Information “Superhighway”?

More like Super-Cowtrak

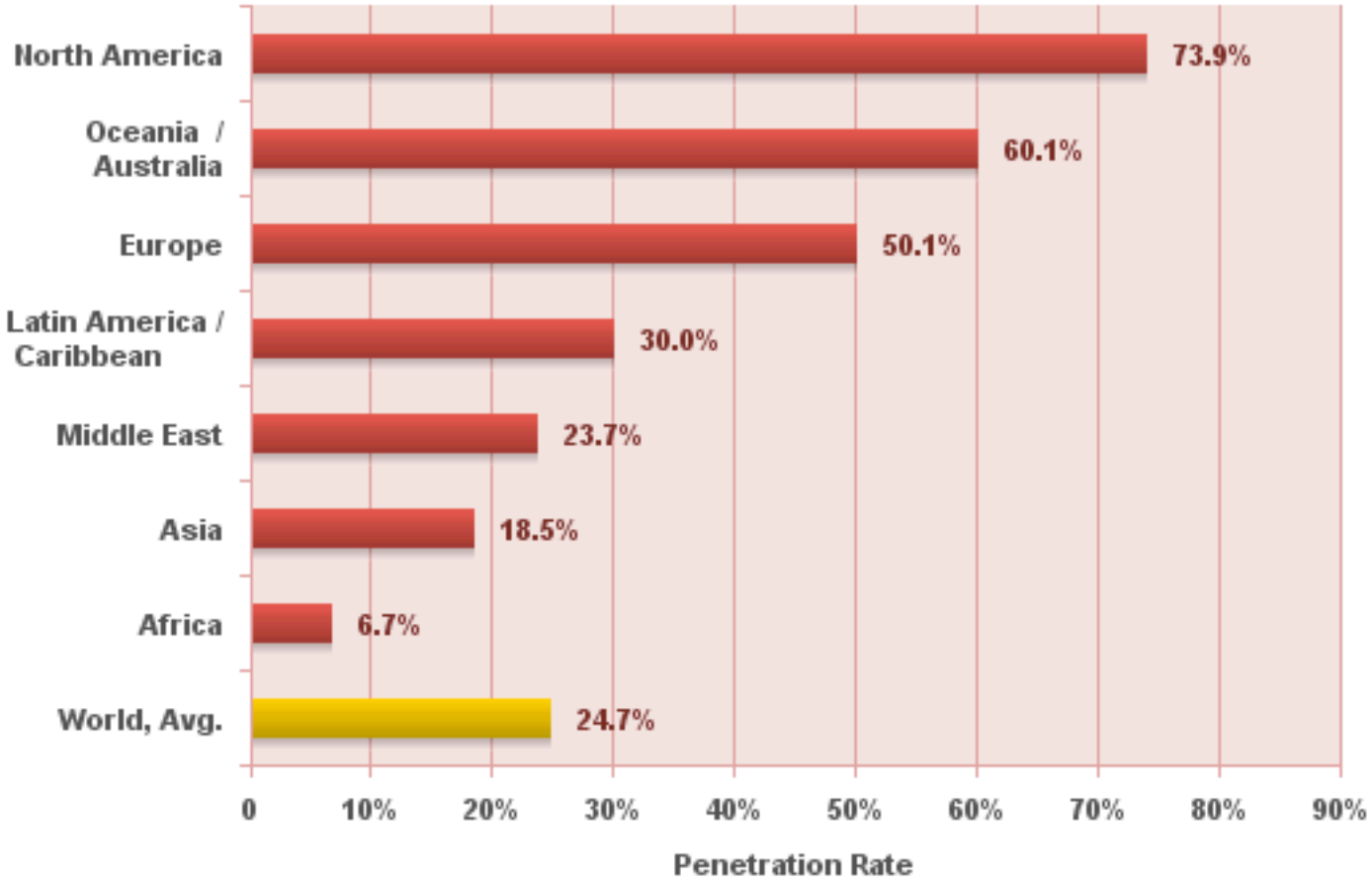
AUSNOG-2009

Access and Speed

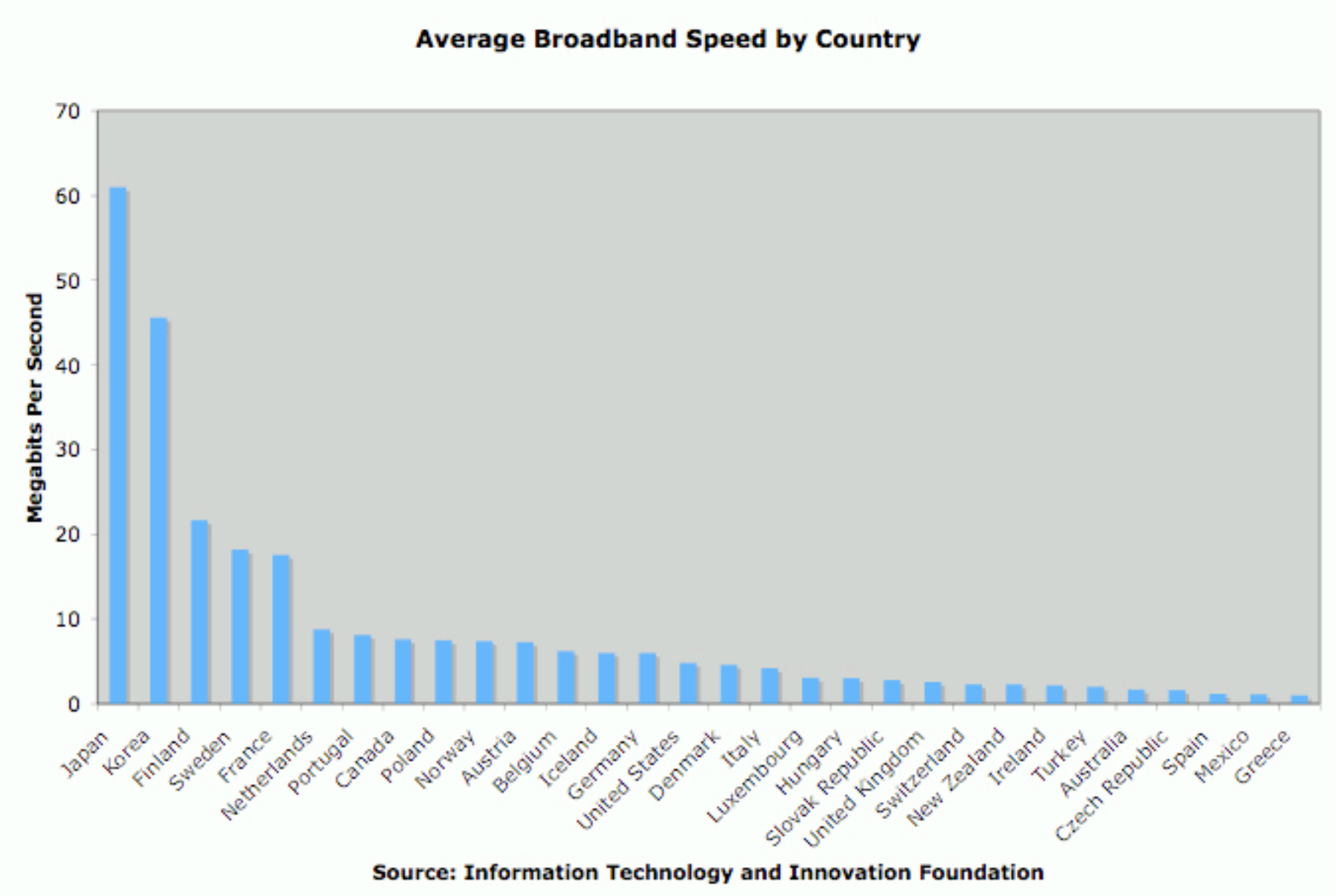
- Millions and Millions Served
- More Ubiquitous
- Faster

- Better?

