Latency to the Eyeballs: A Holistic Approach

Todd Underwood, <toddunder@gmail.com>
John Van Oppen, <john@vanoppen.com>

Problem: Sh*t's Too Slow

- throw new BuzzWordWarning(new.String("cloud"));
  - https://cloudsleuth.net/global-provider-view
  - Summary: sample app takes >10s on most "clouds"
- Users get distracted after 1s and leave after 4-6s.
- Network:
  - Oh my goodness, the first IP hop is 60 ms away from my house but only 5 miles away... Yikes!
  - Is this the problem?
- Review the contributors to latency, stack rank them, and solve the most serious ones.

Real Contributors to Latency

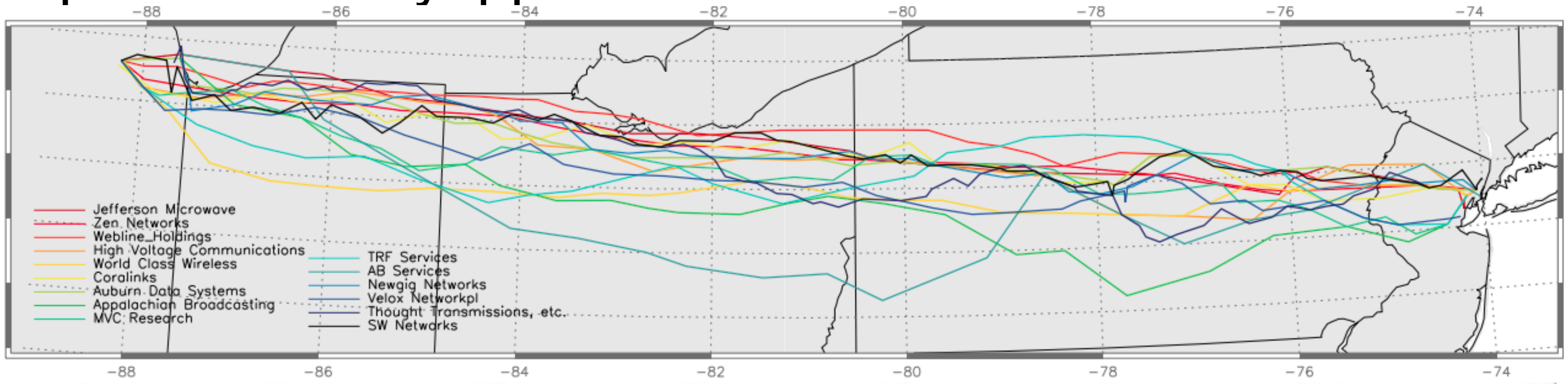
- Network design, traffic engineering
- CPE/Last Mile
- Bad Geolocation
- Crappy Web 2.x Ruby on Rails on MongoDB to serve a static file nonsense sites
- Insufficient RAM on servers
- Fiber Propagation Delay (index of refraction)
- c[1]

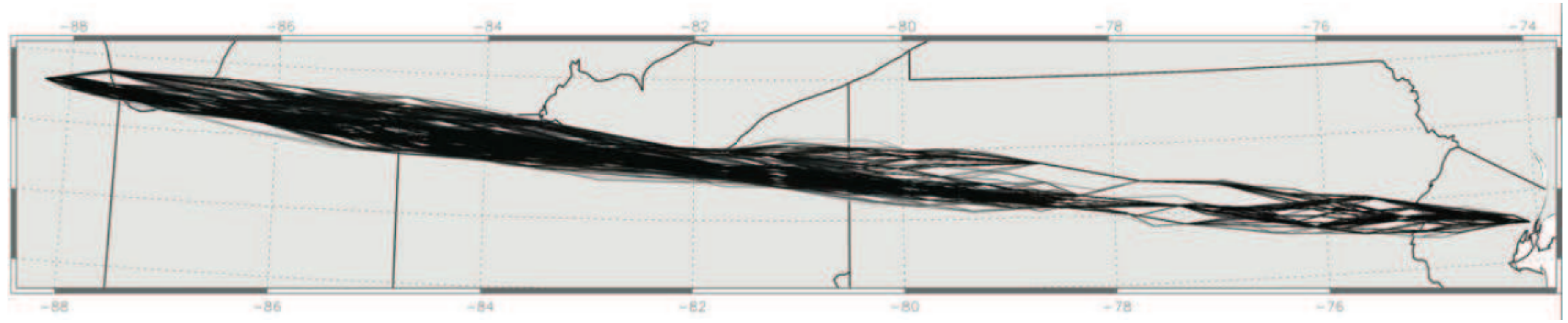
Let's address these in turn.

