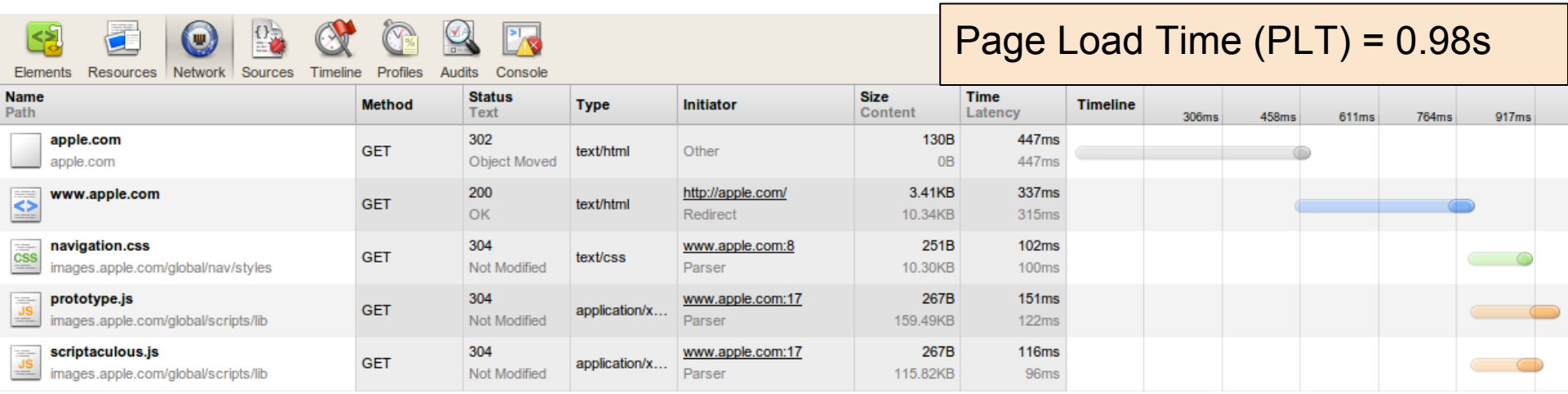
Improving Web Performance on Mobile Browsers

bengr@google.com

Desktop browsing is fast, relatively speaking.

- Desktops and laptops are over-provisioned to display web pages
  - The have GBs of RAM, fast processors, fast networks
- The experience is homogeneous
  - Pages load in a small number of seconds (3-4s average in US, 6 s worldwide)



- We know we want subsecond loads, and desktop is close in some markets

Mobile Web performance is bad: Pages don't usually load in under a second

Average mobile page load is 9s

http://www.nytimes.com/2012/03/01/technology/impatient-web-users-flee-slow-loading-sites.html?pagewanted=all&_r=0

Often much longer.

Mobile is ~10X behind

- ~10X less processing power
  - Sun Spider Javascript Benchmark (lower score is faster)

Chrome on 2.5 GHz quad core desktop	295
Android Browser on Galaxy Nexus	1988

- ~10X less memory (256 MB vs 2G)
- ~10X slower network
  - 3G vs cable, LTE vs FIOS in high-end markets, EDGE vs DSL

Why do pages load slowly?

There are several smoking guns:

For some page loads, high RTT is the bottleneck.

For others, slow transfer rates.

For still others, limited CPU has the dominant impact.