World Internet Penetration Rates by Geographic Regions



Source: Internet World Stats - www.internetworldststs.com/stats.htm
Penetration Rates are based on a world population of 6,767,805,208 and 1,668,870,408 estimated Internet users for June 30, 2009.


Average Broadband Speed by Country
Megabits Per Second
70
60
50
40
30
20
10
0
Japan
Korea
Finland
Sweden
France
Netherlands
Portugal
Canada
Poland
Norway
Austria
Belgium
Iceland
Germany
United States
Denmark
Italy
Luxembourg
Hungary
Slovak Republic
United Kingdom
Switzerland
New Zealand
Ireland
Turkey
Australia
Czech Republic
Spain
Mexico
Greece
Source: Information Technology and Innovation Foundation

- “If my network is so fast, how come my ftp is so slow?” - Dykstra

- “the plural of anecdote is data” - Wolfinger

What is performance?

- “Performance” might mean …
  - Elapsed time for file transfers
  - Packet loss over a period of time
  - Percentage of data needing retransmission
  - Drop outs in video or audio

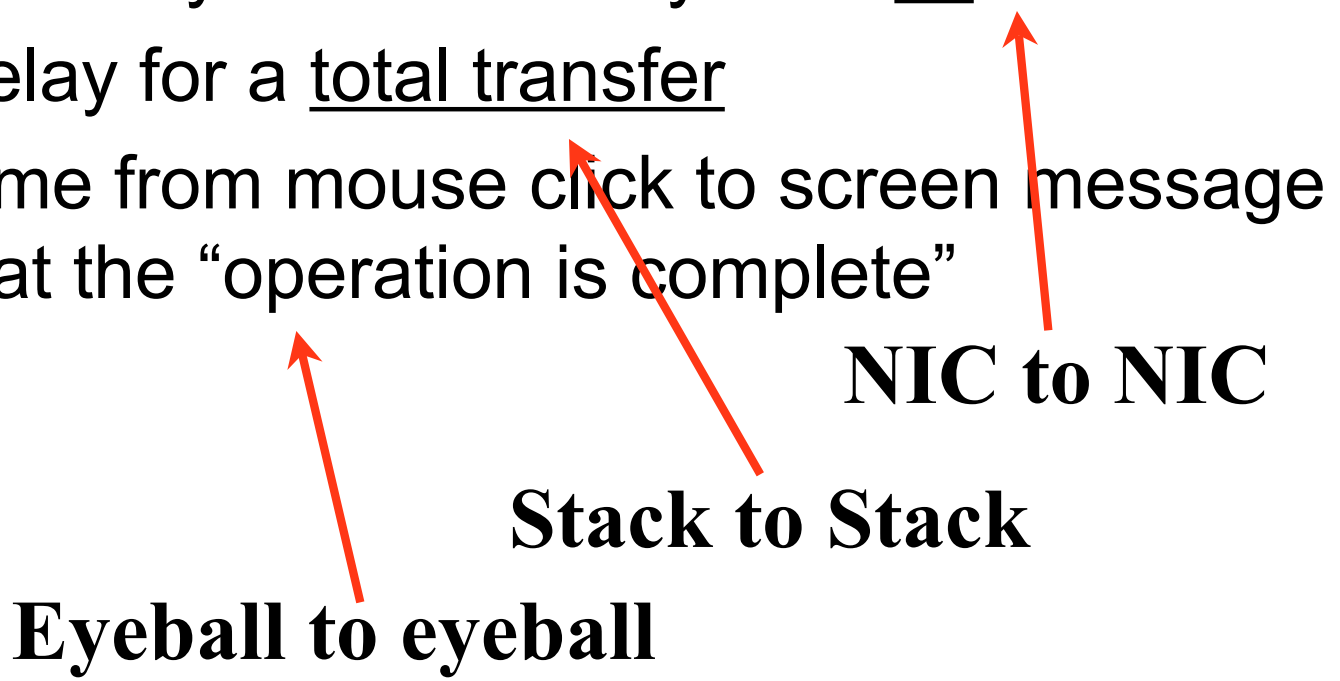
  - Subjective “feeling” that feedback is “on time”

Throughput

- Throughput is the amount of data that arrives per unit time.
- “Goodput” is the amount of data that arrives per unit time, minus the amount of that data that was retransmitted.

Delay

- Delay is a time measurement for data transfer
  - One way network delay for a bit in transit
  - Delay for a total transfer
  - Time from mouse click to screen message that the "operation is complete"



Jitter

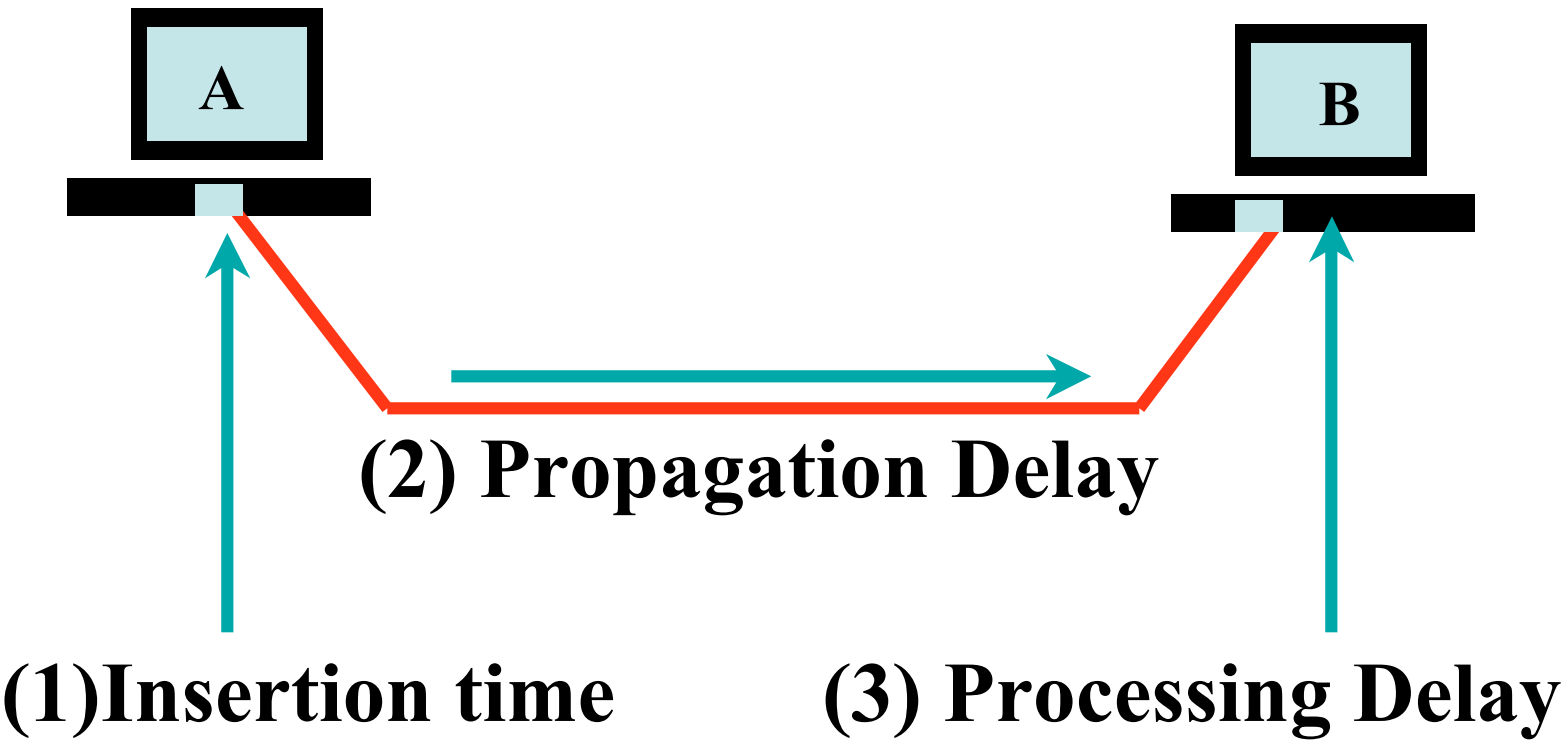
- Variation in delay over time
  - Non-issue for non-realtime applications
  - May be problematic for some applications with real-time interactive requirements, such as video conferencing

