Understanding and Improving TCP for Web Performance

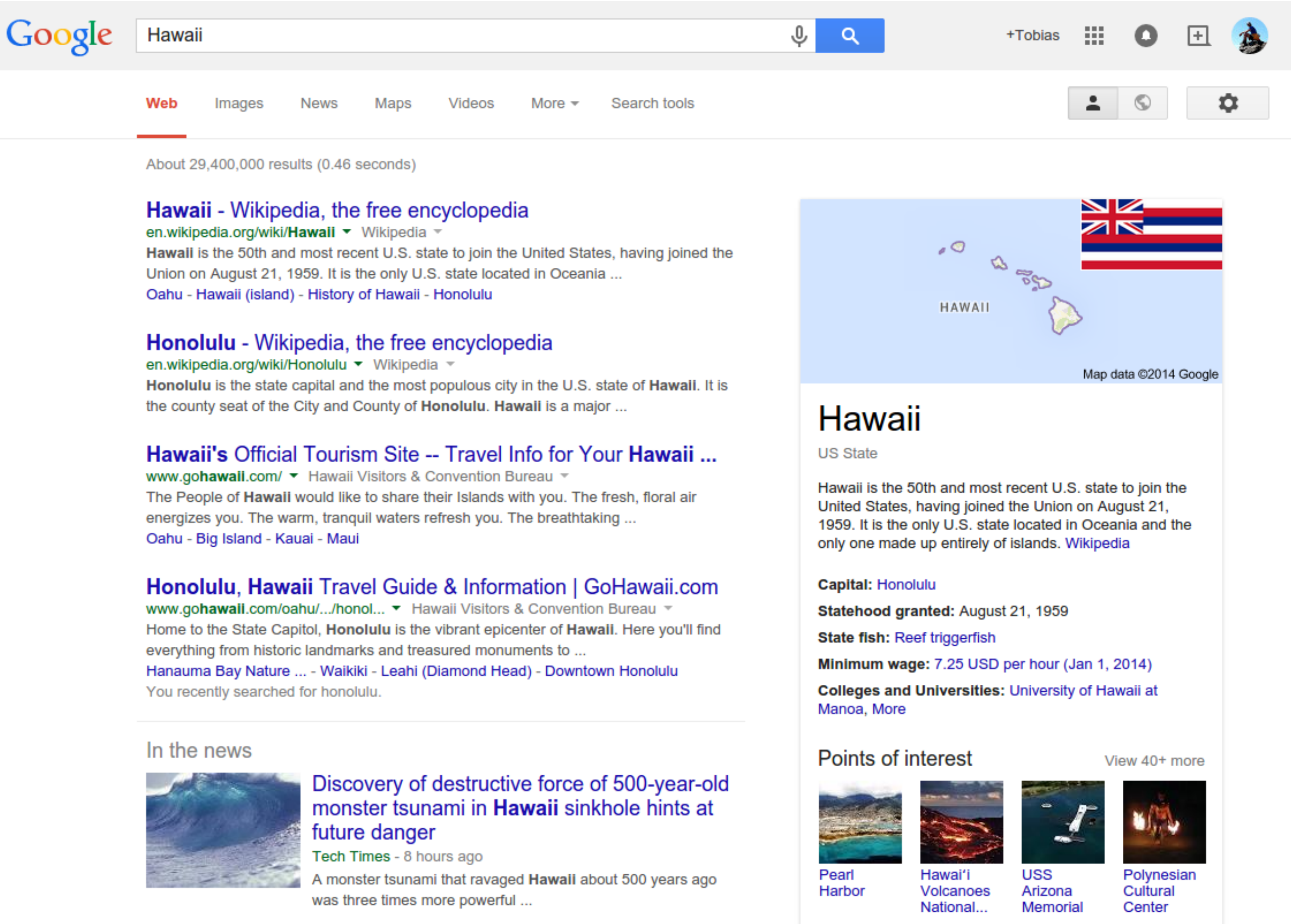
Tobias Flach

Nandita Dukkipati, Andreas Terzis, Barath Raghavan, Neal Cardwell, Yuchung Cheng, Ankur Jain, Shuai Hao, Ethan Katz-Bassett, Ramesh Govindan

USC Viterbi School of Engineering

Google™


Google
Hawaii
Google Search
I'm Feeling Lucky



Google Hawaii

+Tobias

Web Images News Maps Videos More Search tools

About 29,400,000 results (0.46 seconds)

Hawaii - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Hawaii** Wikipedia
Hawaii is the 50th and most recent U.S. state to join the United States, having joined the Union on August 21, 1959. It is the only U.S. state located in Oceania ...
Oahu - Hawaii (island) - History of Hawaii - Honolulu

Honolulu - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Honolulu Wikipedia
Honolulu is the state capital and the most populous city in the U.S. state of **Hawaii**. It is the county seat of the City and County of **Honolulu**. **Hawaii** is a major ...

Hawaii's Official Tourism Site -- Travel Info for Your **Hawaii** ...
www.go**hawaii**.com/ Hawaii Visitors & Convention Bureau
The People of **Hawaii** would like to share their Islands with you. The fresh, floral air energizes you. The warm, tranquil waters refresh you. The breathtaking ...
Oahu - Big Island - Kauai - Maui

Honolulu, **Hawaii** Travel Guide & Information | GoHawaii.com
www.go**hawaii**.com/oahu/.../honol... Hawaii Visitors & Convention Bureau
Home to the State Capitol, **Honolulu** is the vibrant epicenter of **Hawaii**. Here you'll find everything from historic landmarks and treasured monuments to ...
Hanauma Bay Nature ... - Waikiki - Leahi (Diamond Head) - Downtown Honolulu
You recently searched for honolulu.

In the news

Discovery of destructive force of 500-year-old monster tsunami in **Hawaii** sinkhole hints at future danger
Tech Times - 8 hours ago
A monster tsunami that ravaged **Hawaii** about 500 years ago was three times more powerful ...

HAWAII

Map data ©2014 Google

Hawaii

US State

Hawaii is the 50th and most recent U.S. state to join the United States, having joined the Union on August 21, 1959. It is the only U.S. state located in Oceania and the only one made up entirely of islands. Wikipedia

Capital: Honolulu

Statehood granted: August 21, 1959

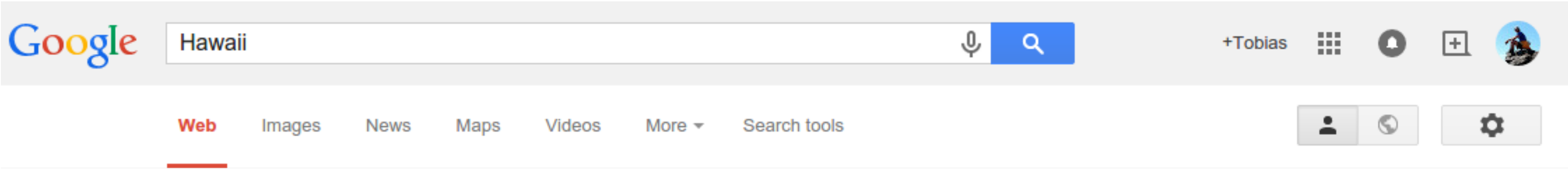
State fish: Reef triggerfish

Minimum wage: 7.25 USD per hour (Jan 1, 2014)

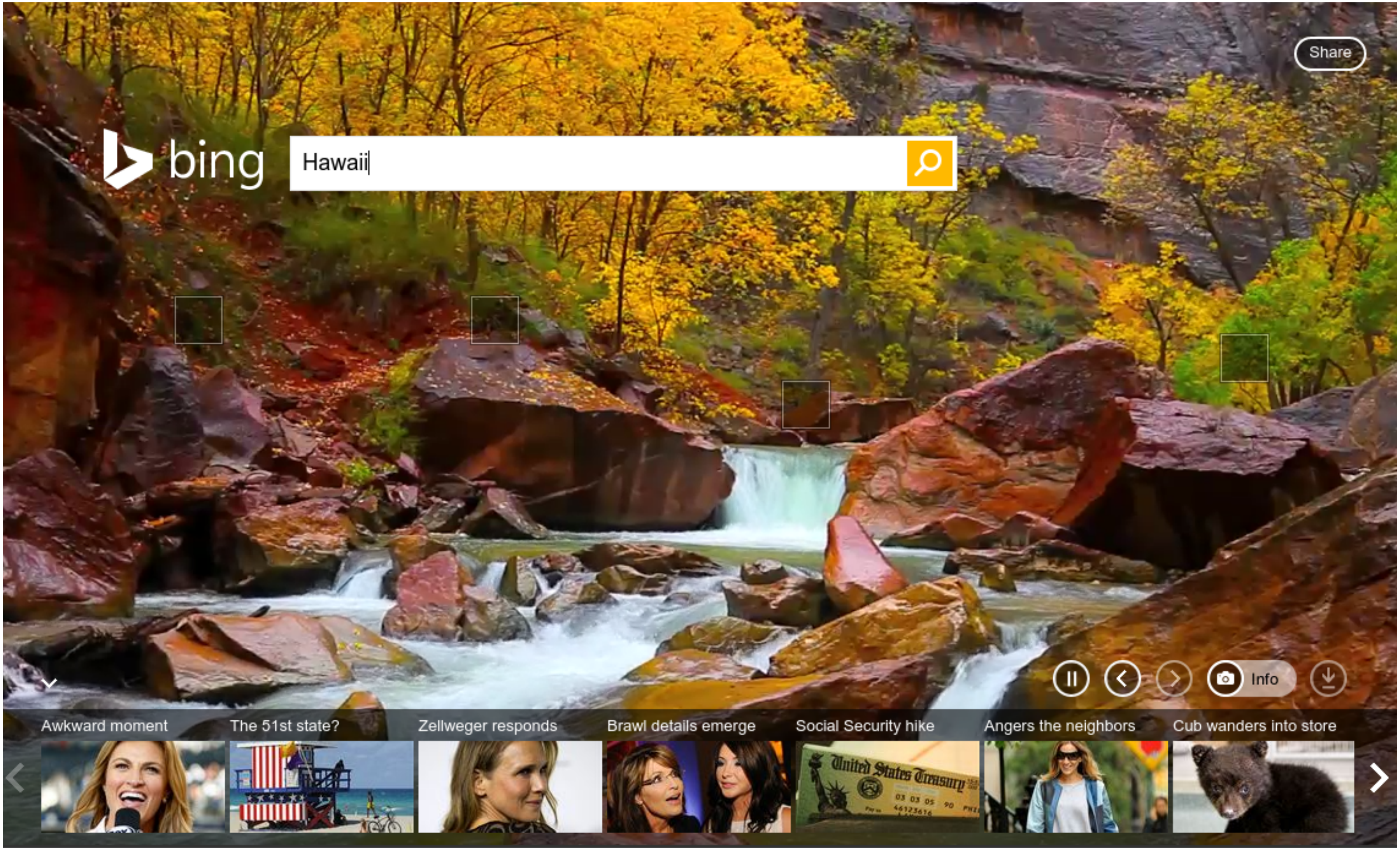
Colleges and Universities: University of Hawaii at Manoa, More

Points of interest View 40+ more

Pearl Harbor
Hawaiʻi Volcanoes National...
USS Arizona Memorial
Polynesian Cultural Center



Sometimes users wait for a long time to get a response



Amazon found that every additional 100ms delay in loading a page costs them 1% of sales

Motivation

- Goal: Optimize communication means to improve web performance (correlated with more revenue)
- Challenges: network and protocol complexity
- Focus here: Troubleshoot and improve TCP

Overview

What aspects of TCP are limiting Web access performance, and how can we overcome these limitations?

