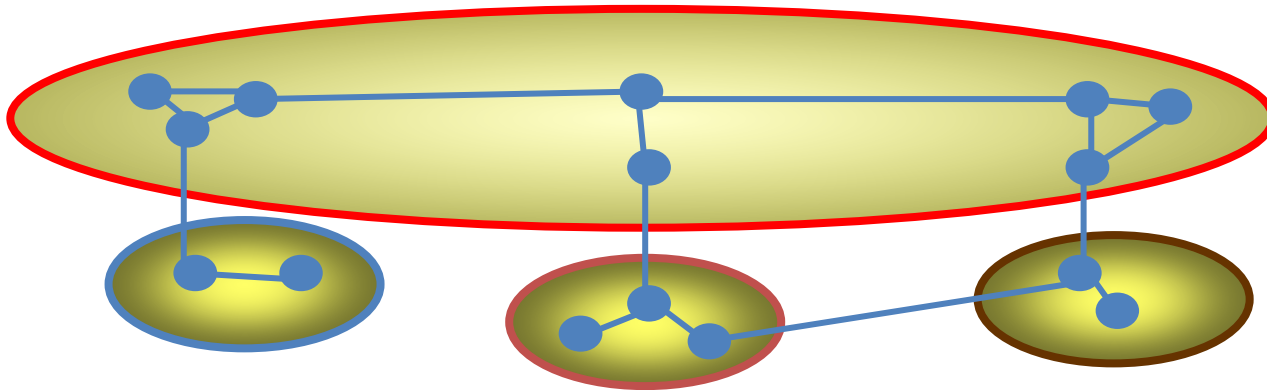
Outline

- Background
- Then, both active and passive examples ...

The Internet Structure

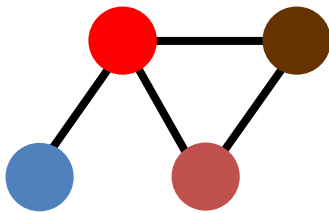
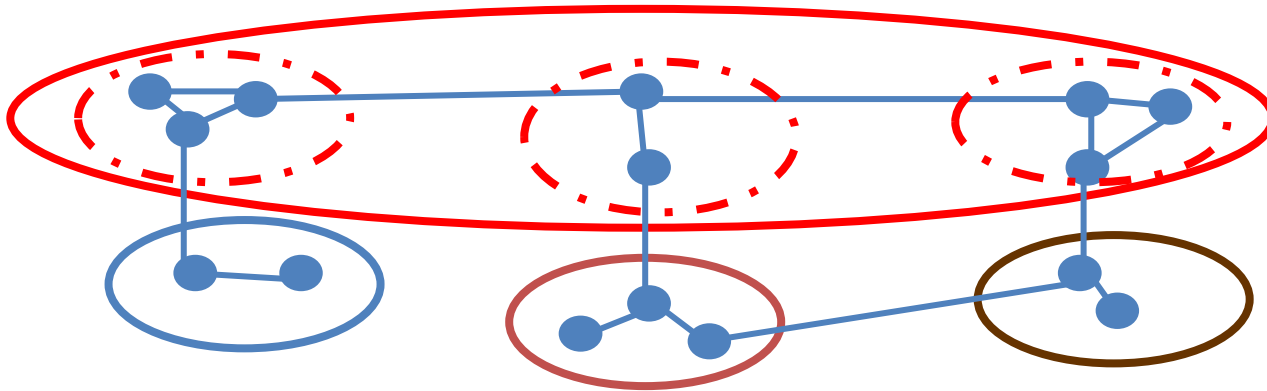


