Toward a Push-Scalable Global Internet

IEEE Global Internet Symposium, IEEE Infocom 2011

Sachin Agarwal [1]

ska@alum.bu.edu

-

Presented by **Oliver Hohlfeld**

—

Deutsche Telekom A.G., Laboratories & TU Berlin
Ernst-Reuter-Platz 7
10409 Berlin, Germany

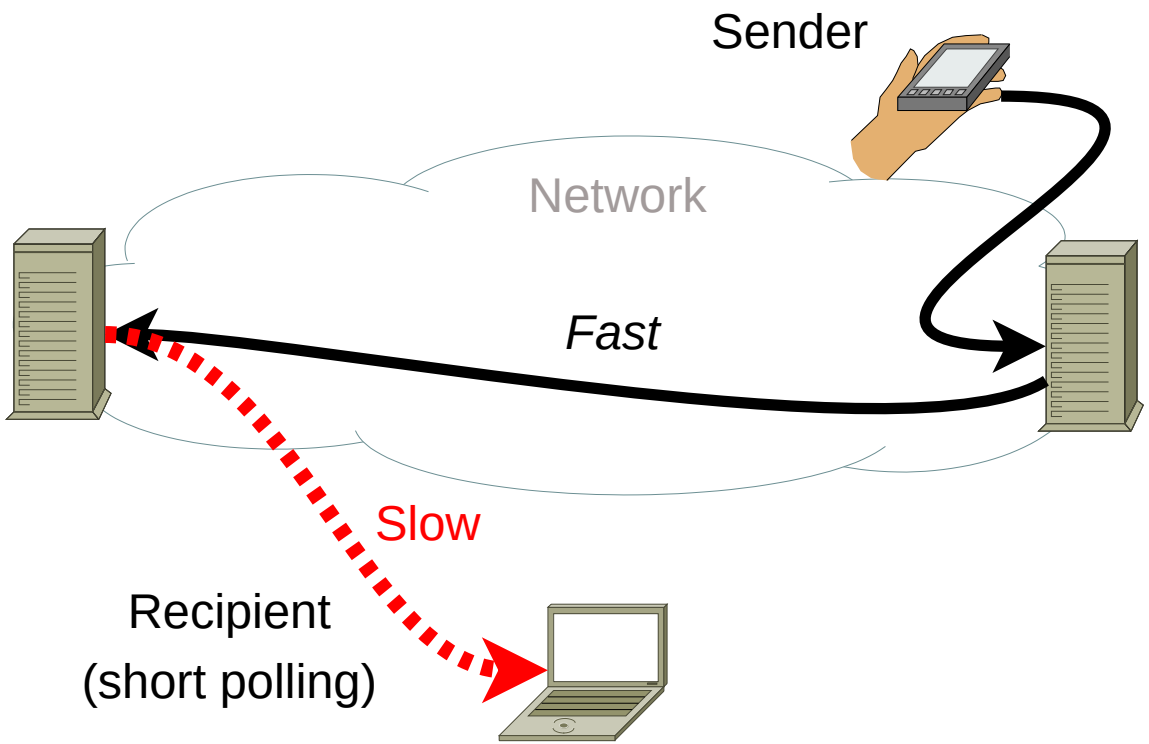
April 15th, 2011

[1] Now with NEC Europe Laboratories, Heidelberg, Germany

