Measuring and Mitigating Web Performance Bottlenecks in Broadband Access Networks

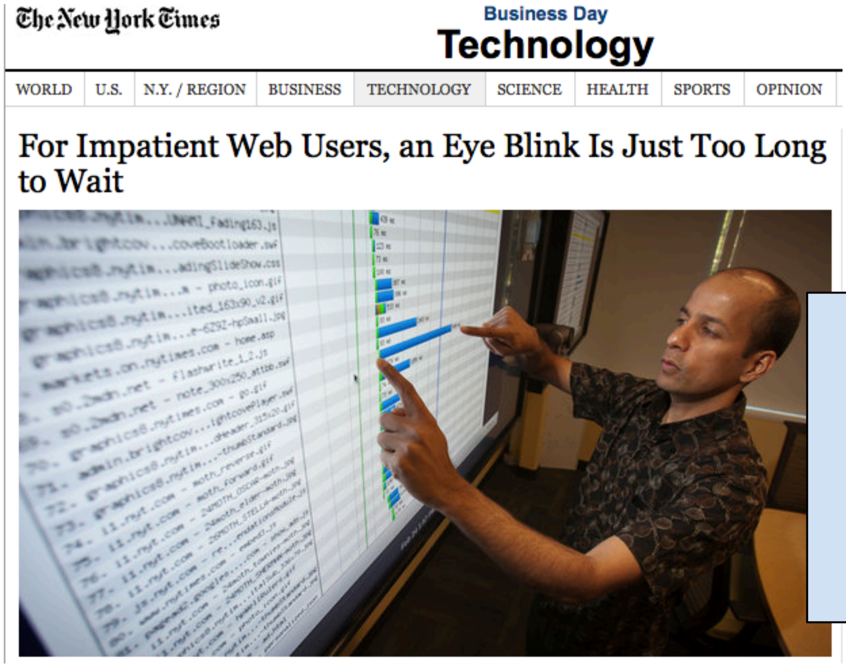
Srikanth Sundaresan, Nick Feamster (Georgia Tech)

Renata Teixeira (Inria)

Nazanin Magharei (Cisco)

http://projectbismark.net/

Every Millisecond Counts



For Impatient Web Users, an Eye Blink Is Just Too Long to Wait

Many Web services companies spend considerable effort reducing Web response time.

500ms delay causes 1.2% decrease in Bing revenue [Souders 2009]

400ms delay causes 0.74% decrease in Google searches [Brutlag 2009]

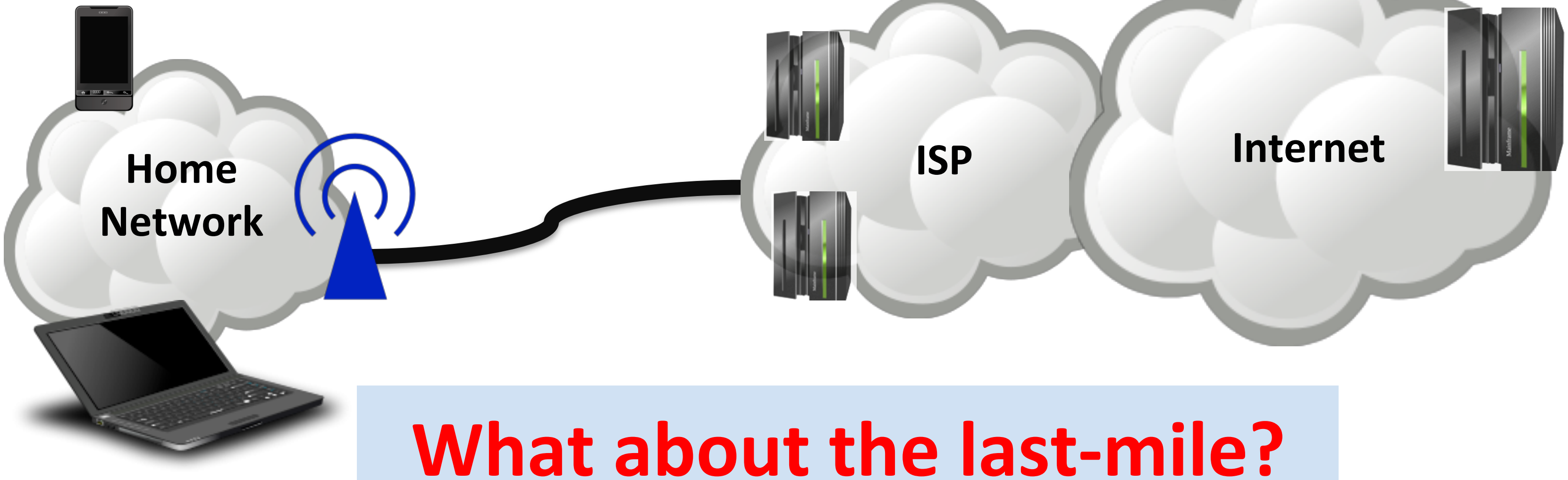
100ms delay causes 1% decrease in Amazon revenue [Linden 2013]

Many Performance Optimizations

Client-Side
- Browser caching
- SPDY/QUIC

ISP Edge
- CDNs
- DNS caching

Server-Side
- Persistent connections
- TCP ICW
- Content optimization



What about the last-mile?