The Internet Structure

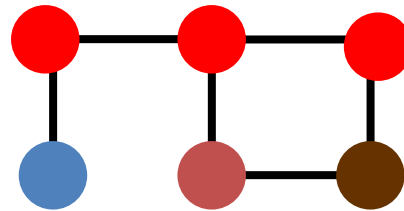


The **AS** graph

The Internet Structure



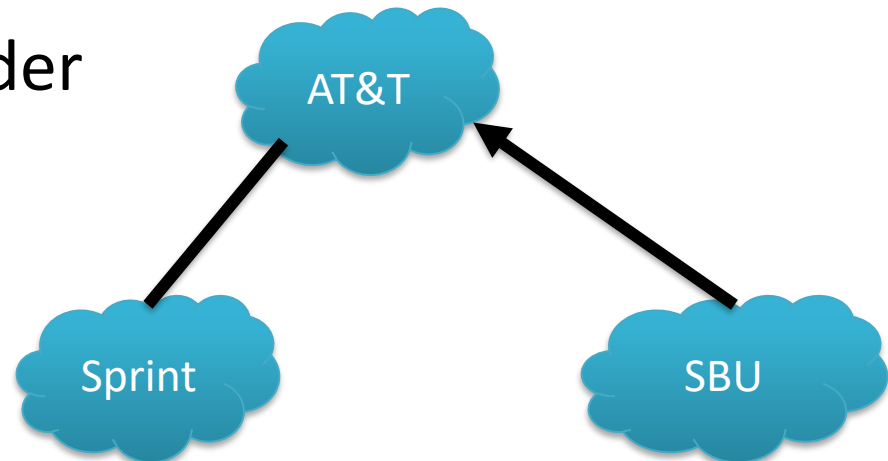
The **AS** graph



The **PoP level** graph

Measuring the Internet's topology

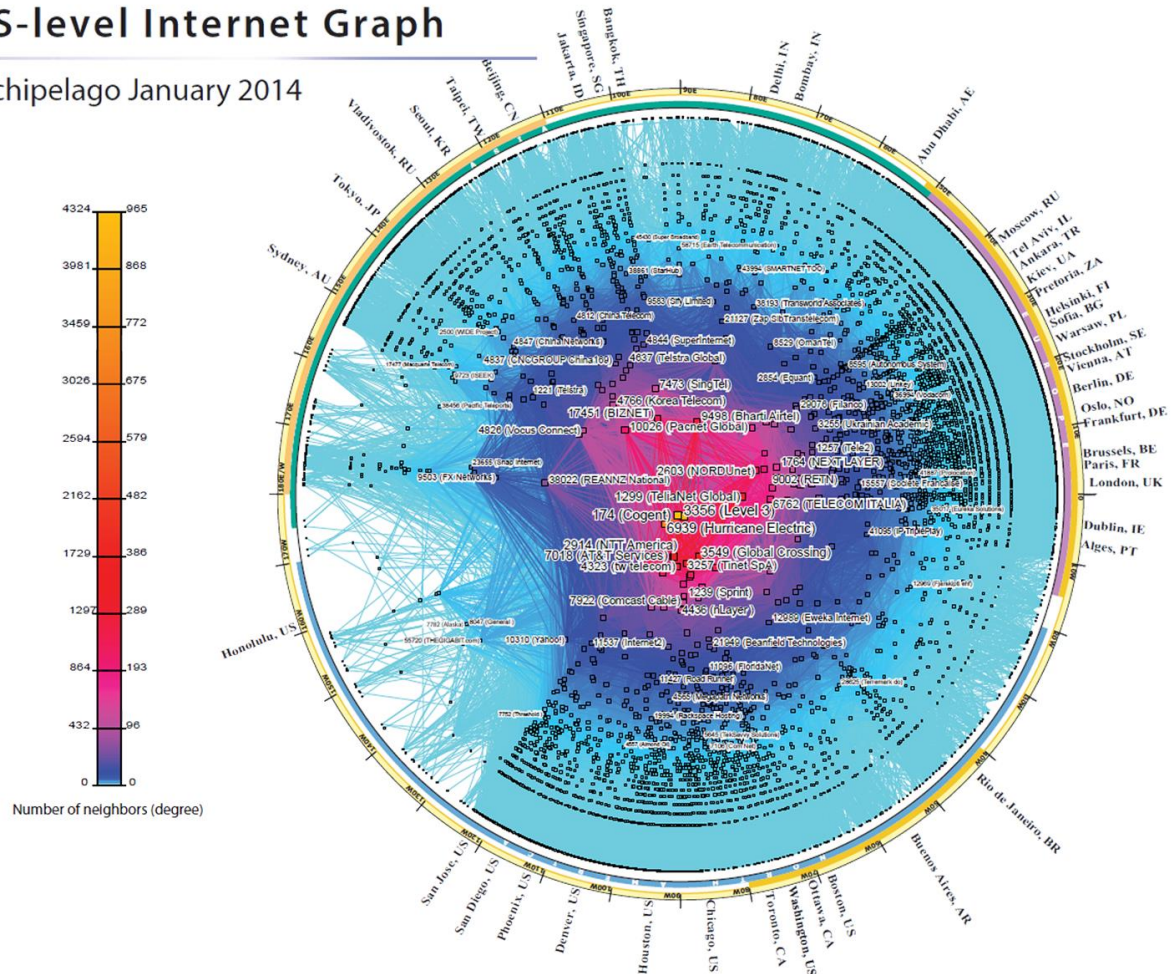
- What do we mean by topology?
 - Internet as graph
 - Edges? Nodes?
 - Node = Autonomous System (AS)
 - Edge = connection.
- Edges labeled with business relationship
 - Customer → Provider
 - Peer -- Peer



The outputs

CAIDA's IPv4 AS Core AS-level Internet Graph

Archipelago January 2014

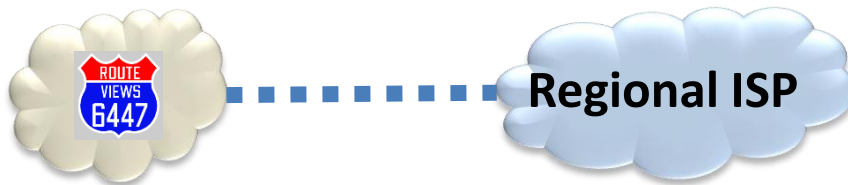


So how do we measure this graph?

- Passive approach: BGP route monitors
 - Coverage of the topology
 - Amount of visibility provided by each neighbor
- Active approach: Traceroute
 - From where?
 - Traceroute gives series of IP addresses not ASes

Passive approach: BGP Route Monitors

- Receive BGP announcements from participating ASes at multiple vantage points



www.routeviews.org

“originally motivated by interest on the part of operators in determining how the global routing system viewed their prefixes and/or AS space”

www.routeviews.org

Going from BGP Updates to a Topology

Example update:

- TIME: 03/22/11 12:10:45
- FROM: 12.0.1.63 AS7018
- TO: 128.223.51.102 AS6447
- ASPATH: 7018 4134 9318 32934 32934 32934
- 69.171.224.0/20

AT&T (AS7018) is telling
Routeviews (AS 6447) about this route.