Implication

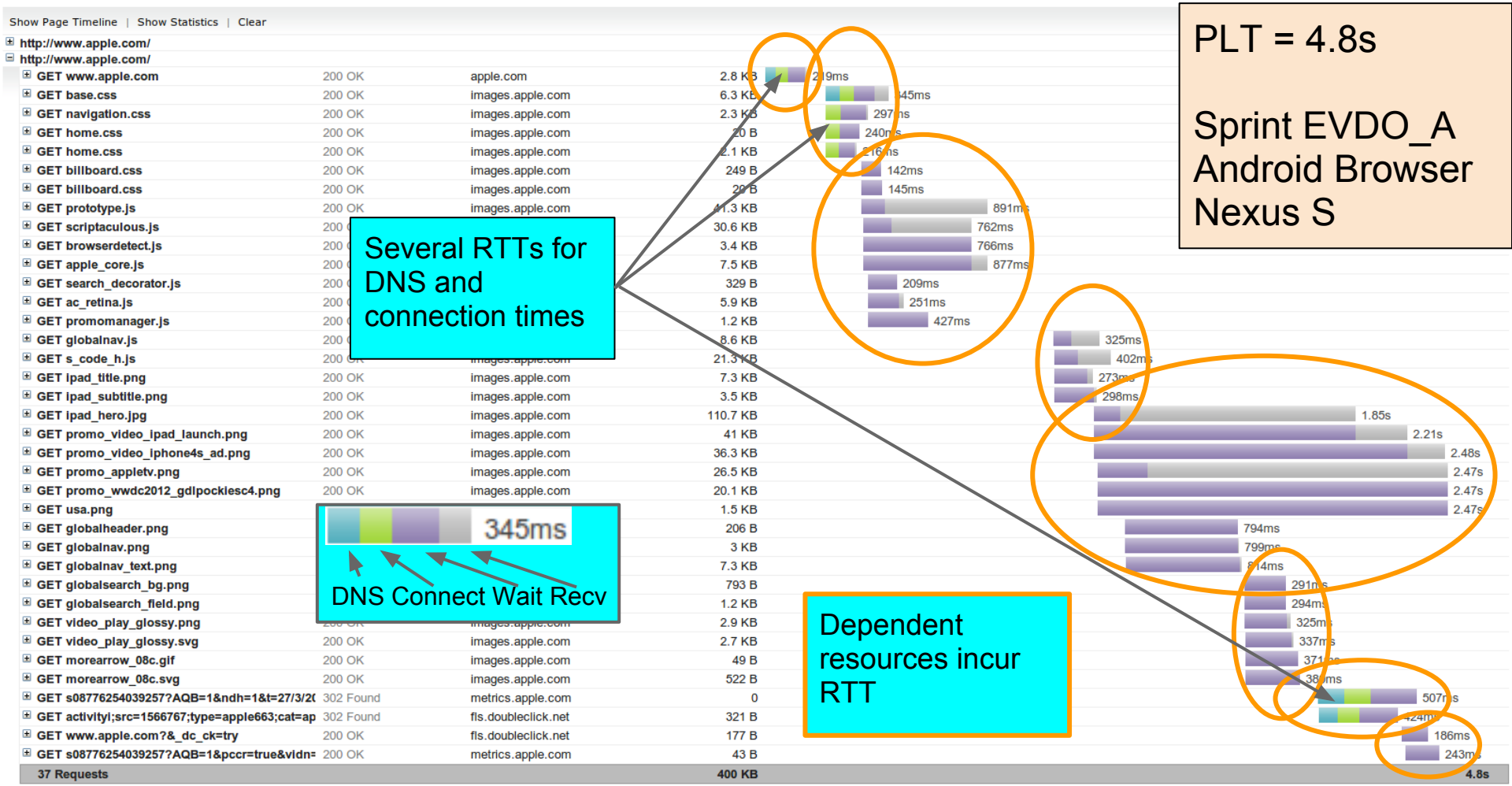
- Need to work towards significant improvements across all of these

First, latency.

- High Latency is the often cited reason for poor mobile performance
  - World averages: 2.4Mbps, 280ms RTT
  - USA Averages: 3.2Mbps, 240ms RTT
  - For reference: 134ms for light to circle the equator
  - Speeds are disproportionately higher than RTTs