The Last-Mile Affects Performance

- Last-mile latency can be significant
  - More than 50% of AT&T DSL users have last-mile latency greater than 20 ms [Sundaresan 2011]
- Optimizations are affected by last-mile performance
- The effects of the last-mile on Web performance has not been specifically studied

This talk: *Measure* and *mitigate* access link bottlenecks in Web performance

Two Contributions

- **Measure** last-mile Web bottlenecks
  - – 5000+ homes, from access point using (SamKnows)
  - – Latency is bottleneck beyond 16 Mbps

- **Mitigate** Web performance bottlenecks
  - – **Popularity-based pre-fetching** in the home
  - – DNS pre-fetching and TCP connection caching in the home can improve page load time by up to 35% (BISmark)

Measuring Last-Mile Effects on Web Performance

- **Challenges:** Instrumenting end-hosts
  - No ground truth: Every browser is different
  - Confounding factors affect measurements
- **Solution:** Measure from router
  - Piggyback on existing deployments (FCC, BISmark)
  - Consistent measurements – similar hardware
  - Continuous measurements – better characterization

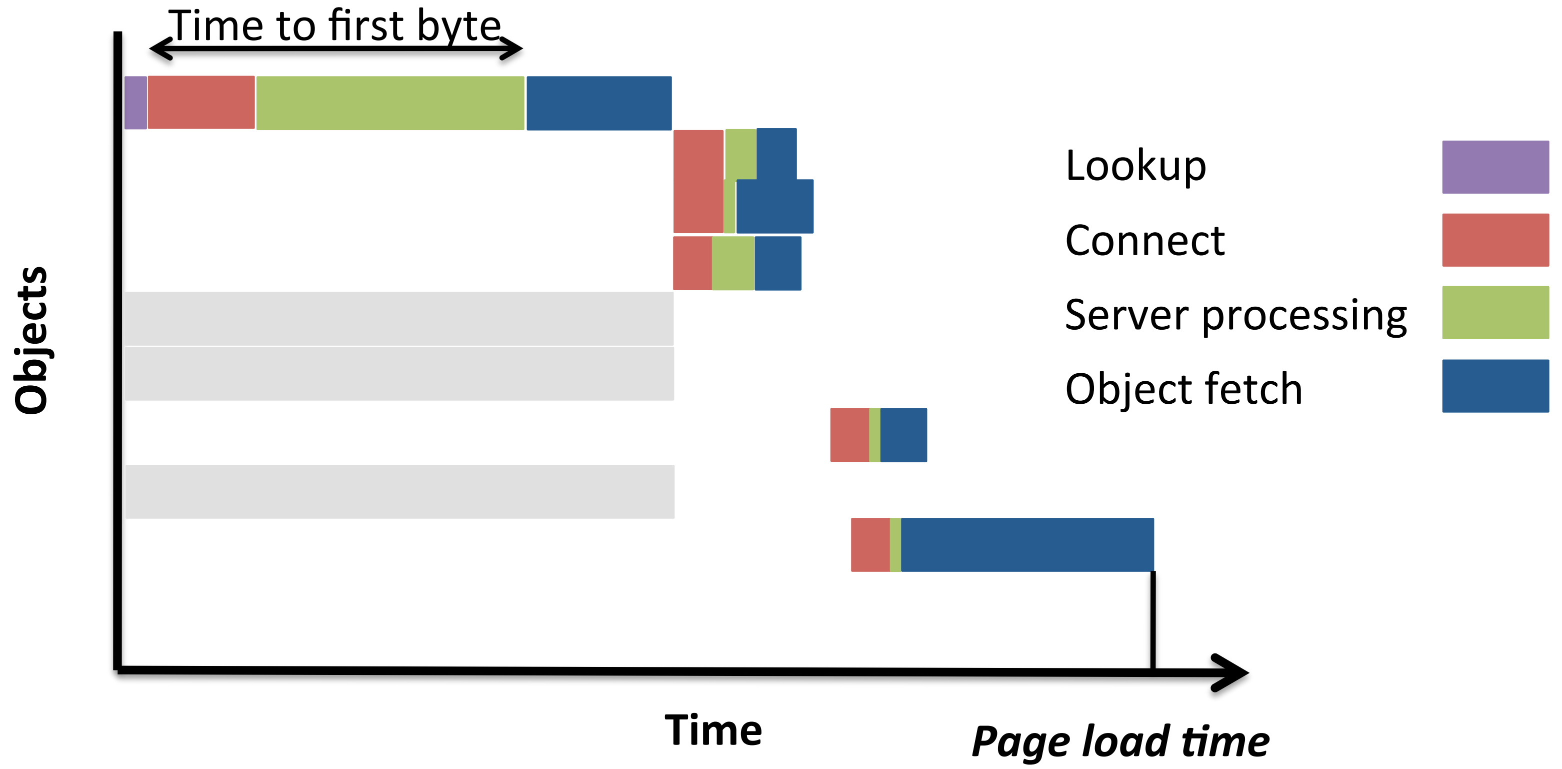
Mirage: Deployment and Data

- Emulates **basic browser function**
  - Estimates page load time
  - Breaks down fetch latencies
- **Deployed** by FCC/Samknows
  - 5,000+ homes in US: from 11 ISPs
  - Profiles 9 popular sites
  - Measurements are **publicly available** [FCC 2012]

