Elephants, Mice, and Lemmings! Oh My!

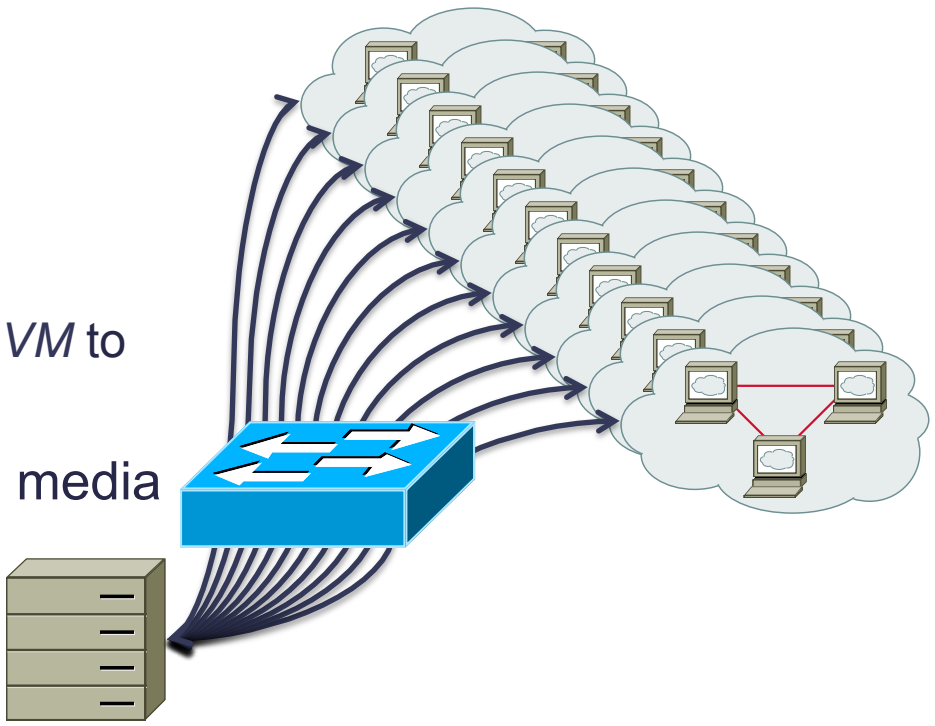
Making life better in data centers and high speed computing

Fred Baker
Fellow

25 July 2014

Data Center Applications

- Names withheld for customer/vendor confidentiality reasons
- Common social networking applications might have
  - $O(10^3)$ racks in a data center
  - 42 1RU hosts per rack
  - A dozen Virtual Machines per host
  - $O(2^{19})$ virtual hosts per data center
  - $O(10^4)$ standing TCP connections *per VM* to other VMs in the data center
- When one opens a <pick your social media application> web page
  - Thread is created for the client
  - $O(10^4)$ requests go out for data
  - $O(10^4)$ 2-3 1460 byte responses come back
  - $O(45 \times 10^6)$ bytes in switch queues **instantaneously**
  - At 10 GBPS, **instant 36 ms queue depth**



© 2013-2014 Cisco and/or its affiliates. All rights reserved.

Taxonomy of data flows

- We are pretty comfortable with the concepts of mice and elephants
  - “mice”: small sessions, a few RTTs total
  - “elephants”: long sessions with many RTTs
- In Data Centers with Map/Reduce applications, we also have ***lemmings***
  - O(10^4) mice migrating together
- Solution premises
  - Mice: we don’t try to manage these
  - Elephants: if we can manage them, network works
  - Lemmings: Elephant-oriented congestion management results in HOL blocking

© 2013-2014 Cisco and/or its affiliates. All rights reserved.

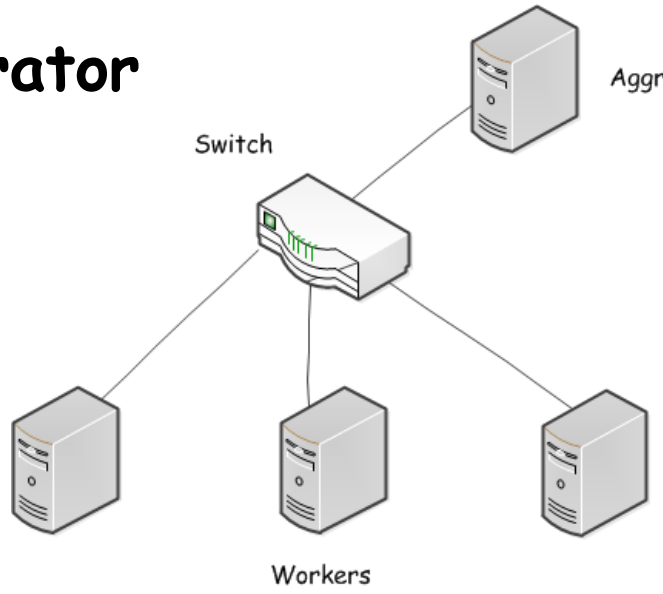
My question

- Most proposals I see, in one way or another, attempt to use AQM to manage latency, by responding to traffic aggressively.
- **What if we're going at it the wrong way?**
  - What if the right way to handle latency on short RTT timescales is from TCP "congestion" control, using delay-based or jitter-based procedures?
- What procedures?
  - TCP Vegas (largely discredited as a congestion control procedure)
  - CalTech FAST (blocked by IPR and now owned by Akamai)
  - CAIA Delay Gradient (CDG), in FreeBSD but disabled by a bug

© 2013-2014 Cisco and/or its affiliates. All rights reserved.