  - **E2E delay of 70 ms +/- 5 ms -> low jitter**
  - **E2E delay of 35 ms +/- 20 ms -> higher jitter**

Some Contributors to Delay

- Slow networks
- Slow computers
- Poor TCP/IP stacks on end-stations
- Poorly written applications

Analysis of Delay



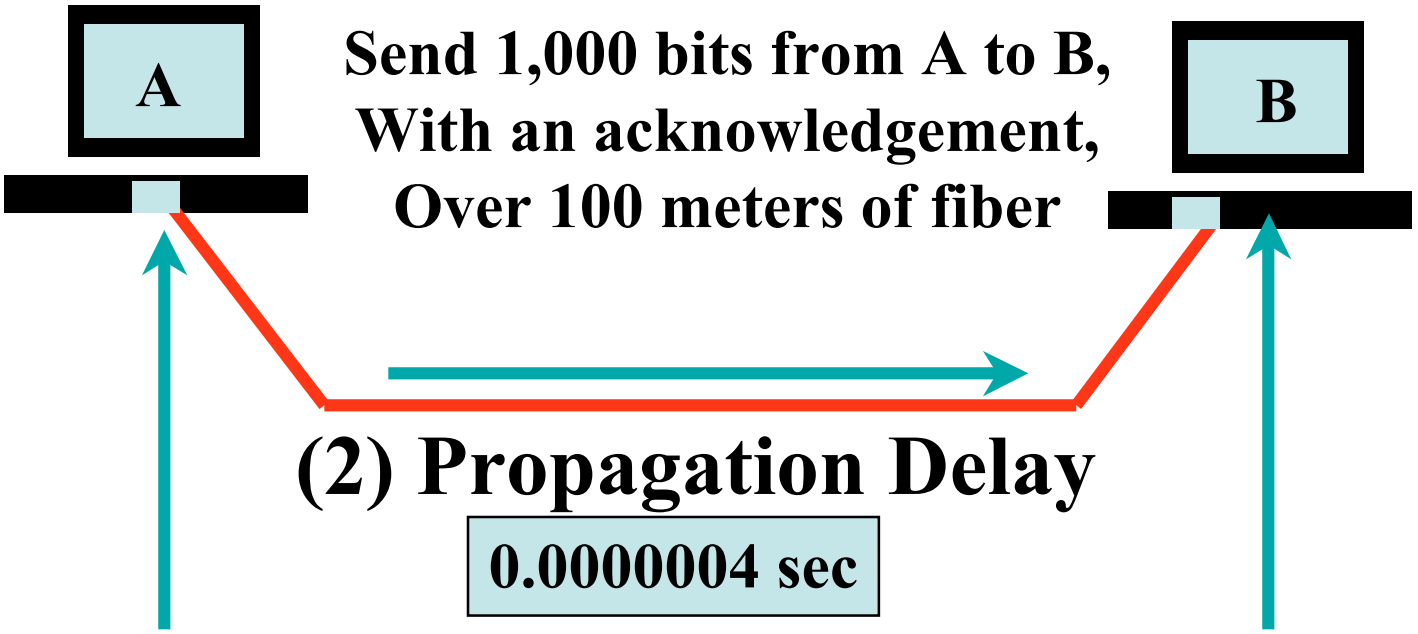
Analysis of

From: Deke
To: Ira
Date: Mon Feb 12, 2002, 11:00AM EST
Subject: Lunch

Hey Ira,

Meet you at the food trucks at noon!