SamKnows Deployment

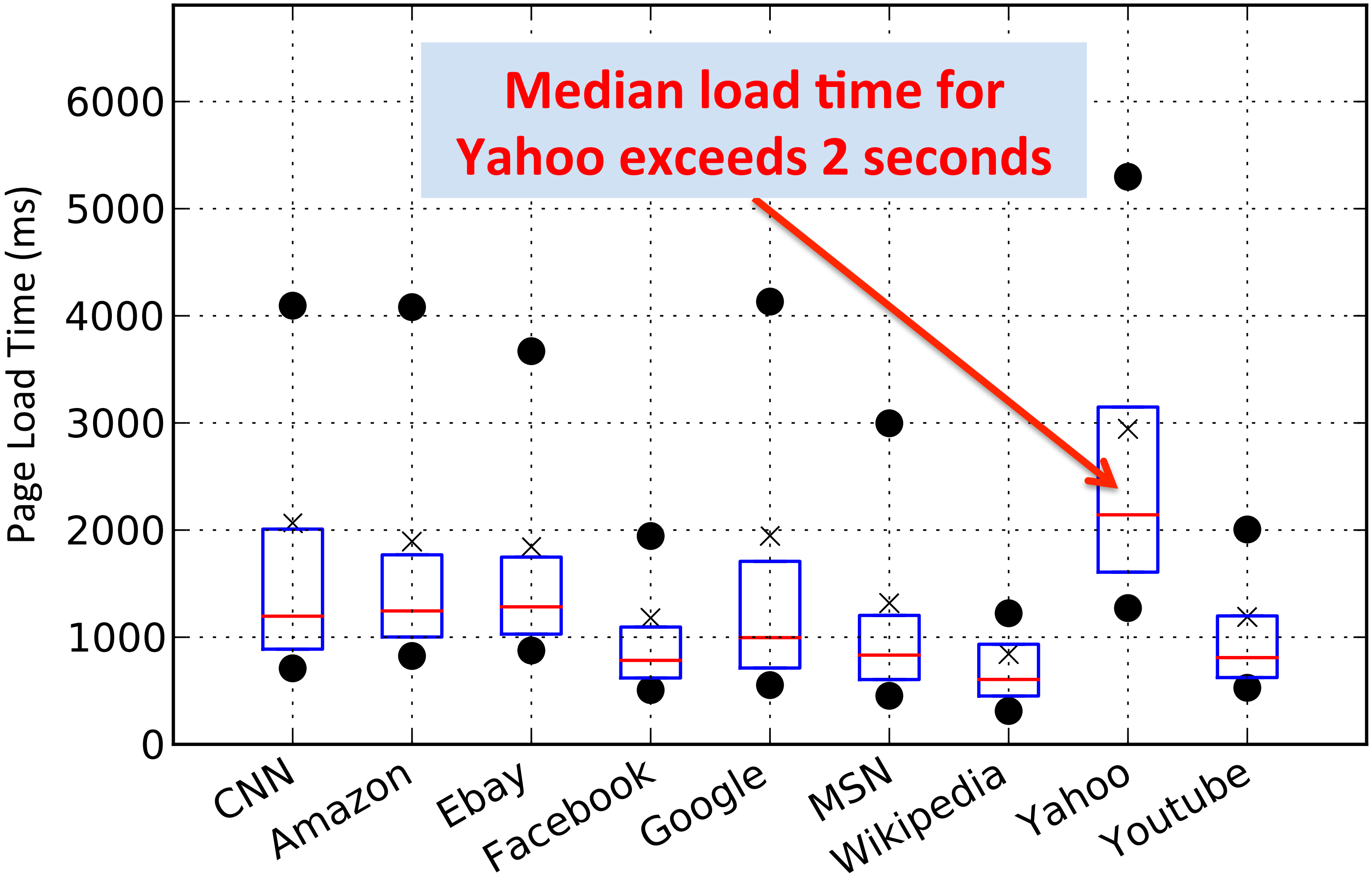
- 5,556 homes in US, Netgear 3500 routers
- 11 ISPs
  - AT&T, Cablevision, CenturyLink, Charter, Comcast, Cox, Mediacom, TimeWarner, Qwest, Verizon, Windstream
- 9 Websites
  - edition.cnn.com, www.amazon.com, www.ebay.com, www.facebook.com, www.google.com/mobile, www.msn.com, www.wikipedia.org, www.yahoo.com, www.youtube.com
- Data from September 2012

