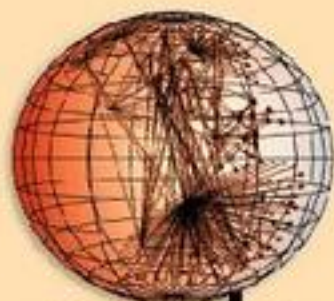




caida

**measurement and analysis
of the root DNS system:**
update

september 2002
ucsd/sdsc/caida
kc@caida.org

<http://www.caida.org/outreach/presentations/>

research problems

main directions of caida's DNS research:

- continuous performance monitoring of root/gtld servers
- investigation and modeling of *bind* algorithm behavior
- analysis of bogus queries and broken name server configurations
- evaluation and optimization of root server placement

types of collected data

- caida started DNS measurement in 2001
- three kinds of data are collected and analyzed
 - passive capturing of DNS packets (netramet, dnsstat/coralreef, tcpdump)
 - log files from root servers
 - active probing of the infrastructure

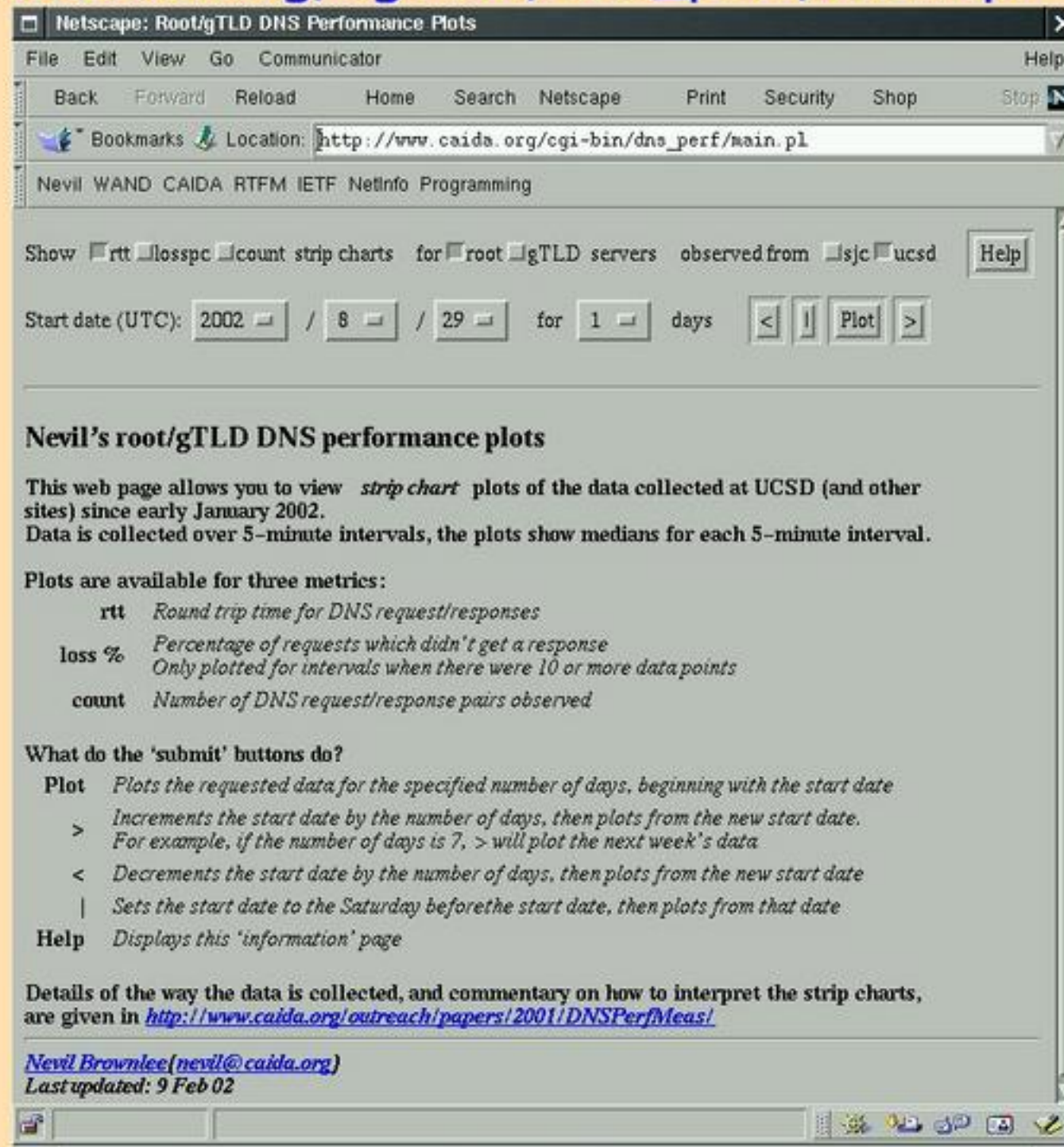
I. monitoring dns root servers performance

(nevil brownlee, caida/u.auckland)

- NeTraMet traffic meter captures DNS request/response packets
 - root servers and gTLDs
- passive observations, January 2002 - present
 - From UCSD - nearly continuous
 - From SJC - best effort
- measurements of:
 - rtt for UCSD and SJC
 - loss% and count for UCSD
- results at:
 - www.caida.org/cgi-bin/dns_perf/main.pl
 - Updated daily after midnight