^Deke



(2) Propagation Delay

0.0000004 sec

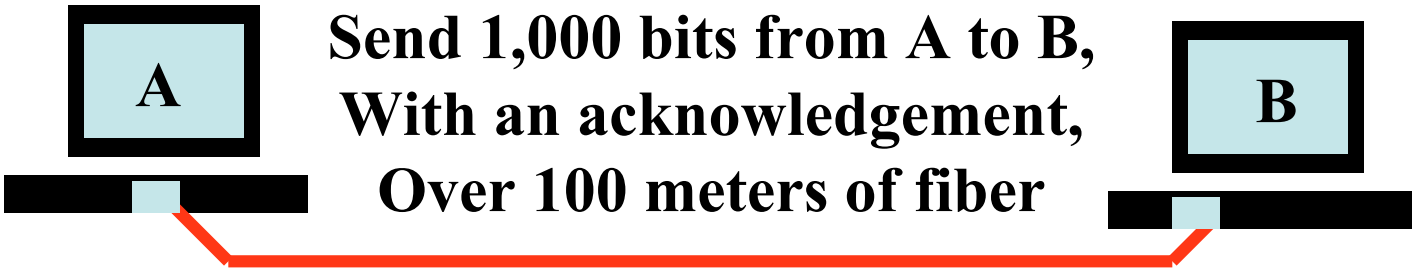
(1)Insertion time

0.0001 sec

(3) Processing Delay

0.01 sec

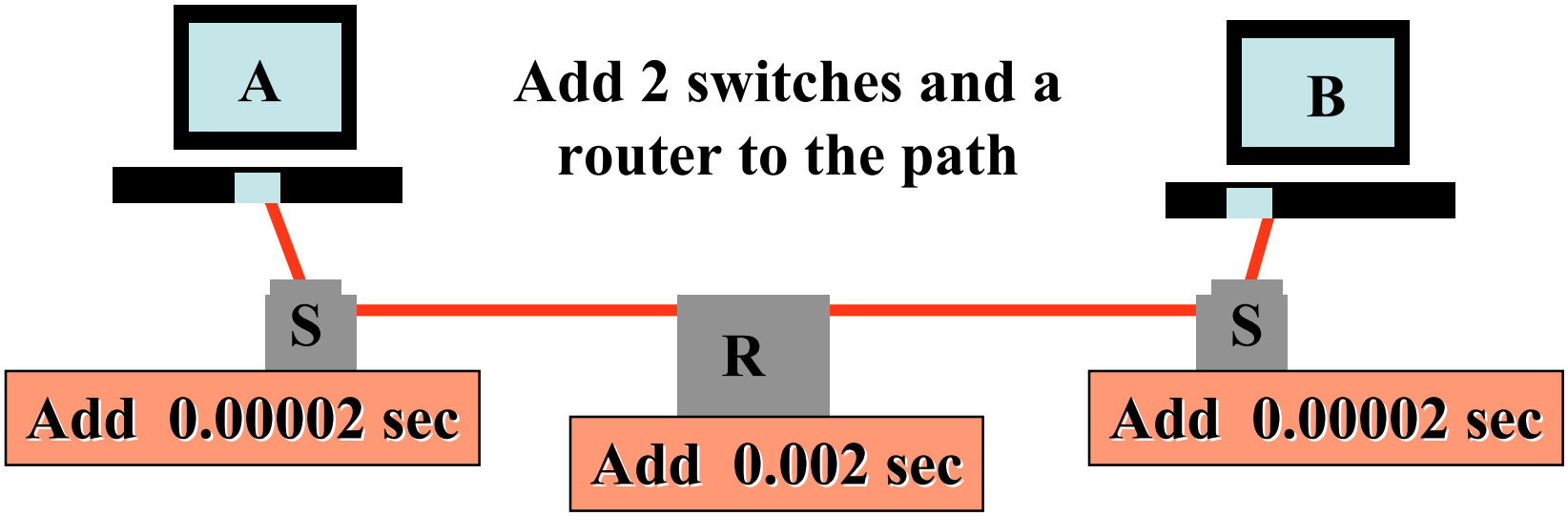
Analysis of Delay



•Data Insertion	• 0.0001 sec
•Propagation	• 0.0000004 sec
•Processing (B)	• 0.01 sec
•Ack Insertion	• 0.001 sec
•Propagation	• 0.0000004 sec
•Processing (A)	• 0.01 sec

Total Elapsed Time: 0.0211008 seconds

Analysis of Delay



New Total Elapsed Time: 0.0231408 seconds

Summary of Delay Analysis

- Propagation delay is of little consequence in LANs, more of an issue for high bandwidth WANs.
- Queueing delays are rarely major contributors.
- Processing delay is **almost always** an issue.
- Retransmission delays can be major contributors to poor network performance.

Why would retransmission occur?

- Truncation
- Outright loss
- Unable to put the pieces back together

- So how big/wide is that path anyway?

PMTU Discovery

- MTU = Maximum Transmission Unit
  - Largest IP packet that a link supports
- PMTU = minimum e2e MTU
  - Sender must keep datagrams no larger to avoid fragmentation
- How does the server know what PMTU is?
- RFC 1191
  - Try a desired value
  - Set DF to prevent fragmentation
  - On receipt of NEED FRAGMENTATION icmp

IP Path MTU

- Path MTU discovery (PMTUD) is a technique in computer networking for determining the maximum transmission unit (MTU) size on the network path between two Internet Protocol (IP) hosts, usually with the goal of avoiding IP fragmentation.

Works like this

- Path MTU discovery works by setting the DF (Don't Fragment) option bit in the IP headers of outgoing packets. Then, any device along the path whose MTU is smaller than the packet will drop it, and send back an ICMP "Fragmentation Needed" (Type 3, Code 4) message containing its MTU, allowing the source host to reduce its path MTU appropriately. The process repeats until the MTU is small enough to traverse the entire path without fragmentation.

Retries

- If the path MTU changes after the connection is set up and is lower than the previously determined path MTU, the first large packet will cause an ICMP error and the new, lower path MTU will be found. Conversely, if PMTUD finds that the path allows a larger MTU than what is possible on the lower link, the OS will periodically reprobe to see if the path has changed and now allows larger packets. On Linux this timer is set by default to ten minutes.

Problems with PMTUD

- Many network security devices incorrectly block all ICMP messages, including the errors that are necessary for PMTUD to work. This can result in connections that complete the TCP three-way handshake correctly, but then hang when data is transferred. This state is referred to as a "black hole connection".

MTU Issues

- Minimum link MTU for IPv6 is 1280 octets (versus 68 octets for IPv4)

  ⇒ on links with MTU < 1280, link-specific fragmentation and reassembly must be used

- Implementations are expected to perform path MTU discovery to send packets bigger than 1280
- Minimal implementation can omit PMTU discovery as long as all packets kept ≥ 1280 octets
- A Hop-by-Hop Option supports transmission of "jumbograms" with up to 2^{32} octets of payload

Issues with Path MTU Discovery

- What set of values should the sender try?
  - Usual strategy: work through the “likely suspects”
  - e.g. 4352 (FDDI), 1500 (enet), 1480 (IP/IP), 298 (modems)
- What if the PMTU changes?
  - Immediate reduction in PMTU
- It FAILS if
  - Routers don’t send ICMP type3 (time exceeded)

Clamping

- A workaround used by some routers is to change the maximum segment size (MSS) of all connections passing through links with MTU lower than the Ethernet default of 1500. This is known as MSS clamping.

Bigger MTUs?

9K MTUs (“jumbo frames”)

- And then there are frames that are six times the size of normal ethernet frames (9180 bytes long), so-called “jumbo frames”.
- 9180 is also noteworthy because it is the MTU of the Abilene backbone and a number of other Research networks

Some benefits of jumbo frames

- Reduced fragmentation overhead (which translates to lower CPU overhead on hosts)
- More aggressive TCP dynamics, leading to greater throughput and better response to certain types of loss.
- See:
  http://sd.wareonearth.com/~phil/jumbo.html
  http://www.psc.edu/~mathis/MTU/
  http://www.sdsc.edu/10GigE/

Are Jumbo Frames Actually Seen “In the Wild”?

The light's green, but...

- The Abilene backbone supports jumbo frames on all nodes under normal operational conditions [one link was recently temporarily constrained to 8192 due to a multicast bug]
- Jumbo frames have been publicly endorsed by I2 (e.g., see: http://www.internet2.edu/presentations/spring02/20020508-HENP-Corbato.ppt)
- But how much jumbo frame traffic are we actually seeing on Abilene? Virtually none.

I2 Netflow Packet Size Data

- For example, if you check http://netflow.internet2.edu/weekly/20030113/#full_packsizes you'll see that out of 144.3G packets, only 704.4K packets were larger than 1500 octets ("<0.00%" of all packets) during that week.
- We really don't know if those packets are 4470 or 9180 octets or … but at one level, that detail really doesn't matter -- what is key is that there's virtually nothing >1500.

Putting the pieces together:

- If we believe:
  -- the Abilene backbone itself (and I2 as an organization) support jumbo frames and
  -- jumbo frames are generally a good idea
  -- but we aren't seeing widespread use of jumbo frames at the current time and
  -- use of jumbo frames doesn't appear to be trending up in any systematic way…