[1] We'll get to this

Usually Not the Problem: Your Network

- We are network engineers. We should fix the network.
- Sometimes (very narrow situations) this makes sense:
- http://www.windyappletech.com/[1]

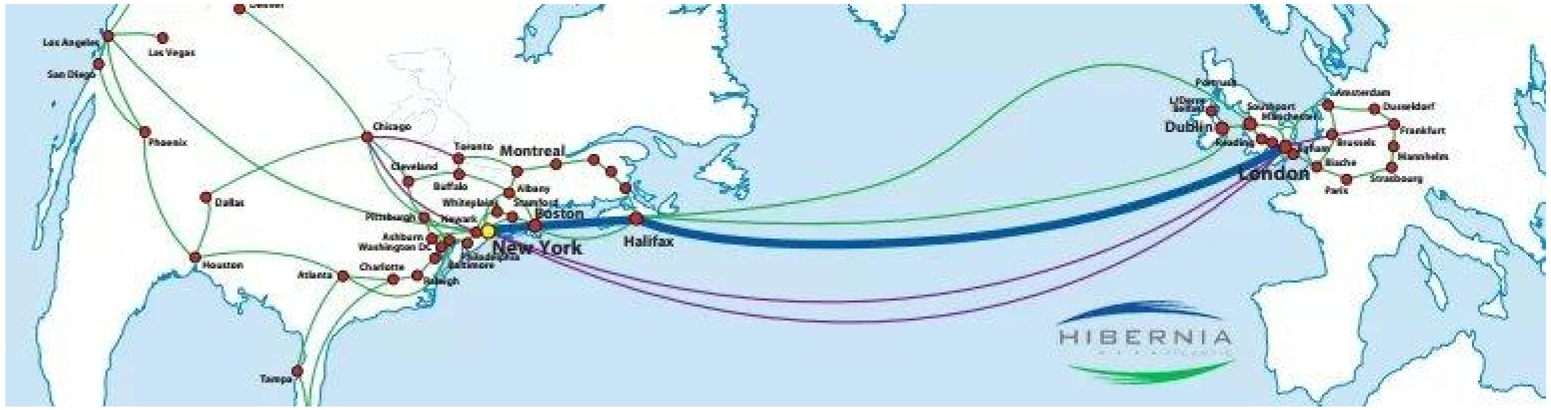




[1] http://arxiv.org/pdf/1302.5966.pdf

Usually Not the Problem: Your Network

- $80m per ms saved across the Atlantic[1]:



- Worth it?
- What is the problem being solved here?
- Is there any more problem to fix?

[1] http://www.igvita.com/2012/07/19/latency-the-new-web-performance-bottleneck/

Solution: CPE - the last mile

Two Problems:

- Interleaving/error-correction
  - Is the line quality so bad that we need this much error correction?
- Buffering
  - What the hell is that 15ms buffer *for*?
  - Great data on how terrible this is in mobile networks as well

Solution: everyone roll out non-stupid CPE right away, please.
Ancillary benefit: Proper, modern, universal v6 support

Solution: Fix Geolocation

Occasionally users are mis-mapped to the wrong place.

Anecdata suggests this is rare. Consequences can be significant, however.

Geolocation is hard. Let's go shopping (or move on).

Solution: none. We're punting on this one.

Solution: Fix Web Technologies

Problem: Your web 2.x site is badly built.

Solution: Don't do that.
Related Solution: Cache a bunch of stuff in RAM (but don't use RAM as a capacity cache or you cannot cold start)
Related Solution: Don't use leaky/unreliable VMs
Related/related solution: Treat web app architecture seriously.

Solution: Fix Propagation Delay

- Can we do better than an index of refraction of 1.468?
  - Vacuum core fiber?[1]
  - something else?

- SMF28 refractive index of 1.4682 (204,332 KM/sec), just slightly over 2/3 of C. (or 54ms round trip london to NYC on the most direct route possible with no slack or electronics)

- Some types of fiber can get to slightly lower numbers, but nothing dramatic (gaining as much as 10,000 KM/sec perhaps)

1. http://www.nature.com/nphoton/journal/v7/n4/full/nphoton.2013.45.html

Where Does This Get Us

- Starting: 10000 - 12000ms end user delays
- Fix networks: (5-25)ms usually
- Fix CPEs: (15-45)ms
- Fix Geolocation: nothing, man
- Fix web apps: (1000-5000)ms
- Improve propagation delay: (33)ms best case
  - (20000 km / c) - (20000km / (c / 1.468))
- Ending: 4000 - 6000 ms end user delays.
  - Most savings from fixing web app

Can't we do better?!?!?!

Thinking Big: Rethinking C

- Insufficiently bold:
  - End User Latency = $\text{Lat}_{\text{client}}$ + Lat_{edge}+Lat_{net}+$\text{Lat}_{\text{server}}$
  - Lat_{net} includes router, path, propagation latency
  - Latency bounded by (c/ 1.468)
  - Fix 1.468? Sure. Fine?
  - Better: increase c
- No one is seriously working on this yet
- For values of c_{new} in $\{2c_{\text{curr}} \ldots 10c_{\text{curr}}\}$
  - Server latency still matters
  - Improvements in edge and client latency no longer matter
  - Network latency factors out entirely.

Bigger c: Everything Changes

- We build things now based on:
  - cost (space,power,staff,connectivity)
  - location (dominated by proximity to other desired traffic sources/destinations, bounded by C/1.468)
  - geopolitical factors
- If we remove location as a constraint, we rethink to front end, back end and interconnections architecture of the entire Internet
- Limited only by power, space, geopolitical concerns.
- Out of time. Imagine. Think Bigger. Think Bigger c.

Questions?