Large-scale Distributed Systems and Networks

(Storskaliga Distribuerade System och Nätverk)

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

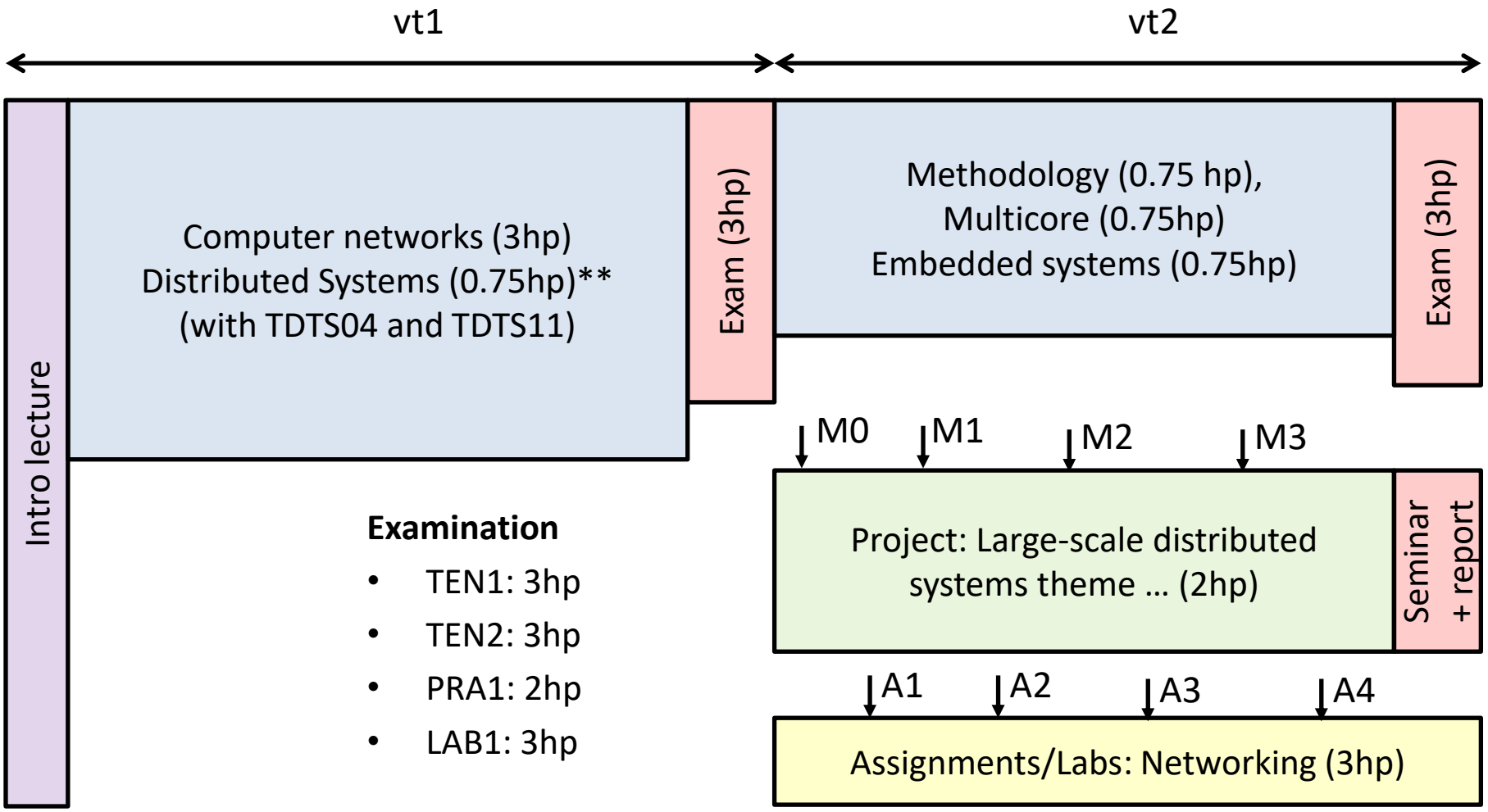
Scalability and systems thinking

- Systems thinking with focus on scalability
  - Holistic perspective (layers, components, etc.)
  - Large distributed systems and services
  - Networks and distributed systems ”hand-in-hand”
  - Single to multicore; single to million machines/users
  - Scalable methods and architectures
  - Modeling and abstraction of big systems (including some basic mathematical modeling)
- Mix of theory and practice
  - ”The knowledge is not yours until you use it”
  - Using experiments and measurements to improve the understanding of real systems in the wild + discuss the future

Subject knowledge

- Networking (vt1)
  - Basics/foundation, similar to TDTS06, TDTS11, and TDTS04 (12-14 lectures). Gives eligibility to TDTS21 (advanced networking).
  - Assignments/labs (at least one for each of the three layers 3, 4, and 5)
- Distributed systems (vt1)
  - Some introductory lectures (4 lectures)
  - Project (groups of 3-4 students)
- Multicore (vt2)
  - Kristoffer Kessler (4 lectures)
- Embedded systems (vt2)
  - Petru Eles (3 lectures)
- Methods to understand and evaluate large-scale systems (vt2)
  - Some introductory lectures (4 lectures)
  - Modeling, abstraction, and data-driven analysis methods for large-scale systems and services

Overview (2024)



Examination

- TEN1: 3hp
- TEN2: 3hp
- PRA1: 2hp
- LAB1: 3hp

Overview (2024)…

vt1

Exam	Teaching/exam block	Comment
TEN1	Networking (Ch 1-4)	Niklas (TEN1)
TEN2	Distributed systems	Niklas (TEN2) **
TEN1	Networking (Ch 5-9)	Niklas (TEN1)

vt2

Exam	Teaching/exam block	Comment
TEN2	Methodologies	Niklas (TEN2)
TEN2	Multicore	Christoff (TEN2)
TEN2	Embedded systems	Petru (TEN2)

Exam	Teaching/exam block	Comment
LAB1	Assignments	A1-A4
PRA1	Milesones	M0-M3
PRA1	Seminars	Mid+End

TEN1 (March 19): 3 ECTS

TEN2 (May 31): 3 ECTS

LAB1 (vt2): 3 ECTS

PRA1 (vt2): 2 ECTS

Projects and assignments

- **2024: Practice working in teams of 2-4 students.**

- Assignments (and lab sessions)
  - Groups of 2 students
  - Register in webreg by Thursday (Mar. 28, 2024)
  - 4 assignments:
    - Split across multiple network layers
    - 2 x wireshark (HTTP + TCP), proxy, and DV
  - Note: Many advantages having labs during vt2 (instead of vt1) – actually think it is better! Try to use them to finish the labs quicker.
- 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 (Mar. 28, 2024)
  - Projects released by Tuesday (Apr. 2, 2024)
  - Request projects by Thursday (priority if on Wednesday)

Course evaluations [focus on complaints]

- Different quality of lectures: For those with most complaints, there are online alternatives + excellent textbook. Also, added new instructor and changed distribution of lectures.
- Shared summary lecture confusing [2023]: Separated + Niklas gives lecture for TDDE35
- vt1 + vt2 split [*Per design. See prior slides + explanations/motivations*]
  - Student prioritize Pintos + envar. [*Explain/motivate*]
  - [some] Time consuming [*Explain/motivate*]
  - Some suggestions [mostly not feasible/reasonable]
- Expected attendance at seminars (2x 6hrs): *Not much time + Important learning opportunity*
- Request for help/suggestions how to succeed in TDDE35 [*Added more examples + pointers*]
- Some comments may have been a personal (defense) response