This /20 prefix can be reached via
the above path

Going from BGP Updates to a Topology

- Key idea
 - The business relationships determine the routing policies
 - The routing policies determine the paths that are chosen
 - So, look at the chosen paths and infer the policies
- Example: AS path “7018 4134 9318” implies
 - AS 4134 allows AS 7018 to reach AS 9318
 - China Telecom allows AT&T to reach Hanaro Telecom
 - Each “triple” tells something about transit service

Why are peering links hard to see?

- **The challenge:**

- BGP announcements *do not reflect complete connectivity* information
- They are an agreement to transit traffic for the AS they are advertised to... Regional ISP won't see the peering e

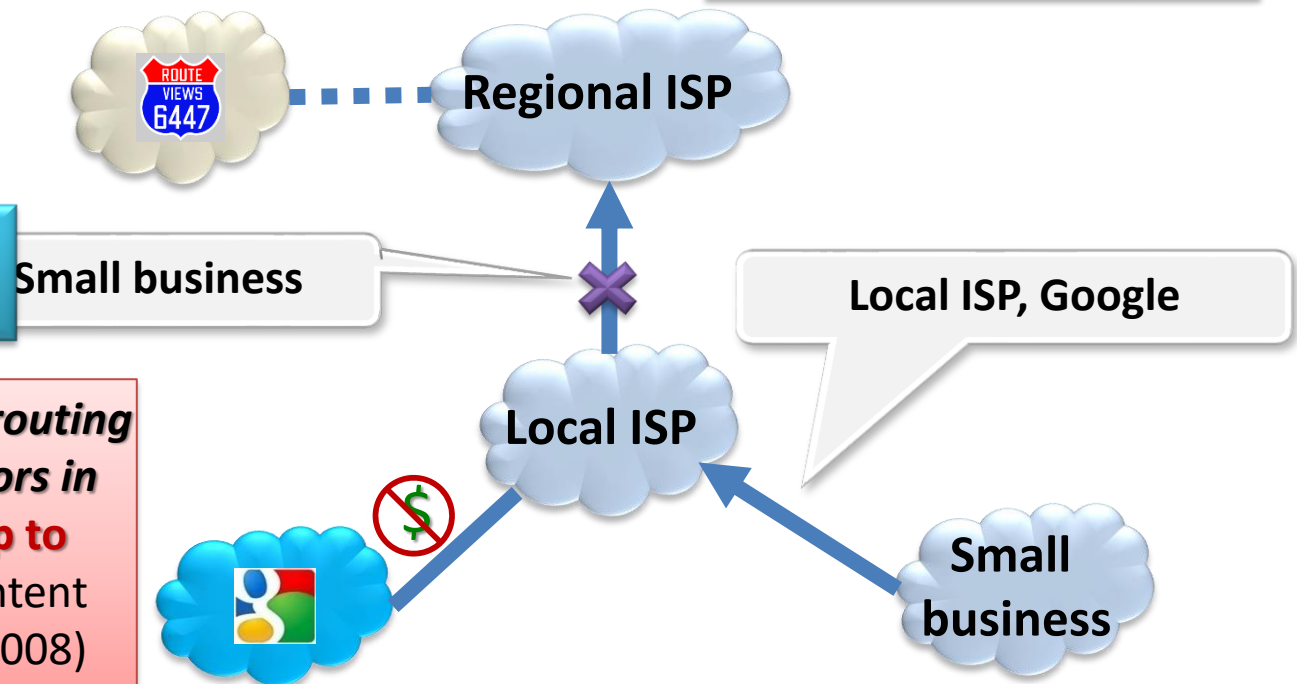
Regional ISP won't see the peering edge!

Neither will Routeviews

Local ISP will **only tell its customers** about the peering link.

(ASes only transit traffic if it generates revenue!)

Combination of ***no valley routing policy*** and a ***lack of monitors in stub ASes*** mean missing **up to 90%** of peering links of content providers! (Oliveria et al. 2008)



Active approach: Traceroute

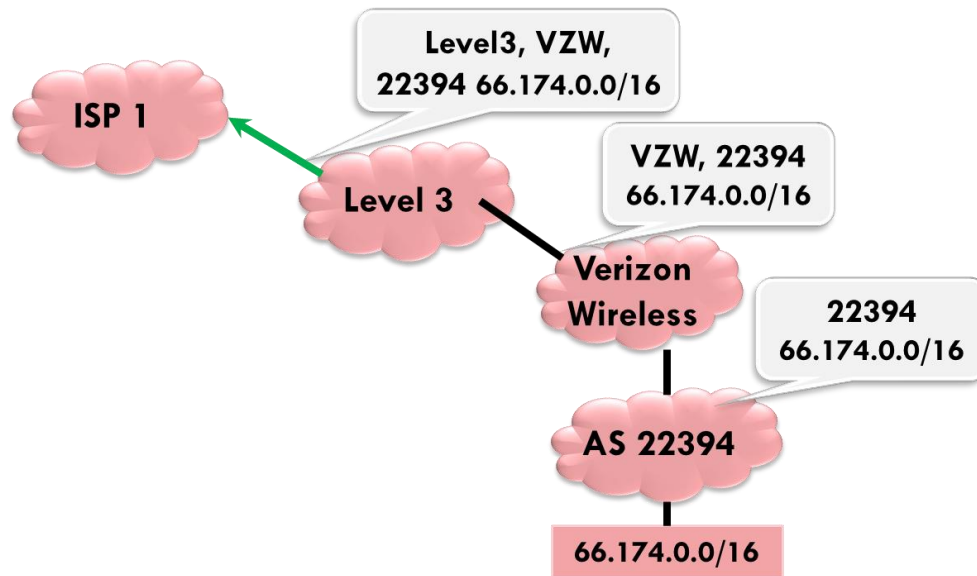
- Issue: Need control over end hosts to run traceroute
 - How to get VPs?
- <http://www.traceroute.org/>
 - Collection of $O(100)$ servers that will run traceroute
 - Hosted by ISPs/other network operators (e.g. universities)
- RIPE Atlas
 - Distribute specialized hardware to volunteers
 - $O(1000s)$ of probes
- Dasu
 - Bittorrent plug in that does measurements
 - $O(200)$ ASes with Dasu clients