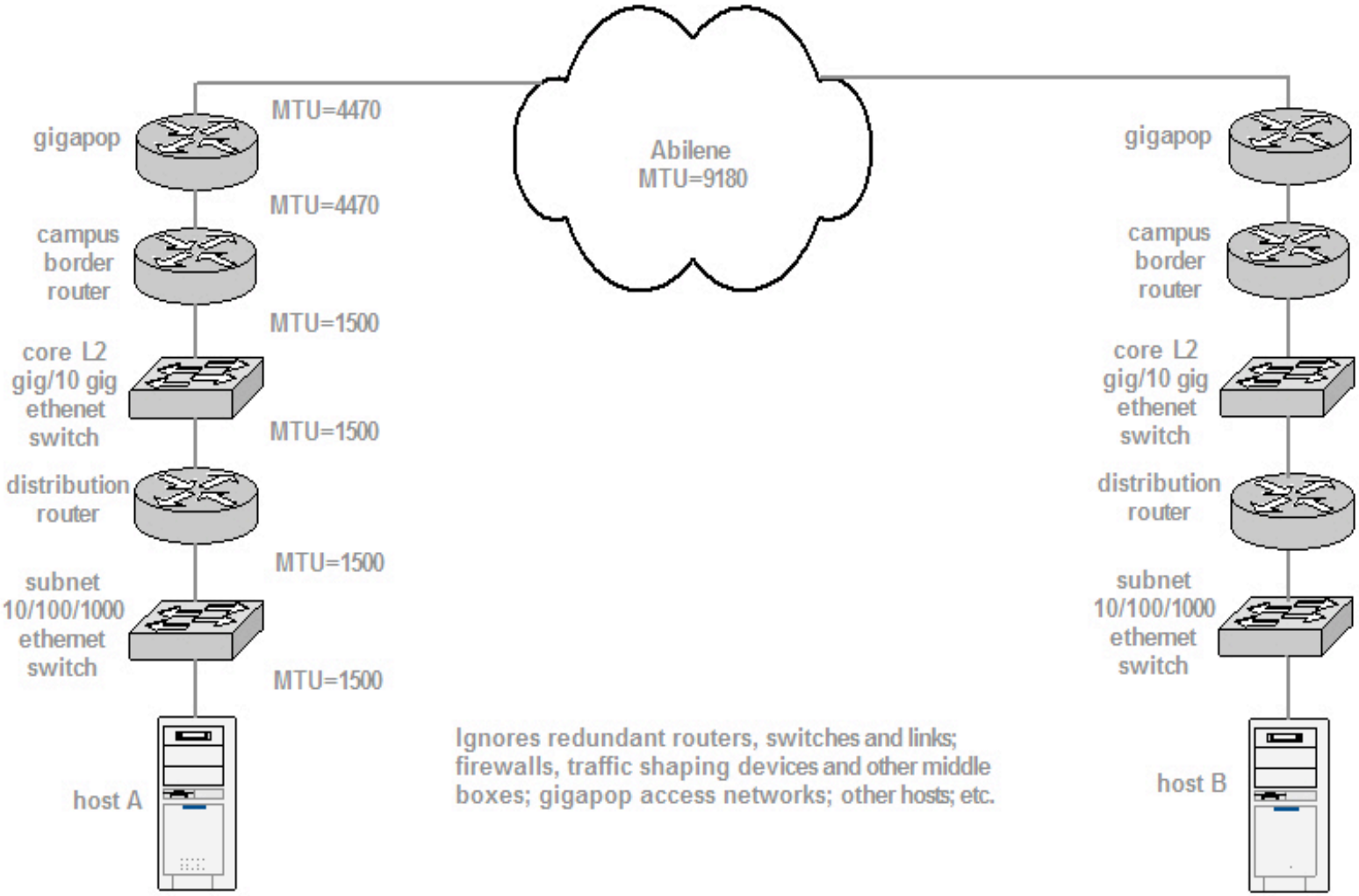
  It is then reasonable to assume that a systematic practical problem exists.

Understanding the Absence of Jumbo Frames on Abilene

Rule #1:

- **The smallest MTU used by any device in a given network path determines the maximum MTU (the MTU ceiling) for all traffic travelling along that path.**
- This principle dominates ANY effort to deploy jumbo frames.
- Consider, for example, a typical idealized conceptual network interconnecting host A and host B across Abilene….

Idealized conceptual network



So, in our hypothetical conceptual network...

- Even though the Abilene backbone can support **9180** byte MTU traffic, and
- Even though our hypothetical router-to-router links are able to support at least **4470** byte MTU traffic,
- The default 1500 byte MTU of the ethernet switches and the ethernet NIC in our hypothetical network means our traffic will have a maximum frame size of **1500** bytes.

And this doesn't even consider the guys on the other end...

- …who will likely also have one or more network devices in the path that use an MTU of 1500 (or less).
- Of course, since Rule #1 applies from end to end, even after you fix your network to cleanly pass jumbo frames, if your collaborators haven't, you will still be constrained to normal frame MTUs to those hosts.

100Mbps, 10Mbps ethernet and subnet MTUs

- A more subtle fact impacting jumbo frame deployment at the campus level is that jumbo frames are rarely supported on 10 or 100Mbps ethernet links. This is relevant because at most campuses:
  -- relatively few hosts are gigabit attached
  -- gigabit hosts often live on the same subnet as 10Mbps or 100Mbps hosts
  -- things get tricky if all hosts on a subnet fail to agree on a common MTU

Cleaning up the neighborhood

- Faced with that reality, the most common option is probably to create a separate gigabit-only jumbo frame subnet, which usually means somebody's going to have to renumber unless you've been very lucky/ systematic in assigning IP addresses.
- You may also need additional gigabit router interfaces (assuming you want to keep the legacy 10/100 hosts downstream of a gigabit uplink).

“If it isn’t broken…”

- The final potential killer roadblock at the campus level is reluctance on the part of many network engineers to screw around with a stable production network just so a few systems can begin [trying] to use a perceived “non-essential” feature.
- You should also be prepared to be asked, “Well, who else on I2 that you work with is using jumbo frames at this point, anyhow?” [the classic chicken-and-egg question that also dogged IP multicast and IPv6 rollout]

Internet2 Participant MTUs

- All that discussion aside, "*How many I2 participants appear to have routine >1500 MTU connectivity, for example to their primary web server www.<whatever>.edu?*"
- Courtesy of Bill Owens and Nysernet, tests were done from ATM-connected Debian box [with at least a 4470 byte-clean path to Abilene] to over 211 Internet2 participant main web sites.

On the choice of primary web servers as an MTU test target

- We know that some may question our choice of the institution's primary web server as our MTU test target -- such a box may not have any need for jumbo frames, for example. True. However, it does provide a convenient, centrally maintained, universally available "important" host to test. (We'd gladly test other better-connected hosts if we knew they existed!)

It's a 1500 byte MTU world out there...

- The most noteworthy thing we found is that none of the tested hosts could accept >1500 byte frames.