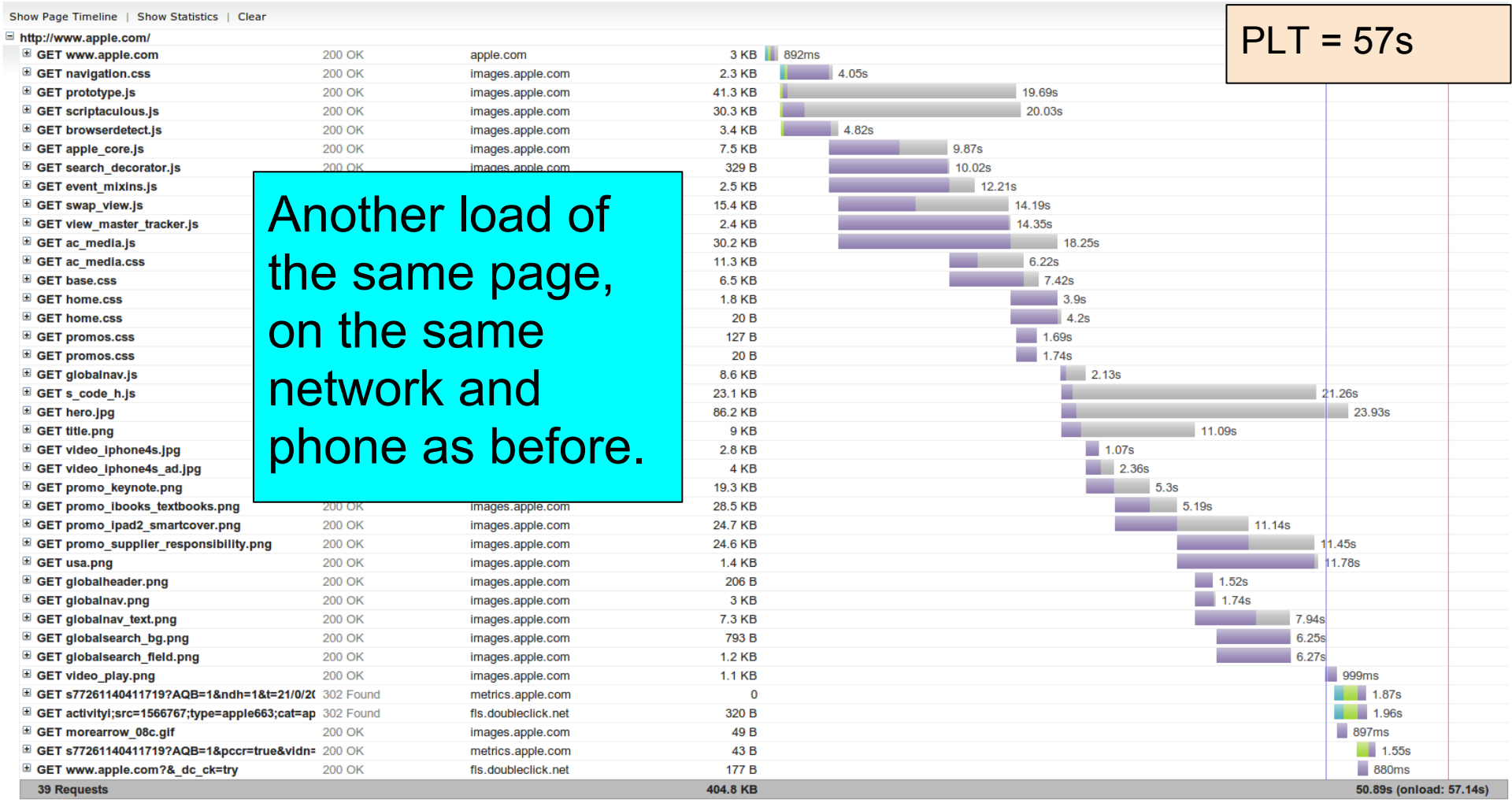
Source: 17M records from speedtest.net, mid 2011

RTTs accumulate (and the network is a little slower). Sounds plausible.



Source: Google testbed archives

Latency can't be the only issue.