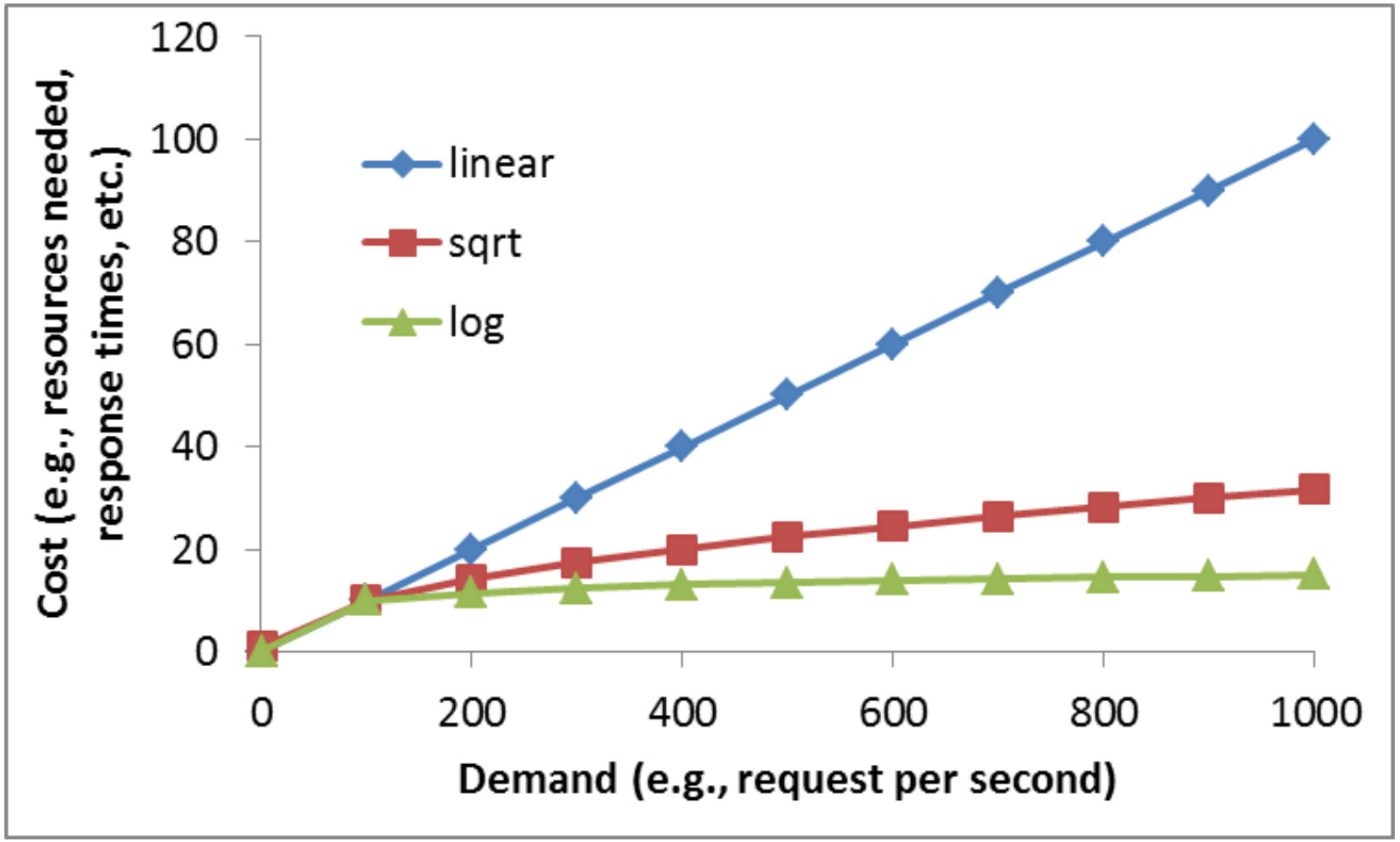
Reminder: Attend lectures …

- 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

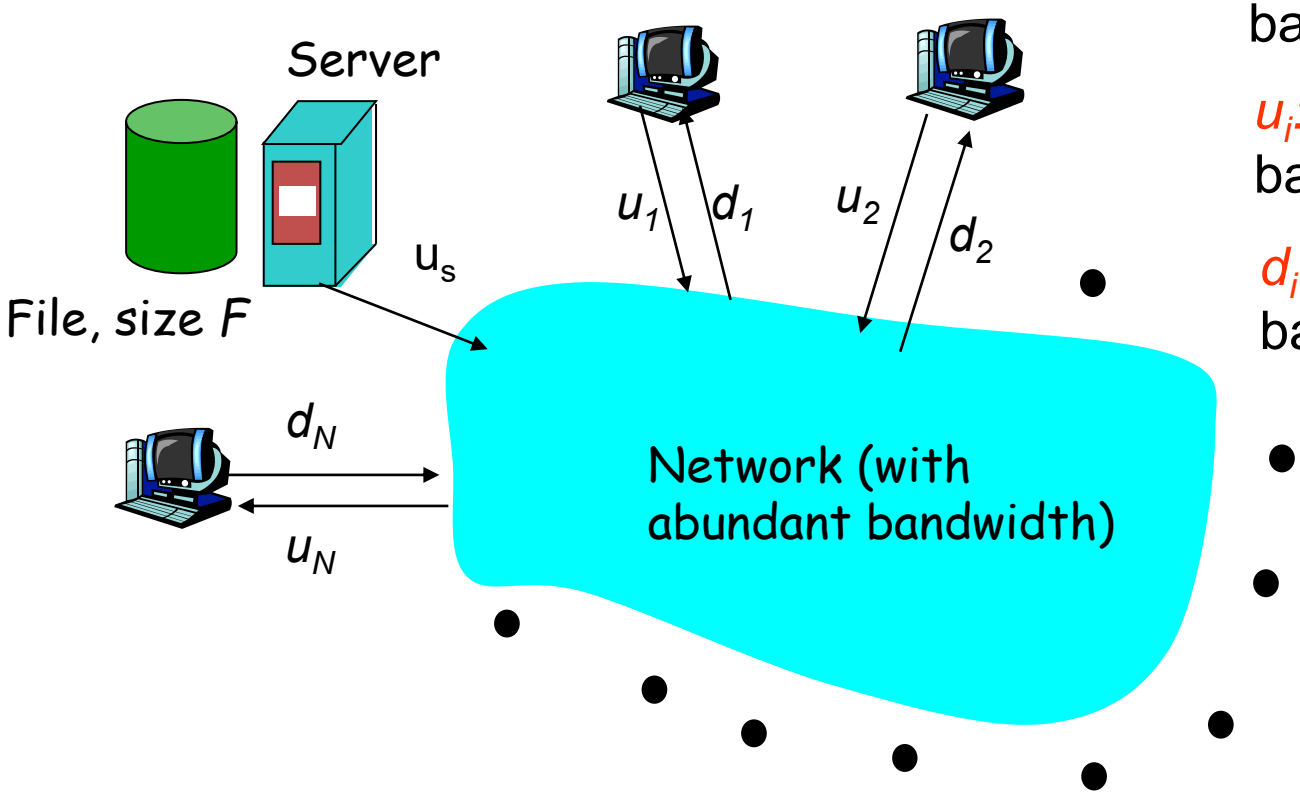
- Typically want solutions that ``scales”
  - Ability of a system, network, or process to handle a growing amount of work effectively
  - Capability to increase its total output under an increased load when resources are added
- Typically want:
  - the costs or resource capacity needed to scale sub-linearly 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*?

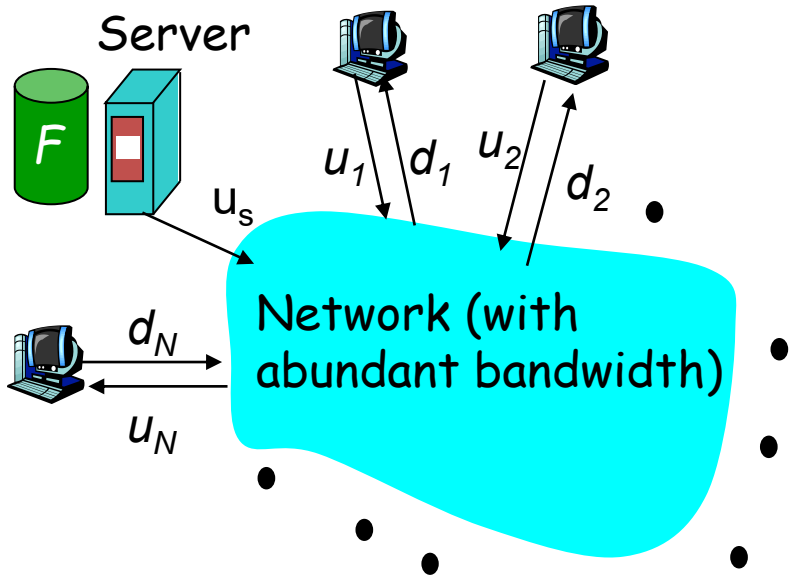


u_s: server upload bandwidth

u_i: peer i upload bandwidth

d_i: peer i download bandwidth

File distribution time: server-client



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

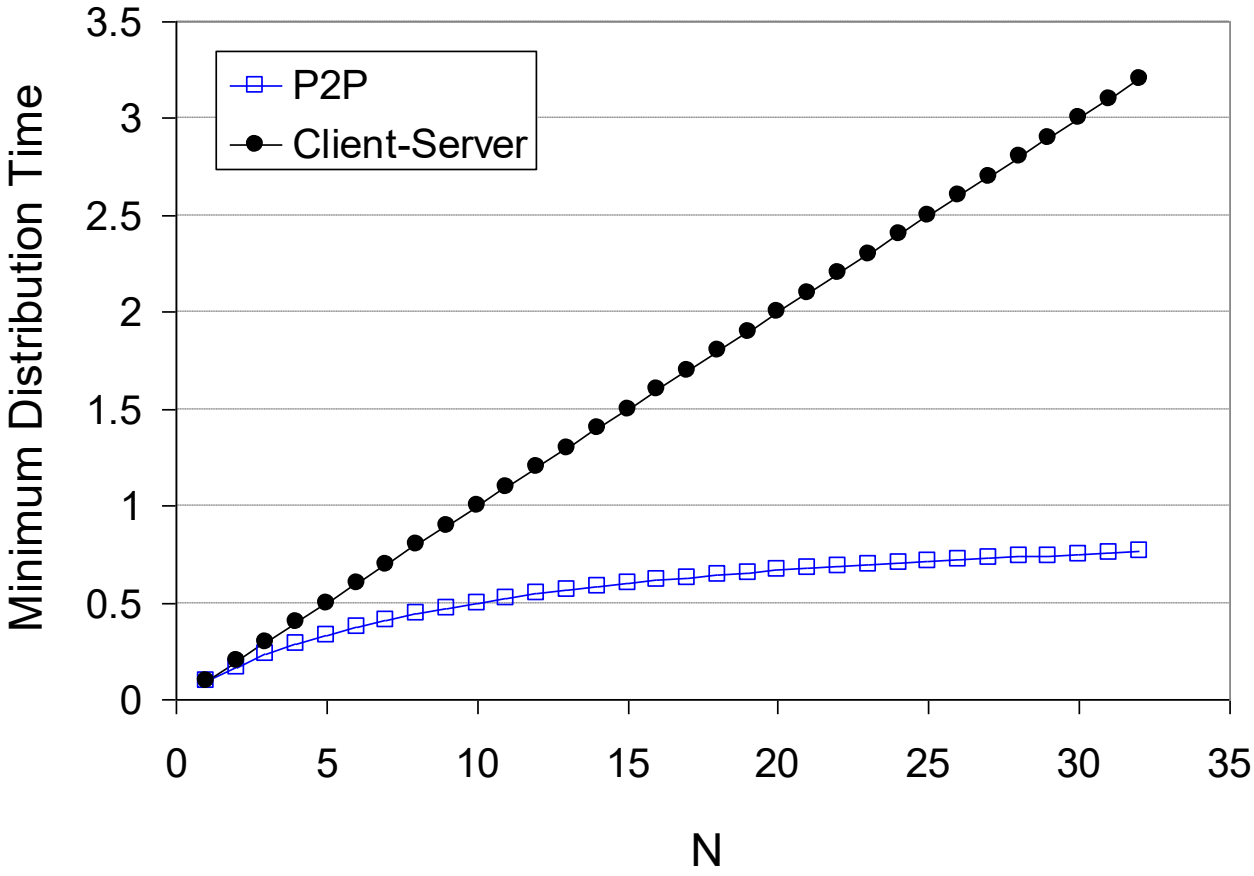
$$= d_{cs} = \max \{ NF/u_s, F/\min(d_i) \}$$

… and using a P2P approach

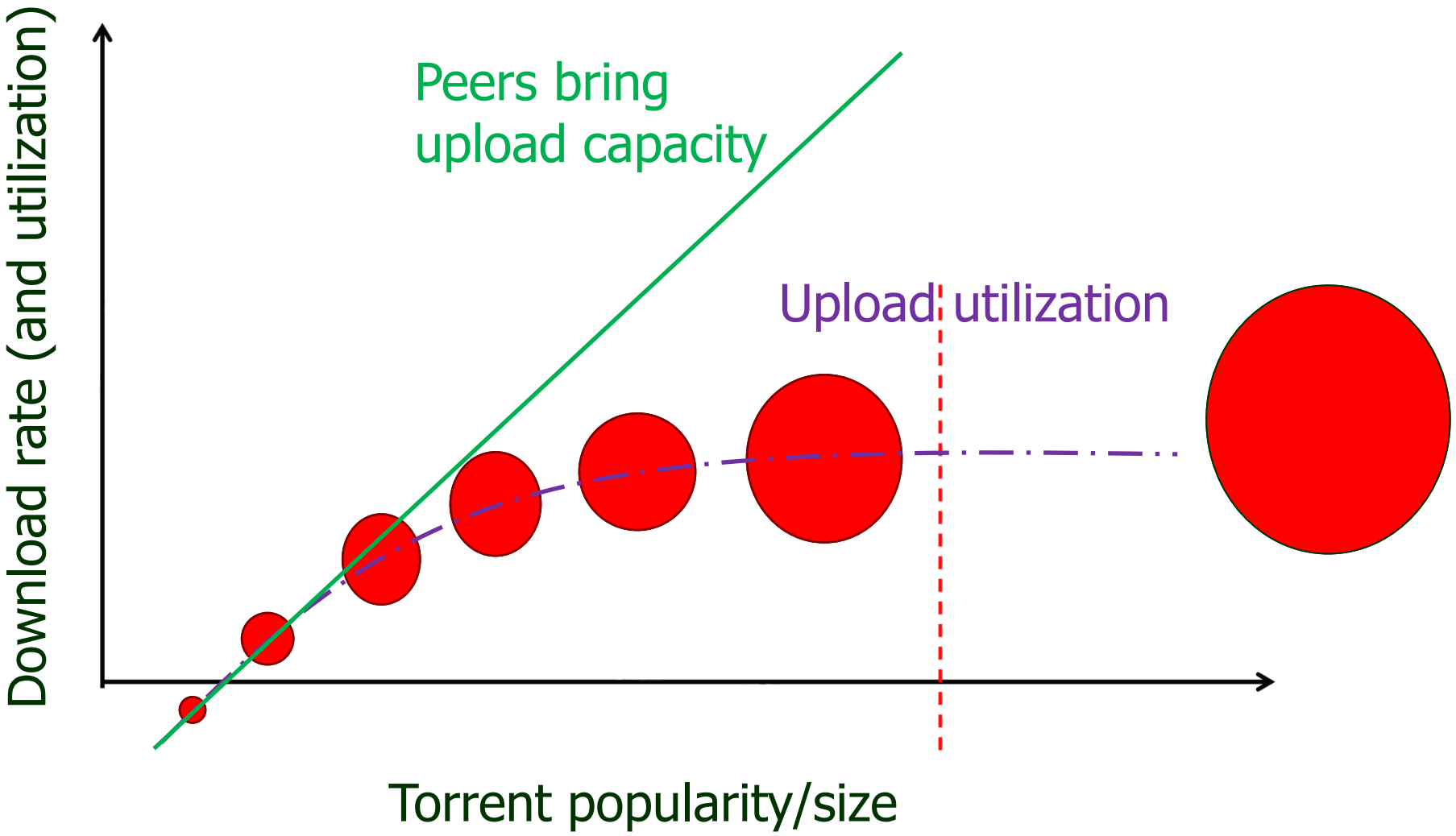
$$d_{P2P} = \max \{ F/u_s, F/\min(d_i) , NF/(u_s + \Sigma u_i) \}$$

Server-client vs. P2P: example

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



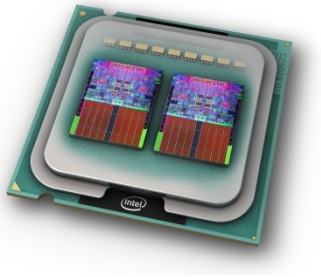
Similarly, BitTorrent upload utilization …



… 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 ...