Mirage Identifies Latency Bottlenecks



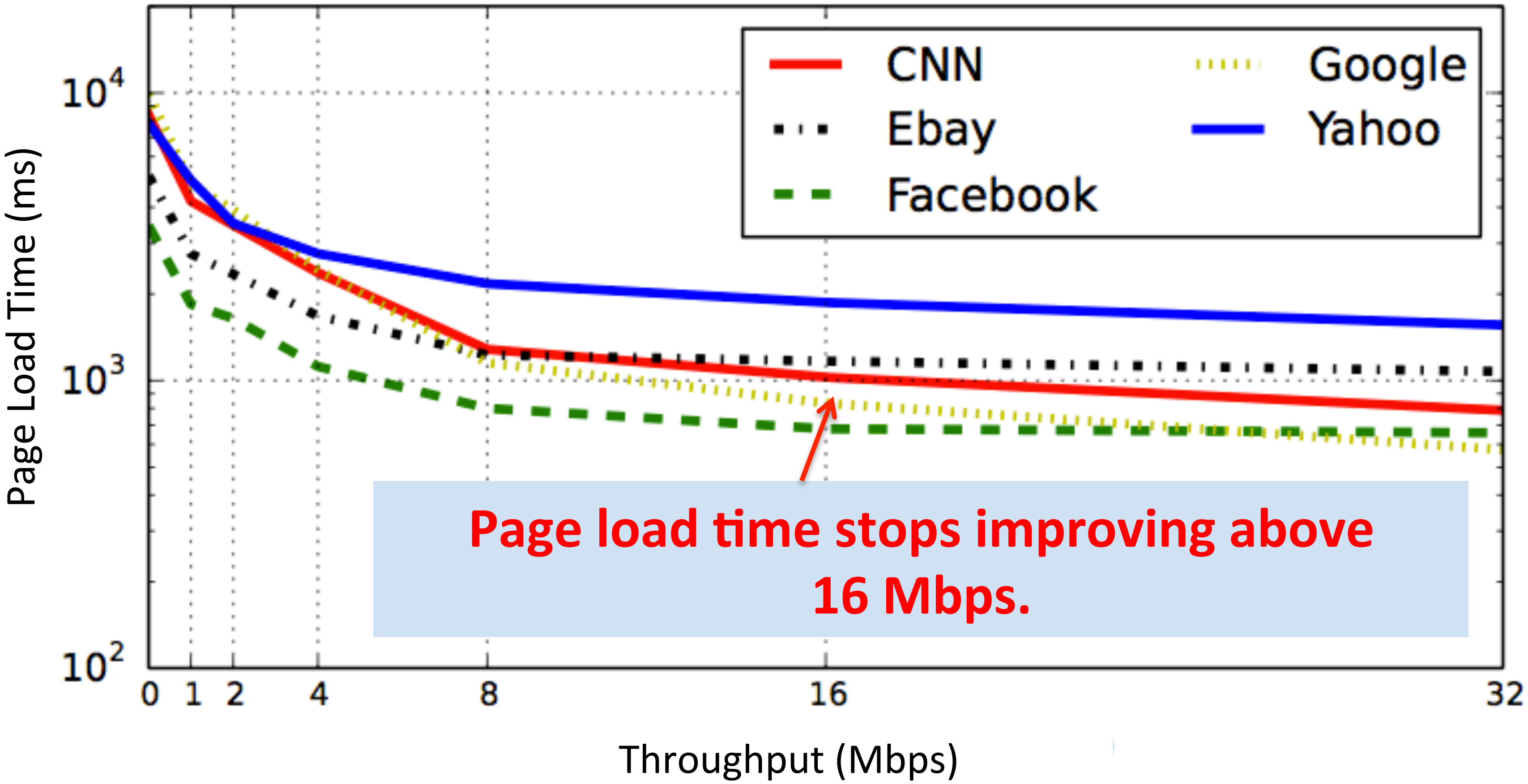
- Does not:
  - Resolve or fetch active objects
  - Establish ground truth (there isn't any)

Popular Sites Have High Page Load Time



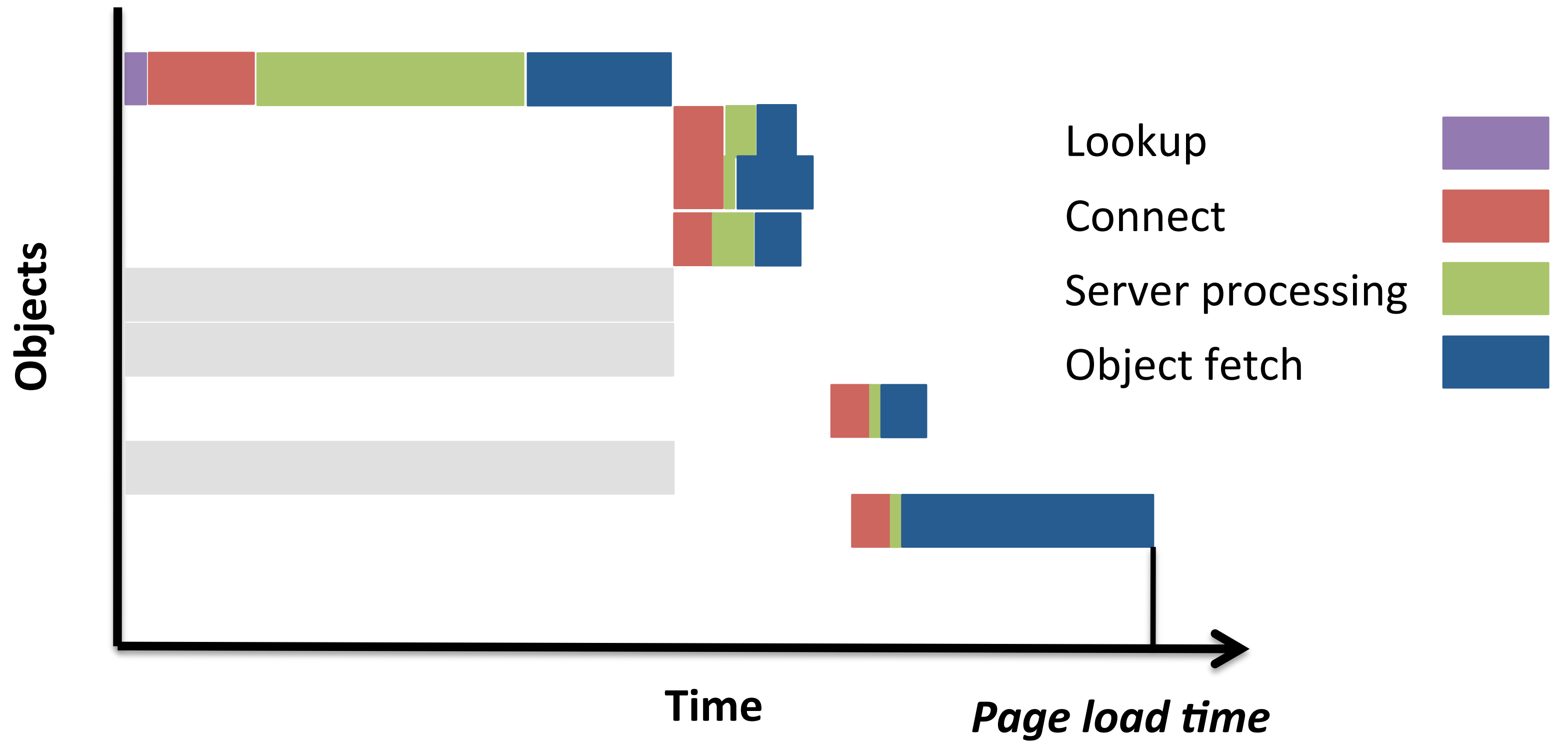
Median load time for Yahoo exceeds 2 seconds