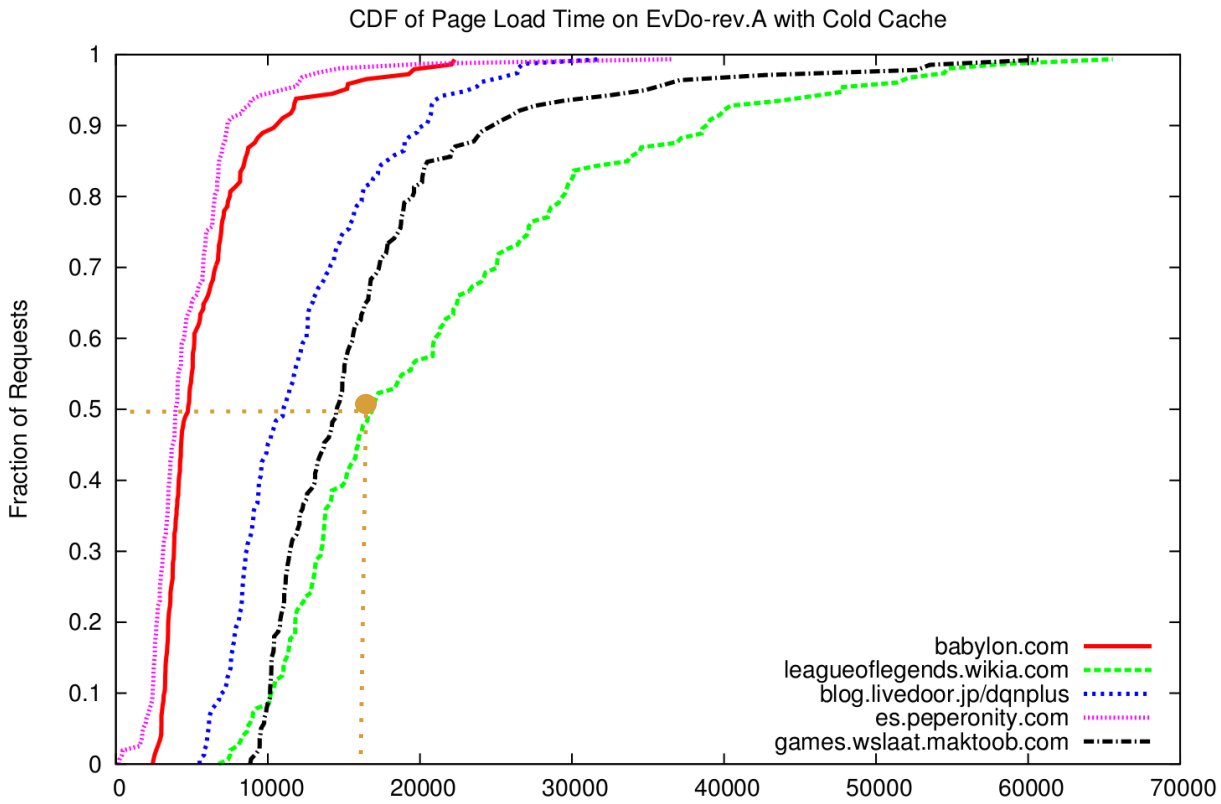


Mobile web performance is highly variable

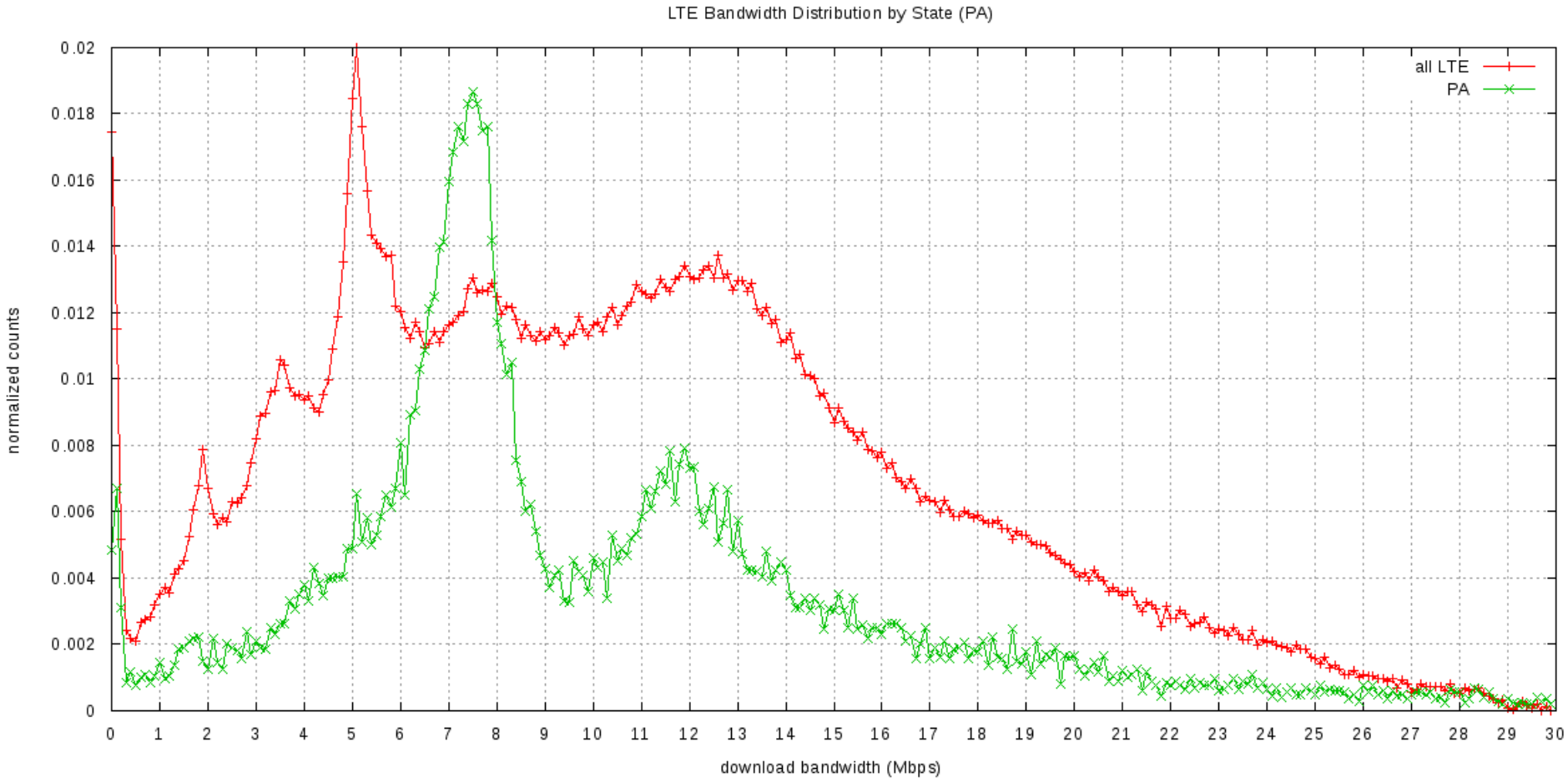
PLTs vary over an order of magnitude

- 3-15s is typical

Lots of undesirable minute-long loads or mobile



Available bandwidth exhibits a wide distribution. Often low. Location matters.