Outline

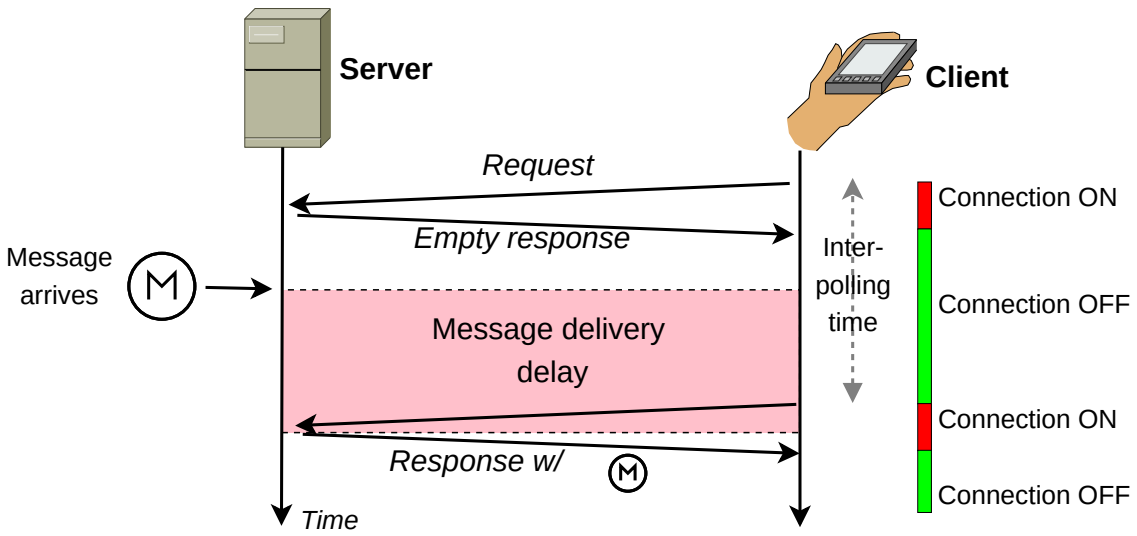
- Motivation - Push and Internet scalability
- Case study: Android's push service
- Our solution: Content-based optimization
- Conclusions and future Work

Information Delivery on the Web



- Recipient's server receives message almost instantaneously...
- ...but short polling leads to delayed delivery