Higher Throughput Doesn’t Always Help



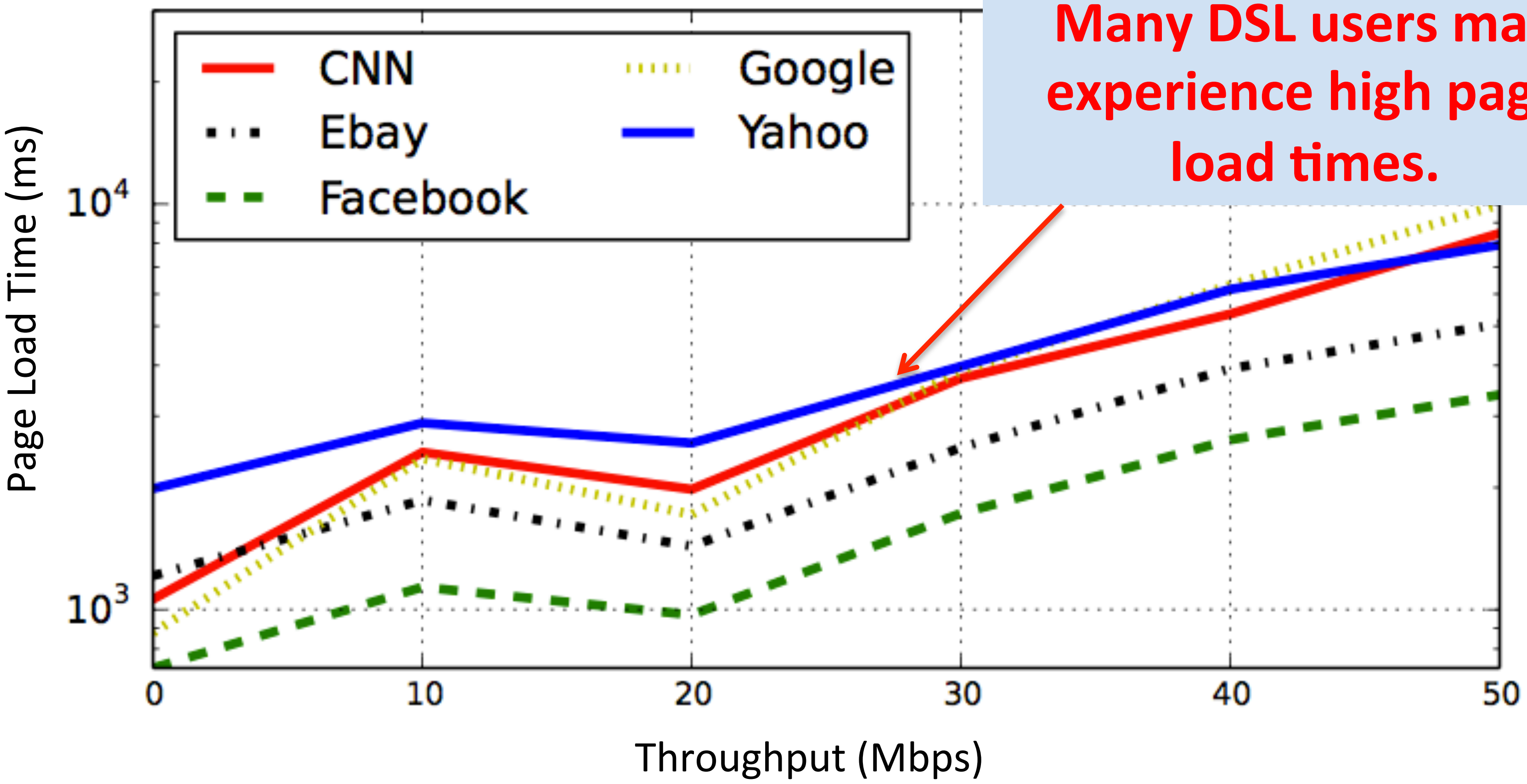
Page load time stops improving above 16 Mbps.