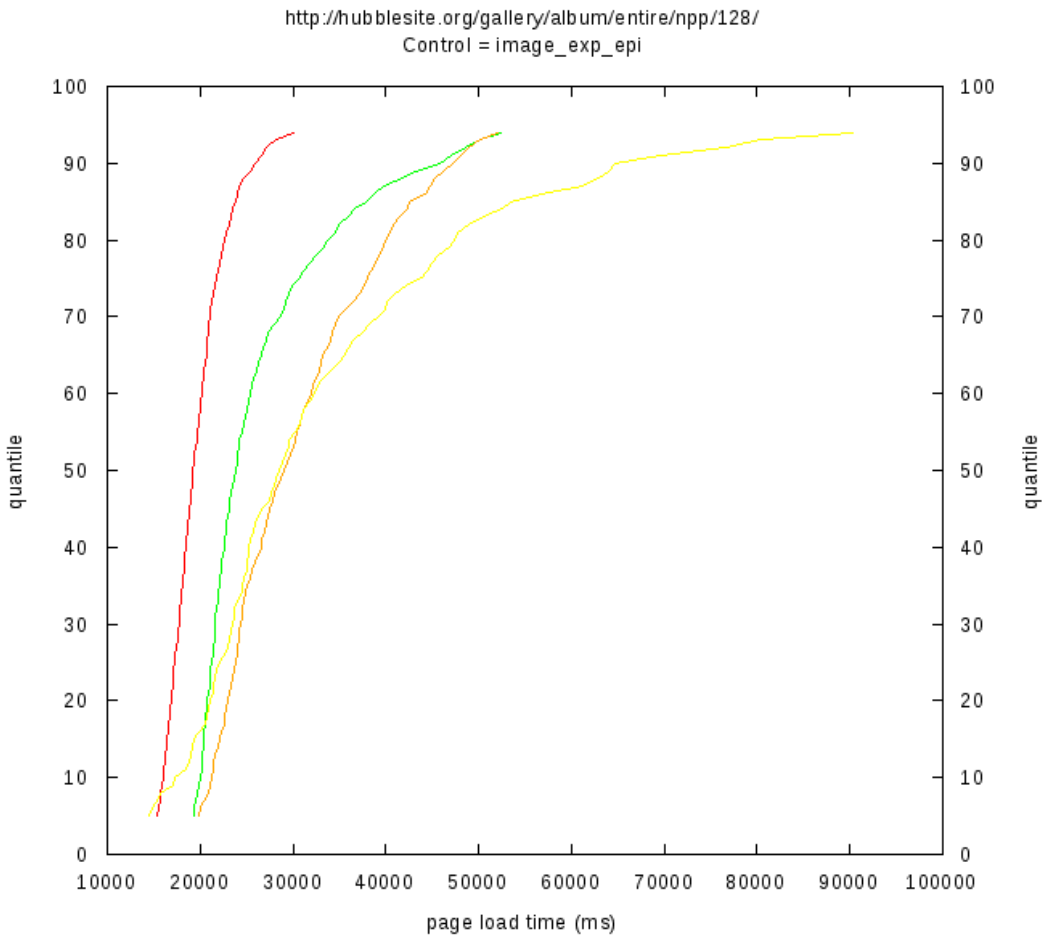


Even small changes to location affect bandwidth.

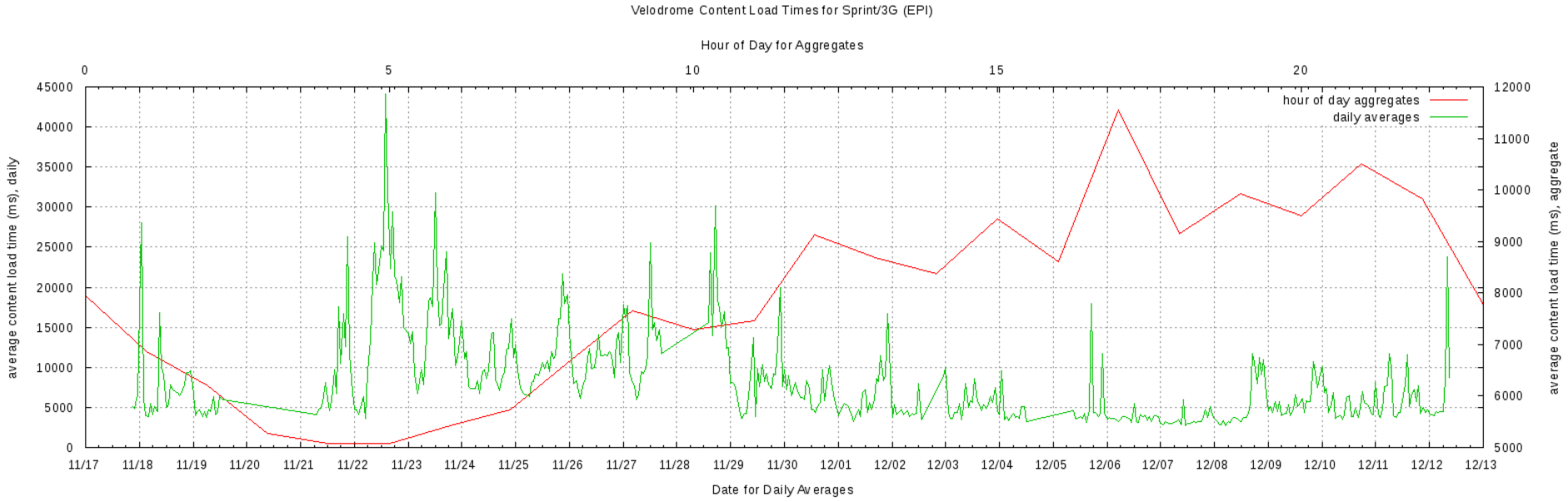
CDFs of page load time at several locations in and near our office

Performance

- Best in location with line-of-sight to cell tower (leftmost curve)
- Worst inside office (rightmost curve)
- Curve in middle is inside office with signal booster



Time of day affects bandwidth.



Performance is different for different carriers.