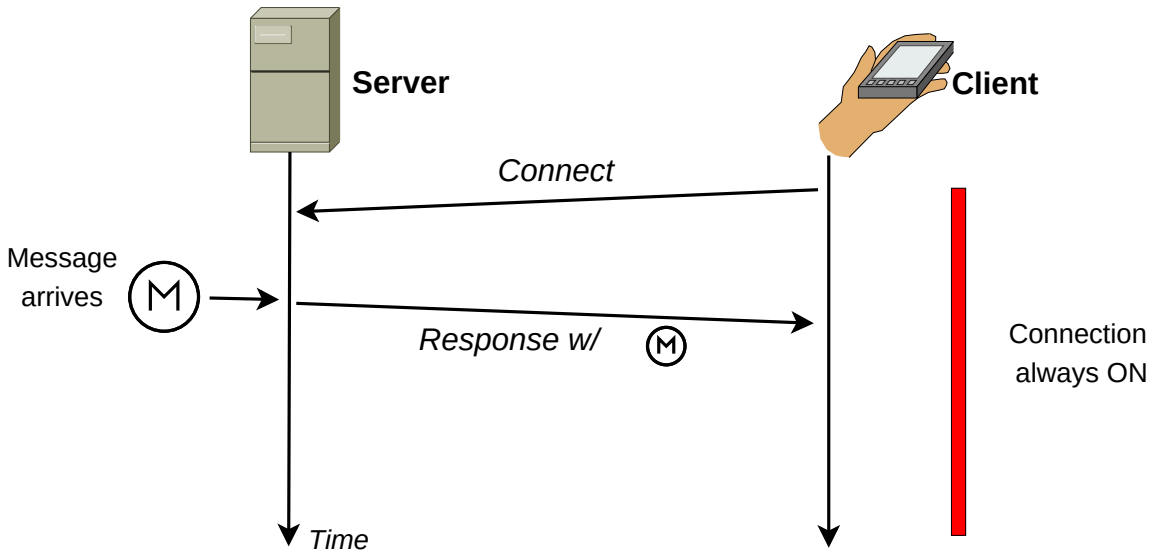
Pulling for Messages: Short Polling



Message delivery delay

No continuous connection

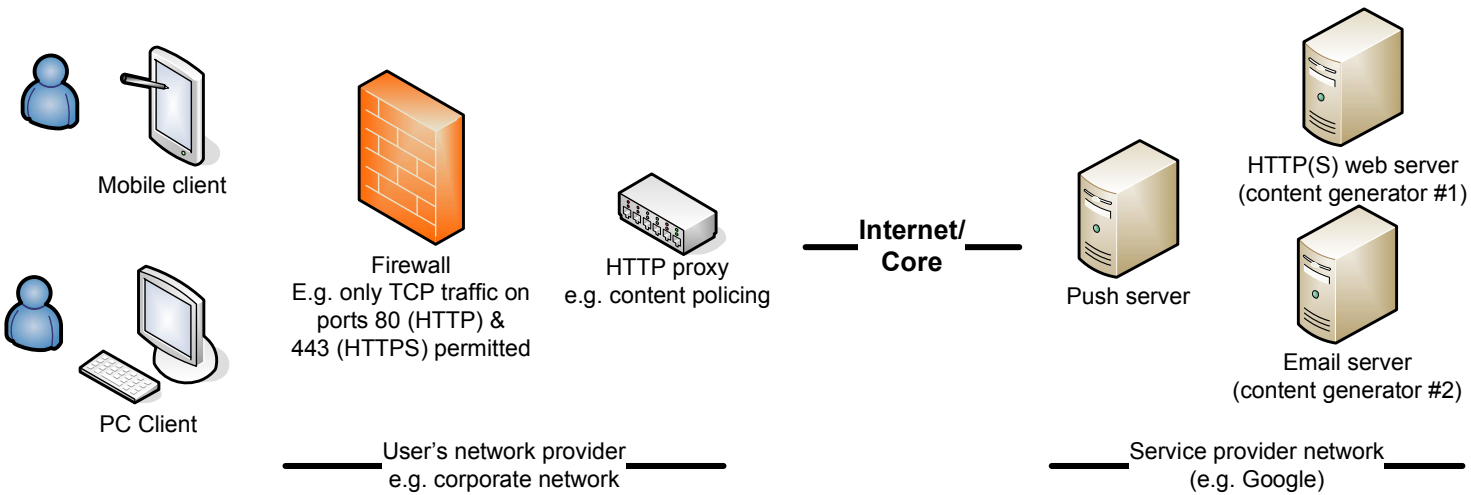
Alternative to Short Polling - Push Message Delivery



Near-zero delay in message delivery

Continuous (TCP) connection

Effects of Long-lived Connections on Web Infrastructure



Tied up network resources (e.g. proxy memory/processing)
End-point scalability limitations (client battery/processing, server capacity)