Mirage Identifies Latency Bottlenecks



Latency overhead can dominate fetch time, particularly for small Web objects

Common Last-Mile Latencies Result in High Page Load Times



Many DSL users may experience high page load times.

Two Contributions

- **Measure** last-mile Web bottlenecks
  - 5000+ homes, from access point
  - Latency is bottleneck beyond 16 Mbps

- **Mitigate** Web performance bottlenecks
  - **Popularity-based pre-fetching** in the home
  - DNS pre-fetching and TCP connection caching in the home can improve page load time by up to 35%

Many Optimizations Don’t Help in Last Mile

Client-Side

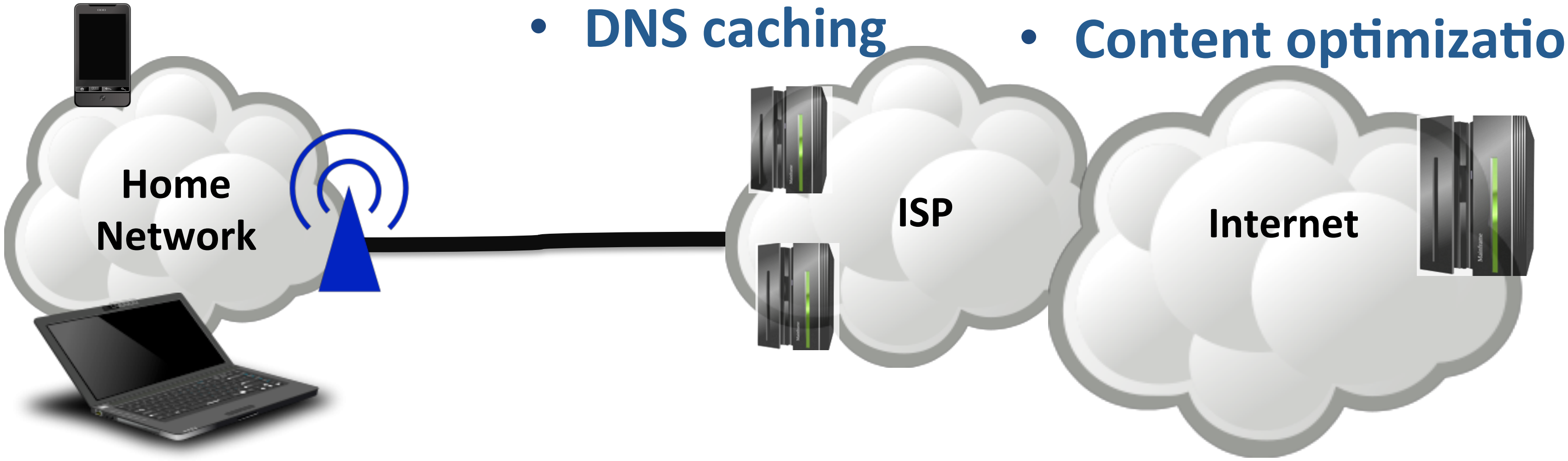
- **Browser caching**
- **SPDY/QUIC**

ISP Edge

- **CDNs**
- **DNS caching**

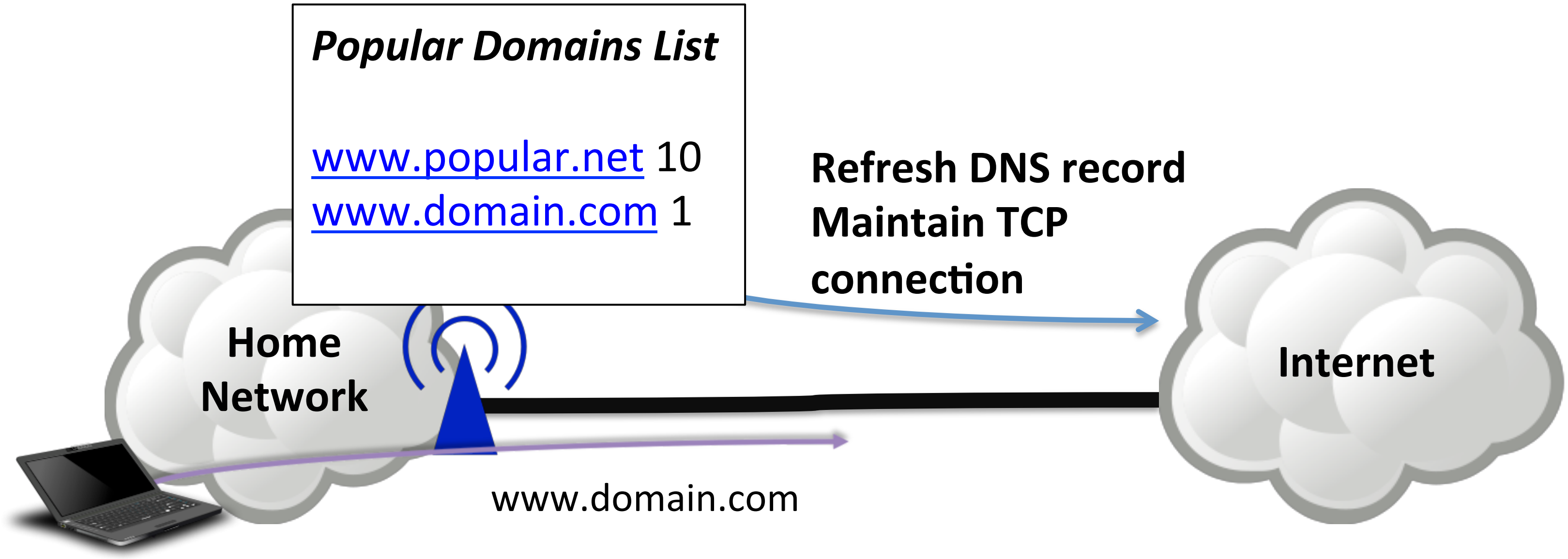
Server-Side