Identify performance bottlenecks and root causes

Mitigate root causes through protocol or structural changes

"Reducing Web Latency: The Virtue of Gentle Aggression" (SIGCOMM 13)

"Understanding TCP Flow Performance at Scale Through Behavioral Signatures"
(in progress)

Overview

A. **Reducing Web Latency: The Virtue of Gentle Aggression**

B. Understanding TCP Flow Performance at Scale Through Behavioral Signatures

We improved Google’s response time by 23%

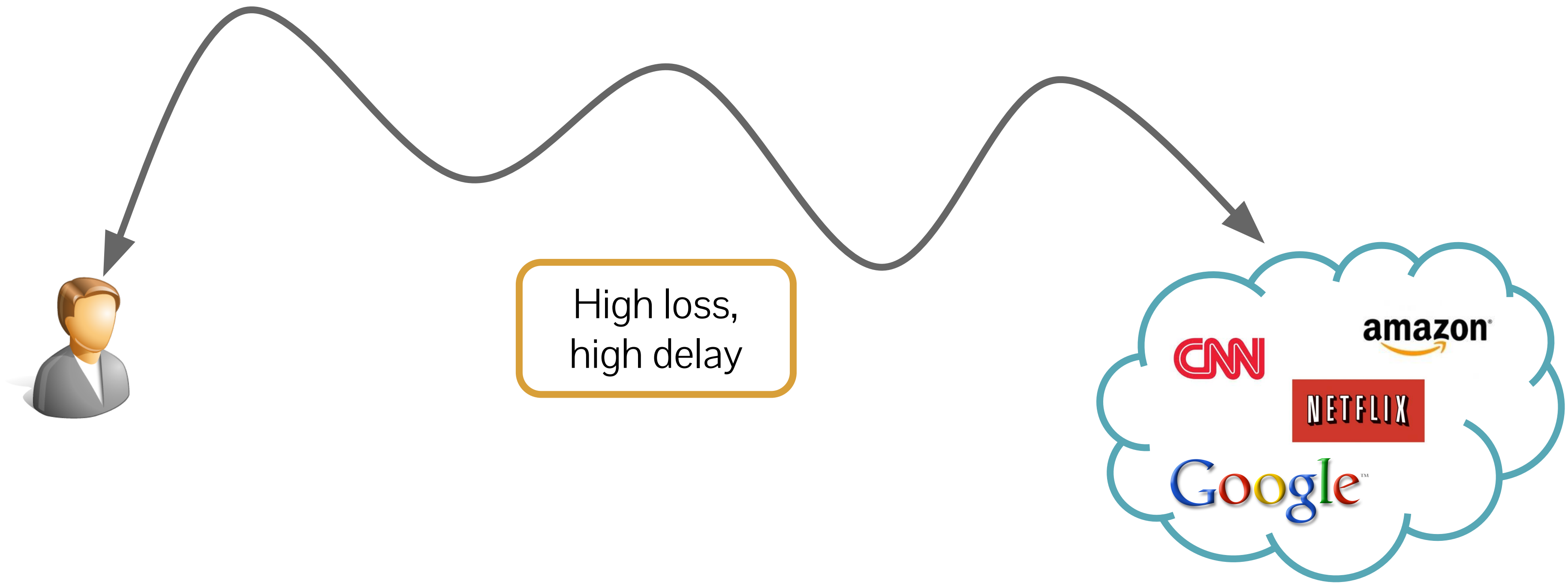
- Across billions of client requests, we improved the mean response time by 23%.

- We achieved this despite ONLY speeding up the 6% of transfers that experienced packet loss.

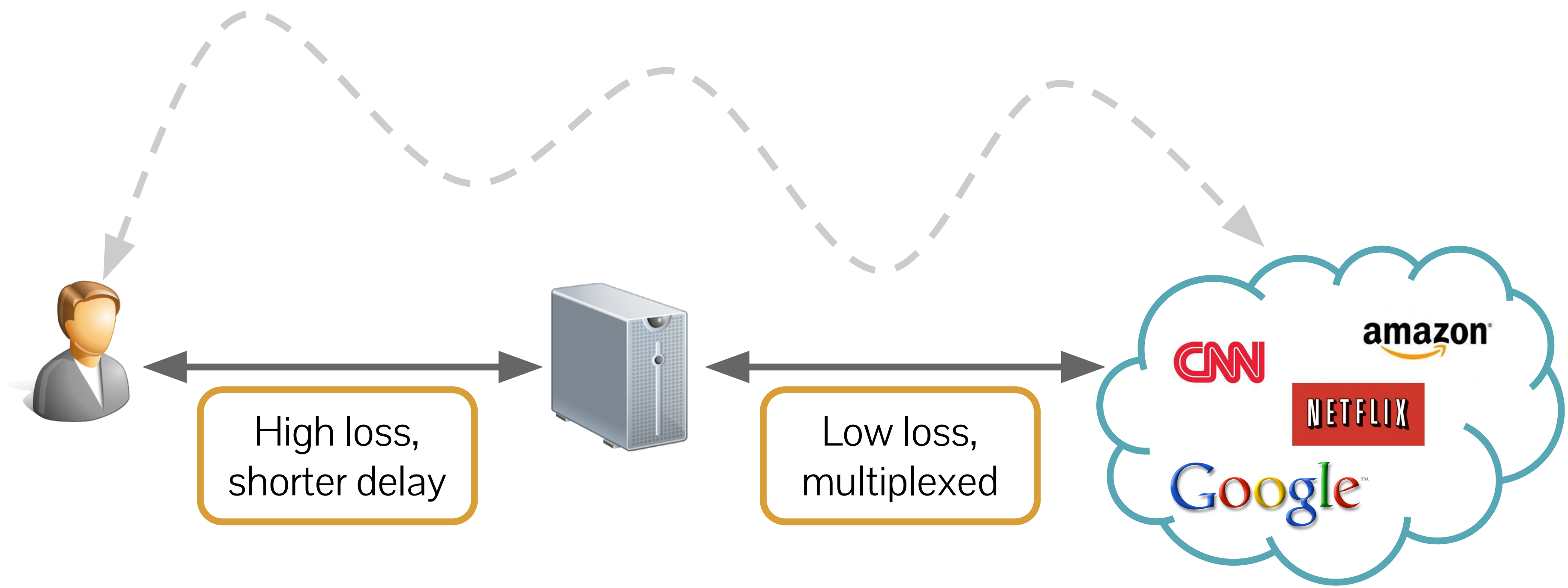
- Improvement is in the tail: We halved latency in the 99th percentile.

- For latency-sensitive services faster transfers mean a better user experience.

Ways to Reduce Latency



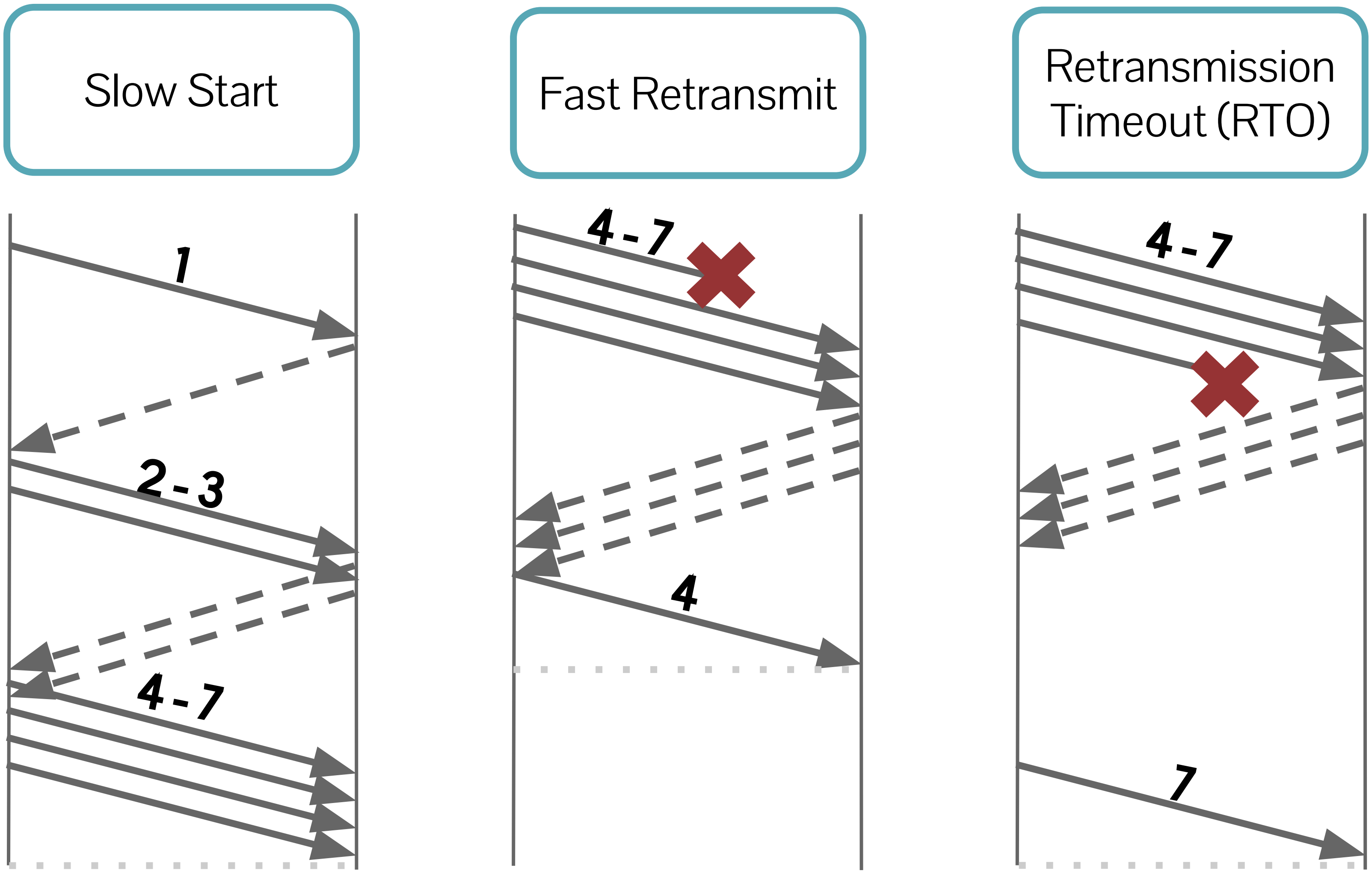
Ways to Reduce Latency



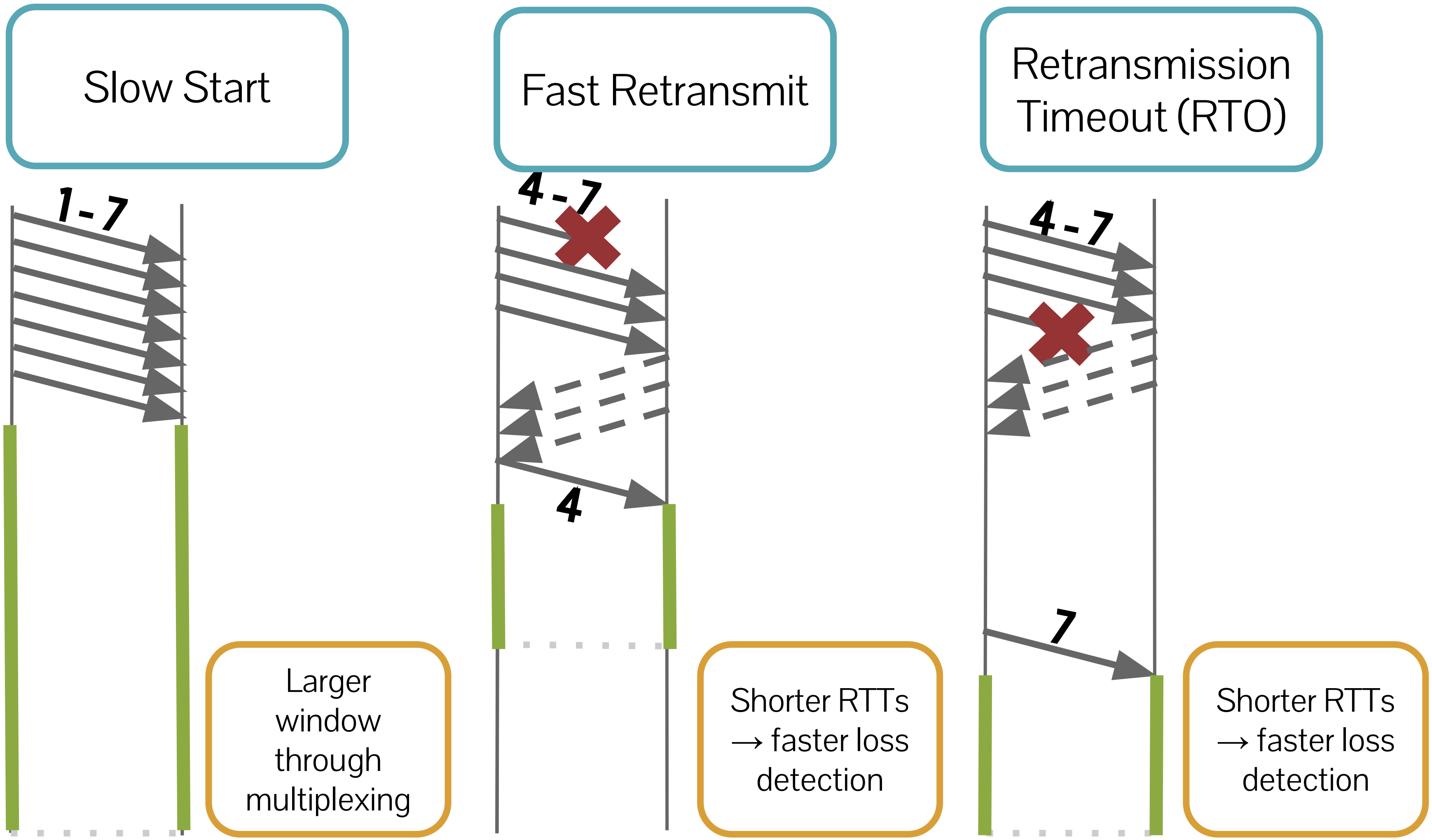
Improve the proximity of services to the user