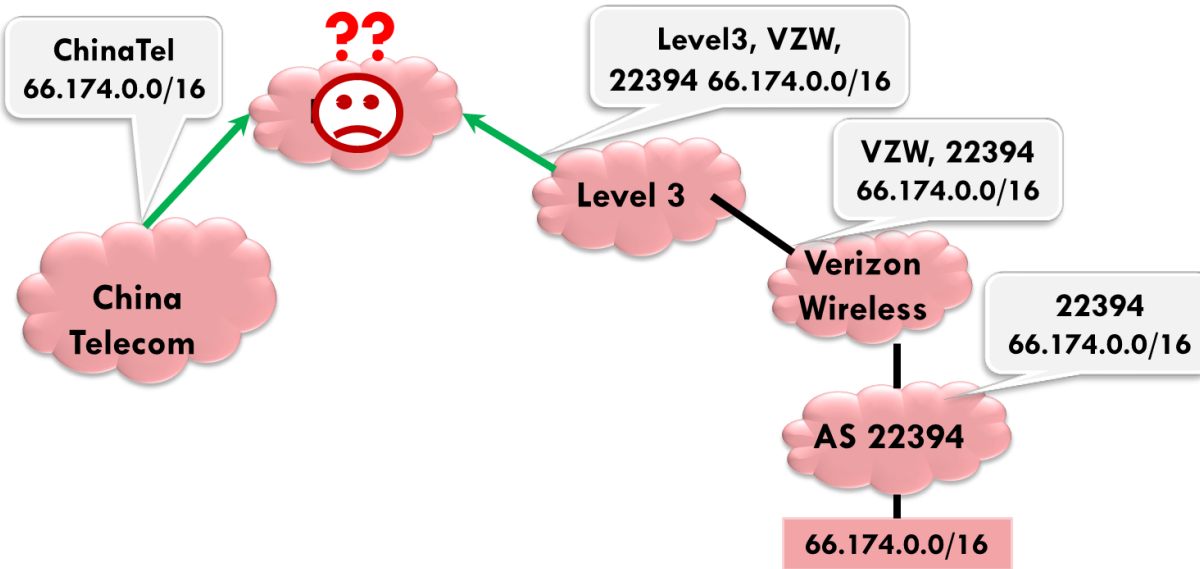
iPlane and PlanetLab

- iPlane
 - Performs traceroutes from PlanetLab nodes and traceroute servers to construct a router interface-level Internet map.
 - Clustering interfaces into PoPs
 - Based on TTL response time
 - Latency prediction (but have also been used for other things)
- PlanetLab
 - Old data: 1080 nodes over 496 (academic) sites
 - Bare bone machines. Load your own tool.
 - Host various measurement projects:
 - DIMES
 - iPlane
 - ScriptRoute: (flexible scripts)
 - ... (including some past TDDE35 course projects)