I. monitoring dns root servers performance

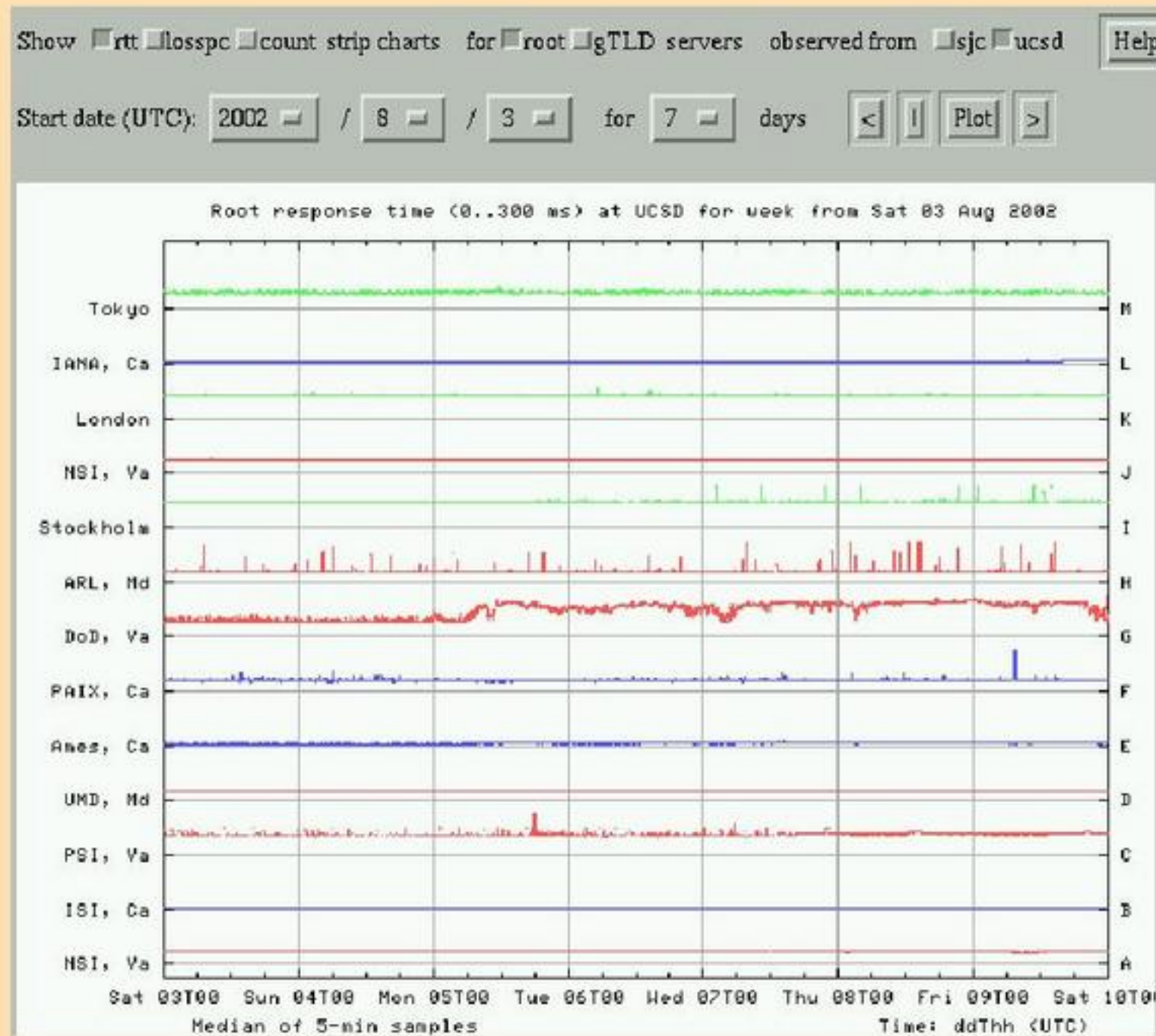
www.caida.org/cgi-bin/dns_perf/main.pl



■ interactive plotting of parameters for comparison and analysis

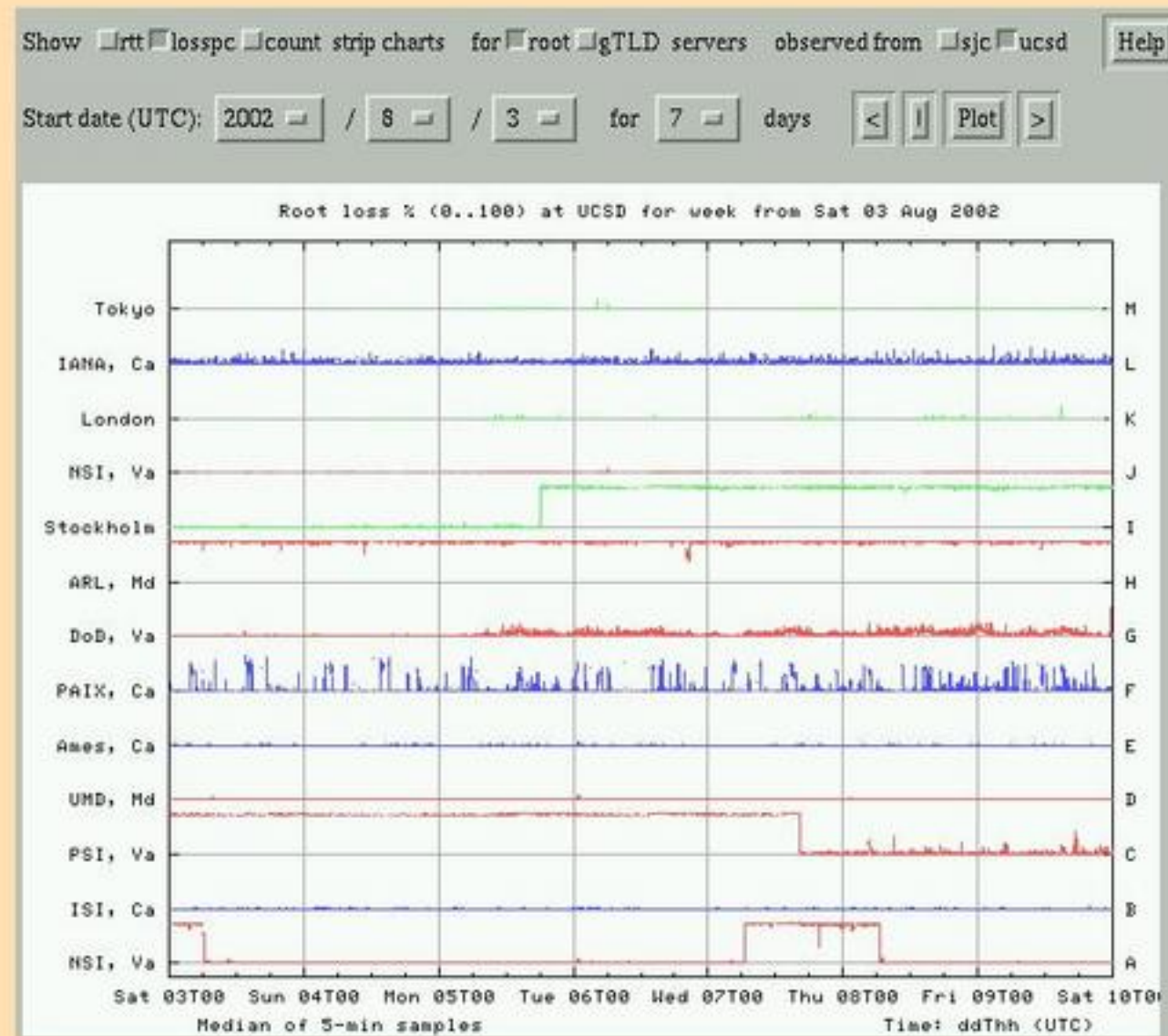
● examples follow

I. root servers performance - RTT



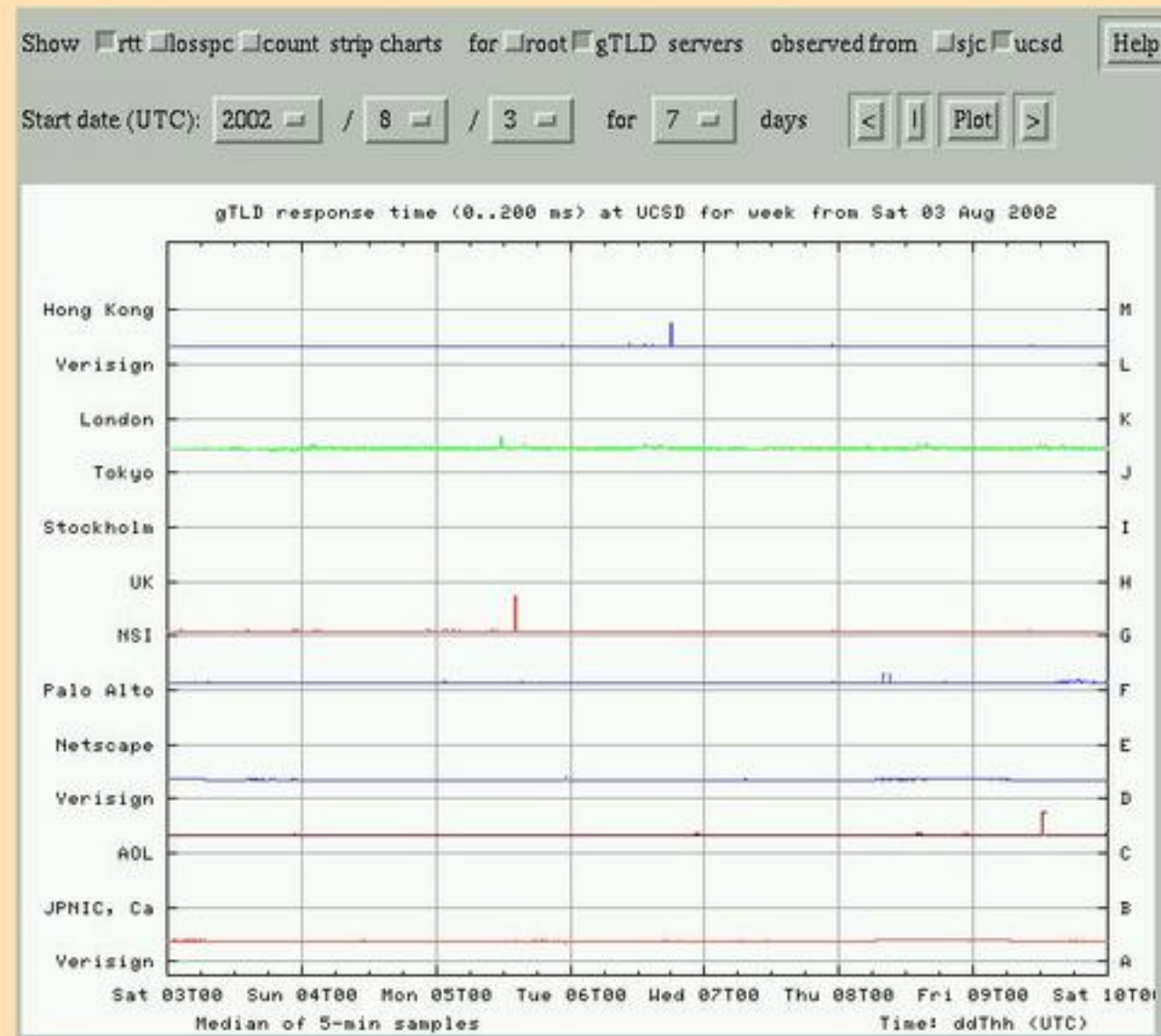
- stable response time for most servers (G overloaded on weekdays)

I. root servers performance - losses



- periods of high loss on A, C, I, and J
 - but note: high losses do not necessarily affect the measured RTT

I. gTLD servers performance - RTT



- gTLDs are consistently more stable than the roots

I. continuous monitoring: future plans

- deploy 3-4 additional NeTraMet meters
 - please contact kc or nevil@caida.org
- strategic locations:
 - europe
 - asia/pacific
 - east coast of US
- start monitoring of country code servers (ccTLDs)
- time-series analysis of the data
 - [lack of] correlation among loss, workload, RTT
- evaluate ICMP as indicator of DNS performance (see task 3)

II. investigation and modeling of BIND

(duane wessels, caida/packet-pushers)

bind certainly works - but **how?**

- who queries root servers?
- how does a client select a root server?
 - presumption: based on RTT measurements
- how does a root server acquire clients?
- do all root servers "see" all clients?
 - Note: they are supposed to...
- can we model all (any of) this?

II. modeling of BIND (cont.)

how does a client select a root server? (courtesy vix)

- 1) rtt sorting
 - try every NS+A until you find best one
 - 2) aged rtt sorting
 - gradually depref the "best NS+A" to force rescans
 - 3) priming
 - only use "root.cache" to do an initial ". NS" query sweep