Leverage multi-stage connections

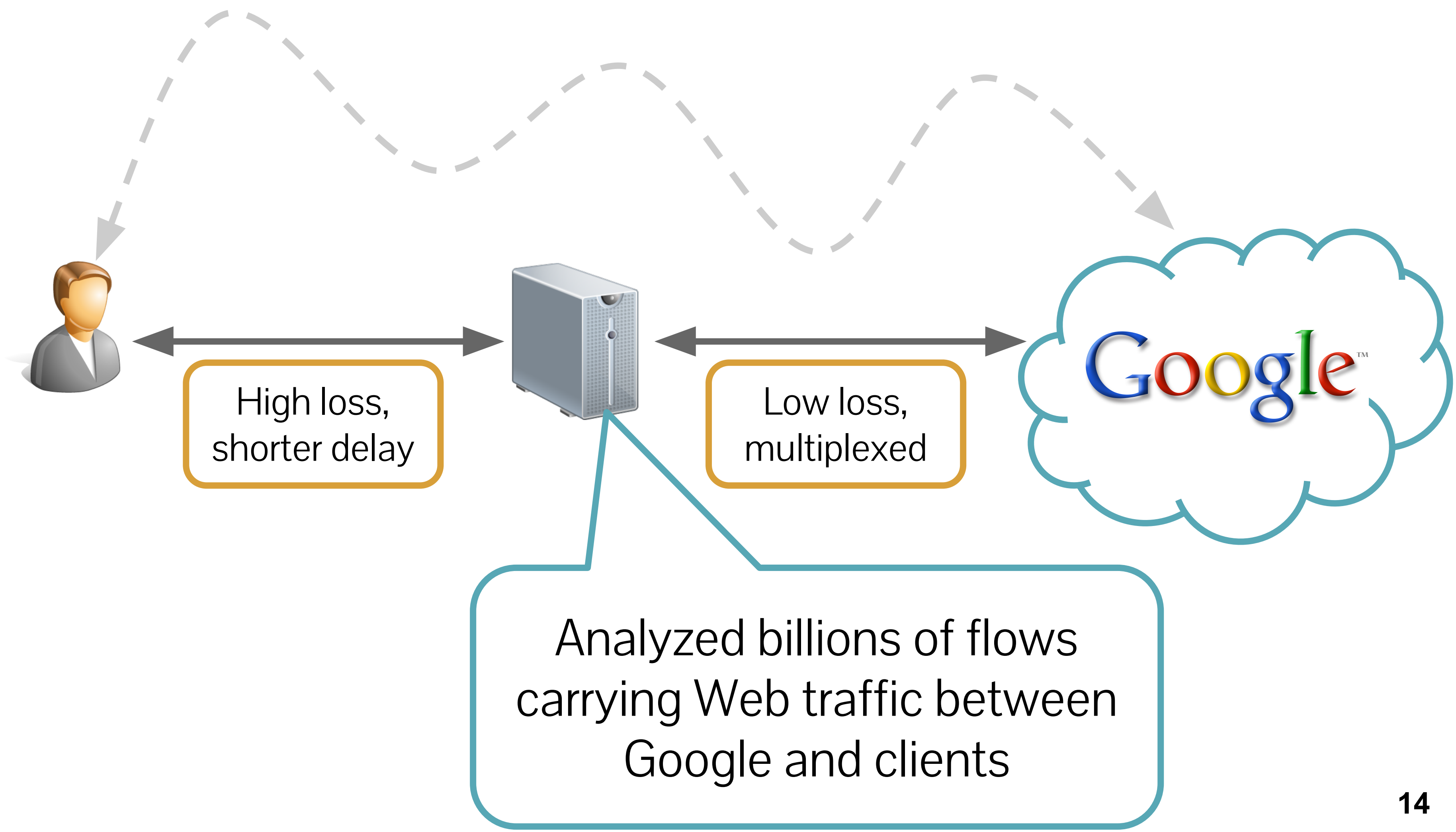
Basic TCP Mechanisms



Basic TCP Mechanisms



Evaluating TCP Performance

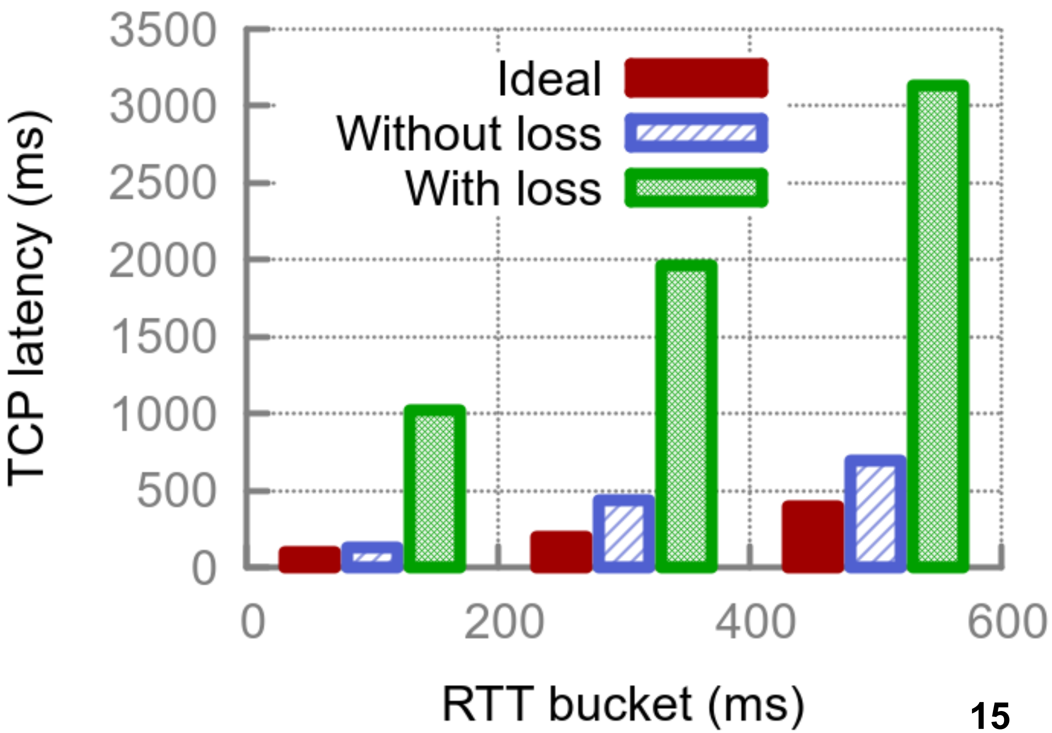


Transfers With Loss Are Too Slow

Loss makes Web latency 5 times slower

Delays caused by TCP loss detection and recovery

6% of transfers between Google and clients are lossy

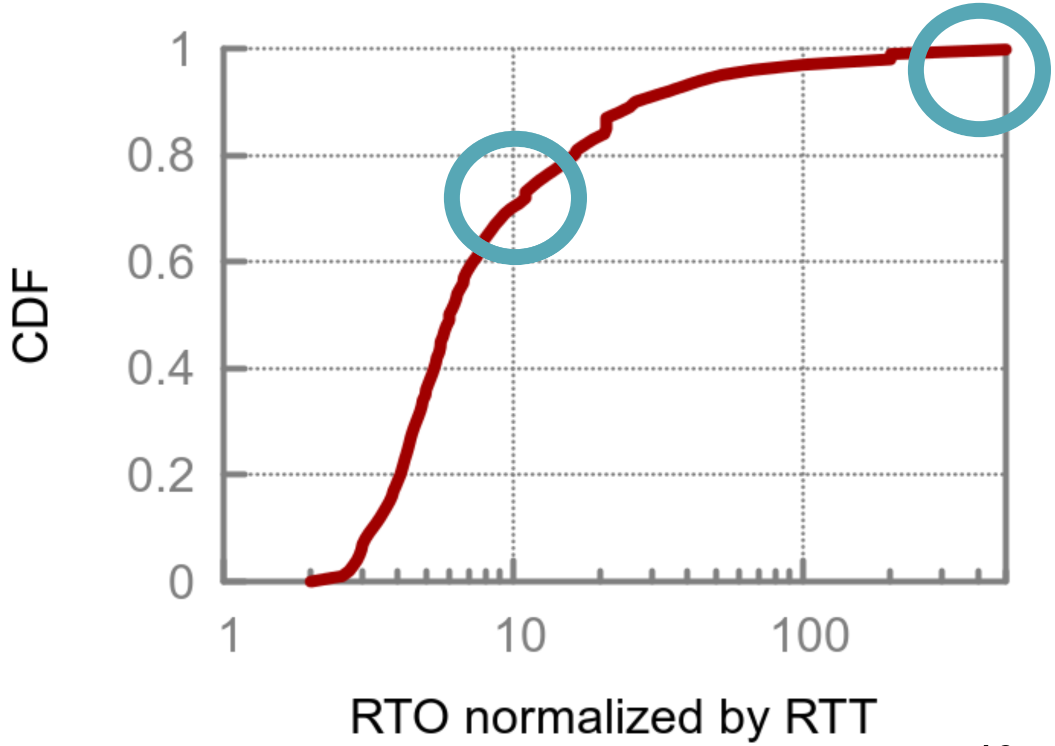


Many Expensive Retransmission Timeouts

77% of losses are recovered by retransmission timeouts

Retransmission timeouts can be 200 times larger than the RTT

Caused by high RTT variance, or lack of samples

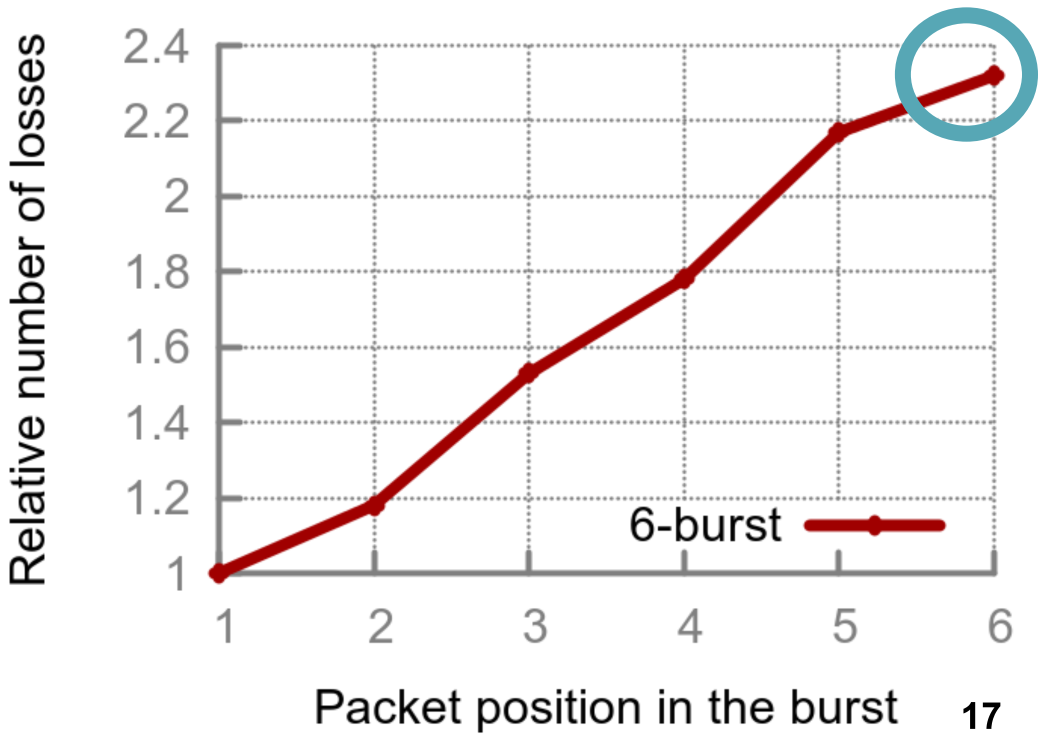


... Caused by Tail Packet Loss

(Single) tail packet drop is very common

Tail packets are twice as likely to be dropped compared to packets early in a burst