Android's Cloud-to-Device Messaging Service

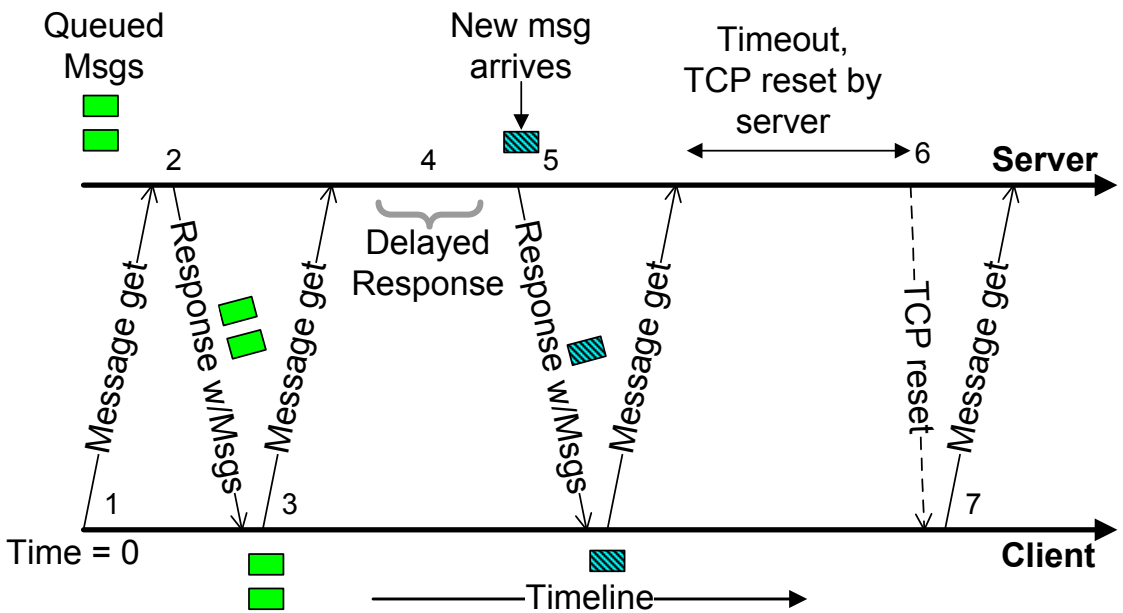


Figure: Time-line of a long polling interaction between client and server.

Always-on connection via long polling

Android: Longevity & Packets per Connection

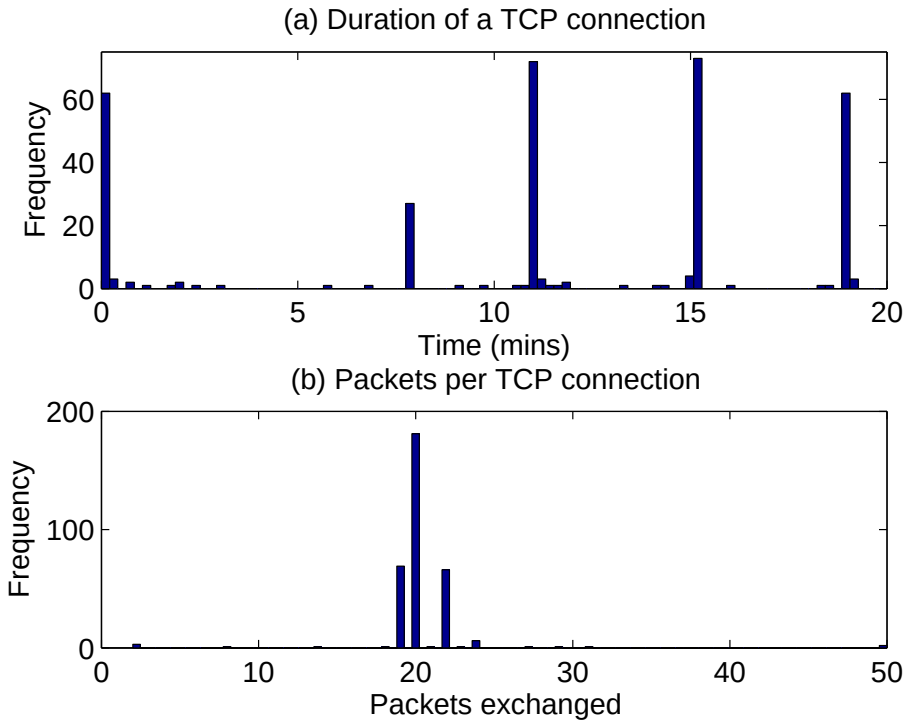


Figure: Durations & packet counts of TCP connection between Android client and C2DM server

Recurring durations & packet counts imply algorithmic control (not random disconnects)