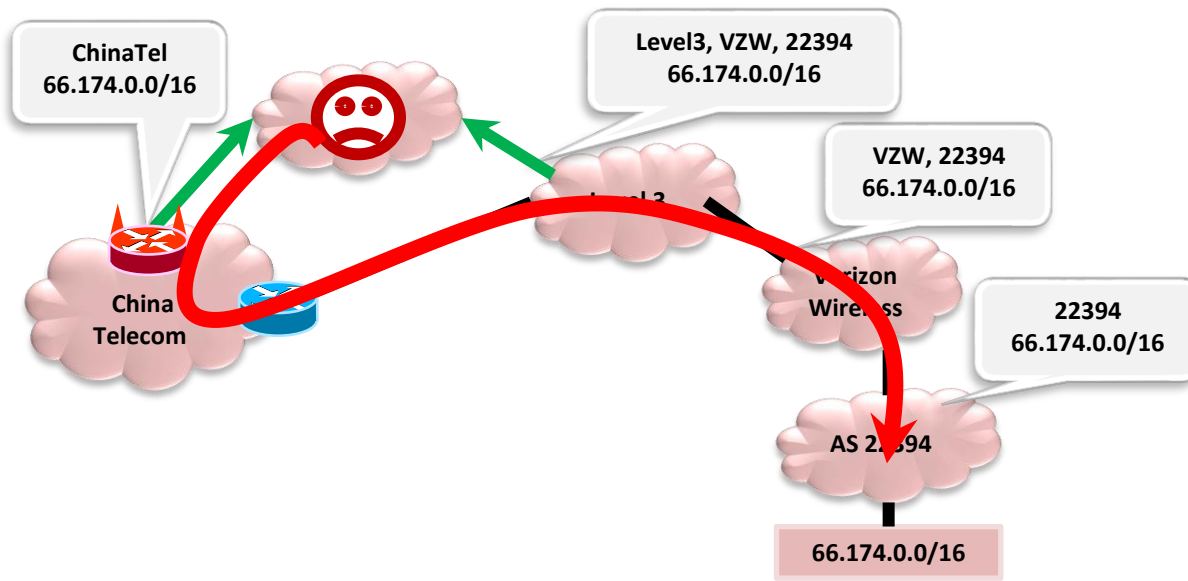
Traceroute vs Announced Path



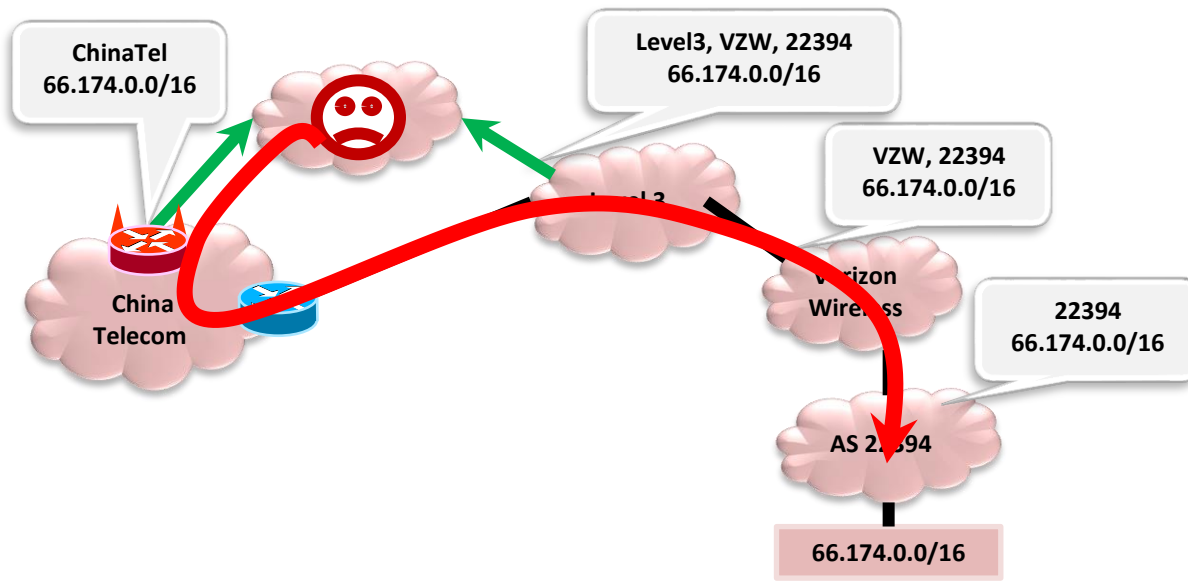
Traceroute vs Announced Path



Traceroute vs Announced Path



Traceroute vs Announced Path



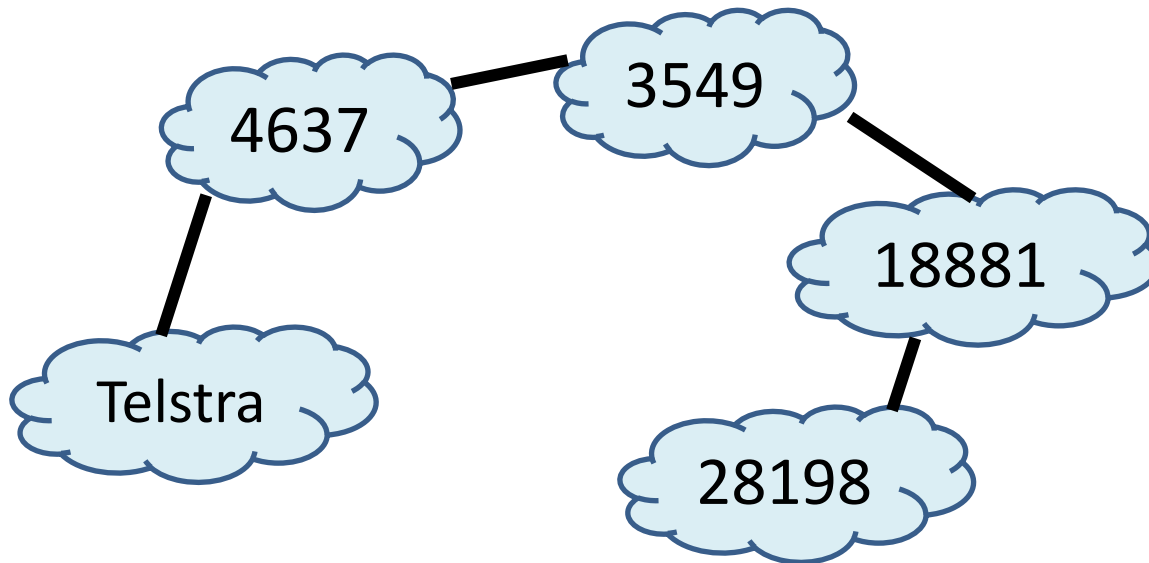
Interception typically results in differences between