Carrier	Download (Mbps)	Upload (Mbps)	Latency (ms)
Verizon	6.403	2.209	174
T-Mobile	3.098	0.86	220
AT&T	2.004	0.707	275
Sprint	1.852	0.523	268
Virgin Mobile	0.612	0.299	386
DTAC (Thailand)	0.147	0.073	929
Vodafone	0.734	0.377	727

All values are averages.

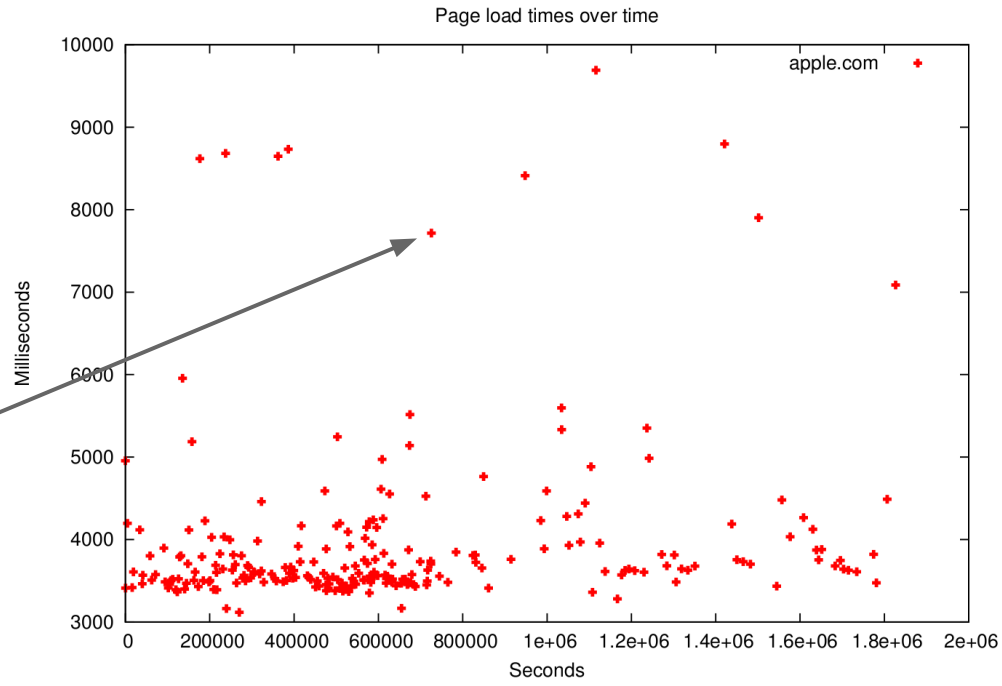
Source: analysis of speedtest.net data, mid 2011.

Huge network variability makes it difficult to characterize page load performance

Different individual loads have different

- Transfer times
- RTTs
- Both

For a particular page load, e.g., this one, network variability may dominate overall performance



The CPU on the phone is sometimes the bottleneck

The CPU a very limited resource on mobiles

Example:

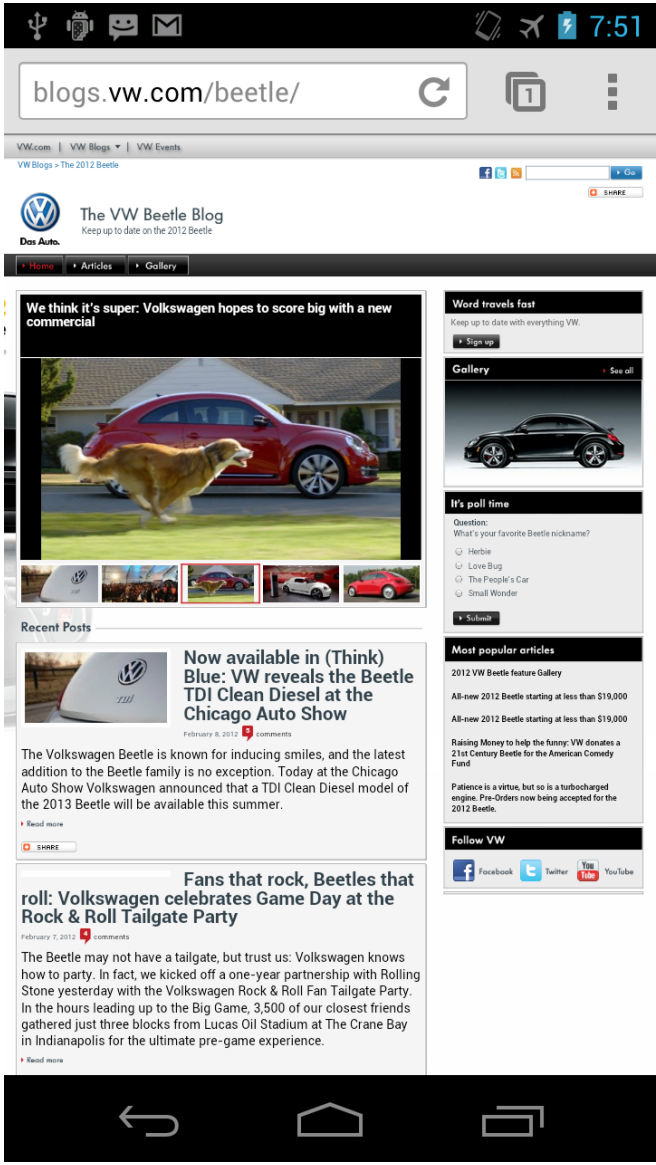
- http://blogs.vw.com/beetle
  Galaxy Nexus (1.2 GHz dual core Arm cortex A9)