35% of lossy bursts observe only one packet loss

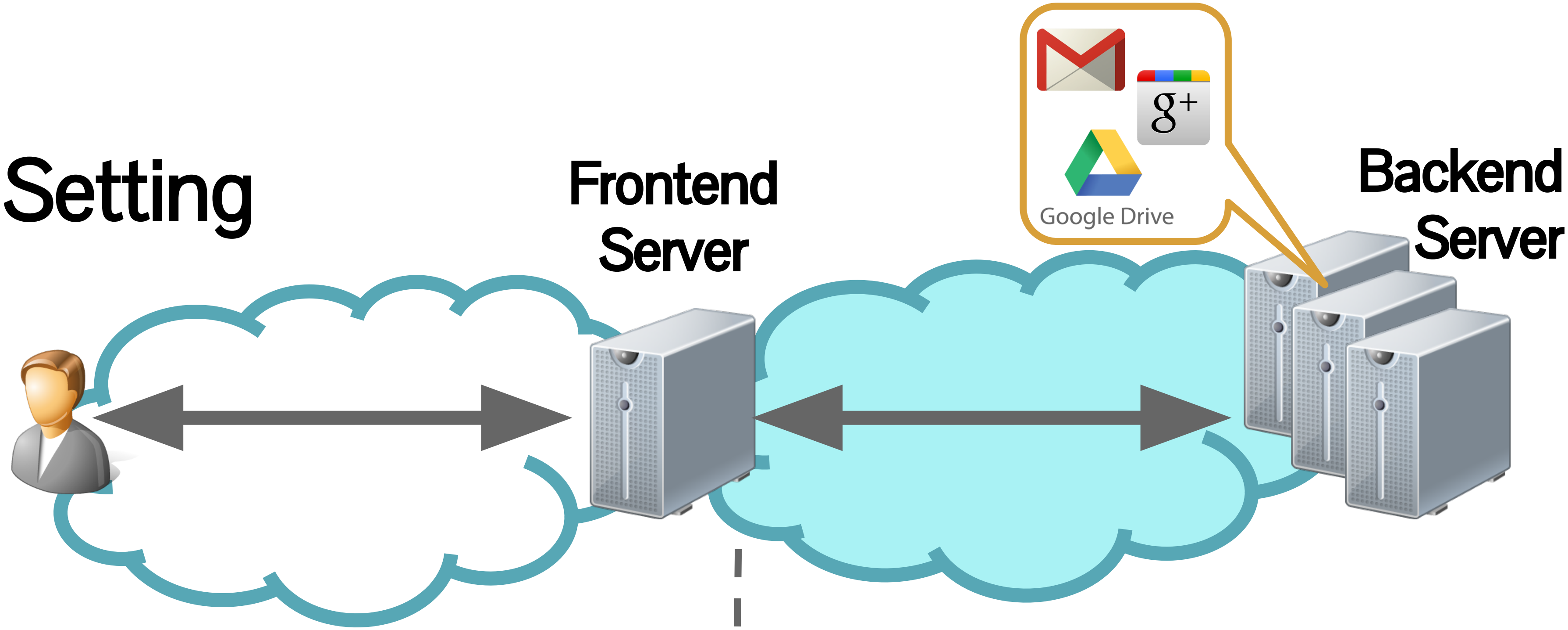


Our Motivation and Goal

Loss significantly slows down transfers.
Due to frequent recovery via slow RTOs.
Caused by tail loss.

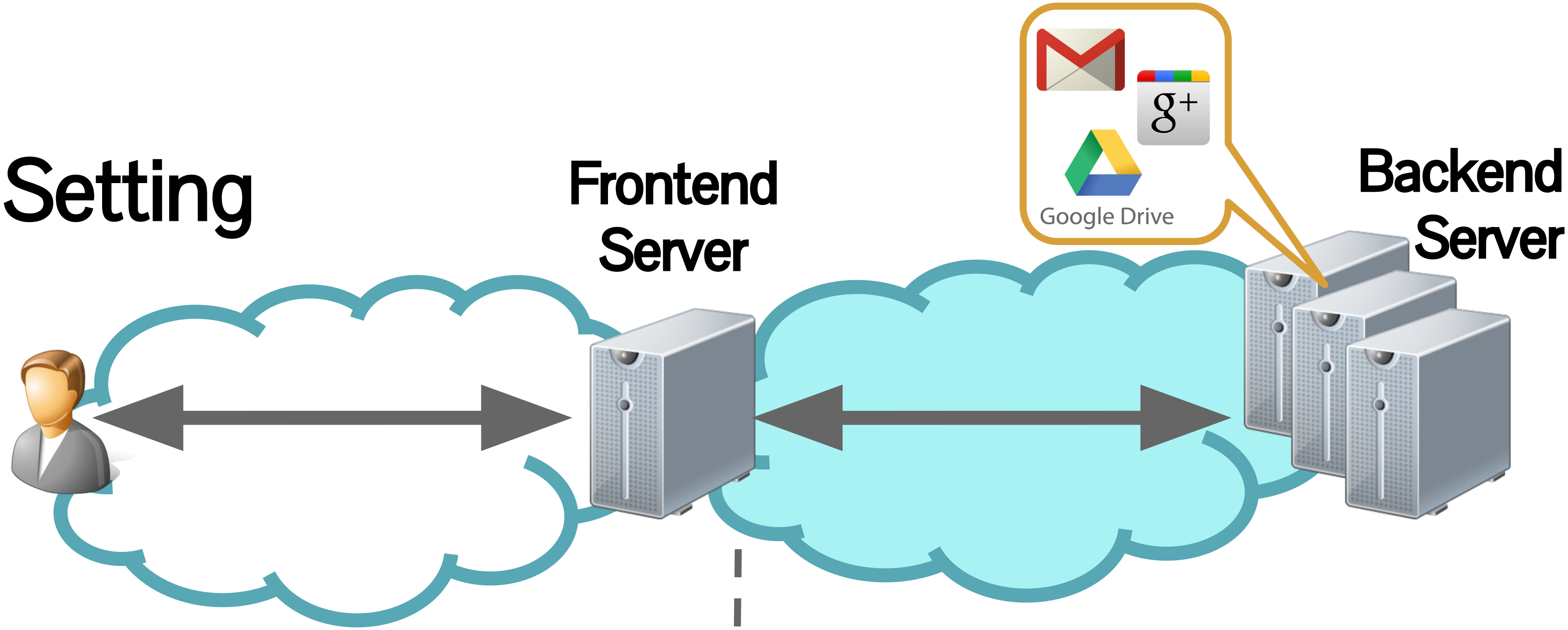
Our Goal: Approaching the ideal of loss detection and recovery without delay.
Without making the protocol too aggressive.

Setting



Public Network	Private Network
Controlling server only	Controlling client, server, and network
Prefer solutions that do not require client changes and are compatible with middleboxes	Can incur more overhead since latency-sensitive traffic is a small portion of traffic mix

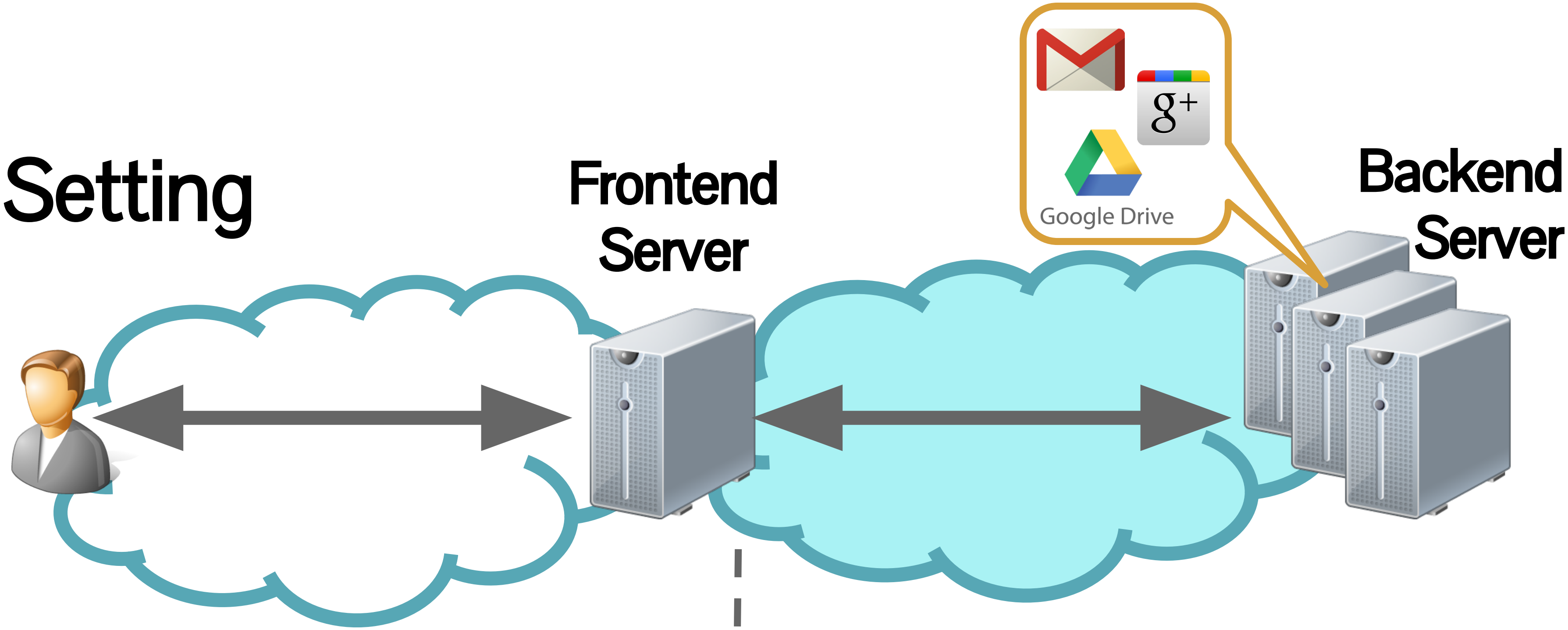
Setting



Public Network	Private Network
Reactive: Trigger fast retransmit by retransmitting the tail packet early	**Proactive**: Avoid retransmissions through packet duplication

Corrective: Add redundancy to enable recovery without retransmission, or trigger fast retransmit

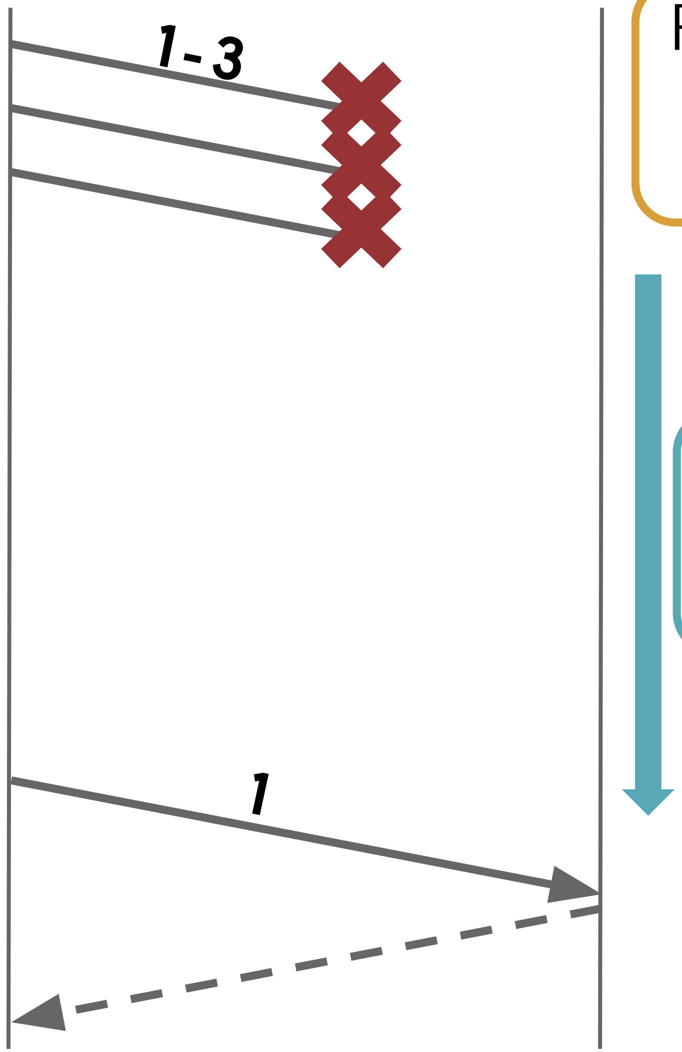
Setting



Public Network	Private Network
Reactive: Trigger fast retransmit by retransmitting the tail packet early	**Proactive**: Avoid retransmissions through packet duplication

Corrective: Add redundancy to enable recovery without retransmission, or trigger fast retransmit