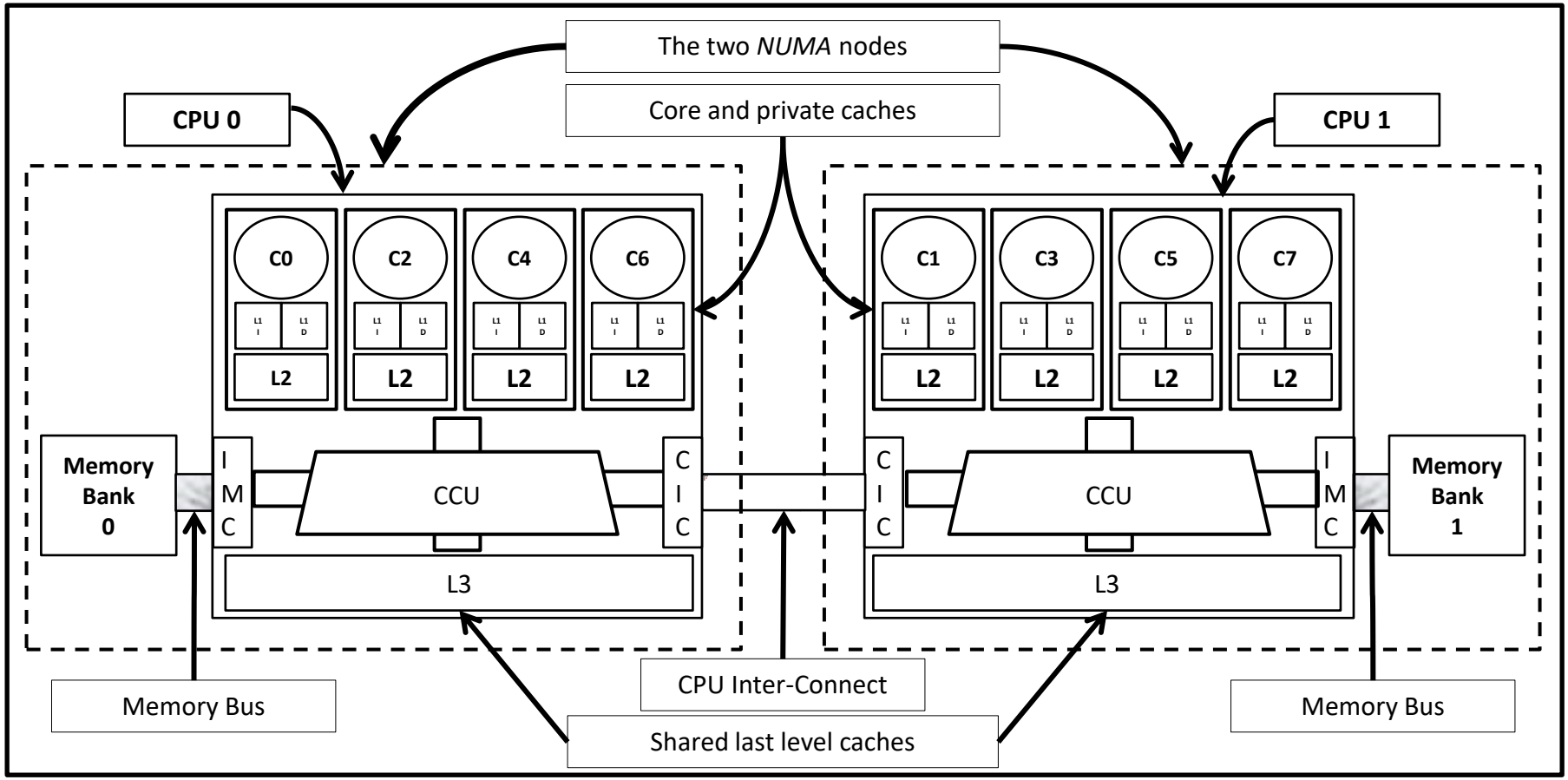
Measurements

- 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

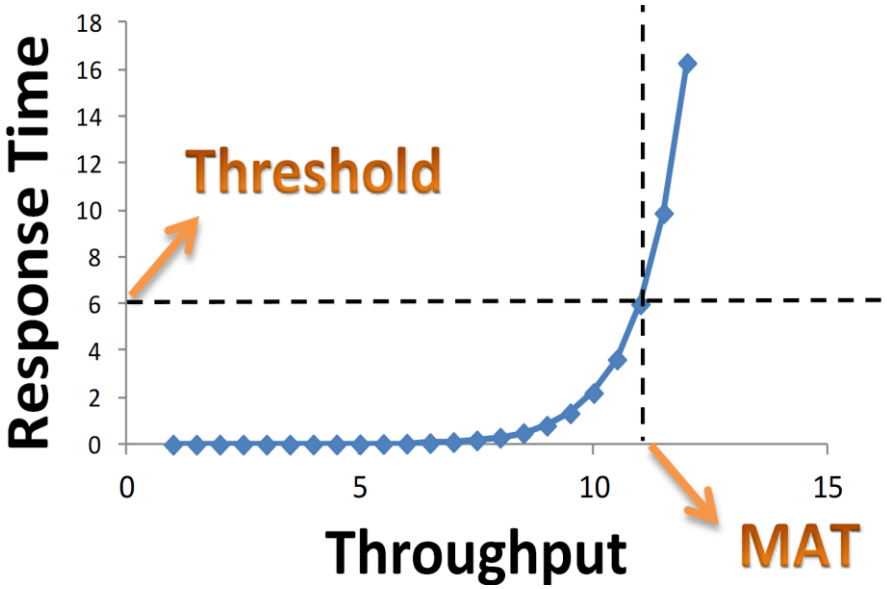
NUMA Architecture

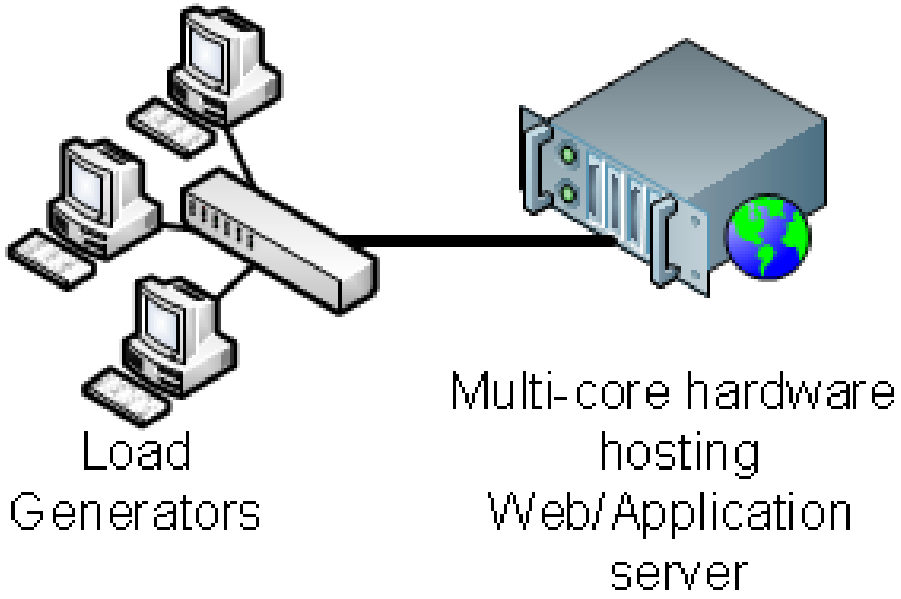
An example of a two processor eight core NUMA system



Scalability Evaluation Measurements

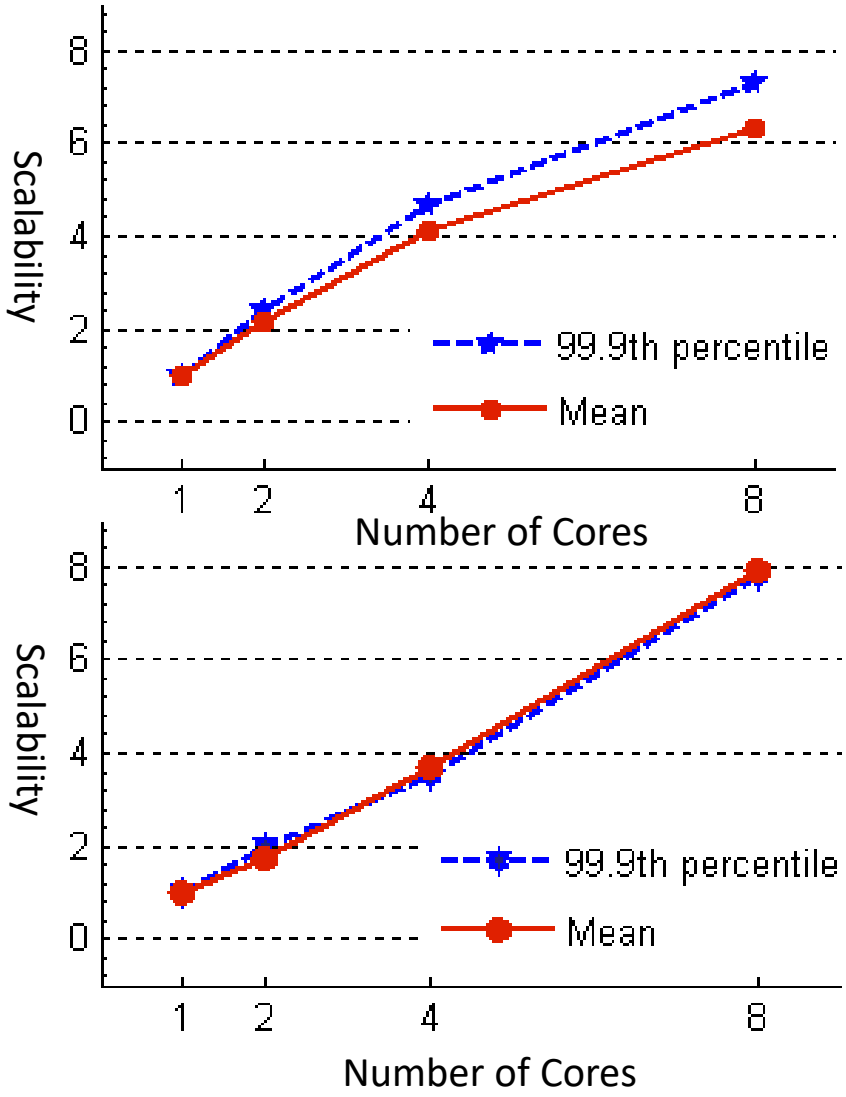
- E.g., Measure Web server scalability for workloads [ICPE '13]
  - Typically want to provide some 99% response time
  - Example scalability measure: Maximum Achievable Throughput (MAT)