Page Size:

- Unmodified page 1022KB
- Minified page: 566KB (45% smaller)

PLT: 9.4s in both cases

- Web inspector confirmed executing JavaScript was the bottleneck

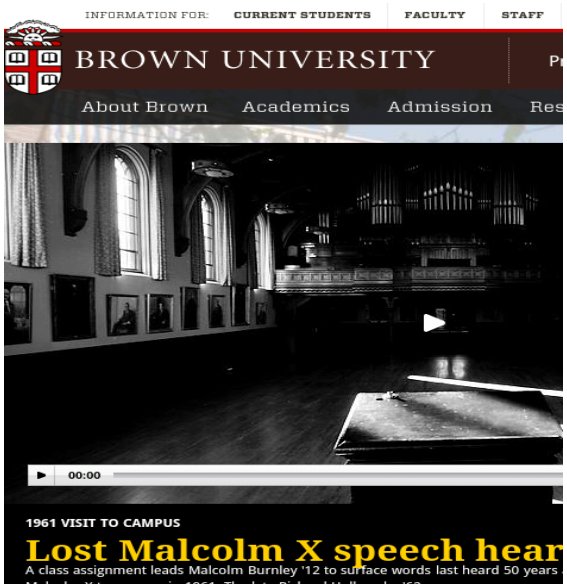


Sometimes it's easy to find performance issues.

E.g., when

- gzip is off
  - 20% of Alexa-1000 HTML/CSS/JS isn't compressed
- Resources have no cache-control headers
  - 57% of resources don't have cache-control headers
- Resources are much larger than necessary
  - e.g.,

124 KB of background image that is completely obscured



But it is usually difficult to characterize how a Web site will perform in practice

- For site developers
- For performance engineers like me

How can this community help?

Provide tools to measure page loads in practice

Need lots of measurements with broad coverage to characterize the distribution

- Across devices, locations, etc.
- With labels along various dimensions: bw, latency, cpu, etc.
- Lognormal, etc., are a great start

Combine the coverage of speedtest.net

- Tests lots of phones in the wild
- But only provides bandwidth and latency

With the measurement details of webpagetest.org

- Provides page load times

Provide techniques to inform origins of expected performance

What indications could we provide to inform of particular conditions?

- To be consumed by origin servers?
- Before a page loads?
- E.g., to shape content to prevailing network conditions?

Network quality (e.g. bandwidth, latency)?

- Might require better communication between radio and browser

Device resources (CPU, memory, available cache, screen size)?

Lots of ways to convey this information.

Help site developers diagnose problems

Currently, no 'right click' for deep network diagnosis

- State of the art (on android):
  - Chrome remote debugging + adb port forwarding
  - + roll your own devtools interpreter
  - + tcpdump
  - + Chrome netlog trace
  - + lots of script writing

Connect network resource use to browser operations

- E.g., resource loads to packets, link quality, etc.

I'd like to learn more about developers' specific needs.