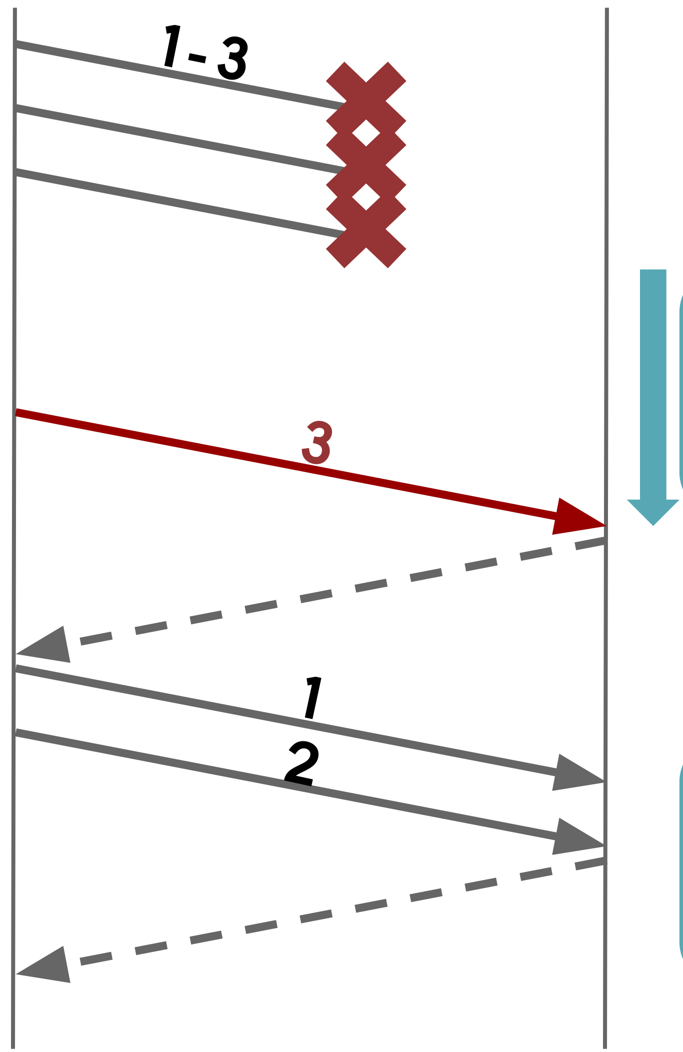
Reactive



Receiver does not know about the loss and therefore cannot send signals back

Wait time until RTO

Reactive



Wait for two RTTs

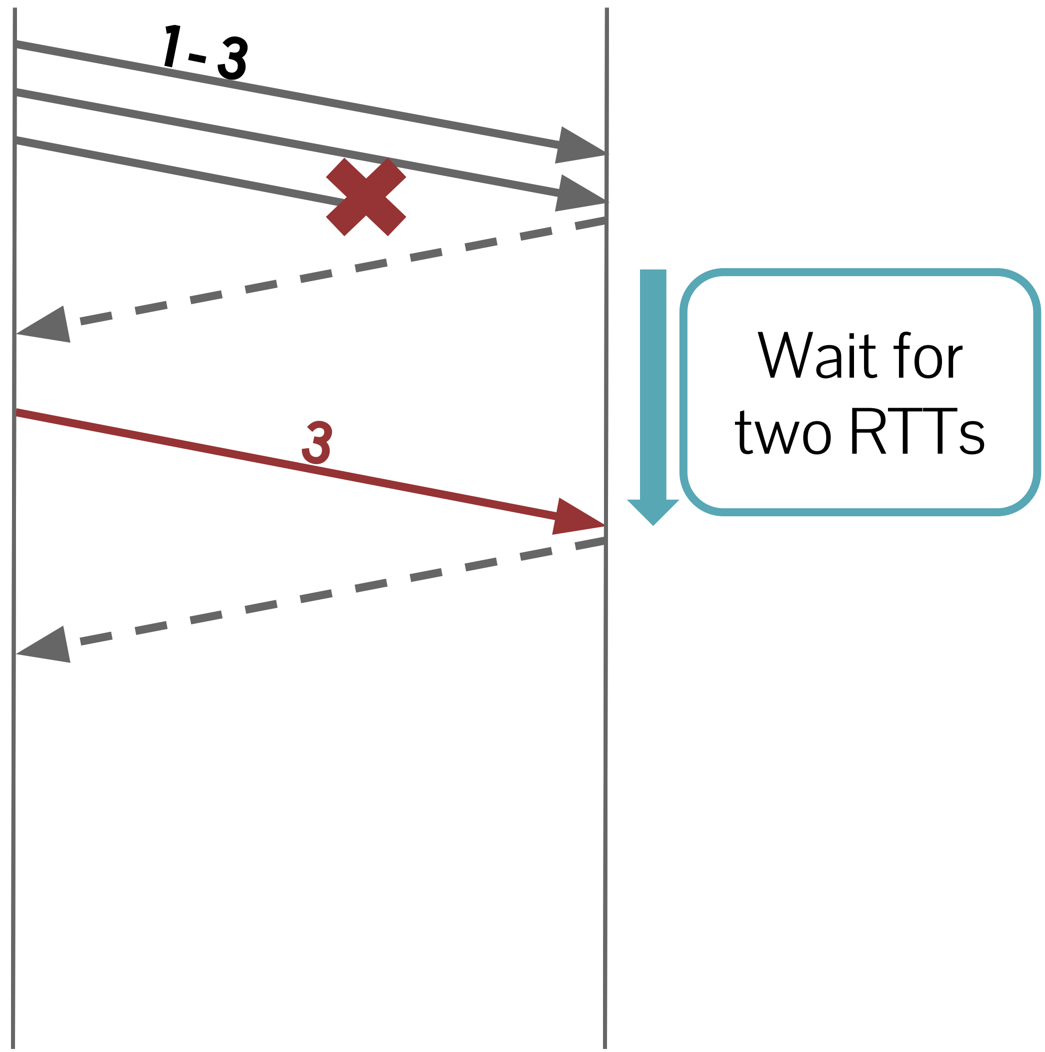
Fast retransmit

Retransmit new packet or previous (tail) packet after two RTTs

Can trigger selective acknowledgement indicating loss

Speeds up loss detection

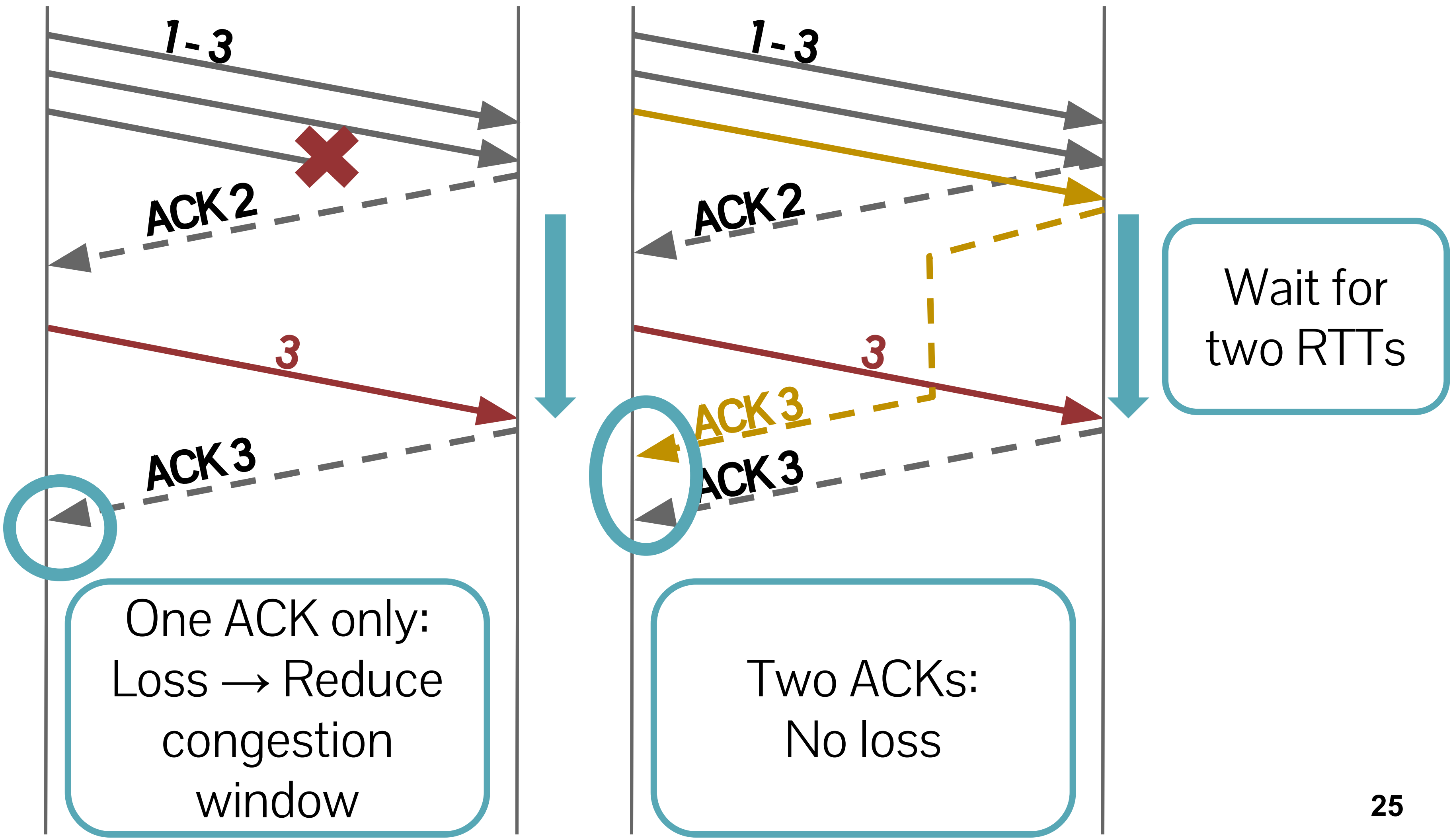
Reactive: Detecting Masked Losses



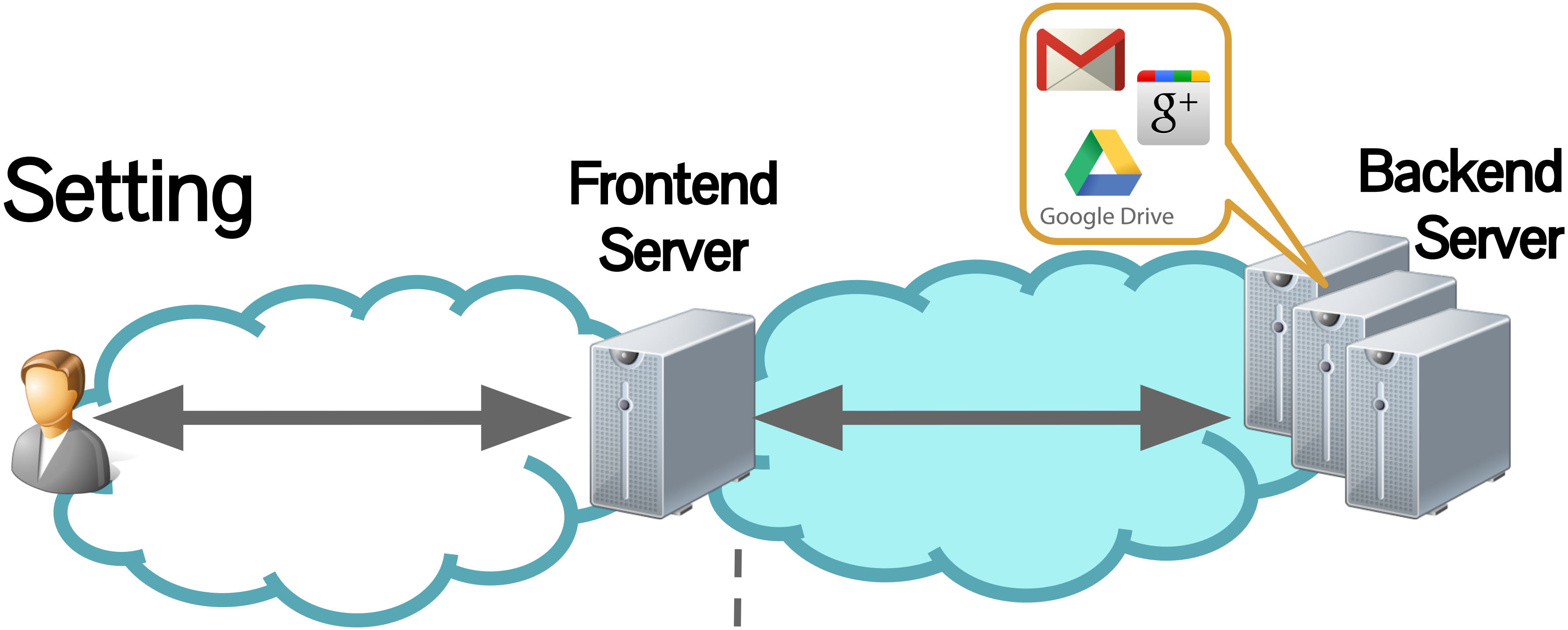
Cannot ignore the case where a packet loss is recovered by the Reactive probe