- **Persistent connections**
- **TCP ICW**
- **Content optimization**



- Many (CDNs, server-side) stop at the ISP edge
- Client-side optimizations are application-specific

Solution: Pre-fetching in the Home



Popular Domains List

www.popular.net 10
www.domain.com 1

Refresh DNS record
Maintain TCP
connection

Home
Network

Internet

www.domain.com

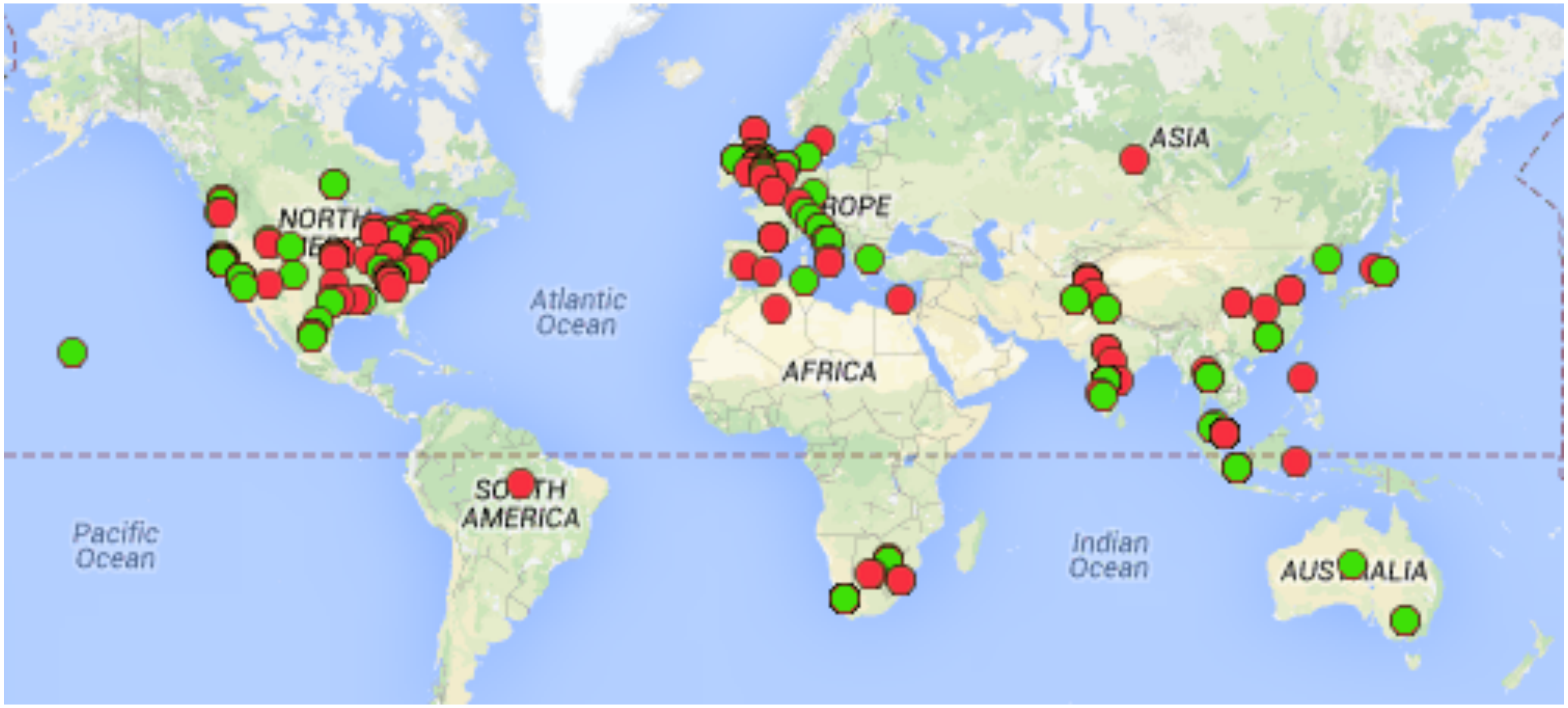
Idea: Refresh DNS records and TCP connections to popular domains at the router

Popularity-based Pre-fetching: Evaluation

- What is the improvement **in the best case**?
  - Mirage on BISmark platform (64 nodes worldwide)
- How do the benefits **complement browser opimizations**?
  - Phantomjs in controlled setting
- How can we make it **practical**?
  - Evaluate caching using real user traces from 12 homes