 - 4) static
 - use servers in order on EVERY query, stop when answer heard
 - 5) random selection among known authoritative (including roots)
 - 6) round robin
 - rotate the NS and/or A list every time one is used.
-
- BIND4/BIND8, and modern BIND9 do (2) and (3)
 - early BIND9 did (1) and (3)
 - djbdns does (5)
 - win2k does (5) [although it may tend to prefer A if it responds quickly]
 - win2003 "(2)-like" [per usoft: tries to balance across NSes per node]
 - nobody does (6)

II. modeling BIND (cont.): asymmetric loads

why does A get 2X the query load of B..M

- if BIND4/BIND8 and modern BIND9 are still dominant query sources then they are hitting A..M somewhat evenly, with small localizations according to RTT variance seen by various client populations.
- A's additional $\sim 3\text{Kq/sec}$ in volume from where?
- as yet uninvestigated

II. investigating/modeling BIND: measurements

CAIDA dnsstat utility

- passive measurements at the root servers
- collects aggregated statistics of queries:
 - source address
 - number of queries
 - type of queries
- does not record query subjects
- can run for days without a problem

II. BIND behavior - dnsstat data

- 26 hours at 10 minute intervals from 14 august 2002

- instrumented:

- e-root (california, us)
- i-root (stockholm, sweden)
- k-root (london, uk)
- m-root (tokyo, japan)
- f-root (california, us)
- a-root (va, us)

need simultaneous runs on all participating servers

need cooperation from US servers

(2.5 non-US servers are quite forthcoming)

II. dnsstat results - number of queries

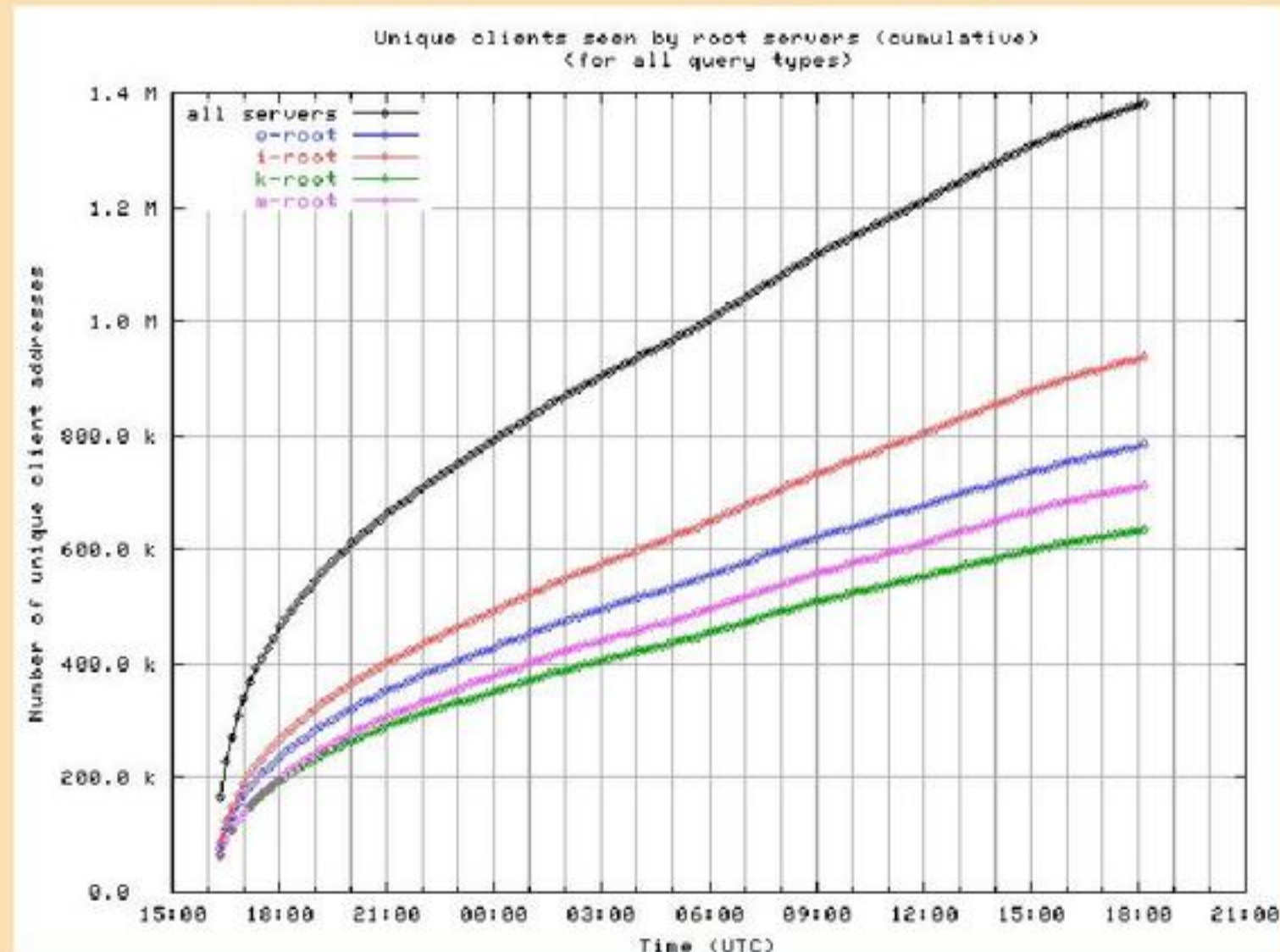
number of queries per 10-minute interval