Count ACKs and reduce congestion window if only one ACK for tail packet received

Reactive: Detecting Masked Losses



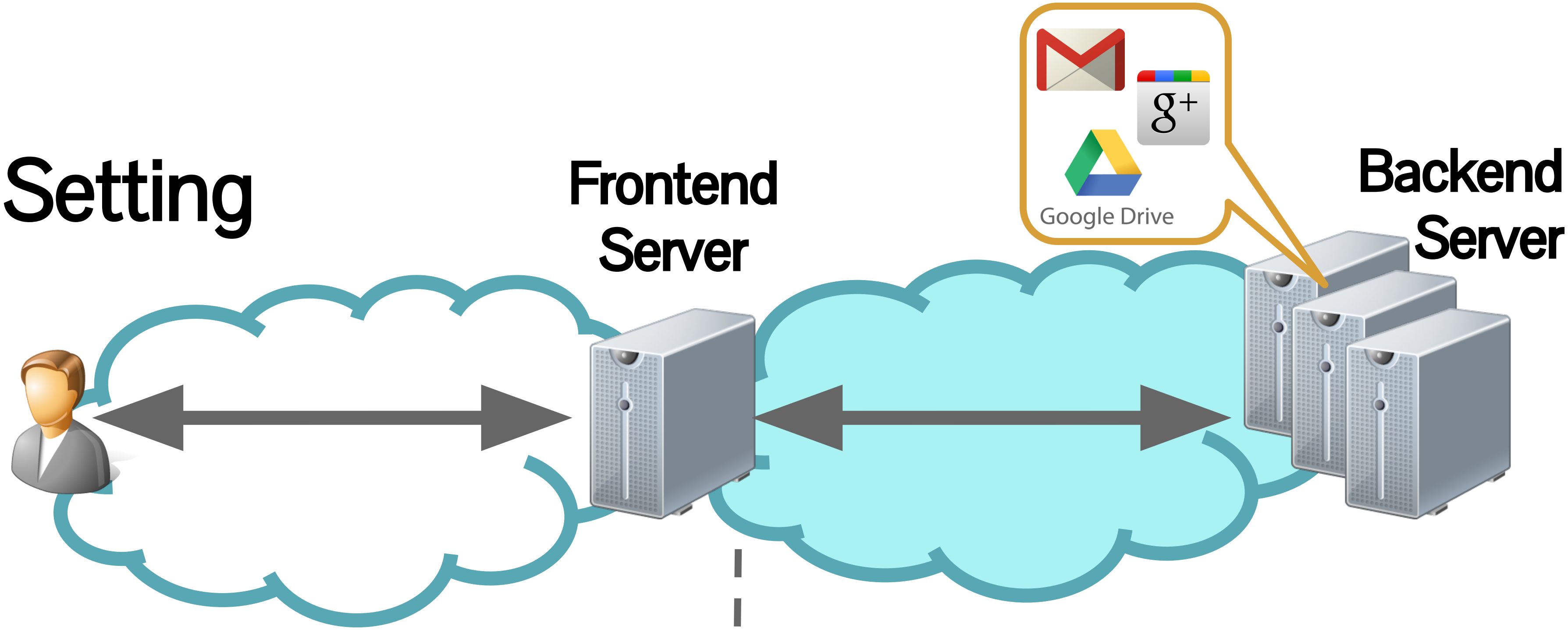
Setting



Public Network	Private Network
Reactive: Trigger fast retransmit by retransmitting the tail packet early	**Proactive**: Avoid retransmissions through packet duplication

Corrective: Add redundancy to enable recovery without retransmission, or trigger fast retransmit

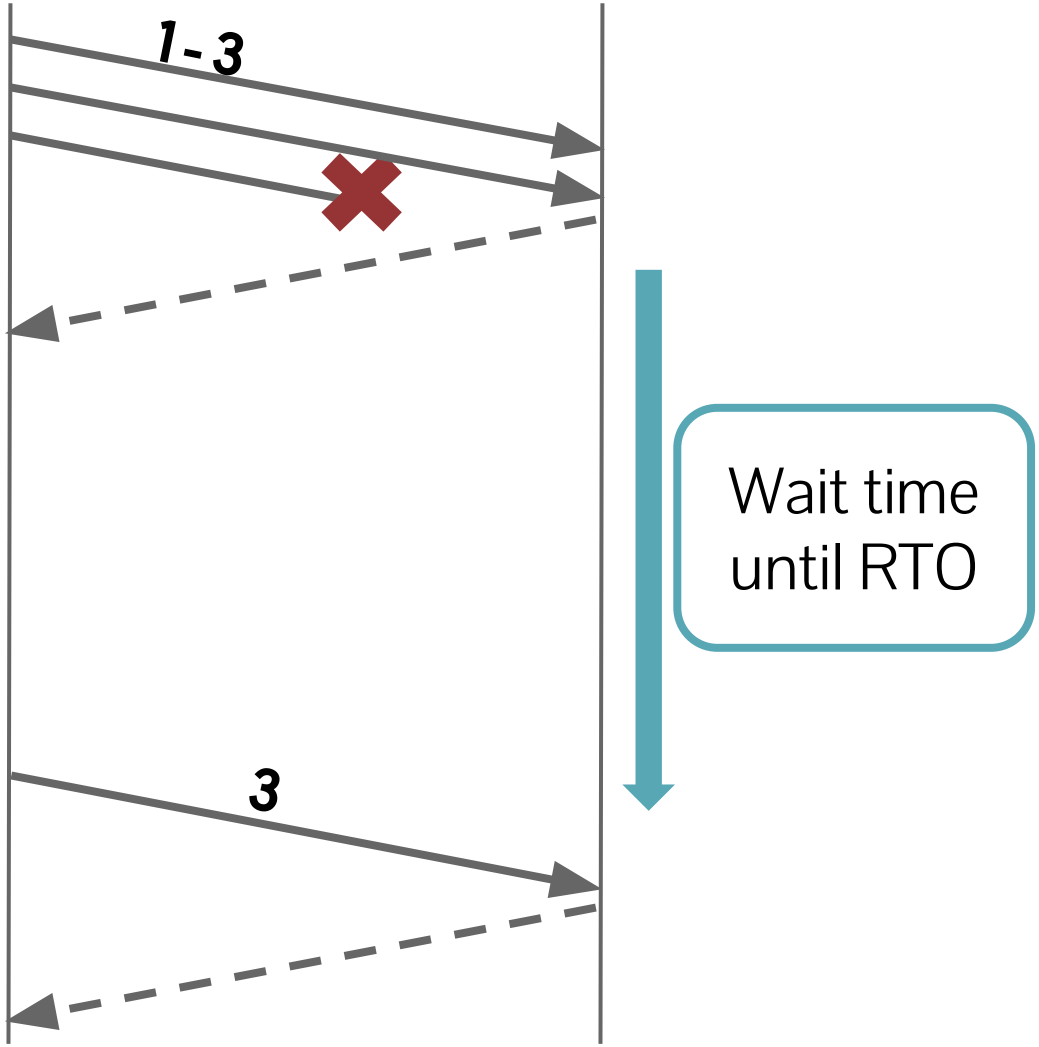
Setting



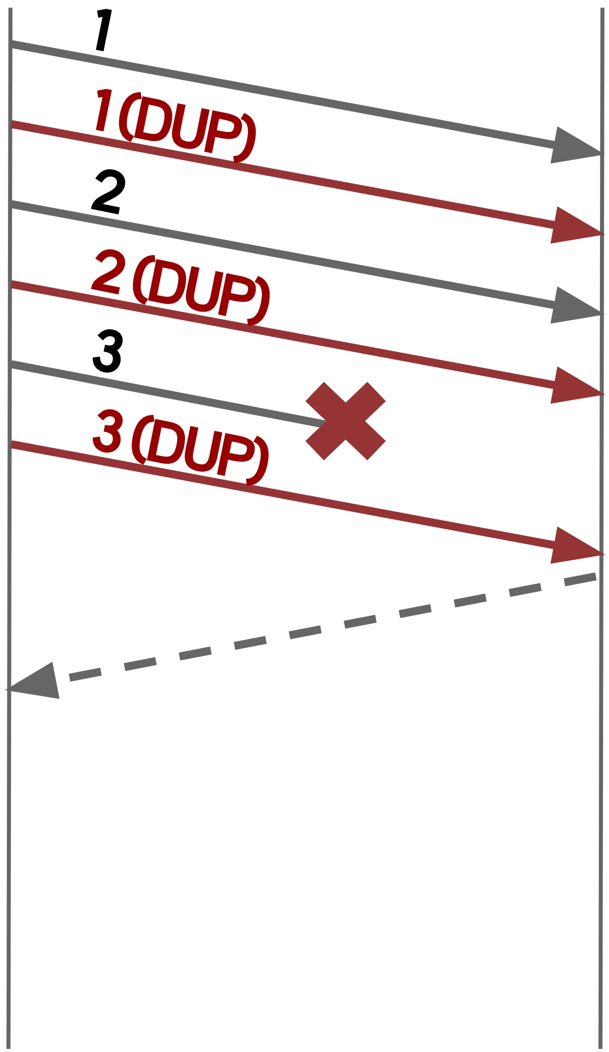
Public Network	Private Network
Reactive — Trigger fast retransmit by retransmitting the tail packet early	**Proactive** — Avoid retransmissions through packet duplication

Corrective — Add redundancy to enable recovery without retransmission, or trigger fast retransmit

Proactive



Proactive

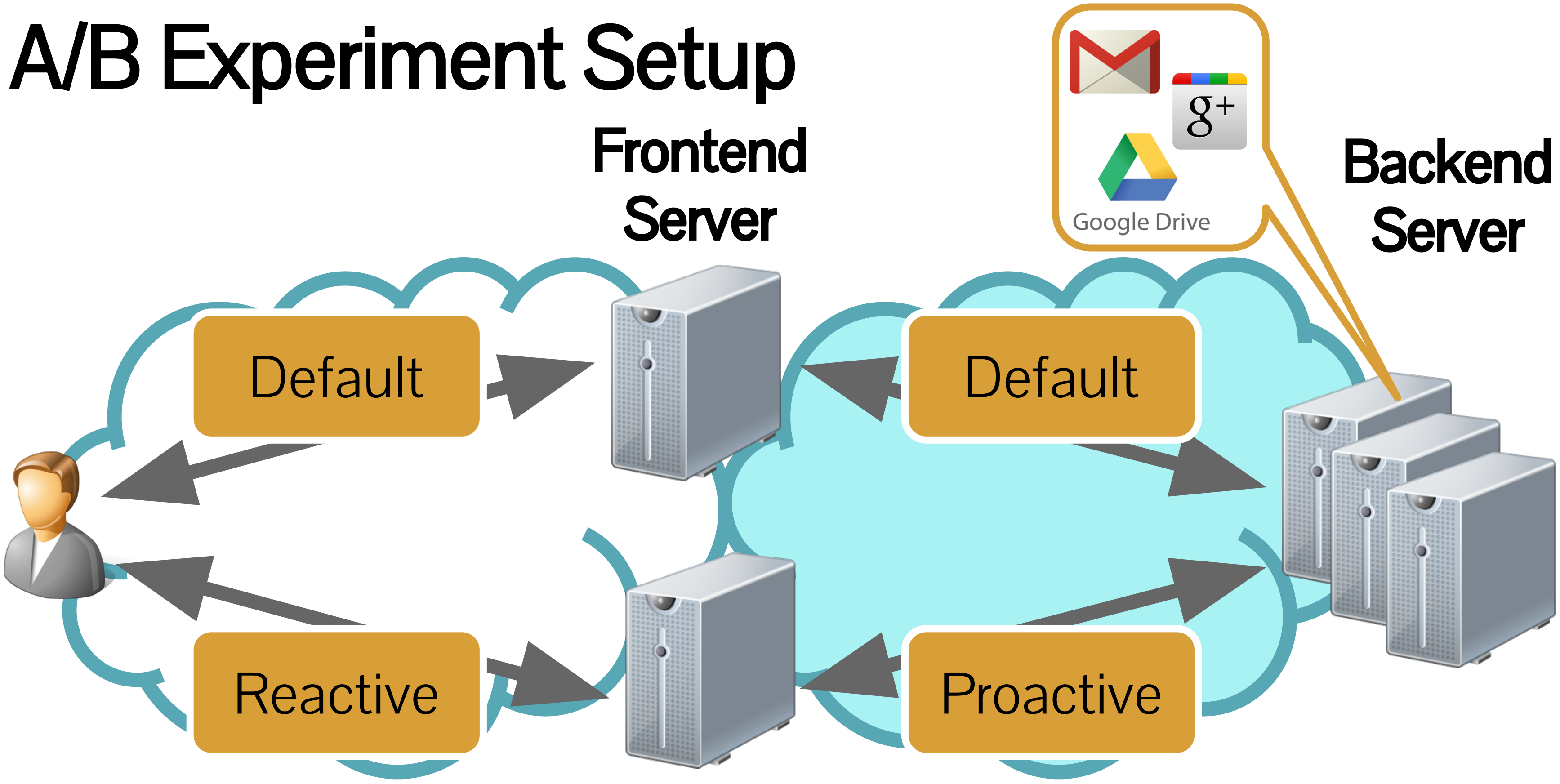


Avoid almost all retransmissions through packet duplication

Duplicates are used if original transmission was lost

Avoids loss detection and recovery

A/B Experiment Setup



Experimented in production environment
serving billions of queries
(millions of queries are sampled)