Android: Concurrent TCP Connections

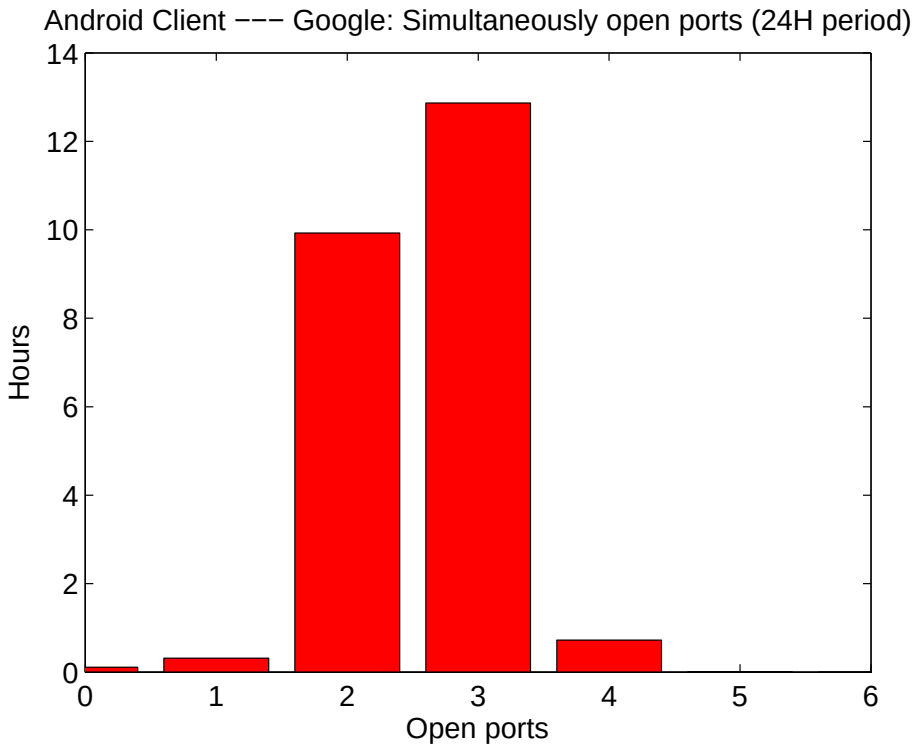
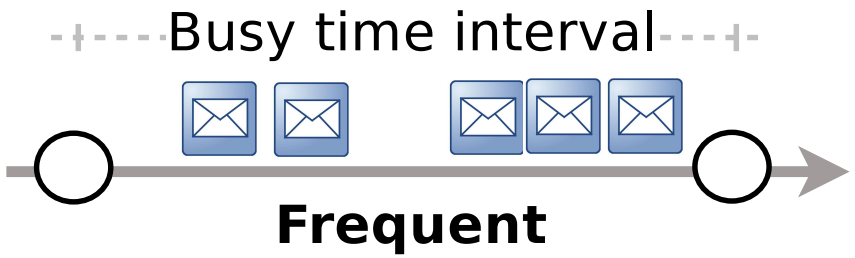


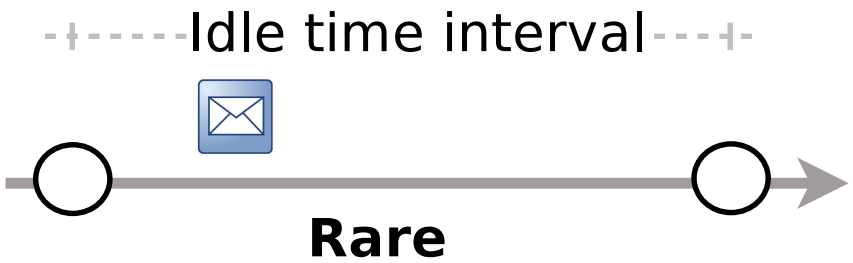
Figure: Concurrent TCP connections - Android client & push server