TCP Steady State

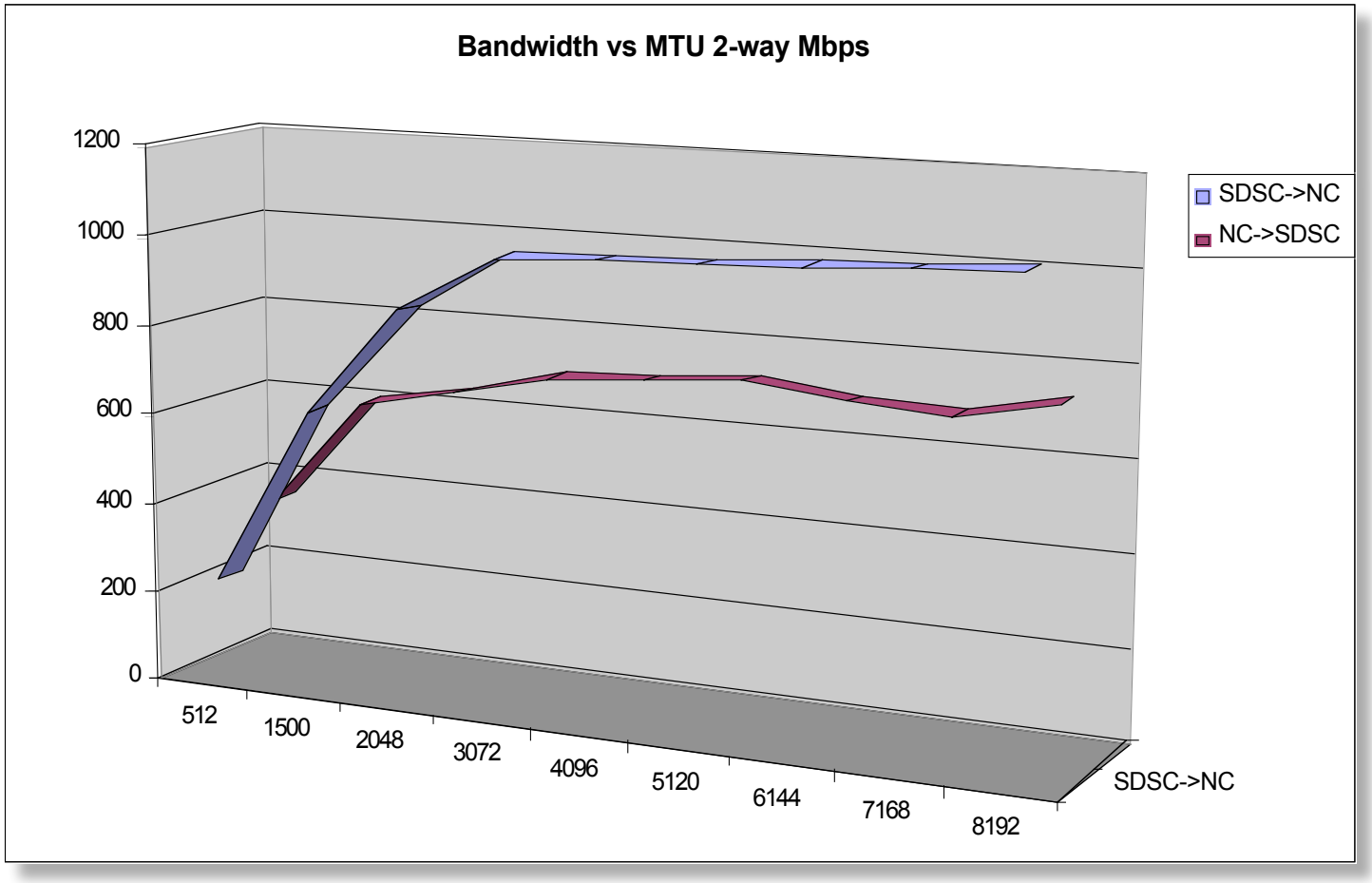
- If TCP window size and network capacity are not rate limiting factors then (roughly):

$$\text{e2e throughput} < \frac{0.7 * \text{Max Segment Size (MTU)}}{\text{Round Trip Time (latency)} \sqrt{\text{loss}}}$$

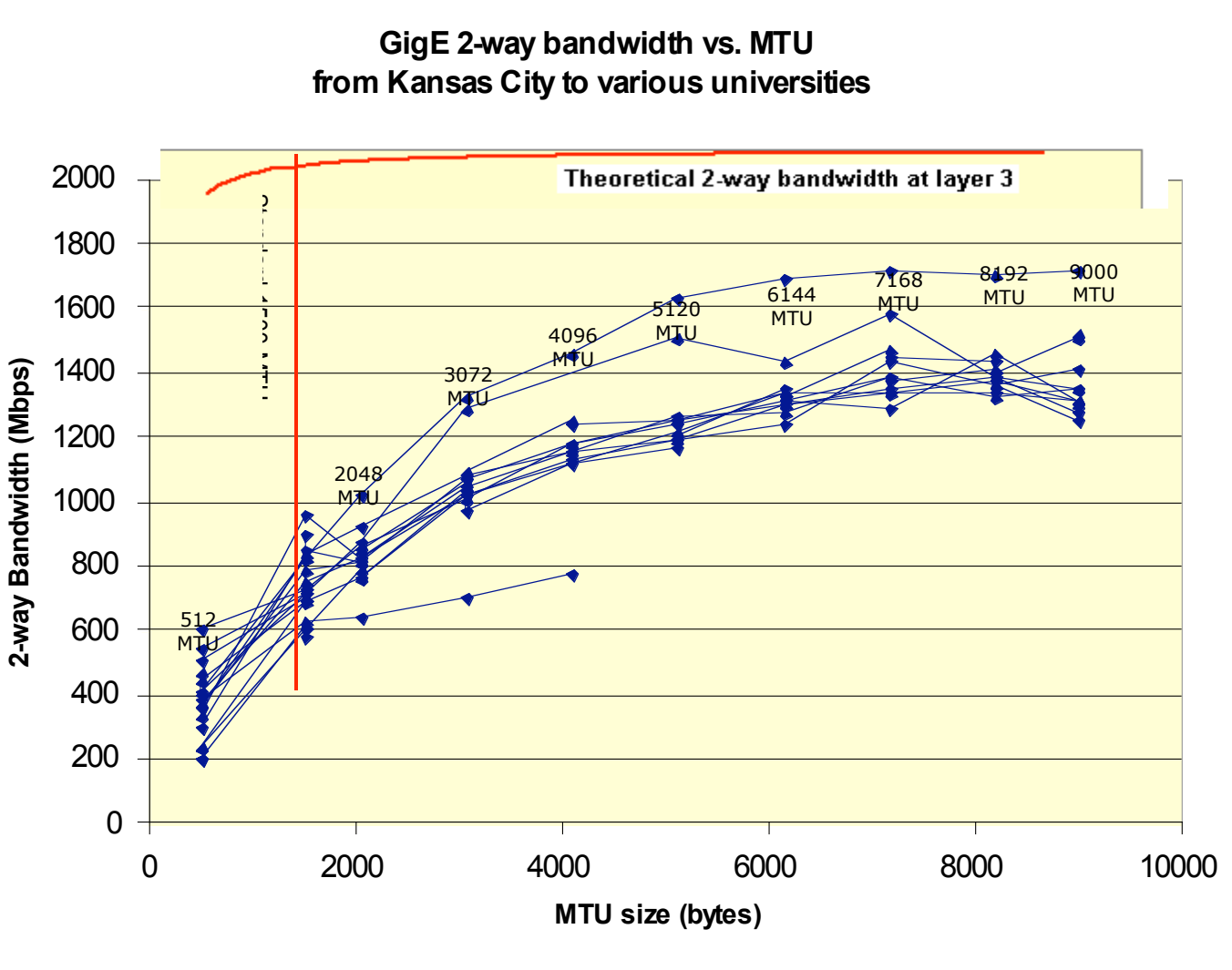
M. Mathis, et.al.

- Double the MSS, double the throughput
- Halve the latency, double the throughput (shortest path matters)
- Halve the loss rate, 40% higher throughput

Abilene Results: iPerf NCSU ← → SDSC



Abilene & CA*net Testing - 2003



UDP v TCP

- Ok, so all this really means is that TCP is going to be a problem with MSS/MTU larger than 1500 bytes. Fragmentation will ensue

- Lets just use UDP …

Fragmentation

- All that discussion about how to avoid/minimize fragmentation was primarily targeted at TCP

- But with DNS - with EDNS0 - can push UDP into a fragmented world too.

Who cares?

Performance Wizards

Database replication

And Firewall Vendors!

DNS protocol limits

UDP - 512 bytes

TCP - 2 MINUTE hold-down

Then … EDNS extensions…

EDNS0 gives us some headroom - 4096 bytes.