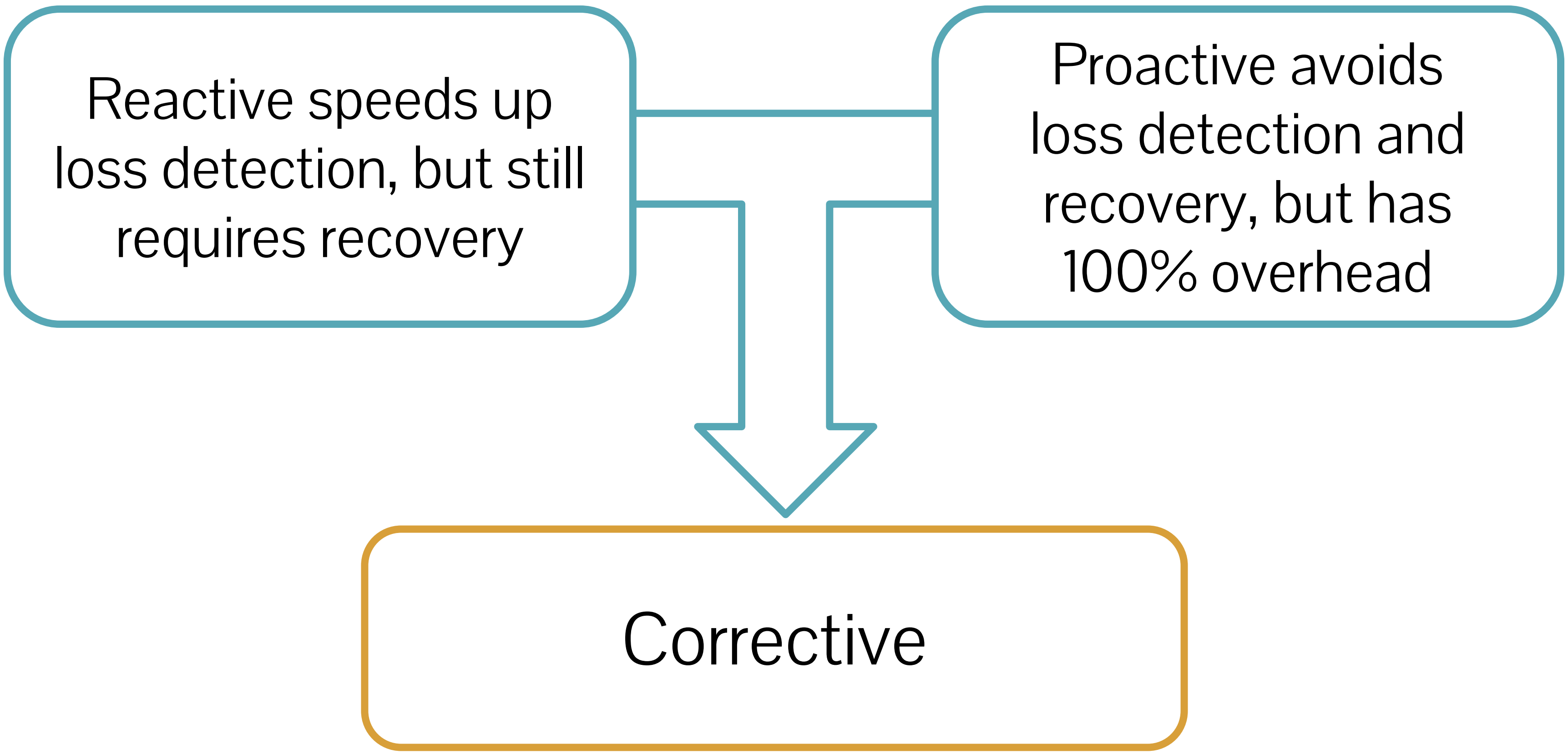
Impact of Reactive and Proactive

15-day experiment, 2.6 million queries sampled:
mean response time reduced by 23%
99th percentile response time reduced by 47%

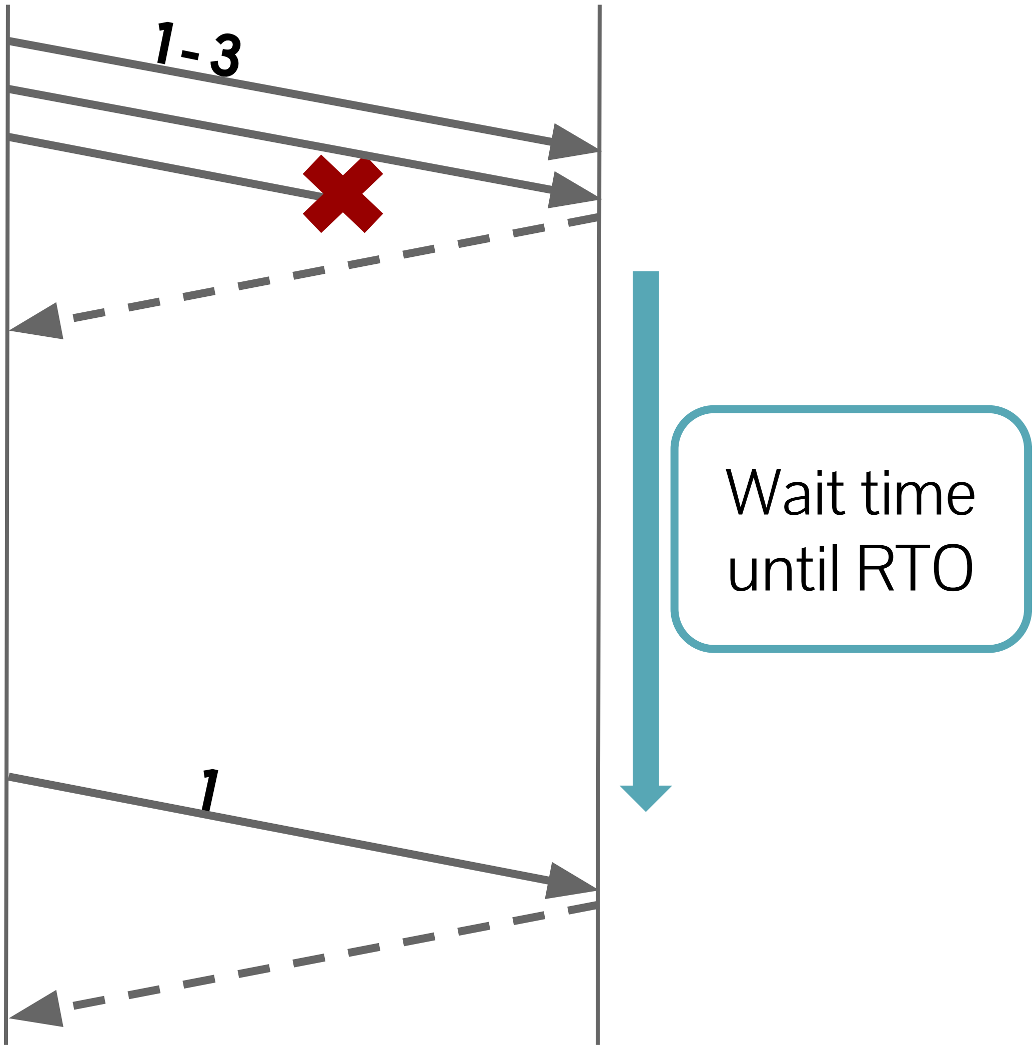
Impact of Proactive:
Retransmission rates on the backend connection dropped from 0.99% to 0.09%

Impact of Reactive:
Almost 50% of retransmission timeouts on the frontend connection are converted to fast retransmits

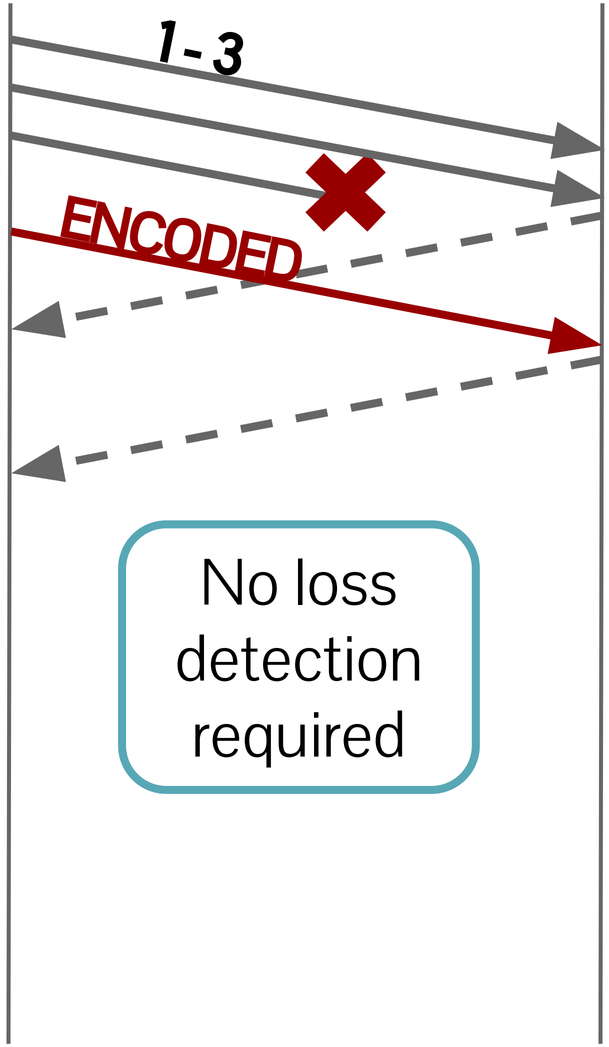
Corrective: The Middle Way



Corrective:
Forward Error Correction in TCP



Corrective:
Forward Error Correction in TCP



No loss detection required

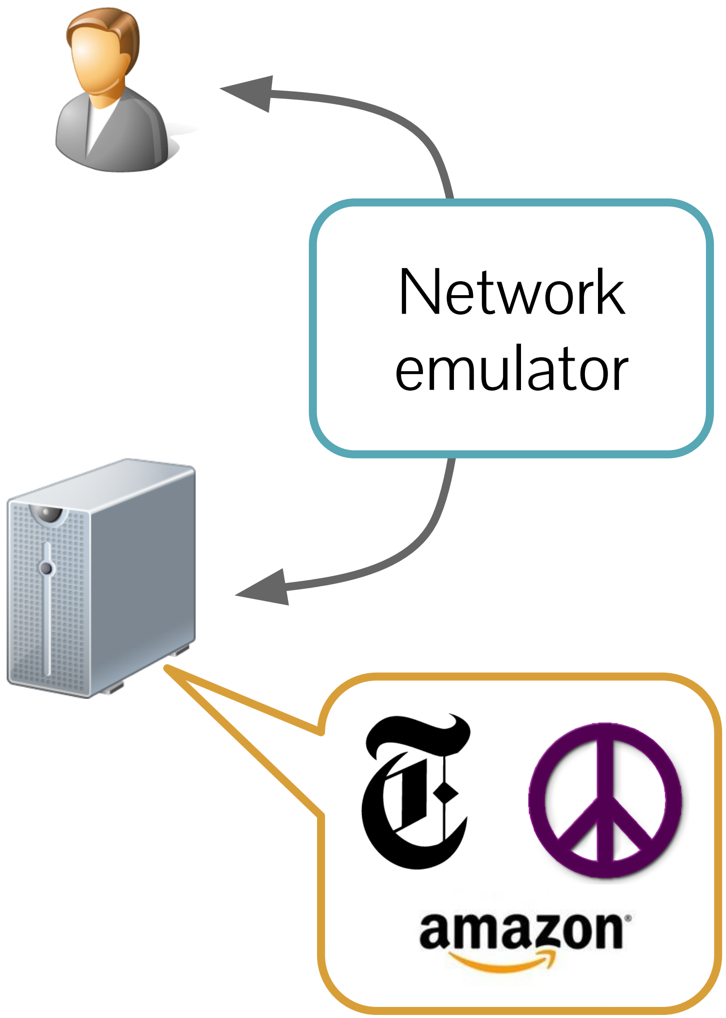
Encodes previously transmitted segments in few coded segments

XOR coding can recover single packet loss at the receiver

Signaling of recovery status to the sender to enforce congestion control or fast retransmit