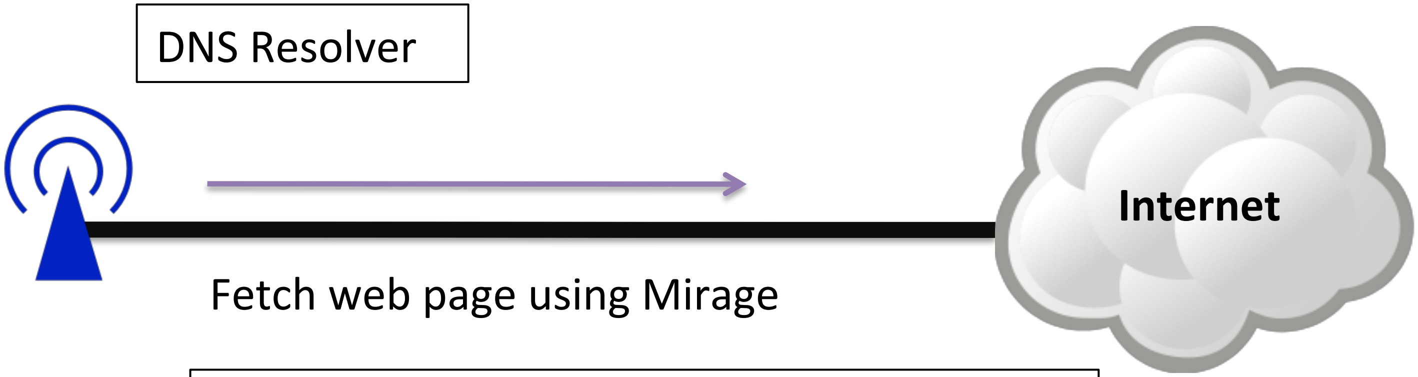
BISmark Deployment

- 64 Homes for study, Netgear WNDR 3800
  - 175+ active routers now
  - Open data, open development



http://projectbismark.net

Effect of DNS Pre-fetching



DNS Resolver

Internet

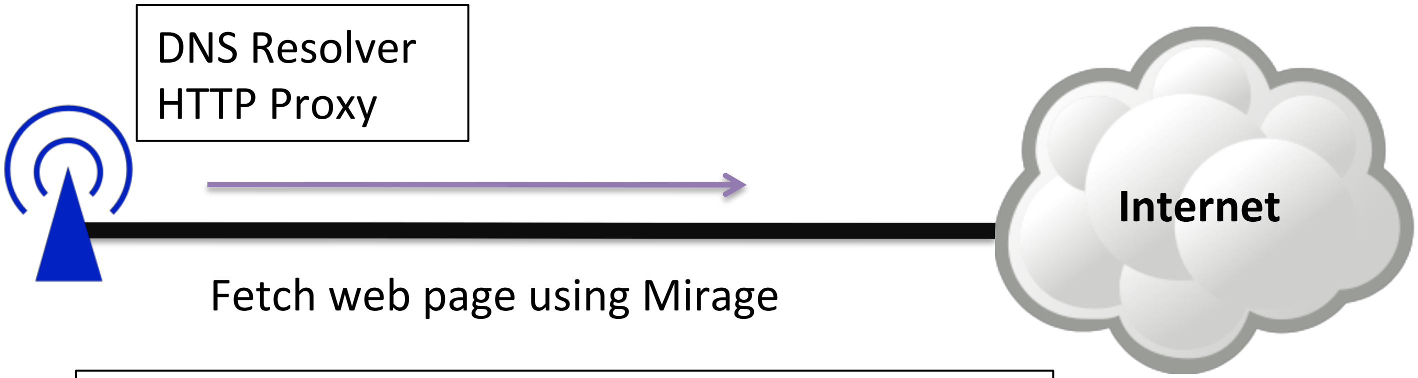
Fetch web page using Mirage

DNS Pre-fetching

1. Clear caches
2. Fetch page
3. Fetch page again

DNS pre-fetching improves page load time by up to 10%

Effect of TCP Connection Pre-fetching



DNS Resolver
HTTP Proxy

Internet

Fetch web page using Mirage

TCP Connection caching

1. Clear caches
2. Fetch page through HTTP proxy
3. Clear content cache
4. Fetch page again

TCP connection caching improves page load time by up to 35%

Popularity-based Pre-fetching: Evaluation

- What is the improvement **in the best case**?
  - Mirage, using BISmark (65 nodes worldwide)
- How do the benefits **complement browser optimizations**?
  - Phantomjs in controlled setting
- How can we make it **practical**?
  - Evaluate caching using user traces from 12 homes

Reducing the overhead of pre-fetching

- **Solution:** Pre-fetch only popular sites with timeout
- Analysis of passive usage traces in 12 homes
  - Simulation based on traces
  - Test list size and timeout intervals
- Cache hits improve list size of 20 and timeout of 2 minutes
  - DNS hit rate improves from 11-50% to 19-90%
  - TCP hit rate improves from 1-8% to 6-21%

Conclusion

- **Page load times are high for popular sites**
  - – Latency is a bottleneck when downstream throughput is > 16 Mbps
- **Popularity-based prefetching** improves performance by up to 35%.
  - – Complementary to existing optimizations
- Data and code are **publicly available** at http://projectbismark.net/