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

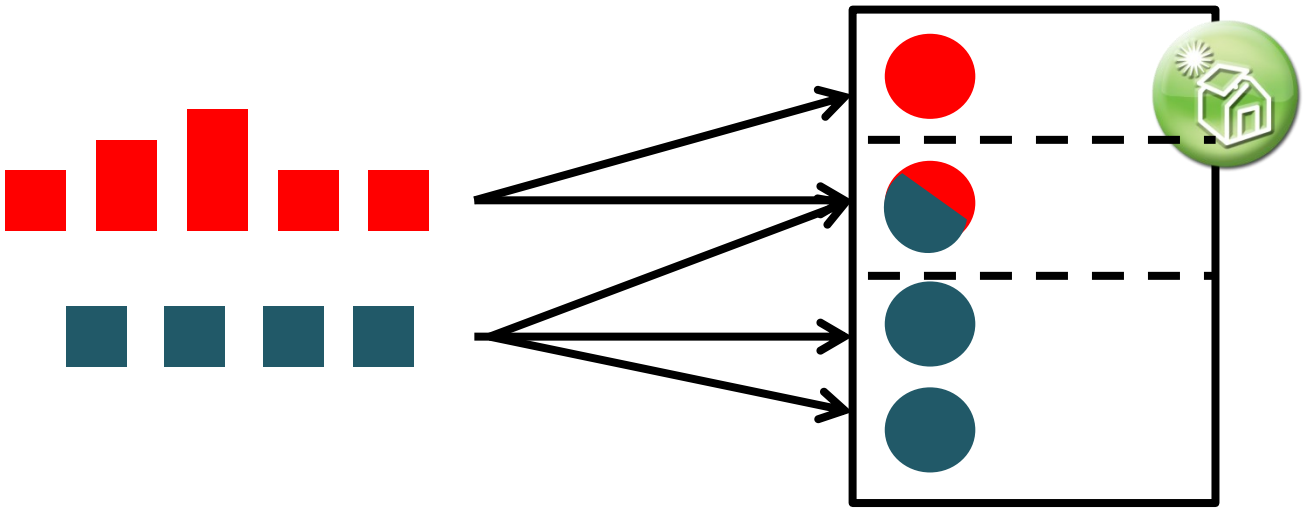


Identification of bottlenecks

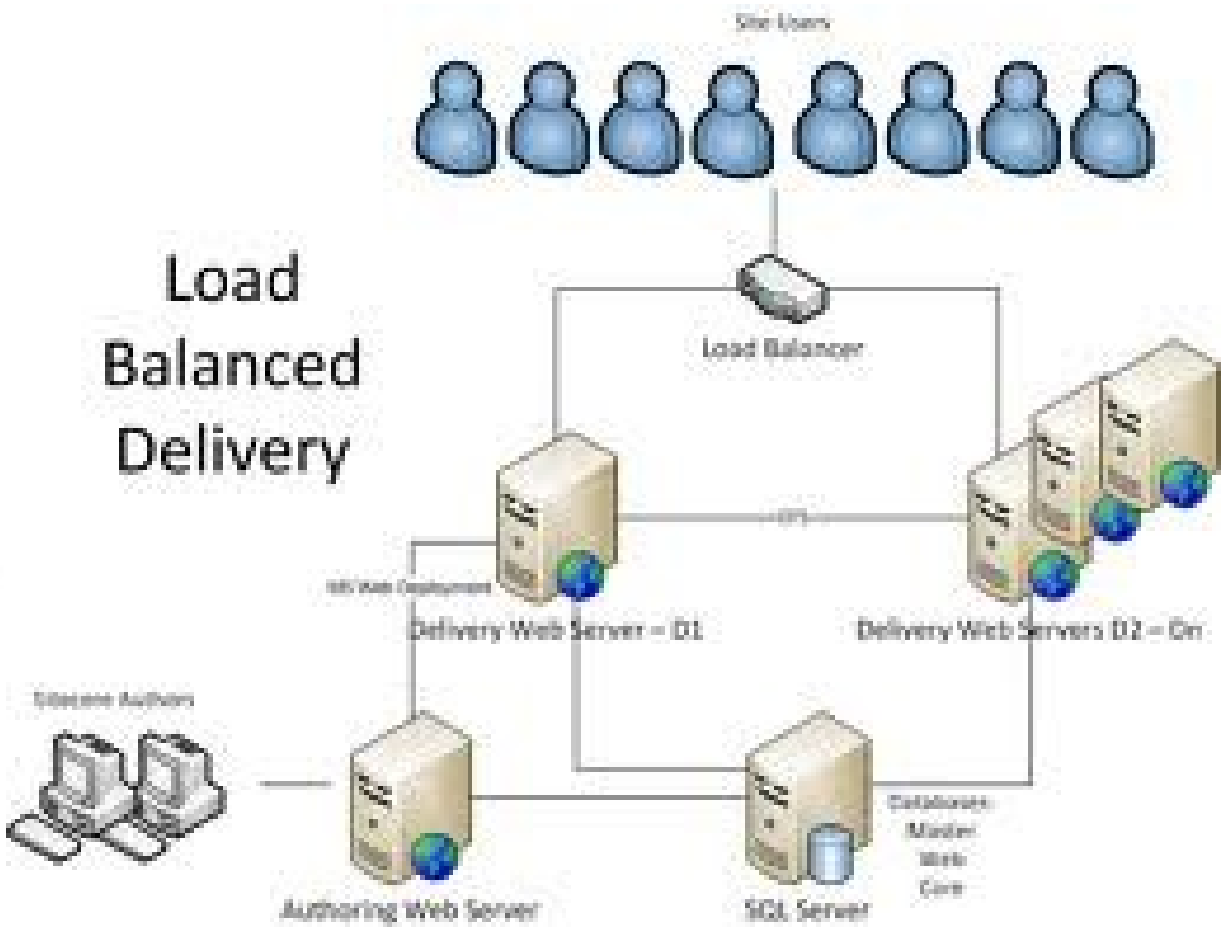
- E.g., memory, CPU, network, cache hierarchy, interconnect bus, scheduler, …
  - Black-box testing
  - Low-level instrumentation

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 …

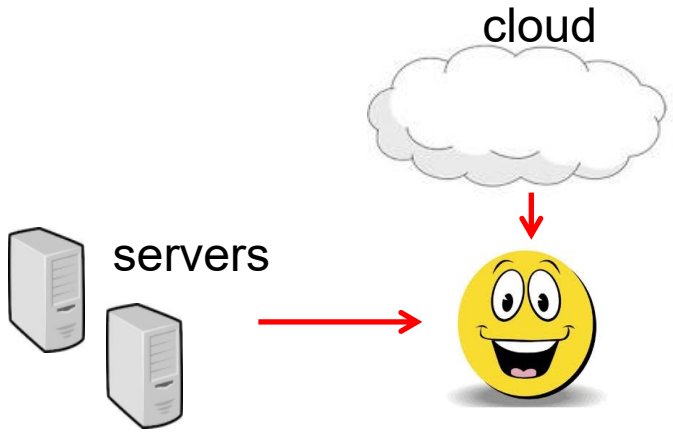
… cost-efficient delivery …

… 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” …]
  - ~ O(1B) hosts in DNS, billions of users (and routers), ~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 on 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 (and vulnerabilities)
  - Attack patterns, port scans, …

Where can we measure the Internet?

How to chose 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 2024 (2 weeks ago, on YouTube soon ...)

Active Measurement Techniques

Active Probes

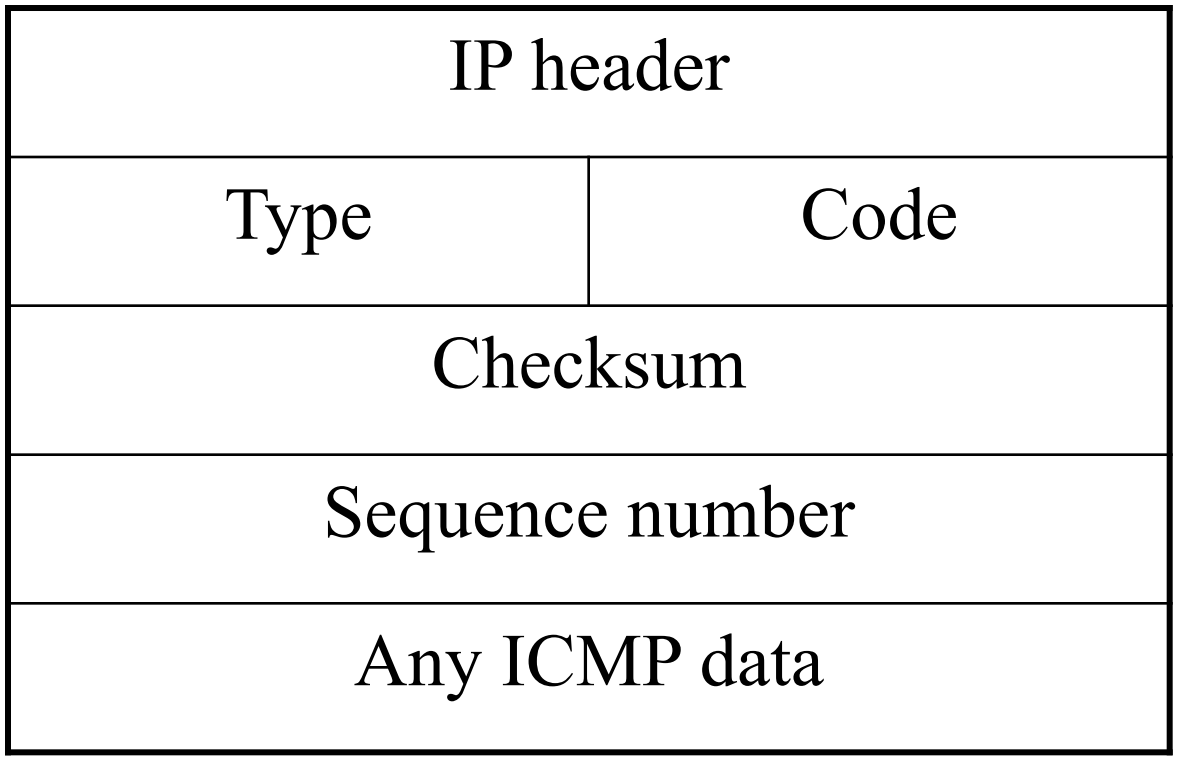
- Active probes send stimulus (packets) into the network and then measure the response
  - Done on network, transport and application layers

- Active probes are useful to measure various things:
  - Delay, delay jitter, and loss
  - Topology and routing behavior
  - Capacity, bandwidth, and throughput

Example: RTT

ICMP

ICMP is the IP error diagnosis protocol.