Speeds up loss detection and recovery

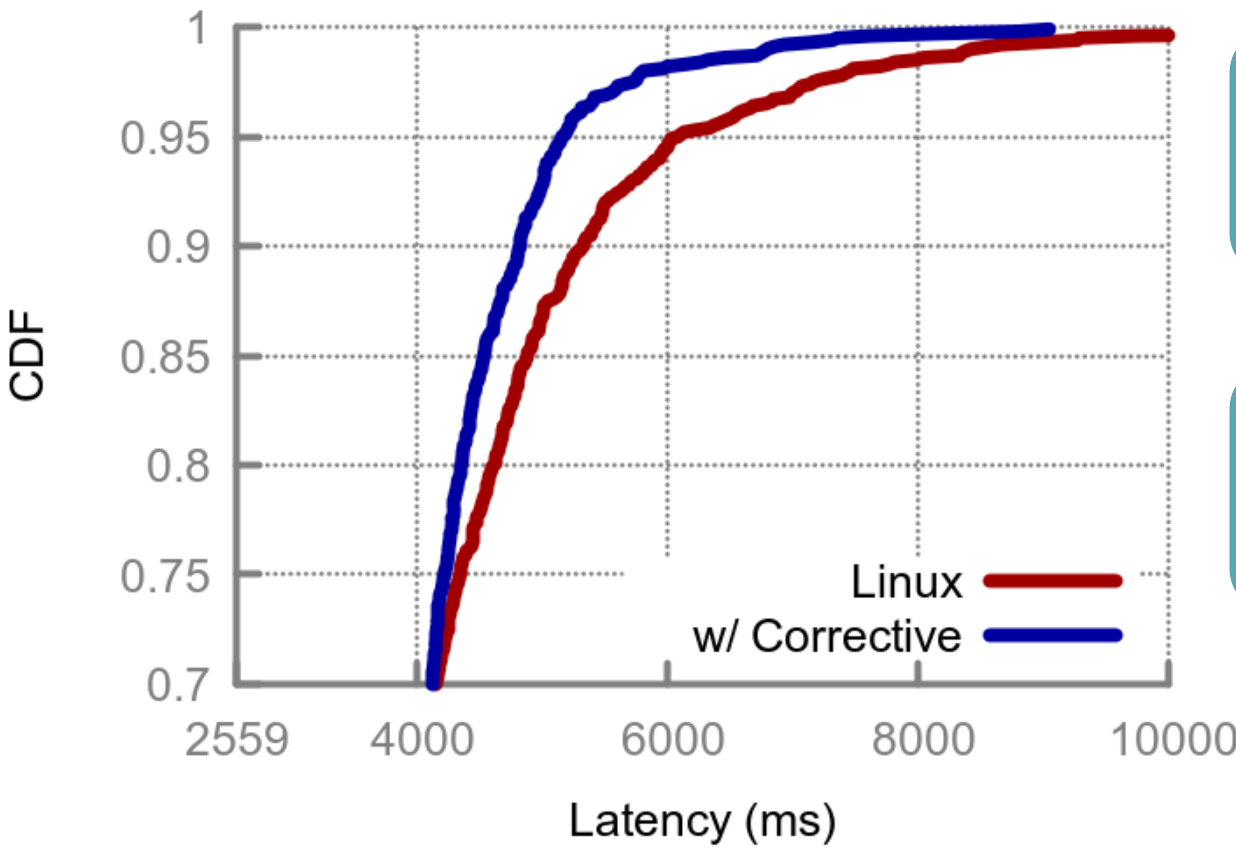
Evaluation: Corrective



Synthetic workloads
(fixed-size single queries)

Web page downloads
(complex multi-resource queries)

Loading nytimes.com with Corrective



Tail latency reduced by more than 20%

But: performance slightly worse on loss-free connections

Dealing with Middleboxes

Protocol changes need to account for middlebox interference

We designed our modules for middlebox compatibility or graceful fallback to standard TCP

Dealing with Middleboxes

Unknown option in data packet is stripped → Require option in all packets

ACK number is rewritten for unseen sequences → Resend lost segment to update middlebox state

Modified retransmission payload is rejected → Detect tampering through checksum

Summary: Reducing Web Latency

- In a measurement study analyzing billions of flows in Google's production environment, we found that performance deteriorates drastically when encountering packet loss

- Analysis of loss patterns motivated three designs to improve latency: Reactive, Proactive, and Corrective

- Reactive and Proactive improved Google's mean response time by 23%

- Reactive and Corrective are IETF Internet Drafts; Reactive is implemented and enabled by default in Linux 3.10

What aspects of TCP are limiting Web access performance, and how can we overcome these limitations?

Overview

A. Reducing Web Latency:
   The Virtue of Gentle Aggression

B. Understanding TCP Flow Performance at Scale
   Through Behavioral Signatures

Understanding TCP Flow Behavior

… can be difficult, because:

- TCP's complexity has increased dramatically over the last couple of decades, e.g. added features like:
  - Window scaling
  - PAWS
  - Segmentation offload
  - Prevention of bursty transmissions
  - Early congestion indicators

- Often working with packet captures only → have to reverse engineer the protocol behavior solely based on packets observed on the wire

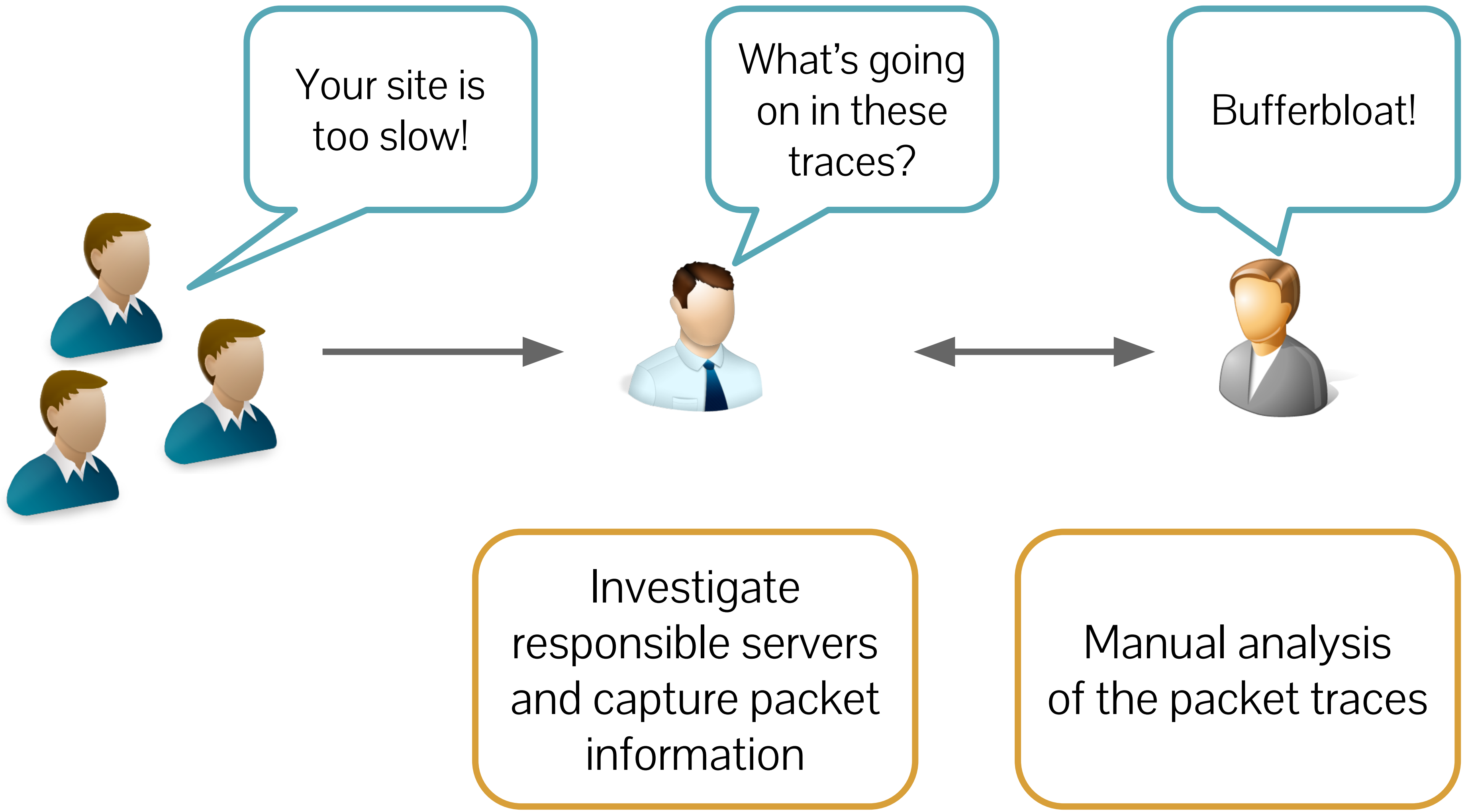
Same Rtx Ratios, Different Impact

- Could observe same retransmission ratios in two scenarios but the underlying causes and the impact on performance are widely different.

- <u>Bufferbloat:</u> Routers use large queue buffers to absorb traffic bursts, but a single flow can fill up the buffer leading to excessive queuing delays → slow recovery in case of packet loss

- <u>Reordering induced by packet-based load balancers:</u> Multiple paths with different delays → out-of-order delivery causing spurious retransmissions which require time to recover and impact throughput

Analyzing Performance Anomalies

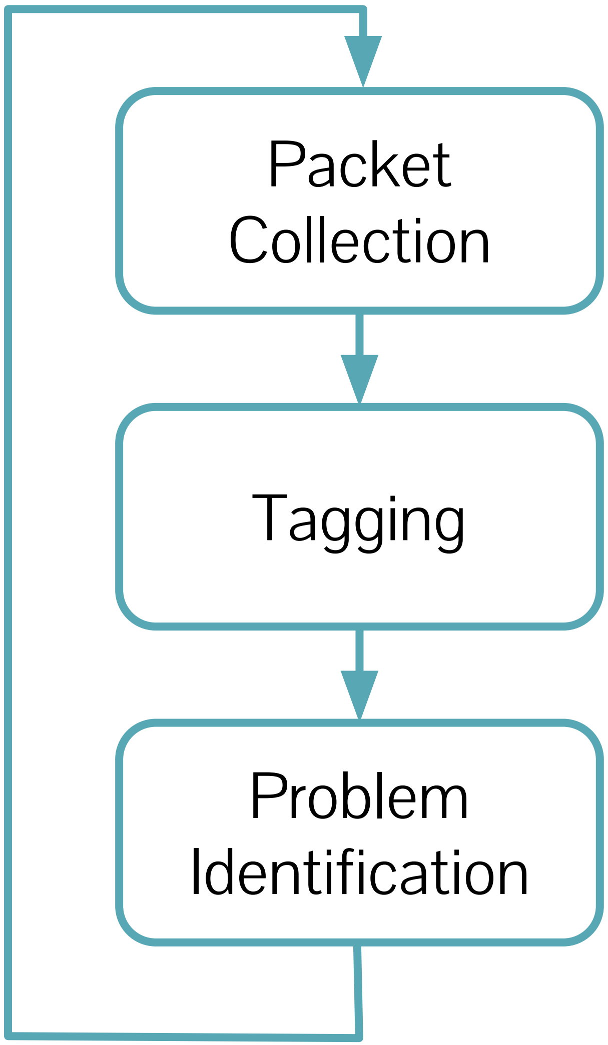


Analyzing Performance Anomalies: Automatic, at Scale

Sample servers capture packet information

Automatic analysis of the packet traces

Automatic Packet Analysis



- Monitoring engine capturing all packets for sample set of connections

- Use behavioral signatures to tag packets, connections, and queries

- Correlate tags with performance metrics
- Manual analysis through visualization and database queries

Analysis Building Blocks: Behavioral Signatures

- Per packet
  - Basic signatures based on header information extracted from few packets
  - Examples: acknowledgements, packet loss
- Per flow
  - Aggregates data from many packets revealing behavior observed over longer time periods
  - Examples: bufferbloat, traffic policing
- Per application entity (e.g. HTTP query)
  - Incorporates non-TCP information, e.g. payload content, application-layer feedback
  - Examples: query response time, video stalling

Next Steps

- Collect packets at scale on web service front-ends for multiple weeks to record transient and persistent performance anomalies

- Identify root causes for traffic patterns associated with tail performance (e.g. tail query response time)

- Derive a set of protocol and/or network changes to address the performance problems and possibly quantify their impact

Conclusion

- Good Web performance requires good TCP performance

- Need to understand the causes of poor tail performance enabling the design of new solutions

- Our approach:
    - Automate the analysis of packet traces at scale to find problems
    - Design solutions tailored to the architectures of modern Web services
    - Deployed loss recovery mechanisms have large impact on Google performance

Understanding and Improving TCP for Web Performance

Tobias Flach
http://nsl.cs.usc.edu/~tobiasflach

Nandita Dukkipati, Andreas Terzis, Barath Raghavan, Neal Cardwell, Yuchung Cheng, Ankur Jain, Shuai Hao, Ethan Katz-Bassett, Ramesh Govindan

USC Viterbi School of Engineering