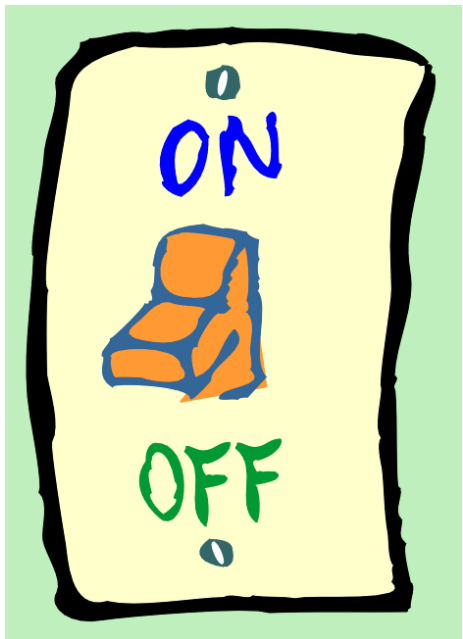
Scalability concerns
Multiple concurrent TCP connections per client, 24×7
Uses HTTPS for security and transparent proxying
(further burden on endpoints)

Our Idea: Content-based Optimization

Turn-off always-on connections when message arrivals are rare.







Question: Learning when to flick the switch
Answer: Message arrival patterns (over time)

Enron Email Data-set

- Publicly released by the US Federal Energy Regulatory Commission
- Contains $\tilde{5}$00,000 email messages of 150 senior Enron employees over $\tilde{4}$ years
- Email headers also available (e.g. email sending times)
- Convenient database representation of data-set available via

  A.Fiore and J.Heer, UC Berkeley, Enron Email Analysis http://bailando.sims.berkeley.edu/enron email.html

Example: Day of week Email Message Arrivals

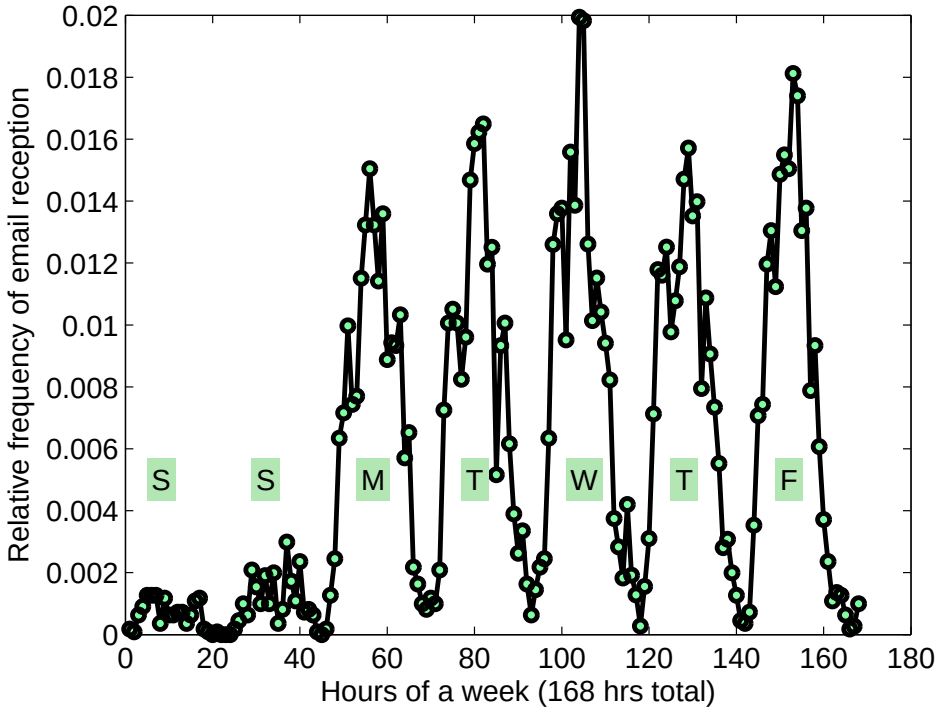


Figure: A user's weekly email reception (averaged over 110 weeks).