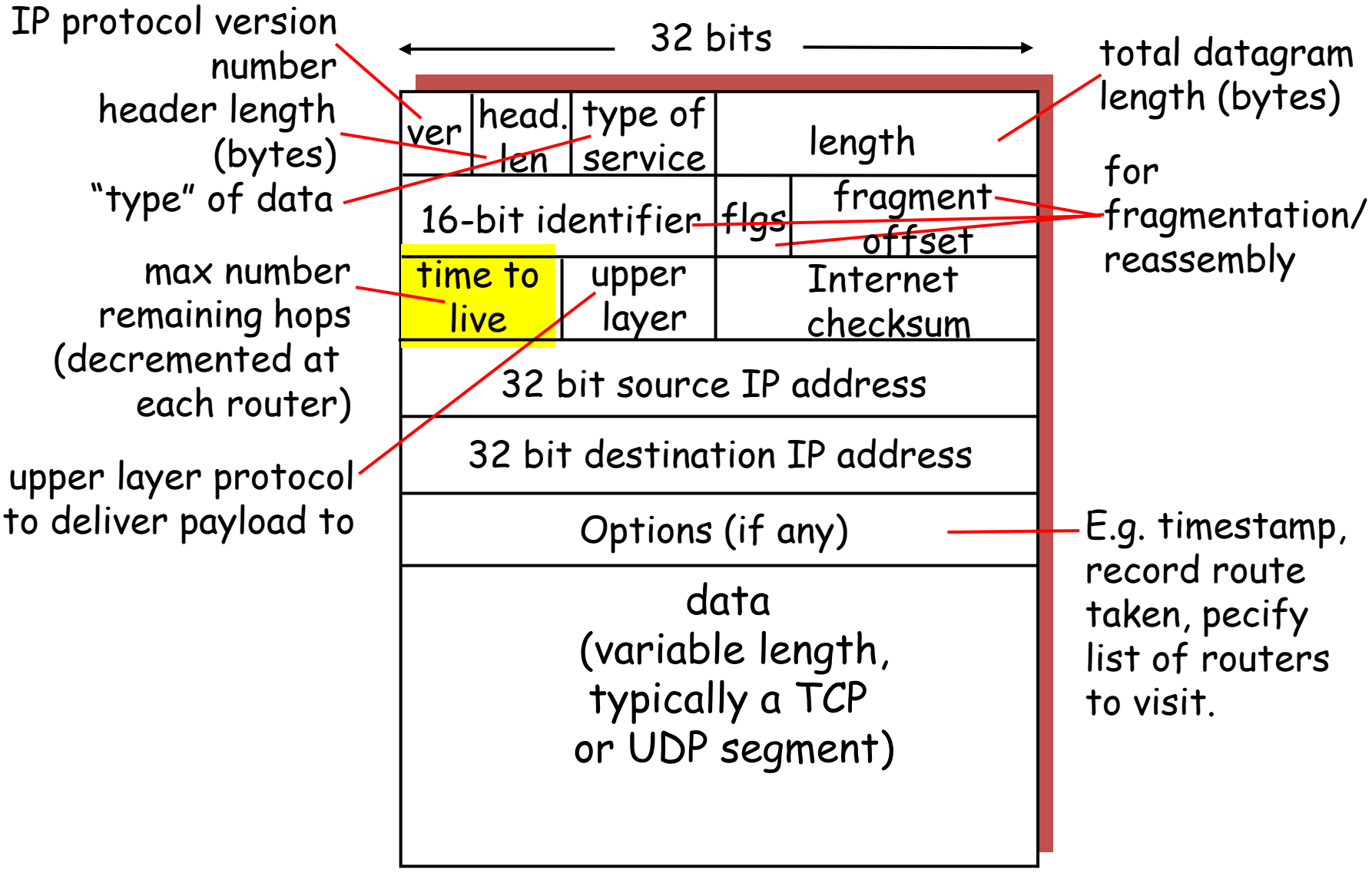
ICMP Message Types	
Type No.	Meaning
0	Echo reply
3	Destination unreachable
4	Source quench
5	Redirect
8	Echo
9	Router advertisement
10	Router solicitation
11	Time exceeded
12	Parameter problem
13	Timestamp
14	Timestamp reply
15	Information requeste
16	Information reply

Application layer “ping”

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

Example: Path

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

Type	Code	description
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

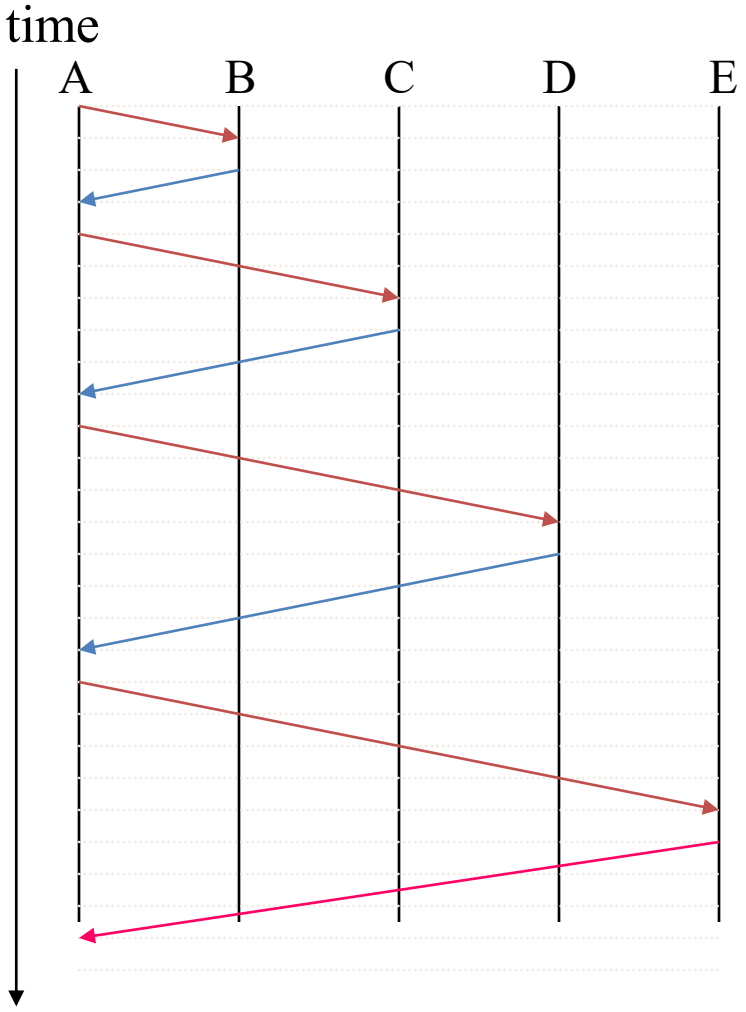
traceroute

Regular UDP packets

- successive TTLs

ICMP “TTL expired” message

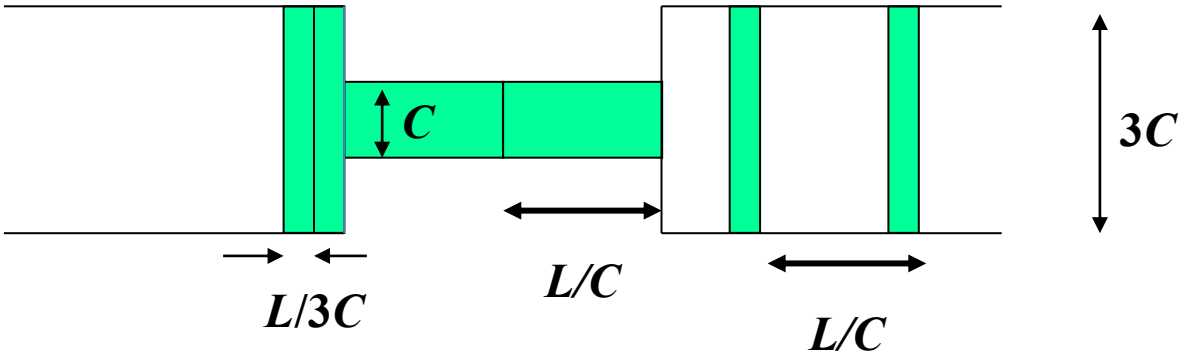
ICMP “port unreachable” message



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

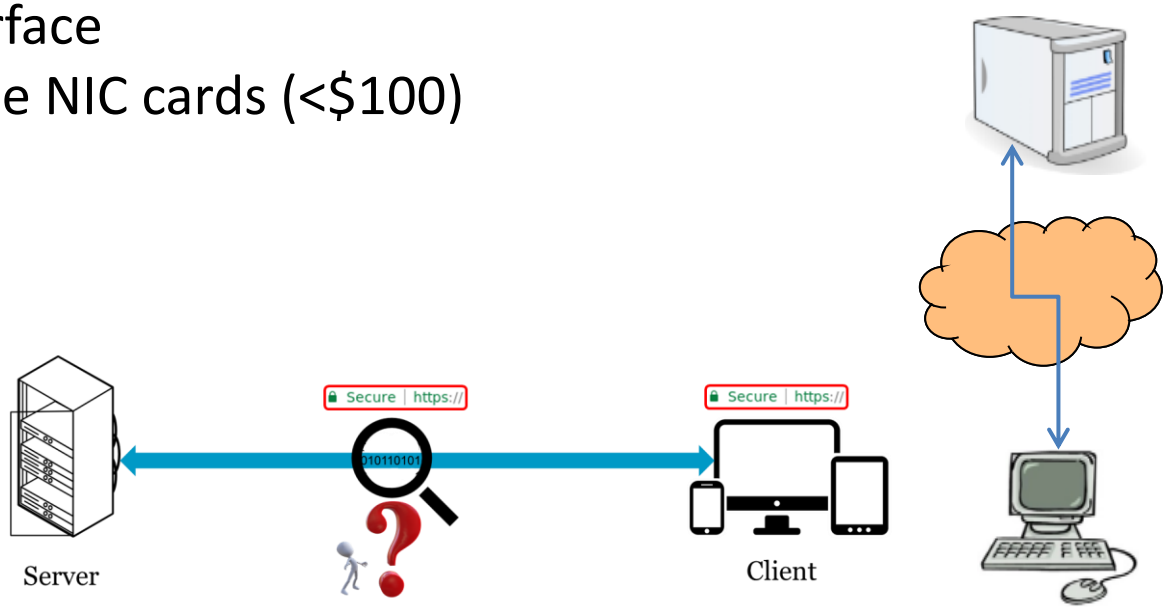


- But cross-traffic 'noise' can effect Δ.
- E.g., patchar "allows any user to find (estimate) the bandwidth, delay, average queue and loss rate of every hop between any source & destination on the Internet"