- no clear pattern, between 5000 and 12000 queries per second

II. dnsstat results - new clients

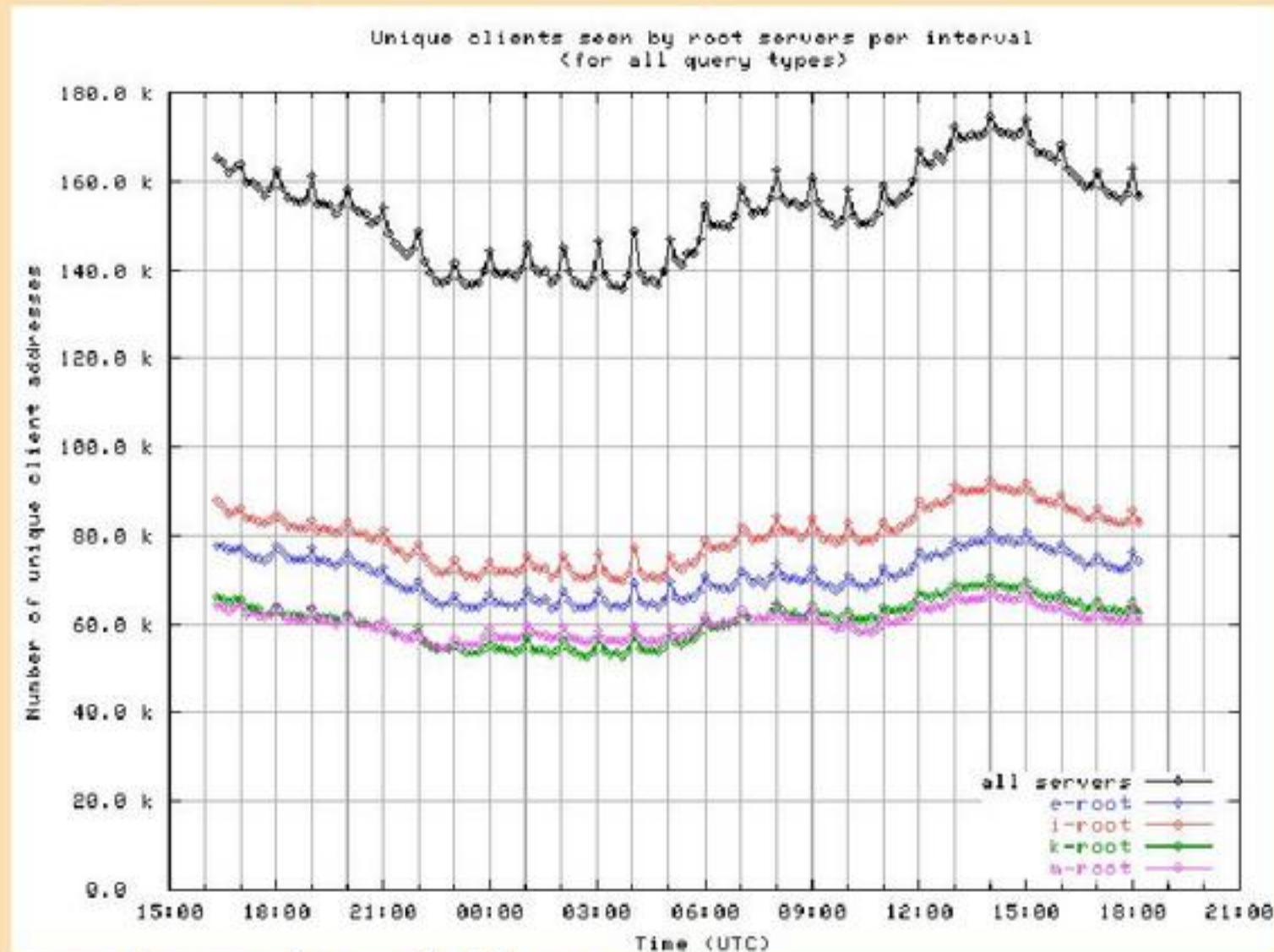
accumulated number of unique clients



- no conversion or slowdown after 1 day

II. dnsstat results - new clients (cont.)

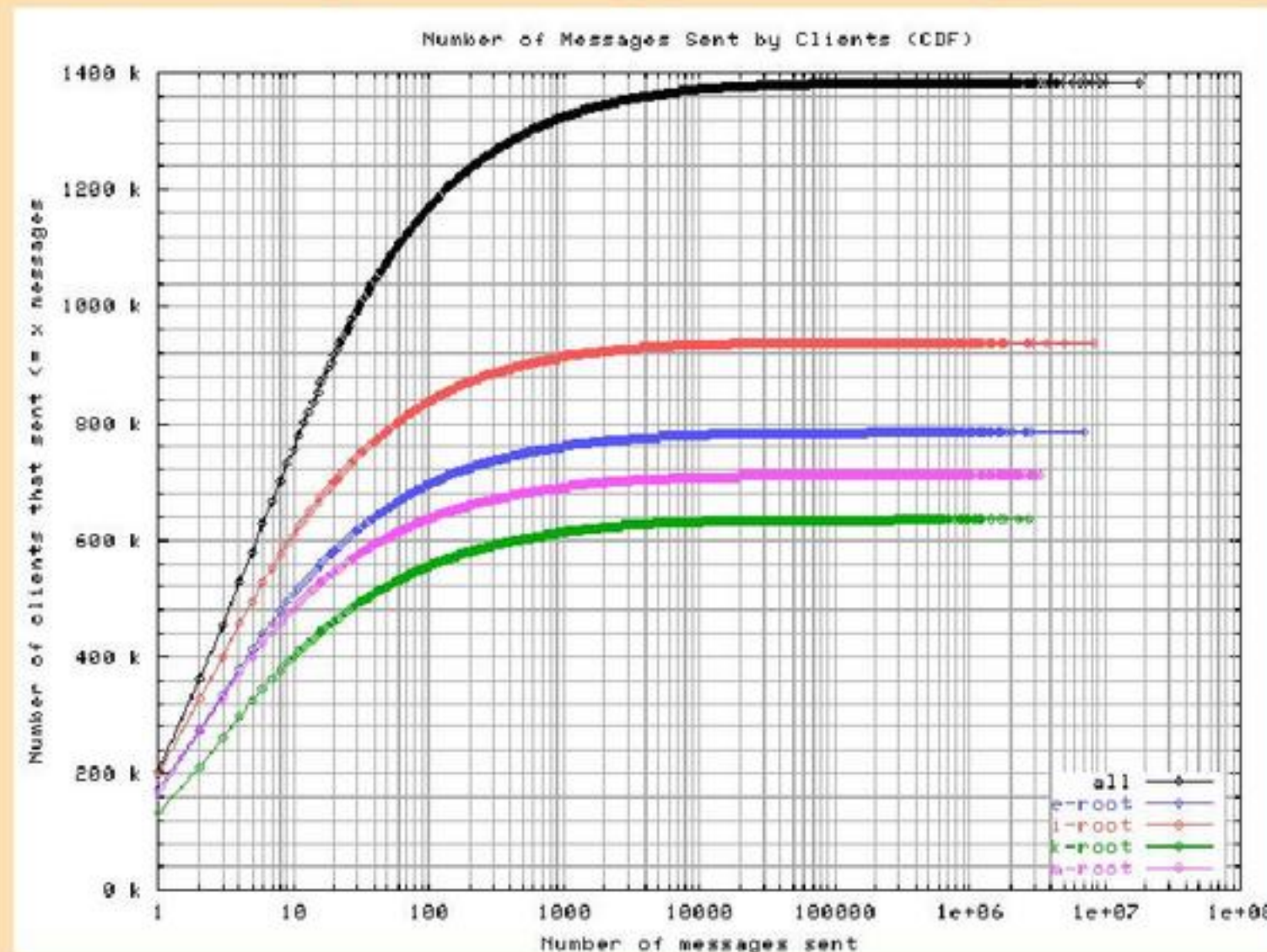
number of unique clients per 10 minute interval



- apparently, no diurnal variations
- peaks at hourly boundaries - why?
 - some popular software's cron behavior?
 - still investigating

II. dnsstat results - number of messages

number of messages sent by clients



- half of the clients sent 8 or fewer messages (in 26 hrs)
- one client sent about 18M messages (192 per second)

II. BIND behavior analysis - future plans

- continue collection and characterization of log files
 - interarrival rates
 - popularity (some names are more popular than others)
 - correlations between popularity and TTLs
 - message sizes
 - response codes
 - duplicate queries
 - invalid queries
 - percentage of caching/non-caching clients
- develop simulating software
- run controlled experiments
- `icmp as indicator of dns' calibration (again)

III. bogus queries

(evi nemeth, caida/sailboat; andre/ken/duane, caida)

misuse of root servers

- root servers receive a large amount of invalid queries
- possible classification:
 - stupid (e.g. lookup the IP address of an IP address)
 - invalid TLD (i.e. "foo.ntdomain")
 - repeat queries for the same data (new meaning to `persistent software')