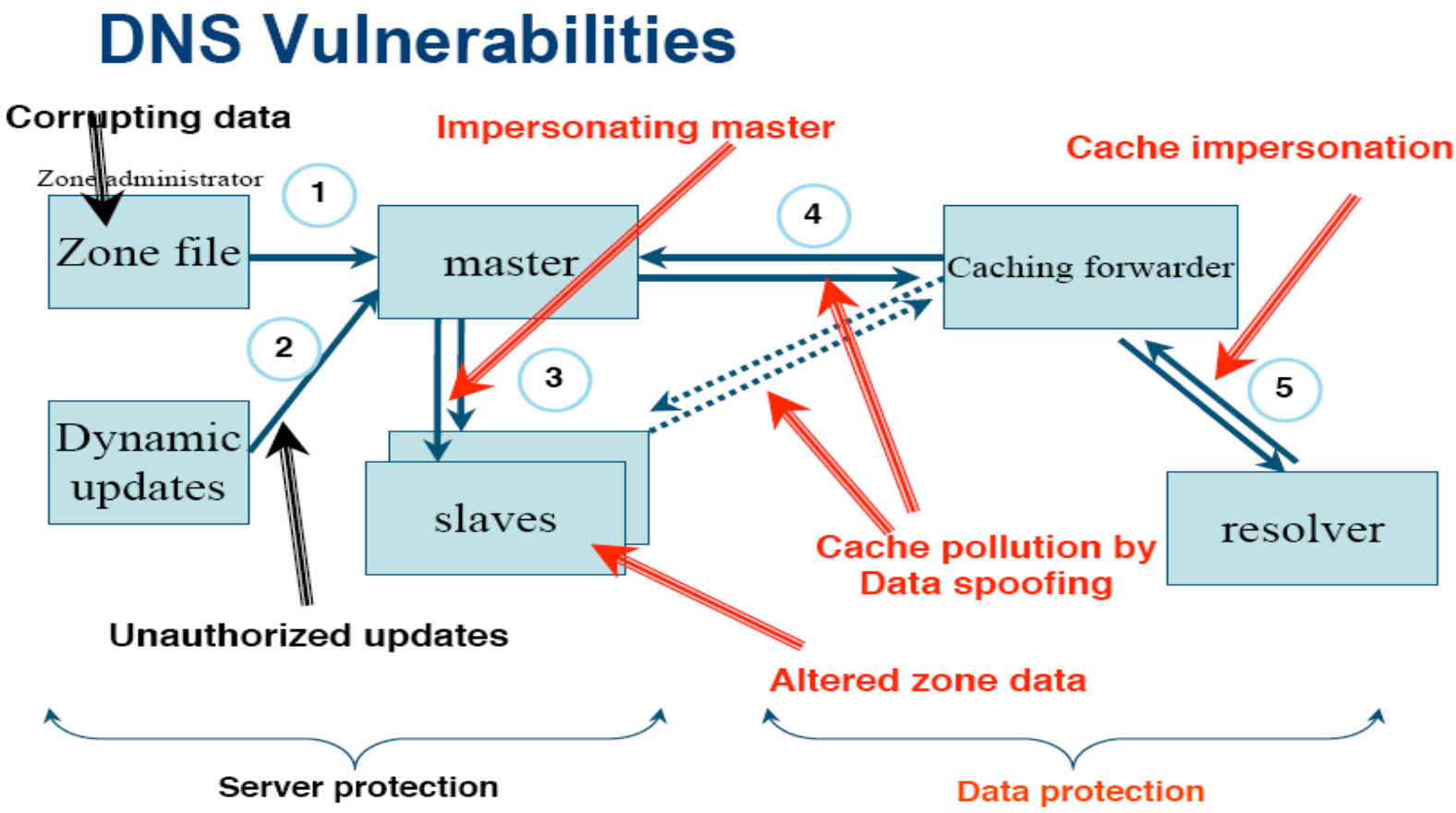
Except -some- applications

DNS with DNSSEC is a serious contender

The Application changes the default MTU for UDP datagrams

from 512 to a max of 4096 bytes.

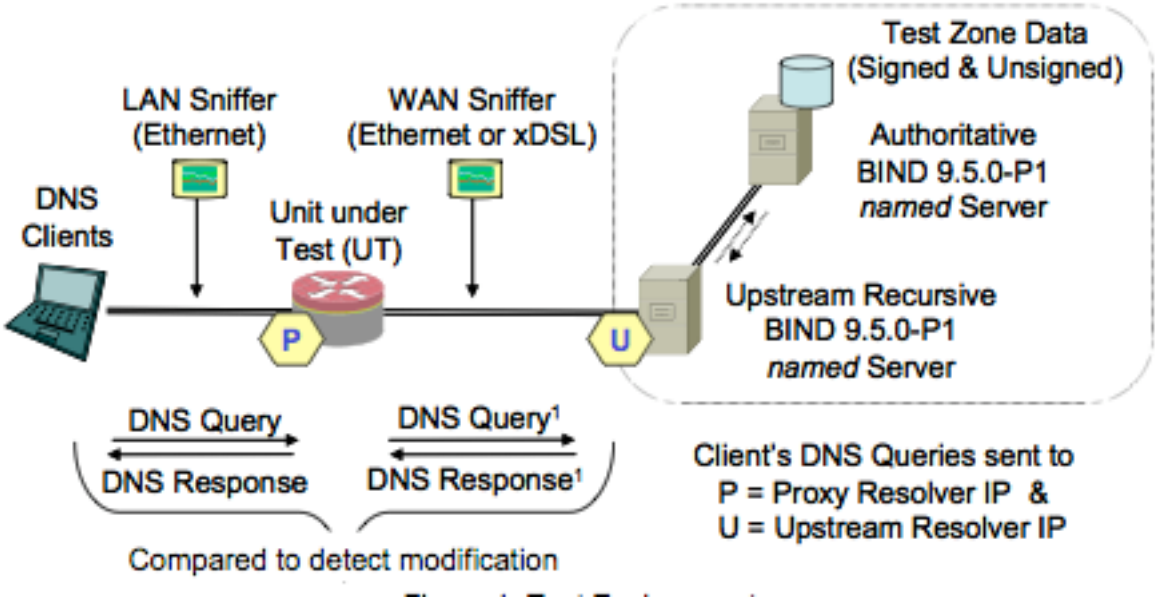
Why DNSSEC?


DNS Vulnerabilities
Corrupting data
Zone administrator
Zone file
1
master
Impersonating master
4
Caching forwarder
Cache impersonation
2
3
5
Dynamic updates
slaves
resolver
Cache pollution by Data spoofing
Unauthorized updates
Altered zone data
Server protection
Data protection

Where it hurts

- Between the IMR and the authoritative server
- Between the IMR and the resolver.

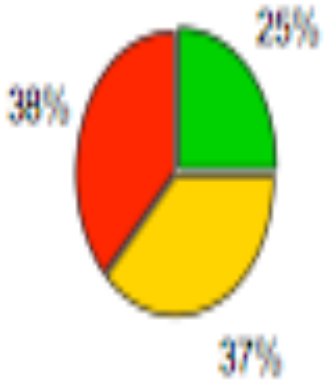
And where are they?



What do firewalls do

ssac-35 - tested 24 CPE units

As a consequence, we conclude that just 6 units (25%) operate with full DNSSEC compatibility "out of the box." 9 units (37%) can be reconfigured to bypass DNS proxy incompatibilities. Unfortunately, the rest (38%) lack reconfigurable DHCP DNS parameters, making it harder for LAN clients to bypass their interference with DNSSEC use.