— **Announced AS-PATH**

— **Data path (traffic)**

Policy checks if legit reason(s)

Traceroute vs Announced Path



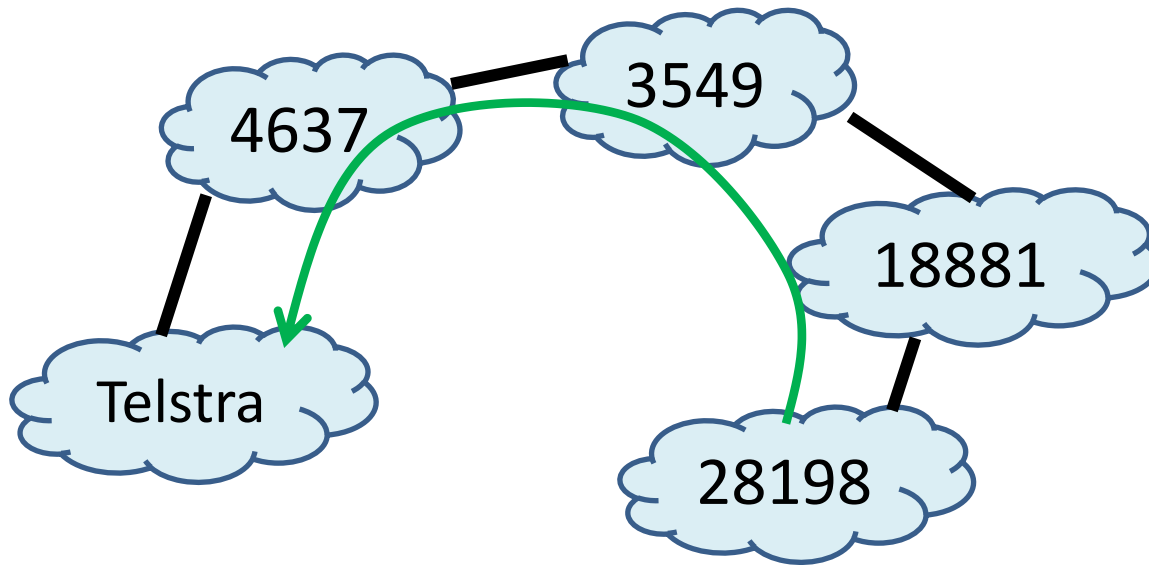
Sometimes differences

— Announced AS-PATH

— Data path (traffic)

Many legit reason(s)

Traceroute vs Announced Path



Sometimes differences

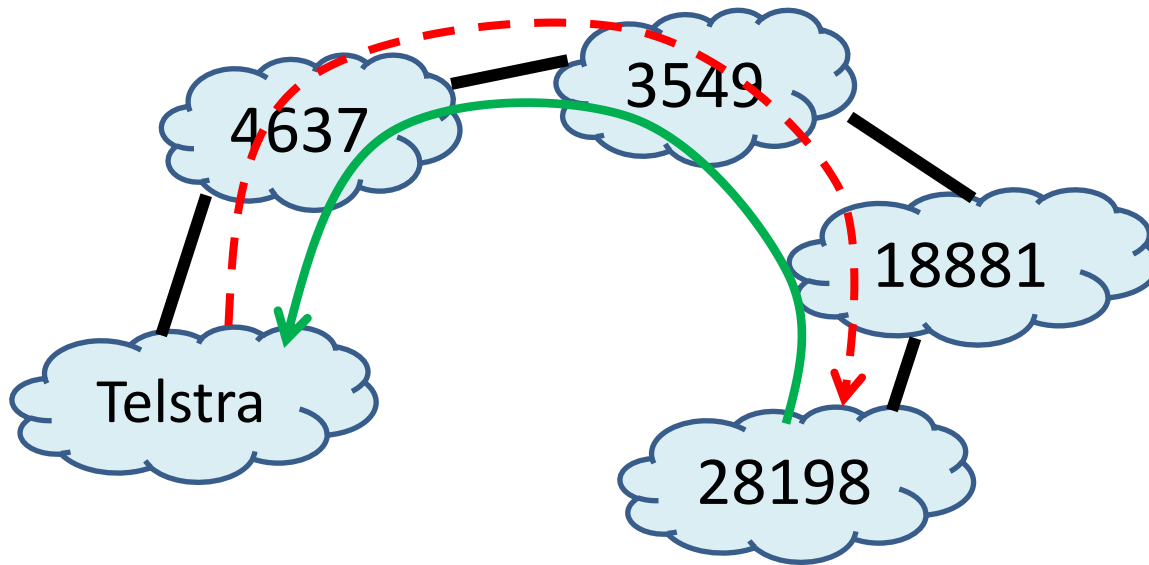
— Announced AS-PATH

— Data path (traffic)

Many legit reason(s)