our goals:

- identify clients that do not cache referrals
 - would more consistent caching reduce load significantly?
- determine the nature of high load clients
 - misconfigured name server installations?
 - unknown DNS implementations?
 - viruses?
- suggest possible fixes to reduce the load

III. bogus queries

data from our earlier study:

- bogus A queries to root servers for a few hours at f-root in 2001
 - A queries ask for the IP address of a hostname
- malformed A queries were 14% of the load at F.root
 - asking for the IP address of an IP address
 - ▶ example: "A 206.168.0.4" - should not happen
 - guilty: Microsoft Win2k resolver, viruses (win95/98/nt), macOSX resolver
(good news: with our help, Microsoft found and fixed this bug in Win2k (although the way to turn off a bad default configuration is 6 or so menus deep...))
- 20% of queries asking for non-existent TLD
 - lots of internal microsoft names (active directory)
 - lots ending in .local, .localhost, .workgroup, .msft, .domain, etc.
- hard to track down, nameservers just relay clients queries
 - cannot see back to the actual client that asked the question

insidious problem: private (rfc1918) addresses

workload myth:

private addresses do not appear in the core

reality:

- private addresses appear all over the place including (consistently) in queries to root name servers

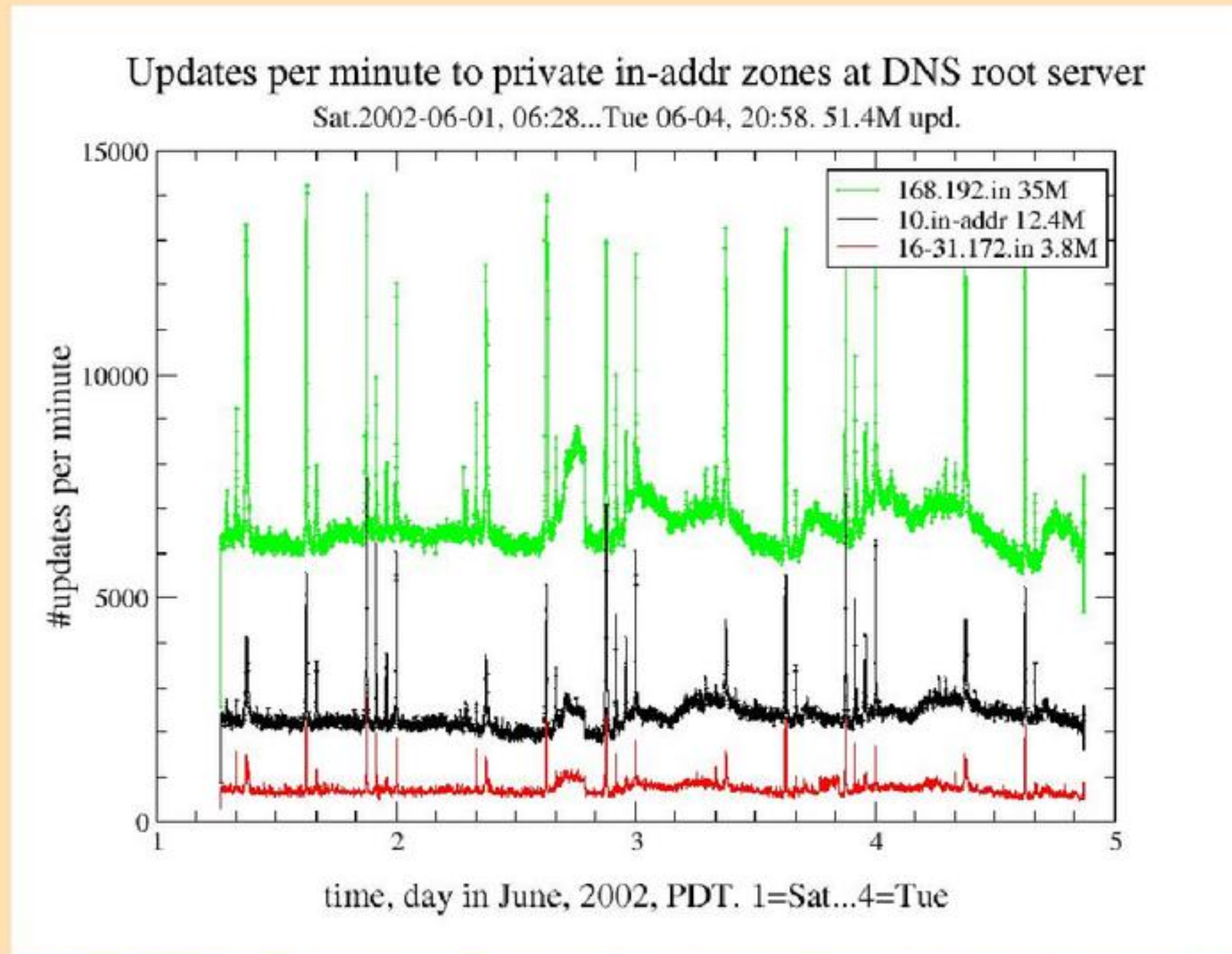
Broido's 1st Law:

'what should not be seen in the Internet will appear 1% of the time'

data:

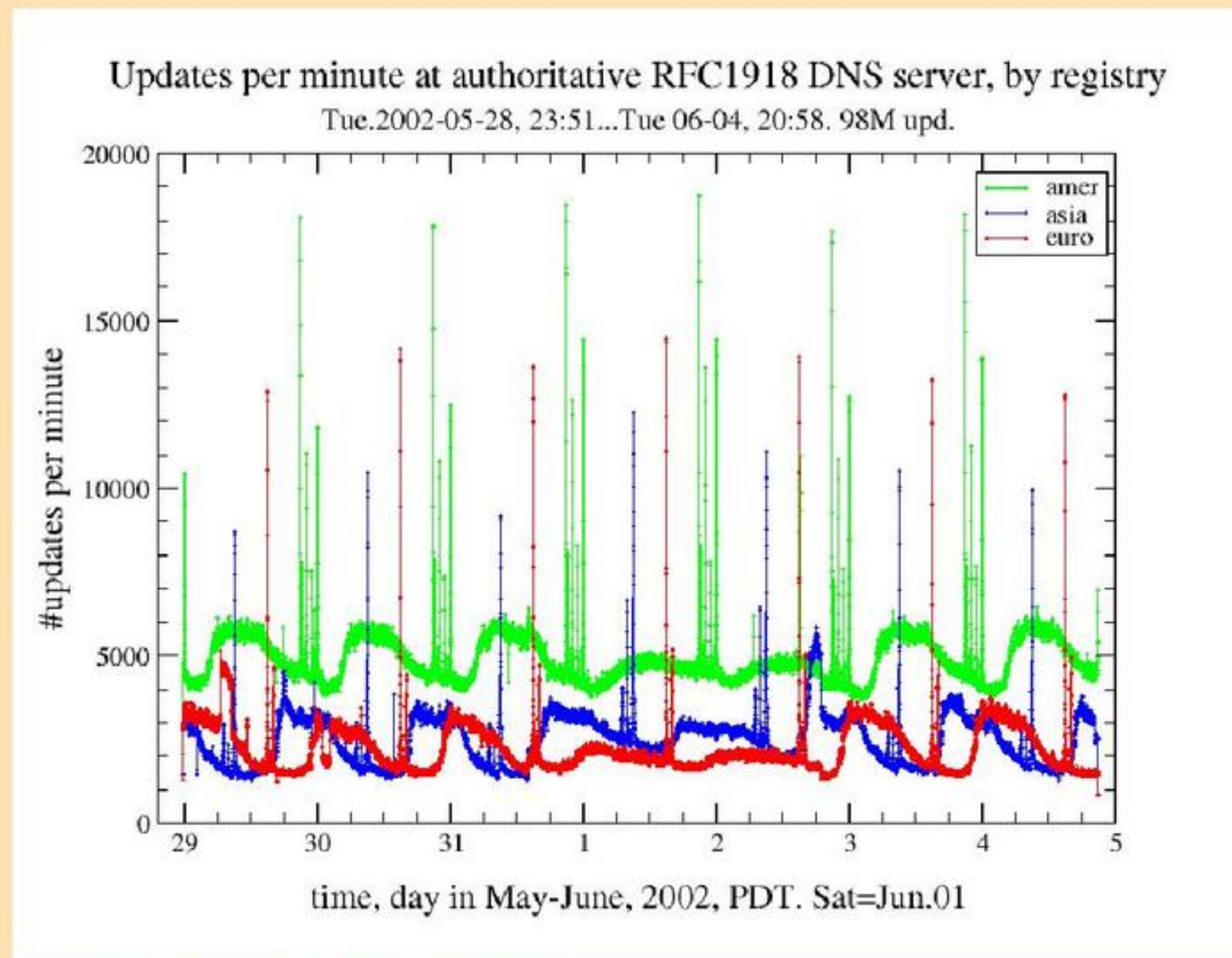
- log files from an authoritative RFC1918 (AS112 project) name server hazel
- bogus PTR record updates
 - attempts to modify a PTR record
- 51.4M updates in 86.5 hr = 10,000 per minute = 165 per second

Private addresses (cont.) - workload



- 192.168.0.0/16 is the most popular in networks using cable modems and DSL connections