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

```
[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

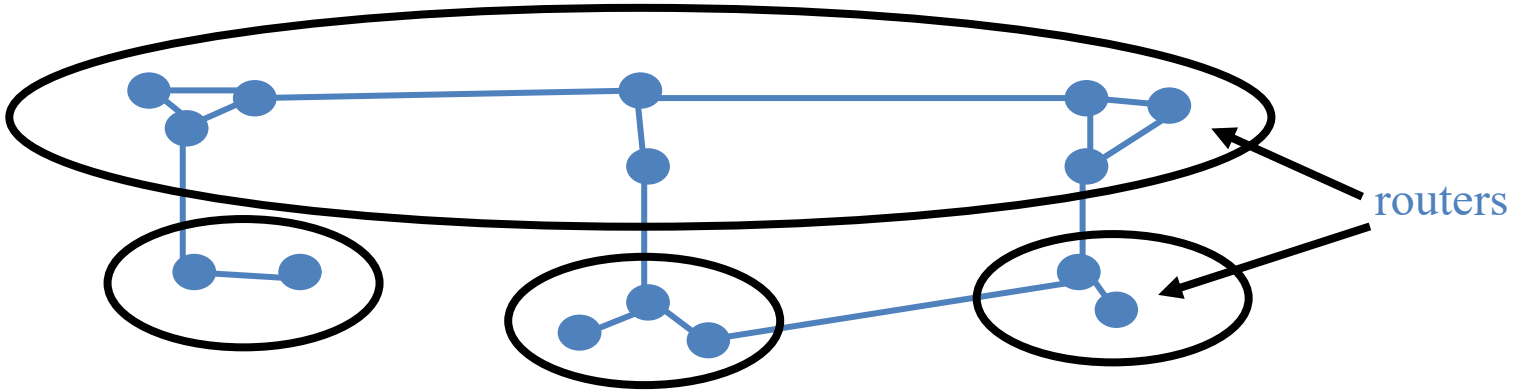
- Zeek (formerly Bro)
  - Open-source network security monitoring tool that allows easy extraction of information from the network traffic
  - Flexible and powerful when wanting to extract information from the various network layers
  - Typically use scripts to create logs
- Wireshark (used in labs)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

Measuring the Internet's topology

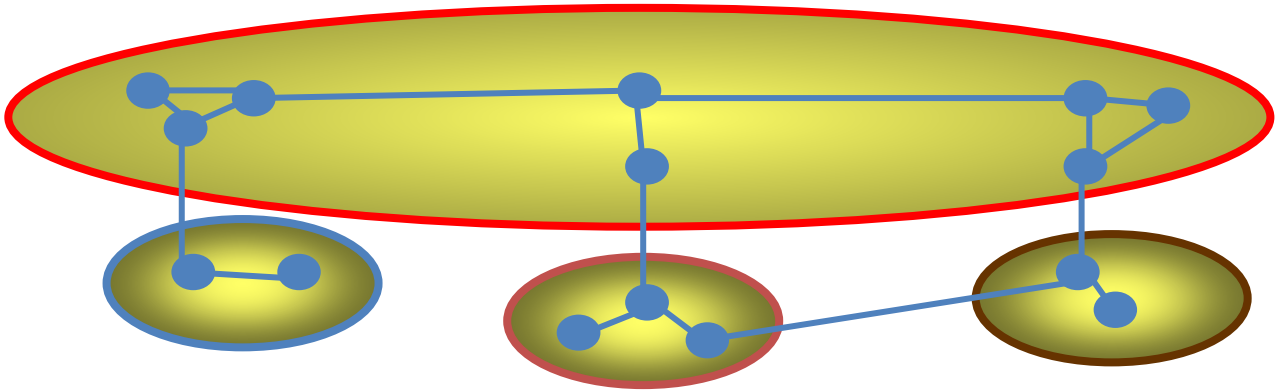
Outline

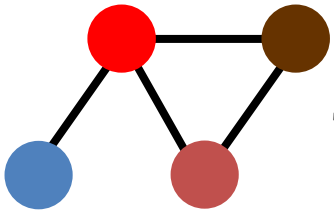
- Background
- Then, both active and passive examples ...

The Internet Structure



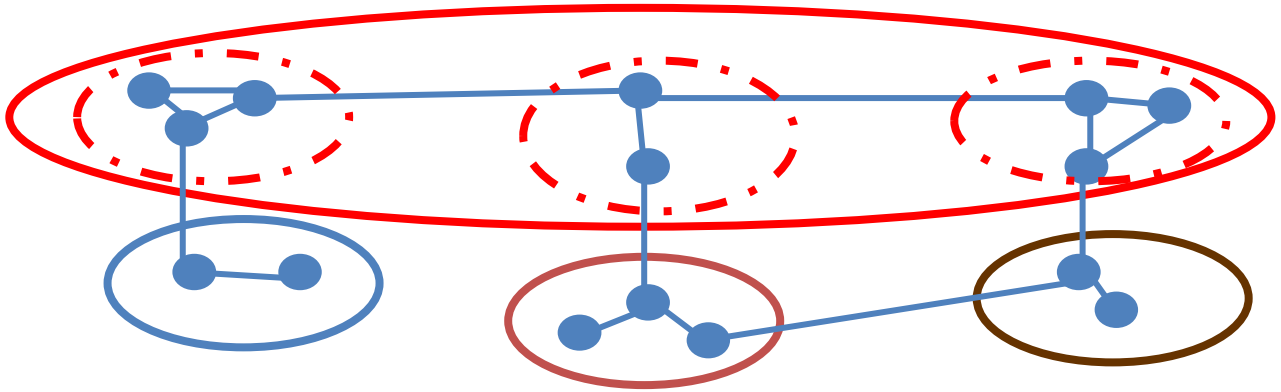
The Internet Structure

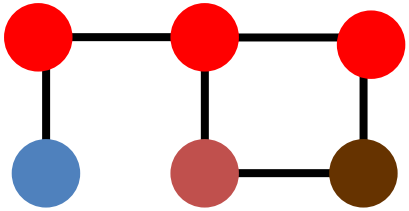


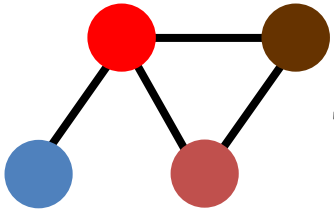


The **AS** graph

The Internet Structure





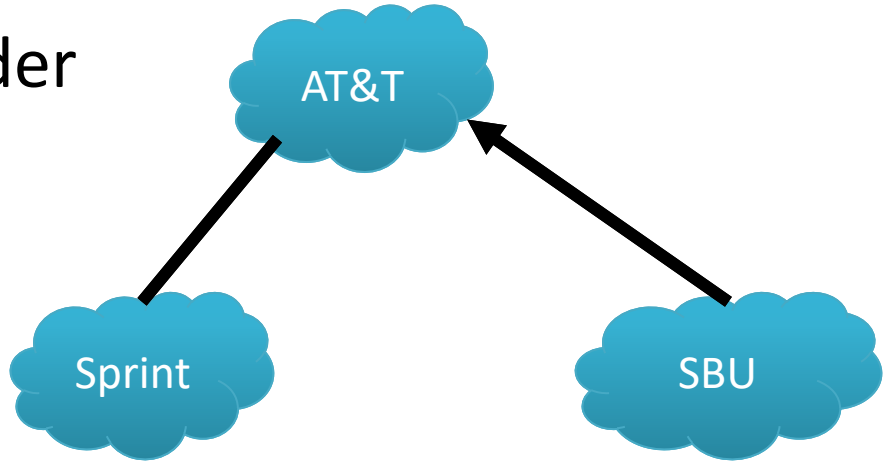


The **AS** graph

The **PoP level** graph

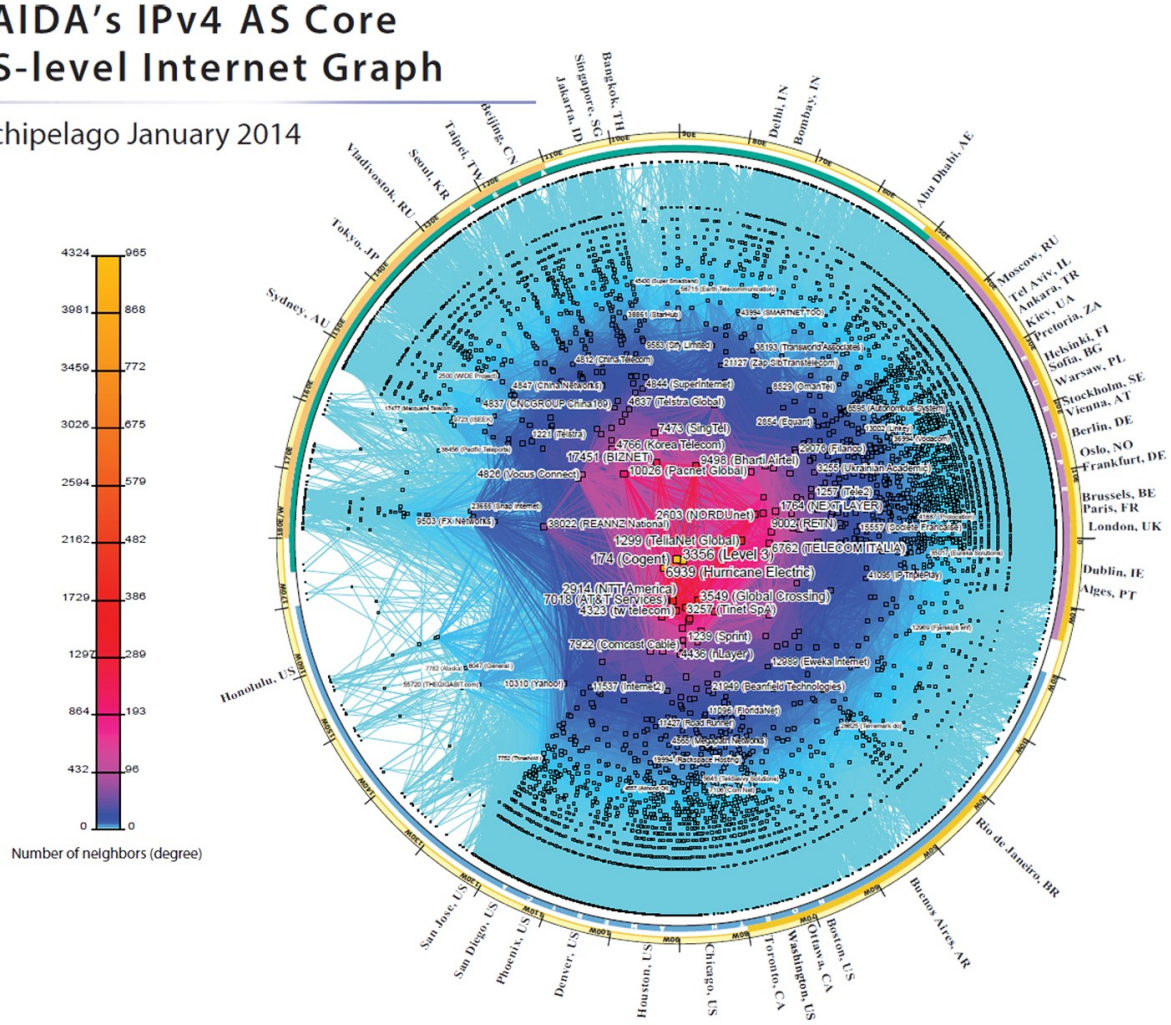
Measuring the Internet's topology

- 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 ….

15412	12041	p2c
15412	12486	p2c
15412	12880	p2c
15412	13810	p2c
15412	15802	p2c
15412	17408	p2c
15412	17554	p2c
15412	17709	p2c