Rare messages arrivals during several hours of a week

Message inter-arrival times

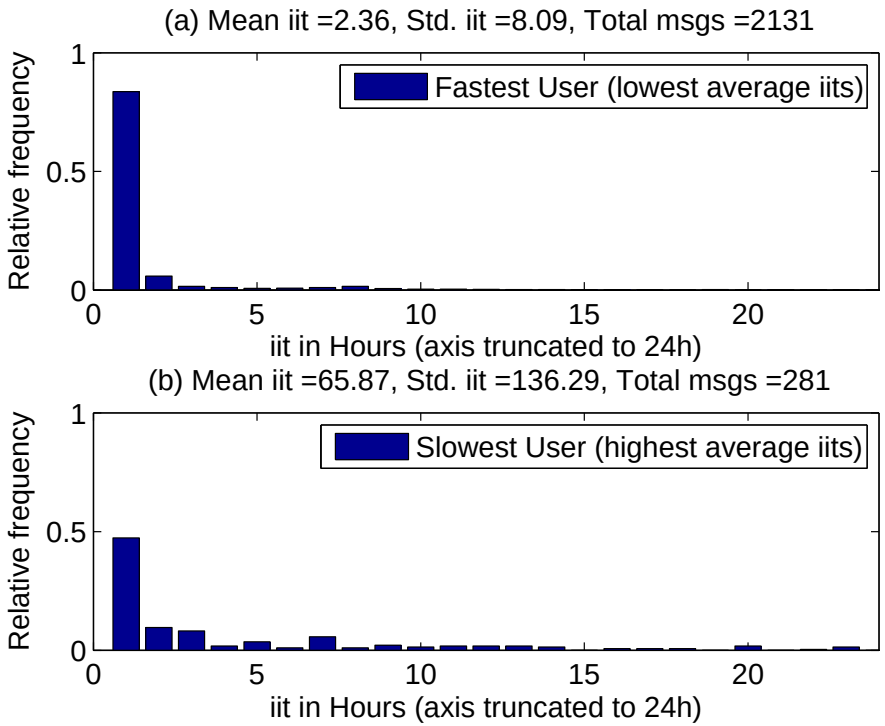


Figure: Relative frequency of inter-arrival times (iits) in hours for (a) the fastest user and for (b) the slowest user.

Non-Poisson message arrivals

When to Flick the Switch - Simple Machine Learning (ML)

Fixed Learning For each user, use a small fraction of arrival-times to rank hours of the week according to the relative frequency of email arrival.
Adaptive Learning Use all previous arrival-times, weigh more recent arrival-times more.

$$F_{i+1} = \alpha F_i + (1 - \alpha) f_i \tag{1}$$

F_i - 168-element vector of hour rank vector of week i
f_i - 168-element vector of message arrival frequencies for week i