private addresses (cont.) - by continent, time

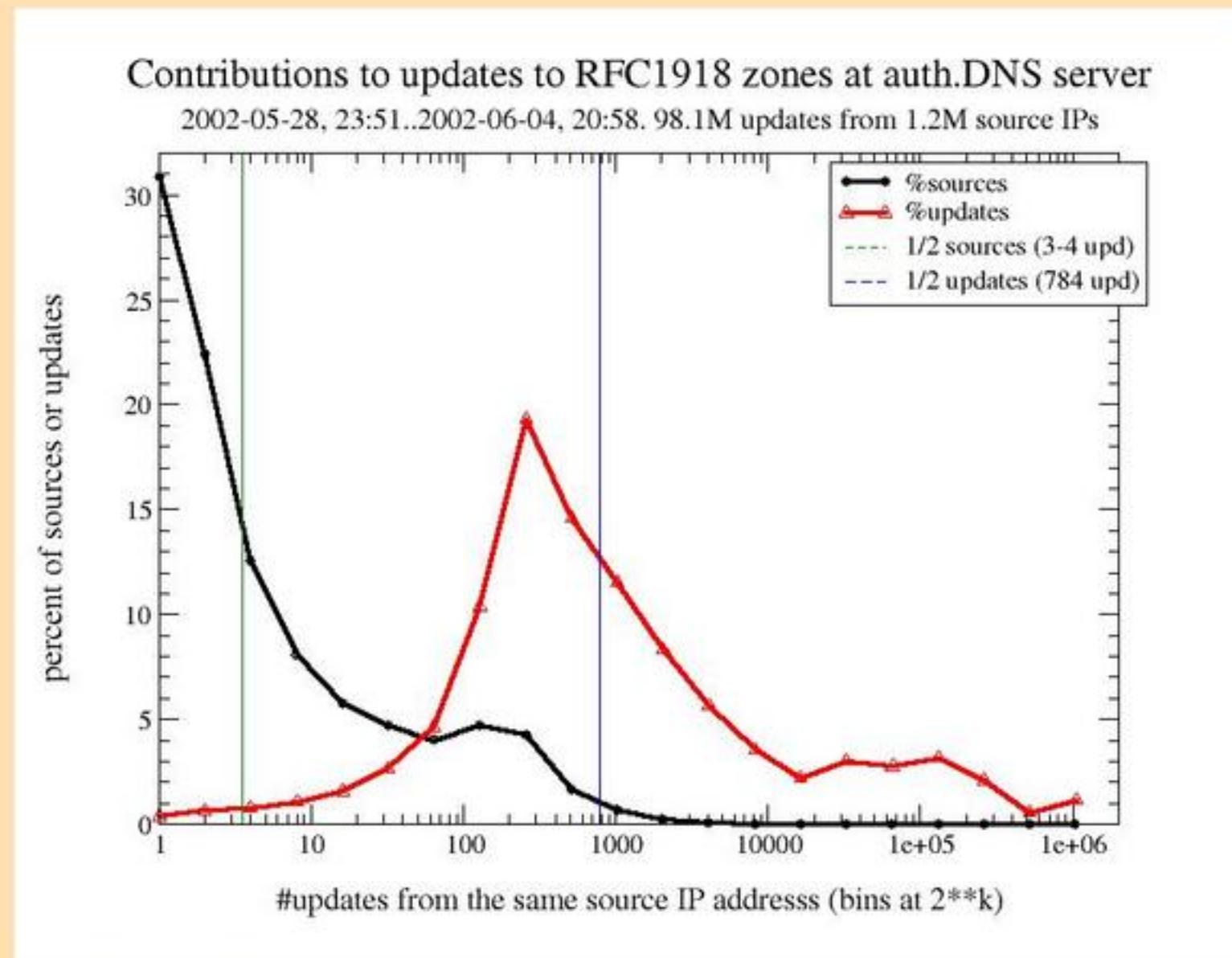


- clear diurnal patterns (by time zones)
- sharp peaks at midnight of each time zone
 - ▶ hypothesis: expiration/renewal of DHCP leases?

private addresses (cont.) - by ASes

- clients belong to 3309 origin ASes
- 20 ASes cause more than 50% of RFC1918 PTR updates
- top offenders:
 - 4134 Chinalink (China)
 - 3352 Ibernet (Spain)
 - 7132 SW Bell (USA)
 - 5673 Pac Bell (USA)
 - 5676 Pac Bell (USA)
 - 4813 China Telecom (Guandong, China)
 - 4812 China Telecom (Shanghai, China)
 - 852 Telus (Canada)
 - 6128 Cablevision (USA)
 - 2828 XO (USA)
 - 1142 Road Runner (USA)
 - 7843 Adelphia (USA)
 - 4760 Netvigator (Hong Kong)
 - 2914 Verio (USA)
 - 1221 Telstra (Australia)
 - 11509 Pajo (USA)
 - 4436 SantaCruz Community I't (USA)
 - 11426 Road Runner (USA)
 - 10994 Time Warner (USA)
 - 2548 Business Internet (USA)

who contributes most of RFC1918 workload?



- a week of RFC1918 PTR upates
- bulk are from hosts sending btw. 256 and 512 updates per week
 - (andre: neither mice, nor elephants - but `workhorses')

private addresses (cont.): identifying OSen

dynamic probing of offending addresses

- used xprobe utility
- very limited: a few samples, 100 to 500 addresses each
- no dominant operating system found
 - mixture of Windows- and Unix- based OS
 - no Apple systems....
- need better diagnostic tools

IV. evaluation/optimization of server placement

(bradley huffaker, caida/u.auckland)

■ 13 root servers

- 10 in US (6 in Washington DC, 4 in California)
- 2 in Europe
- 1 in Asia

is this arrangement optimal?

■ are some servers redundant?

■ are more servers necessary?

■ how to determine best root server location?

- politics
- prestige
- control

IV. server placement (cont.)

macroscopic topology measurements

■ CAIDA skitter tool

- <http://www.caida.org/tools/measurement/skitter>
 - ▶ traceroute-like methodology
 - ▶ increments Time-To-Live (TTL)
 - ▶ ICMP echo requests
 - ▶ small (52-bytes) probe packets
 - ▶ slow-paced

■ probes measure

- IP forward path information
- round trip time (RTT) to destination
- thousands of destinations

■ resulting data

- hundreds of thousands of paths per day, for years
- most comprehensive macroscopic Internet topology data

IV. server placement (cont.)

skitter measurements for dns root servers

- 11 (out of 13) root servers instrumented w. skitter monitors
 - J co-located with A
 - C has not responded
- measures ICMP RTT from skitter to target destinations
 - not actual DNS response time
 - characteristic of infrastructure
- common destination lists for all monitors