15412	18101	p2c
15412	19806	p2c
15412	19809	p2c
15413…		

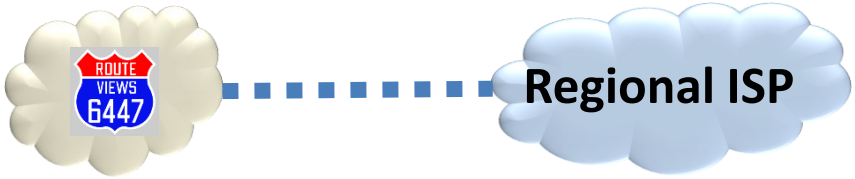


So how do we measure this graph?

- 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

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

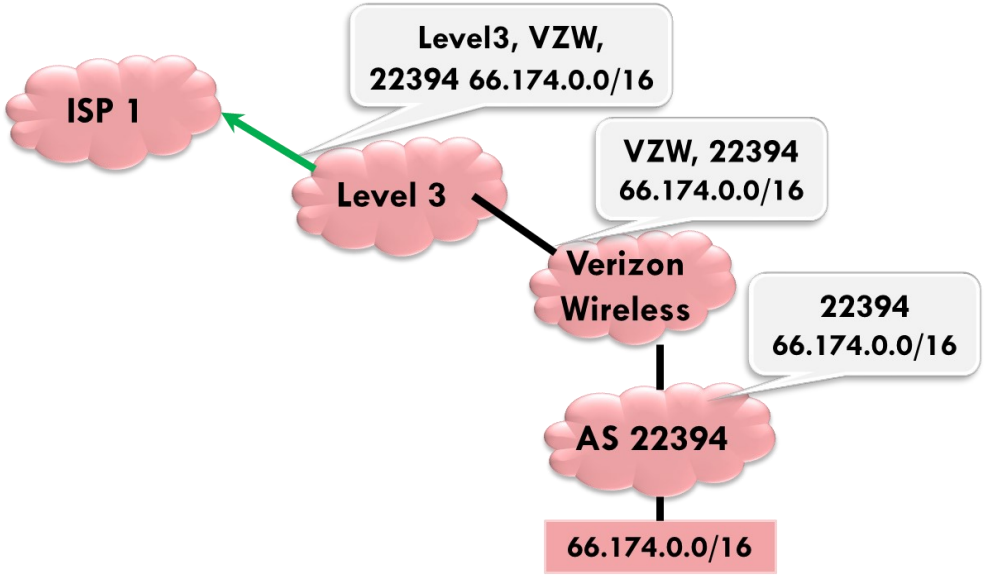
AT&T (AS7018) it telling Routeviews (AS 6447) about this route.

This /20 prefix can be reached via the above path

Going from BGP Updates to a Topology

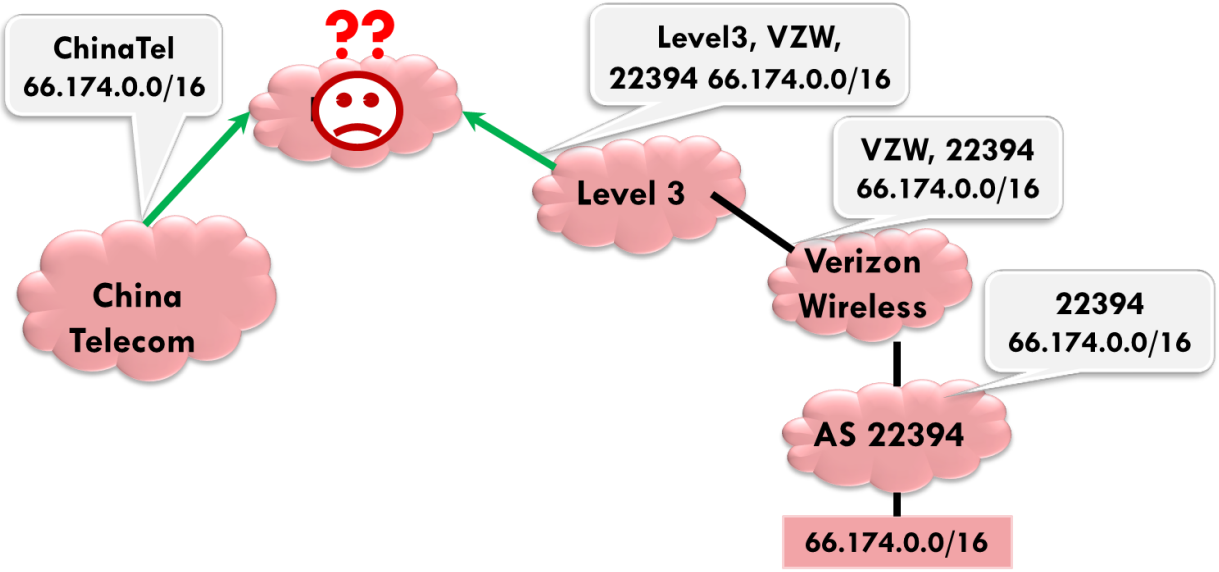
- Key idea
  - The business relationships determine the routing policies
  - The routing policies determine the paths that are chosen
  - So, look at the chosen paths and infer the policies
- Example: AS path “7018 4134 9318” implies
  - AS 4134 allows AS 7018 to reach AS 9318
  - China Telecom allows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

Traceroute vs Announced Path



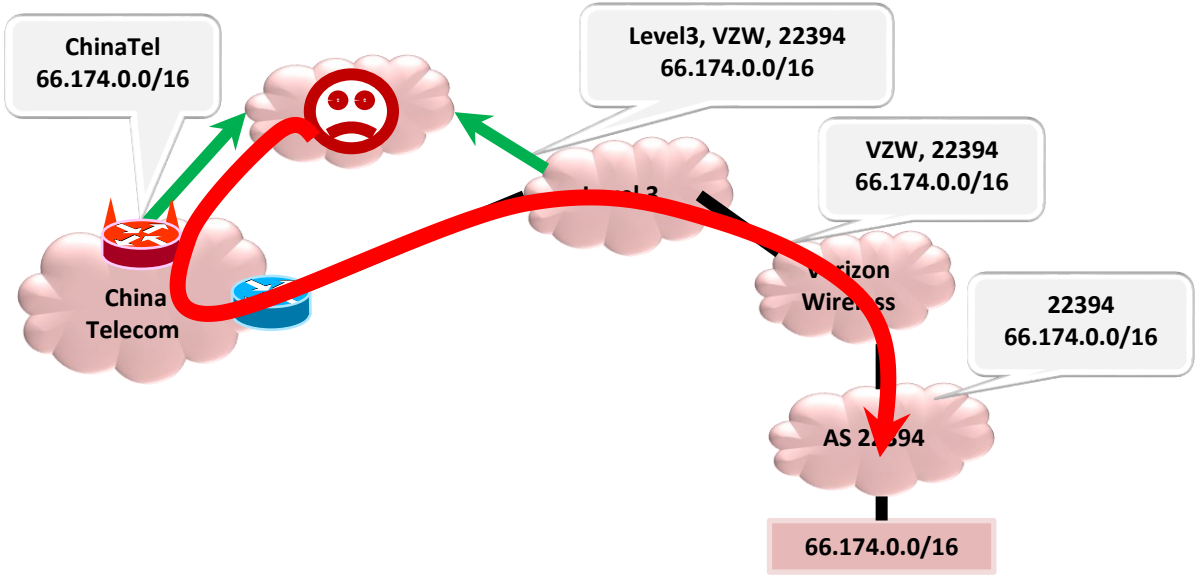
Characterizing Large-scale Routing Anomalies: A Case Study of the China Telecom Incident, R. Hiran et al., Proc. PAM 2013.

Traceroute vs Announced Path



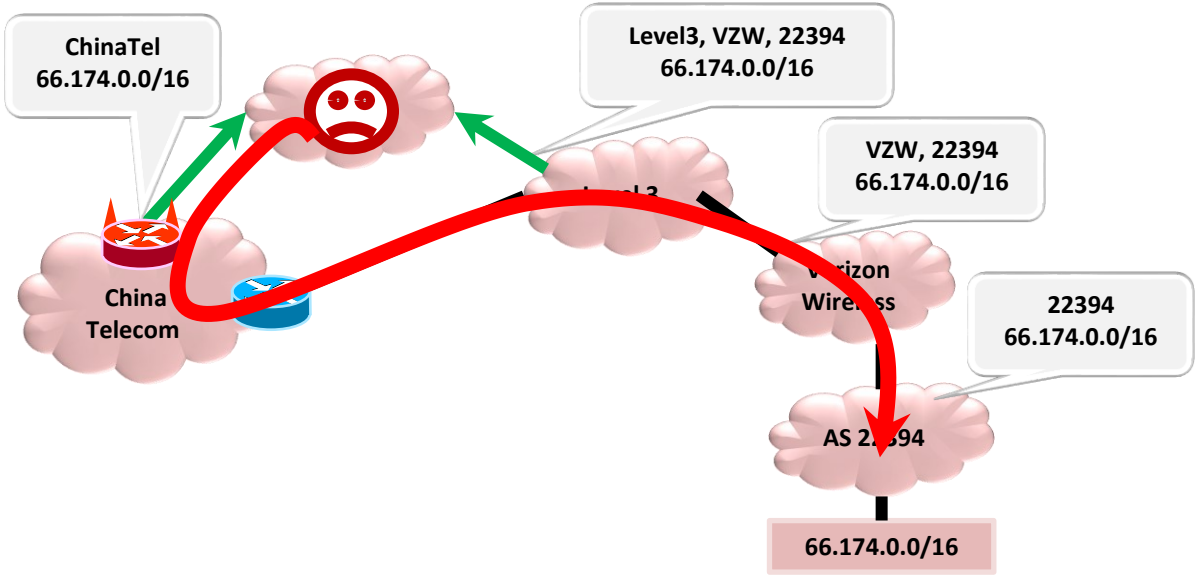
Characterizing Large-scale Routing Anomalies: A Case Study of the China Telecom Incident, R. Hiran et al., Proc. PAM 2013.

Traceroute vs Announced Path