AS-PATH: 177.52.48.0/21 | 1221 4637 3549 18881 28198

Traceroute vs Announced Path



Sometimes differences

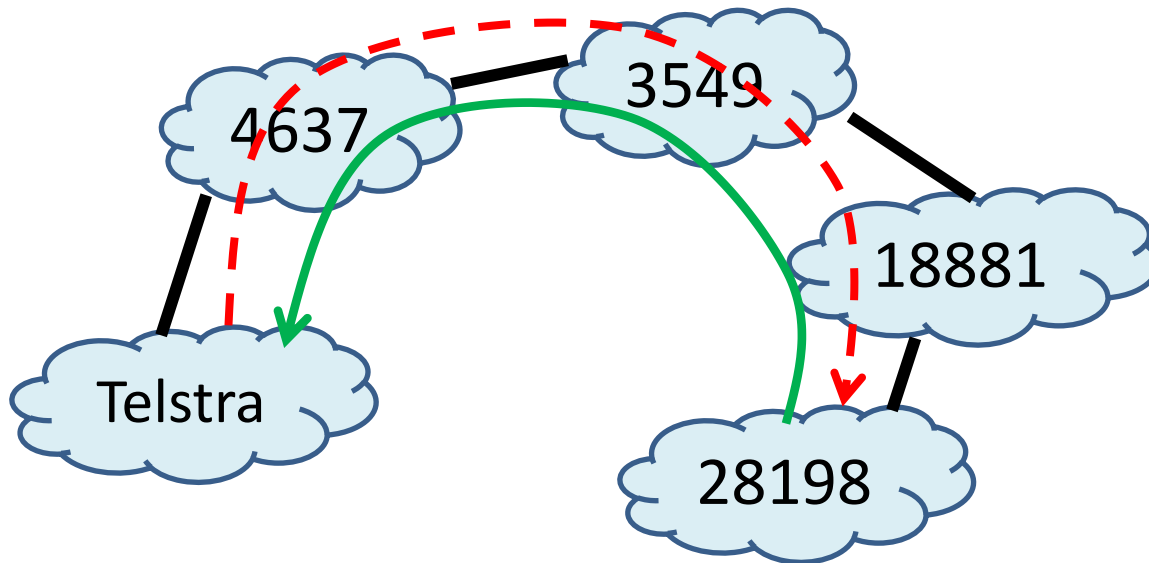
— Announced AS-PATH

— Data path (traffic)

Many legit reason(s)

AS-PATH: 177.52.48.0/21 | 1221 4637 3549 18881 28198

Traceroute vs Announced Path



Sometimes differences

— Announced AS-PATH

— Data path (traffic)

Many legit reason(s)

AS-PATH: 177.52.48.0/21 | 1221 4637 3549 18881 28198

Traceroute:

... (initial hops)

9. telstraglobal.net (134.159.63.202) 164.905 ms

10. impsat.net.br (189.125.6.194) 337.434 ms

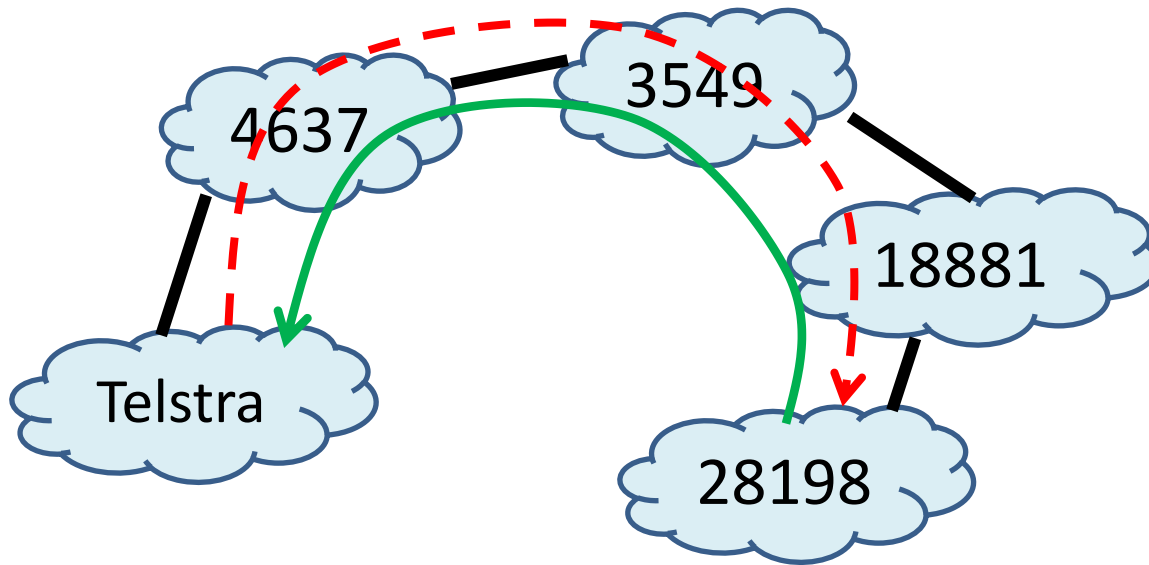
11. spo.gvt.net.br (187.115.214.217) 332.926 ms