Performance - Fastest User

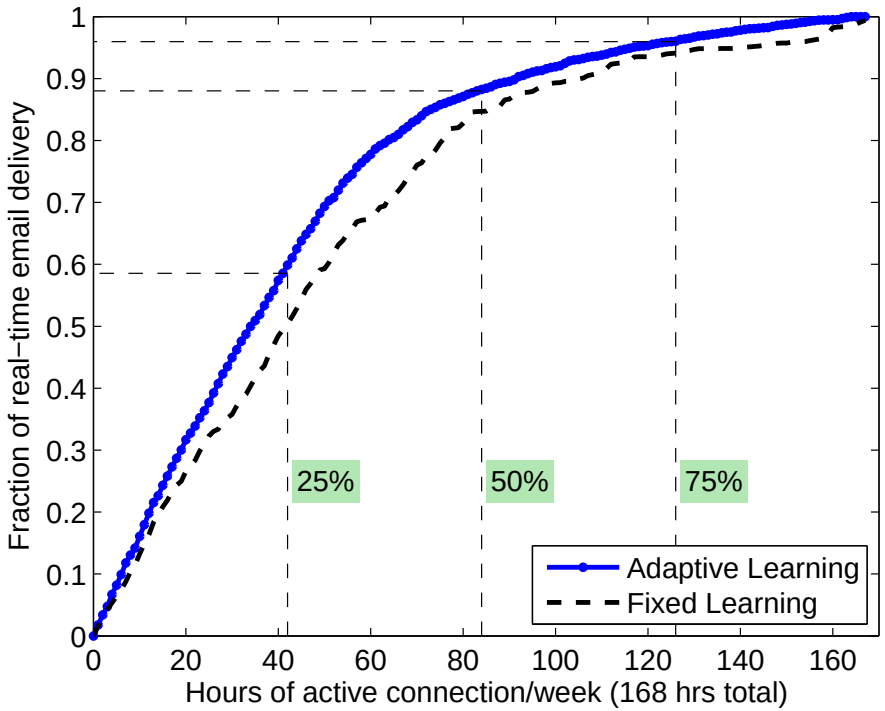


Figure: Fastest user, $\alpha = 0.9$ for adaptive learning

Both learning algorithms perform reasonably well

Performance - Averaged Across 150 Users

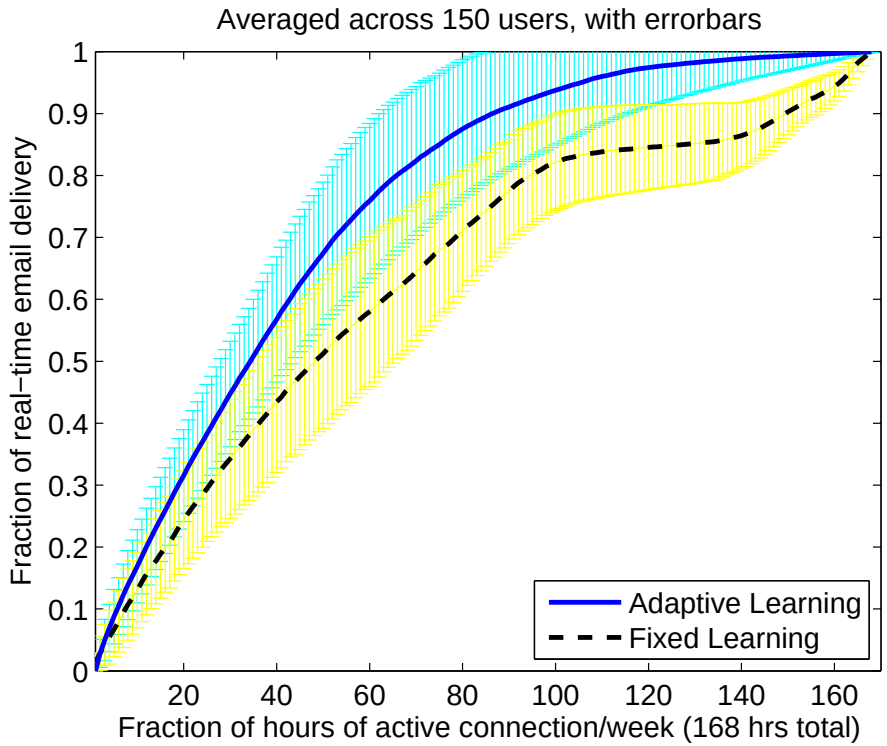


Figure: 150 users, $\alpha = 0.9$ for adaptive learning

Adaptive learning more effective for slower users

Future Work

- Quantitative study of scalability bottlenecks - client battery, network elements, server resources
- Other content information for ML: e.g. semantic meaning, importance of message, sender, spam score, size, attachments, etc.
- Applicability to other pushed information e.g. social network updates
- Sophisticated ML algorithms

Conclusion

- Push messaging on the web is not free
- Content-based optimization may hold the key
- Proposed approach yielded 50% on-time reduction with 90% messages delivered instantaneously
- Multiple future directions possible here - more measurements, ML, other message types, etc.
- High impact research problem - explosion of mobile devices and HTTP (port 80) based communication