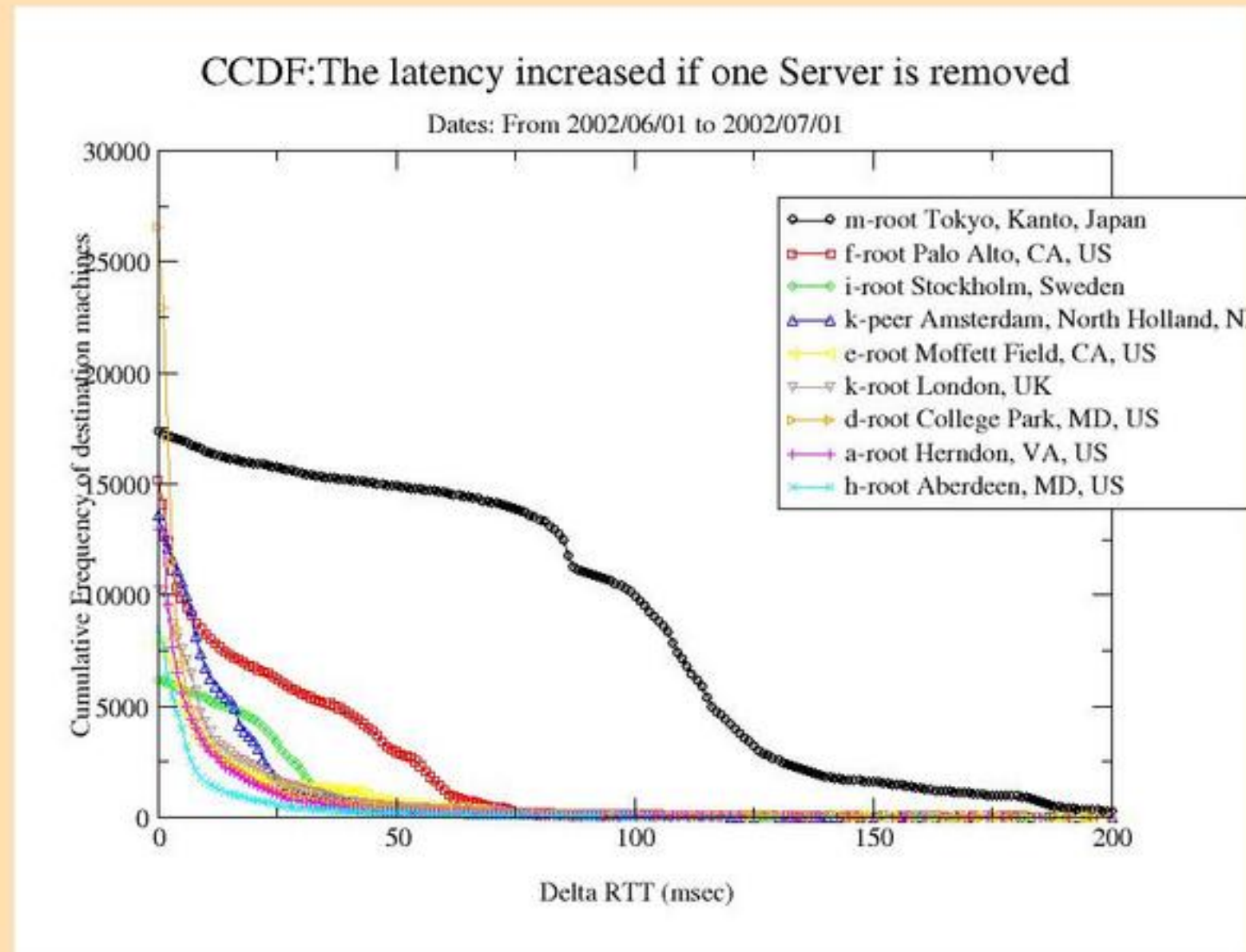
=> dns clients list

IV. server placement (cont.): measurements

dns clients list

- Goal: **representative** list to run on all skitter monitors
 - combine individual clients lists from all root name servers
 - stratify IPv4 address prefix space
 - prefix - independently routable slice of address space
 - no more than 150K destinations
 - so that we can probe 3-5X/day (less sensitive to diurnal variations)
- current *dns clients* list created in March 2002
 - passive collection of addresses at 7 root servers
 - select one host per routable /24 prefix
 - prefer hosts seen by most root servers
- nearly 2M addresses passively collected from root servers
- selected more than 143K addresses
- cover about 50% of prefixes from the global BGP table

IV. server placement (cont.): 'remove one'



- m-root is most crucial server
- f-root is second most crucial server

IV. server placement (cont.)

distance btw. a pair of root servers - definition

- skitter monitor is co-located with each root server
- select a subset of destinations responding to both skitter monitors

distance = the average absolute difference btw. median RTTs

- short distance $\Leftrightarrow$ similar RTT distributions for destinations
- cluster root servers based on their virtual proximity
=> "root families"

IV. Servers' placement (cont.)

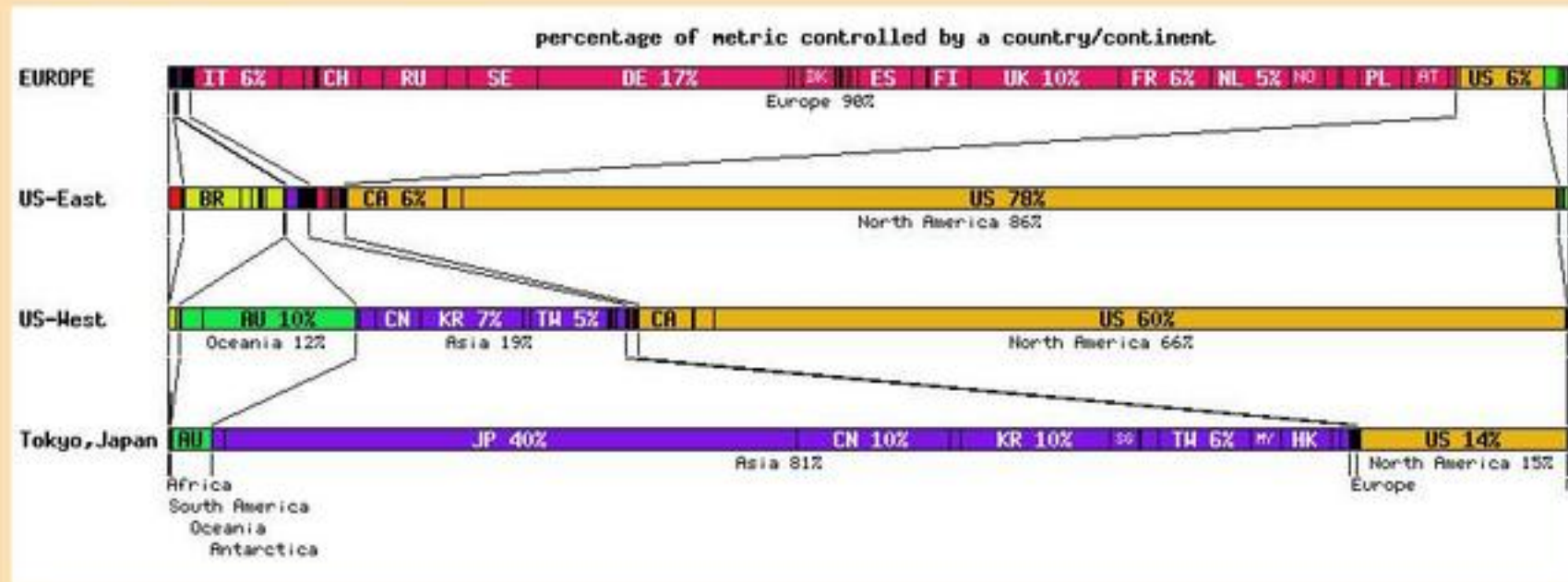
clusters of root servers

	GROUP 1 EUROPE			GROUP 2 US-East				GROUP 3 CA US-West			GROUP 4 Tokyo Japan
	k-root	i-root	k-peer	a-root	g-root	h-root	d-root	f-root	e-root	b-root	m-root
k-root	0	89	105	124	127	125	125	146	138	162	207
i-root	89	0	130	144	158	152	155	174	164	188	218
k-peer	105	130	0	173	164	169	167	186	178	198	233
a-root	124	144	173	0	68	67	64	109	105	129	209
g-root	127	158	164	68	0	64	61	108	103	121	202
h-root	125	152	169	67	64	0	59	94	89	119	195
d-root	125	155	167	64	61	59	0	101	95	120	202
f-root	146	174	186	109	108	94	101	0	65	86	170
e-root	138	164	178	105	103	89	95	65	0	90	165
b-root	162	188	198	129	121	119	120	86	90	0	189
m-root	207	218	233	209	202	195	202	170	165	189	0

- clusters correlate with geography remarkably well
- servers within each cluster are functionally equivalent
- more at the bottom of <http://www.caida.org/projects/dns-analysis/>

IV. server placement: nameservers by geography

clusters of root servers



- 2/3 of NA dsts have lowest mRtts to servers in US-E family, other 1/3 to US-W
- majority of European dsts best-served by Europe family, with some in US-E
- a few Asian dsts favor Europe and US-W family
- Oceania prefers US-W family
- note data missing from 3 root servers but likely not result-changing

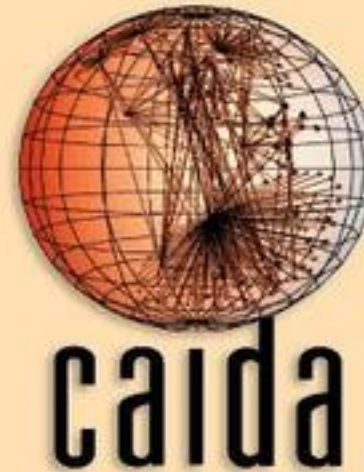
conclusions

- a ton of damage in the root system
- much more dns analysis on caida web site

www.caida.org/projects/dns-analysis/
www.caida.org/outreach/papers/
www.caida.org/cgi-bin/dns_perf/main.pl

- this talk
www.caida.org/outreach/presentations/dns200209/
- lot more to study than cycles to study it
- please send mail if you want to offer monitoring site or analysis cycles (students)

contact info



*the purpose of models is not to fit the data
but to sharpen the questions.*

Samuel Karlin, Samuel Karlin,
11th R A Fisher Memorial Lecture, 20 April 1983.

k claffy
ucsd/sdsc/caida
kc@caida.org
www.caida.org