12. spo.gvt.net.br (189.59.248.109) 373.021 ms

13. host.gvt.net.br (189.59.249.245) 343.685 ms

14. isimples.com.br (177.52.48.1) 341.172 ms

Traceroute vs Announced Path



Sometimes differences

— Announced AS-PATH

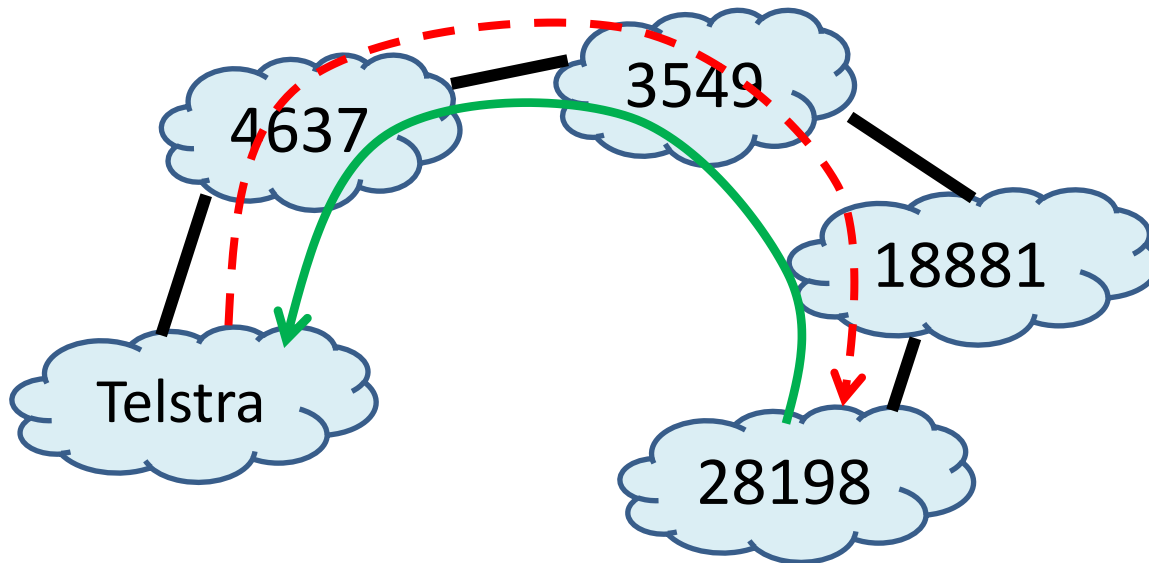
— Data path (traffic)

Many legit reason(s)

AS-PATH: 177.52.48.0/21 | 1221 4637 3549 18881 28198

AS HOPS in traceroute: 1221 1221 1221 1221 4637 4637 4637 4637
4637 3549 3549 3549 18881 18881 18881 18881 28198

Traceroute vs Announced Path



Sometimes differences

— Announced AS-PATH

— Data path (traffic)

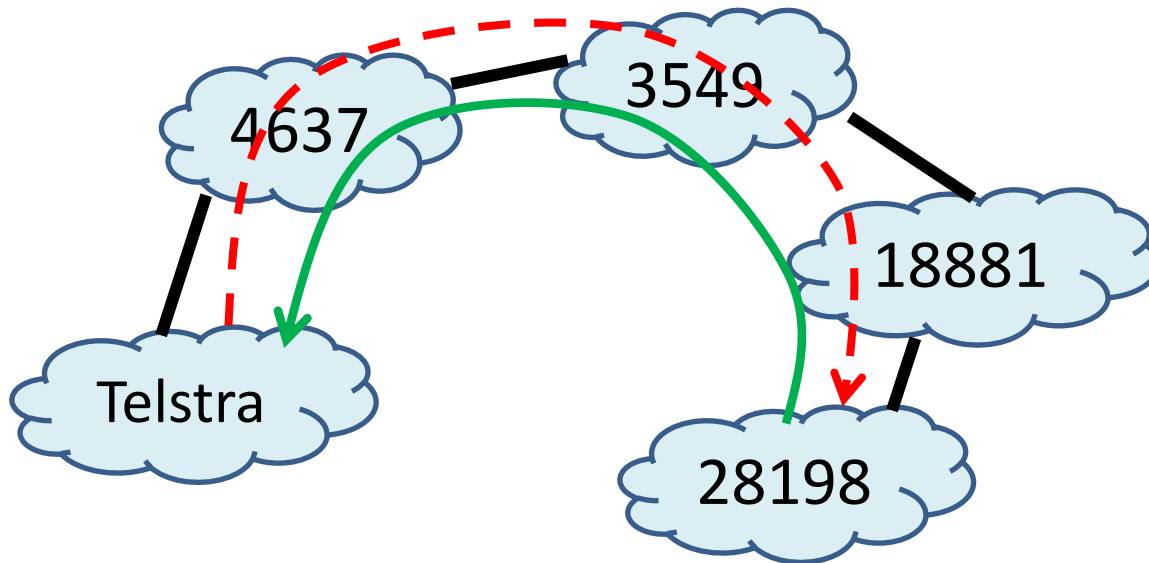
Many legit reason(s)

AS-PATH: 177.52.48.0/21|1221 4637 3549 18881 28198

AS HOPS in traceroute: 1221 1221 1221 1221 4637 4637 4637 4637
4637 3549 3549 3549 18881 18881 18881 18881 28198

Traceroute-PATH: 1221 4637 3549 18881 28198

Traceroute vs Announced Path



Sometimes differences

— Announced AS-PATH

— Data path (traffic)

Many legit reason(s)

AS-PATH: 177.52.48.0/21 | 1221 4637 3549 18881 28198

AS HOPS in traceroute: 1221 1221 1221 1221 4637 4637 4637 4637
4637 3549 3549 3549 18881 18881 18881 18881 28198

Traceroute-PATH: 1221 4637 3549 18881 28198