			Out of the Box Usage Mode	Route DNS to Upstream Resolver	Proxy DNS over UDP	A. EDNS0 Compatibility	B. Signed Domain Compatibility	E. Request Flag Compatibility	D. Checking Disabled Compatibility	C. DNSSEC OK Compatibility	Proxy DNS over TCP
1	2Wire	270HG-DHCP	Proxy	OK	OK	FAIL	OK	OK	FAIL	FAIL	FAIL
2	Actiontec	MI424-WR	Proxy	OK	OK	FAIL > 512	OK	OK	OK	OK	FAIL
3	Apple	Airport Express	Proxy	OK	OK	FAIL > 512	OK	FAIL	FAIL	FAIL	OK
4	Belkin	N (F5D8233)	Proxy	OK	OK	FAIL > 1500	OK	OK	OK	OK	FAIL
5	Belkin	N1 (F5D8631)	Proxy	OK	OK	FAIL > 1500	OK	OK	OK	OK	FAIL
6	Cisco	c871	Route	OK	OK	FAIL > 512	OK*	OK*	OK*	OK*	FAIL
7	D-Link	DI-604	Proxy	MIX	OK	FAIL > 1472	OK	OK	OK	OK	FAIL
8	D-Link	DIR-655	Proxy	OK	OK	OK	OK	OK	OK	OK	FAIL
9	Draytek	Vigor 2700	Proxy	OK	OK	FAIL > 1464	OK	FAIL	FAIL	OK	FAIL
10	Juniper	SSG-5	Route	OK	OK	OK	OK	OK	OK	OK	FAIL
11	Linksys	BEFSR41	Varies	OK	OK	FAIL > 1472	OK	OK	OK	OK	FAIL
12	Linksys	WAG200G	Varies	OK	OK	OK	OK	OK	OK	OK	FAIL
13	Linksys	WAG54GS	Varies	OK	OK	OK	OK	OK	OK	OK	FAIL
14	Linksys	WRT150N	Varies	OK	OK	FAIL > 512	OK	OK	OK	OK	FAIL
15	Linksys	WRT54G	Varies	OK	OK	FAIL > 512	OK	OK	OK	OK	FAIL
16	Netgear	DG834G	Proxy	OK	OK	FAIL > 512	OK	FAIL	FAIL	MIX	FAIL
17	Netopia	3387WG-VGx	Proxy	OK	OK	FAIL > 512	OK	FAIL	FAIL	FAIL	FAIL
18	SMC	WBR14-G2	Proxy	MIX	OK	FAIL > 512	OK	OK	OK	OK	FAIL
19	SonicWALL	TZ-150	Route	OK	n/a	n/a	n/a	n/a	n/a	n/a	n/a
20	Thomson	ST546	Proxy	OK	OK	FAIL > 512	OK	OK	OK	OK	FAIL
21	WatchGuard	Firebox X5w	Varies	OK	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
22	Westell	327W	Proxy	OK	OK	FAIL	OK	OK	FAIL	FAIL	FAIL
23	ZyXEL	P660H-D1	Proxy	OK	OK	FAIL > 1464	OK	OK	OK	OK	FAIL
24	ZyXEL	P660RU-T1	Proxy	OK	OK	FAIL > 1464	OK	OK	OK	OK	FAIL
	Make/Model		DHCP DNS	No Proxy	UDP Proxy Transport Tests		UDP Proxy DNSSEC Tests				TCP Proxy

Table 2. Test Result Summary

DNSSEC moves the performance point

IMR configuration

- BIND 9 ships with default of DO=1
- DO is the “dnssec-ok” bit. When set to one, the authoritative server should send DNSSEC data, if it is available.

IMR fallback

- BIND 9 will first try with bufsize = 4096
- If that fails, then retry with bufsize = 512

- Why will it fail?

IMR to authoritative server

- Back to the Abilene graphic - usually these are on 4096 MTU capable paths
- The IMR asks for and gets signed data

- Except when there is a “guardian” between the IMR and the authoritative server

Issues

Where is your IMR located?

What is the path to your IMR?

The Maginot Line

Firewalls, ALGs, and DPI are always reactive

Vendors take very conservative defaults based on the lemma “only allow exactly what is needed”

But what is needed changes. Can the guardian adapt?

The Timeline

US DoC is committed to signing the root this year - 2009

This will change the default DNS response size from just under 512 bytes to something more.

Remember this is to improve the security and stability of the Internet…

So how big?

- Right now, in an analysis done for the "L" root server replaying modified live data, we see a large jump in TCP queries largely due to the "fall back to EDNS@512" BINDism, but the number of TCP queries is still pretty much in the noise and well within the tolerances of "L" However let's posit an alternative reality in which people actually turn on validation. A root DNSKEY response is 1749 bytes. What does this imply? - David Conrad

The Application Problem

Default resolver behaviour

UDP gt 512 bytes

UDP fragments

TCP

BIND behaviour

BIND first tries bufsize=4096/DO=1 and upon timeout falls back to bufsize=512/DO=1. One can make a case based solely on RFC 3226 that bufsize=512/DO=1 violates that spec.

Why fallback might occur

1. The responding server sends a large response, but there is some device in the path that believes all DNS messages should be 512 or less and Adds Value by dropping the packet.
2. The responding server sets DF, but the resulting IP datagram is too large for some hop and is discarded.
3. The responding server doesn't set DF, the resulting IP datagram is too large for some hop and is fragmented, but some misguided device downstream drops the fragments.
4. The response just disappears as part of "normal packet loss" on the Internet.

Fallback?

- if one falls back directly to 512, the resulting message in a DNSSEC world is almost certain to have TC set, which will result in a retry over TCP.

DNS over TCP

- Ok… but … Webservers can do it

- http://www.litespeedtech.com/docs/webserver/config/tuning/?hilite=tcp,performance,
- Specifies the maximum concurrent connections that the server can accept. It includes both plain TCP connections and SSL connections. It should not exceed the hard limit set by the server: 300 for Standard Edition.

With 2 minute timeout and 20,000 qps and 5% TCP packet loss … that's a lot of open TCP sessions - perhaps more than 300…

Our choices are:

- fix the name server implementations so they do not violate RFC 3226 by not advertising a buffer less than 1220 when DO=1

- revise RFC 3226 to remove/revise the buffer size restriction.

The outcome will be

- pick 1 and trigger hardware/firmware upgrades near the edges of the Internet and residual, persistent edge failure in the DNS for decades to come.

- pick 2, and kowtow to shortsighted hardware/firmware vendors and force a protracted decade-long migration in the changes to DNS transport protocols.

Long Tail

- What is the hardware refresh cycle for CPE gear?
- What is the software upgrade cycle for DNS software?
- … 36-60 months - if your lucky

Open Questions

- Fallback takes time - what are users willing to tolerate?
- If there is something between 4096 and 512, what should it be? 1220, 1490, 1500, 1800, 2048?
- How long is the tail? CPE gear? BIND versions?
- Can you check your path now?

View from “here”

- Pulled the Query logs from “B”
  - 1.27 Billion Queries in 24 hrs
  - 487 million unique IP addresses
- Currently doing PMTU studies on them

- Current guess - double digit % will fail when DNSSEC is enabled in the root
  - 49 million nodes going “dark” is OK?

DNS PMTU tools

- SEC-SPIDER
  - http://secspider.cs.ucla.edu/
  - http://www.vantage-points.org/
- OARC
  - https://www.dns-oarc.net/oarc/services/replysizetest
  - Or .. From the command line:
  - $ dig +short rs.dns-oarc.net txt

FIN?

Questions?

Commentary?

Thanks

Bill Manning

bmanning@ep.net