Characterizing Large-scale Routing Anomalies: A Case Study of the China Telecom Incident, R. Hiran et al., Proc. PAM 2013.

Traceroute vs Announced Path



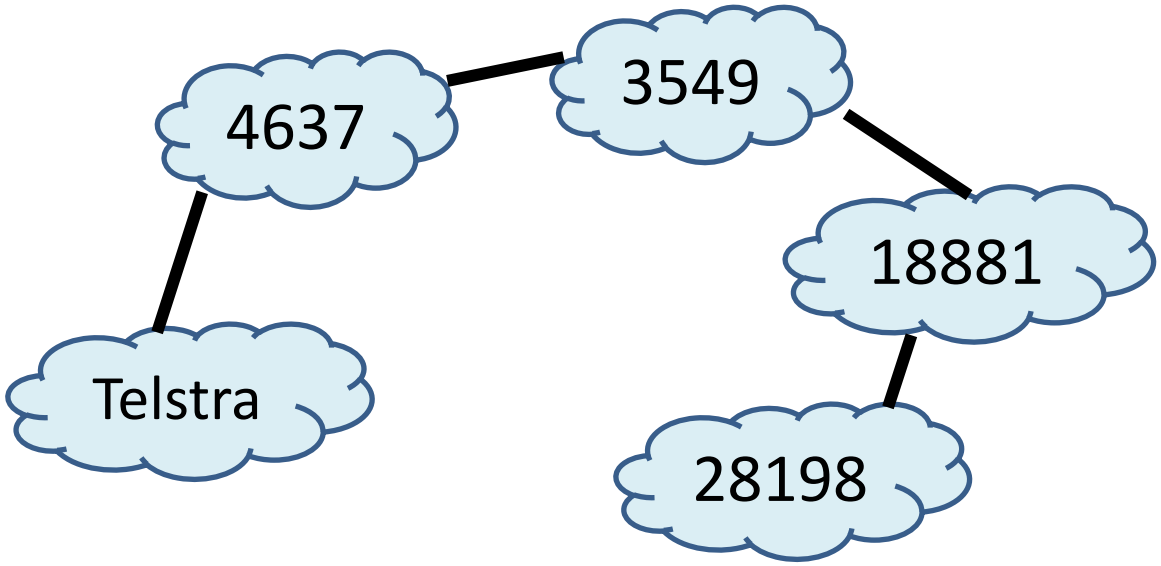
Interception typically results in differences between

- Announced AS-PATH
- Data path (traffic)

Policy checks if legit reason(s)

Characterizing Large-scale Routing Anomalies: A Case Study of the China Telecom Incident, R. Hiran et al., Proc. PAM 2013.

Traceroute vs Announced Path

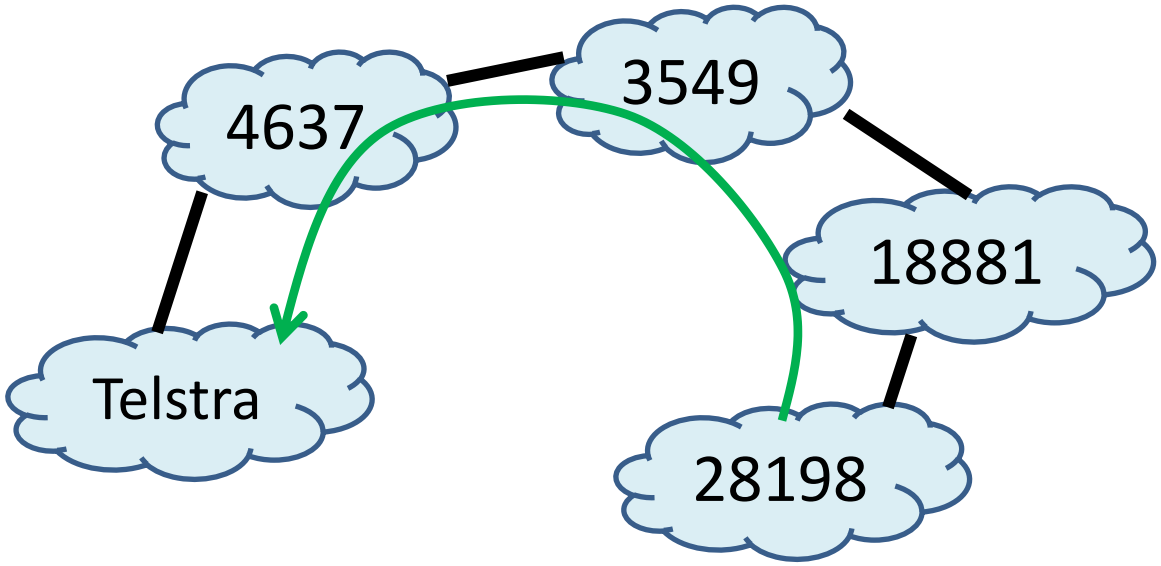


Sometimes differences

- Announced AS-PATH
- Data path (traffic)

Many legit reason(s)

Traceroute vs Announced Path



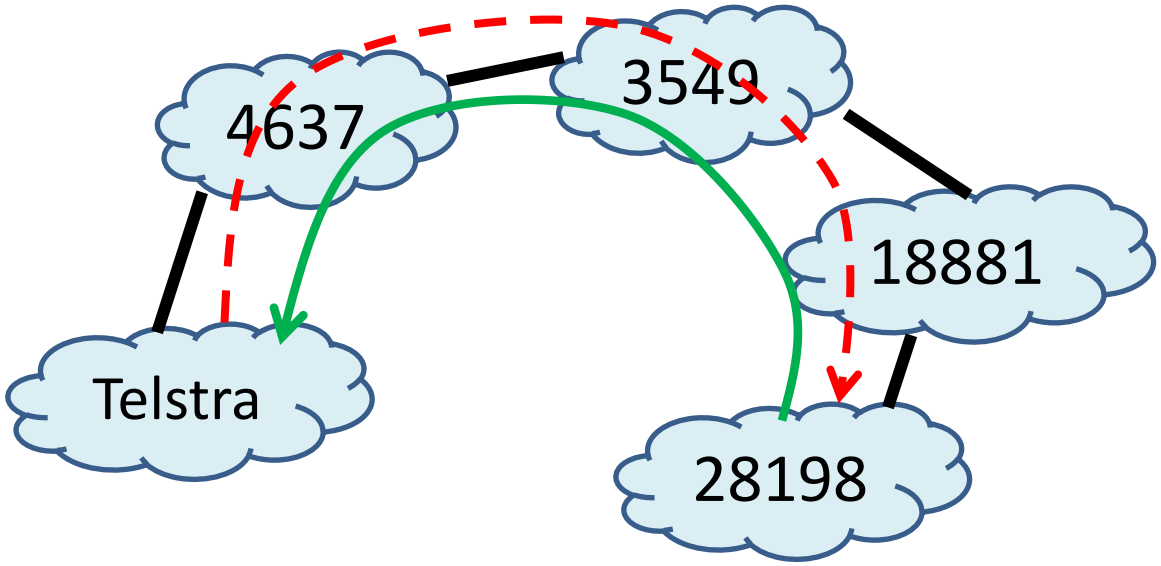
Sometimes differences

- 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



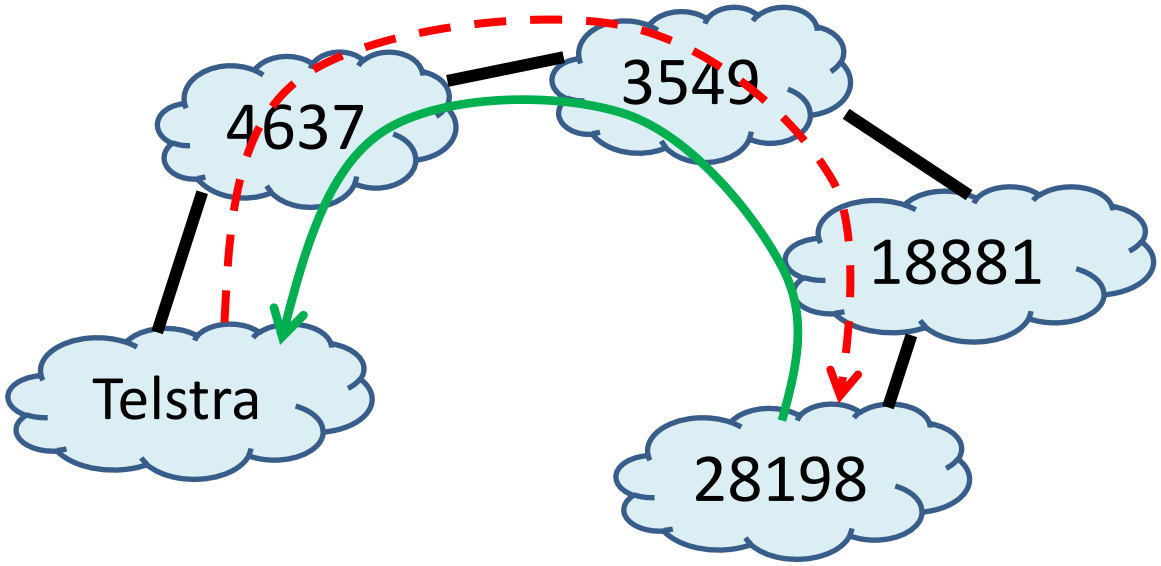
Sometimes differences

- 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)

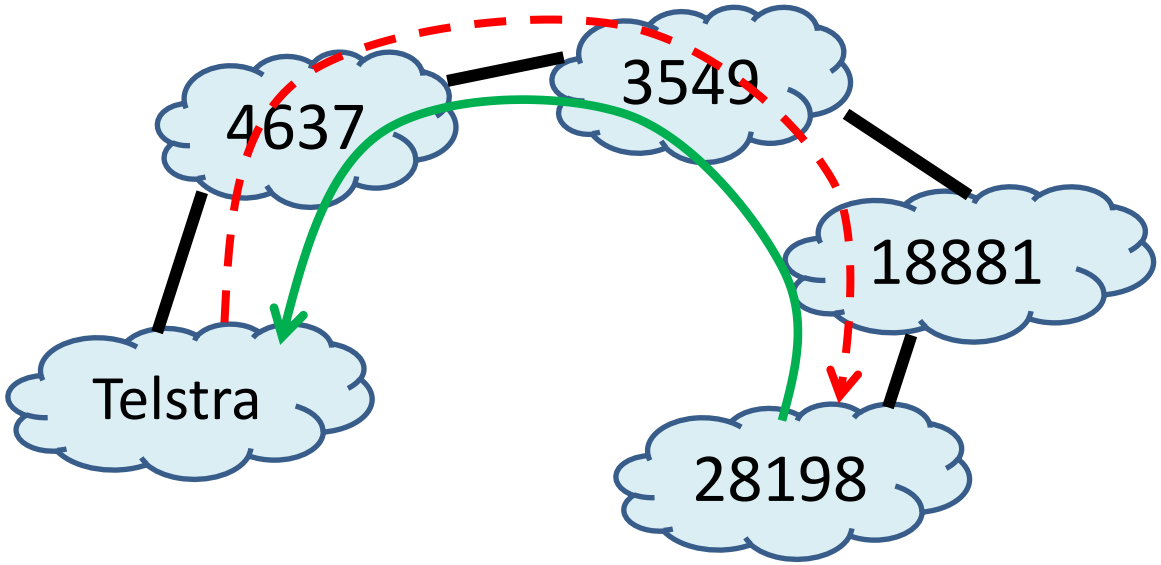
Many legit reason(s)

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

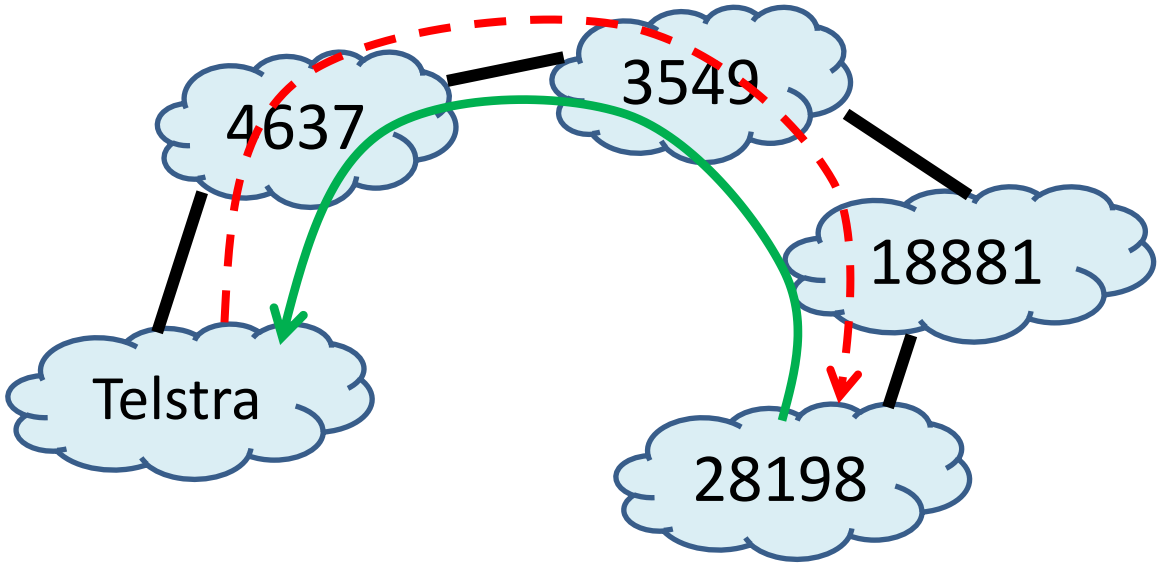
- Announced AS-PATH
- Data path (traffic)

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)

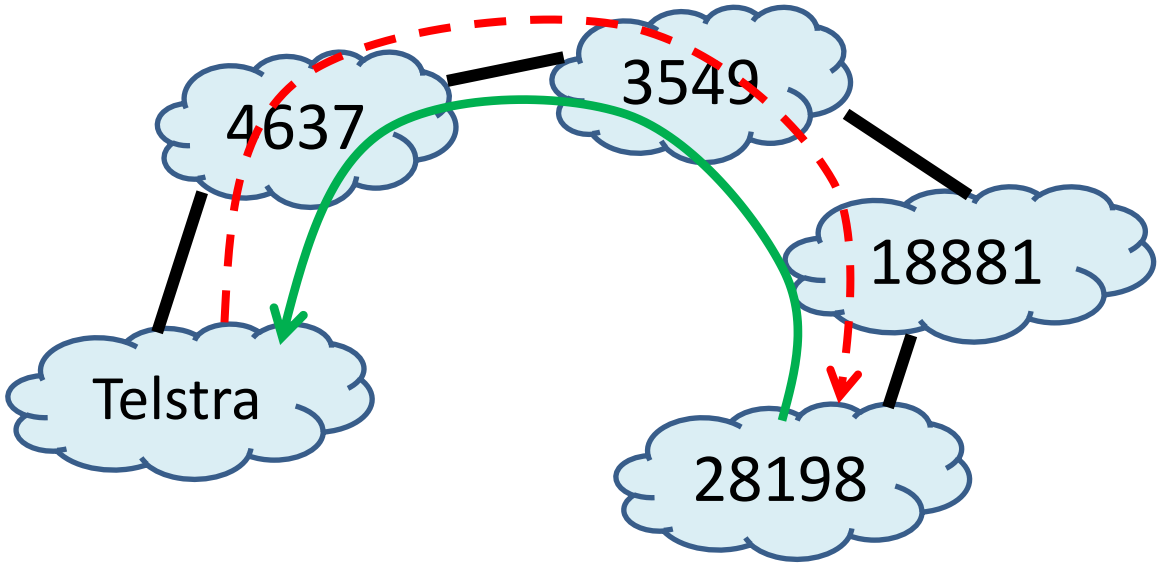
Many legit reason(s)

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