Technical Platform

- Machines
  - Hosts with 3.1GHz CPU, 2GB RAM and 1Gbps NIC (4)
  - NetFPGA
  - Freebsd 9.2-prerelease
- Multi-thread traffic generator
  - Each responses 64KB
  - Buffer: 128KB

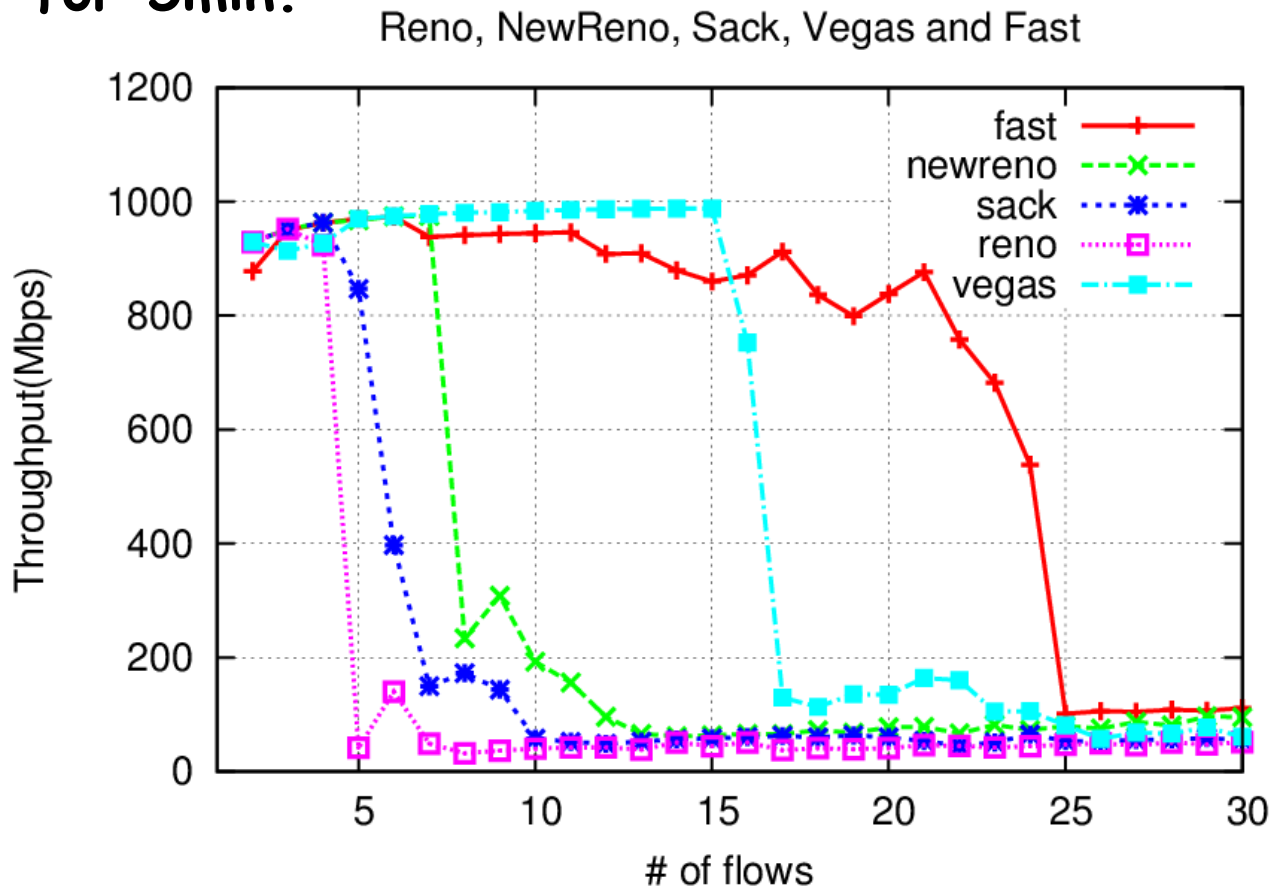


Courtesy Tsinghua University
Cisco/Tsinghua Joint Lab

© 2013-2014 Cisco and/or its affiliates. All rights reserved.

TCP Performance on short RTT timeframes

- Each flow responses 100KB data
  - Last for 5min.

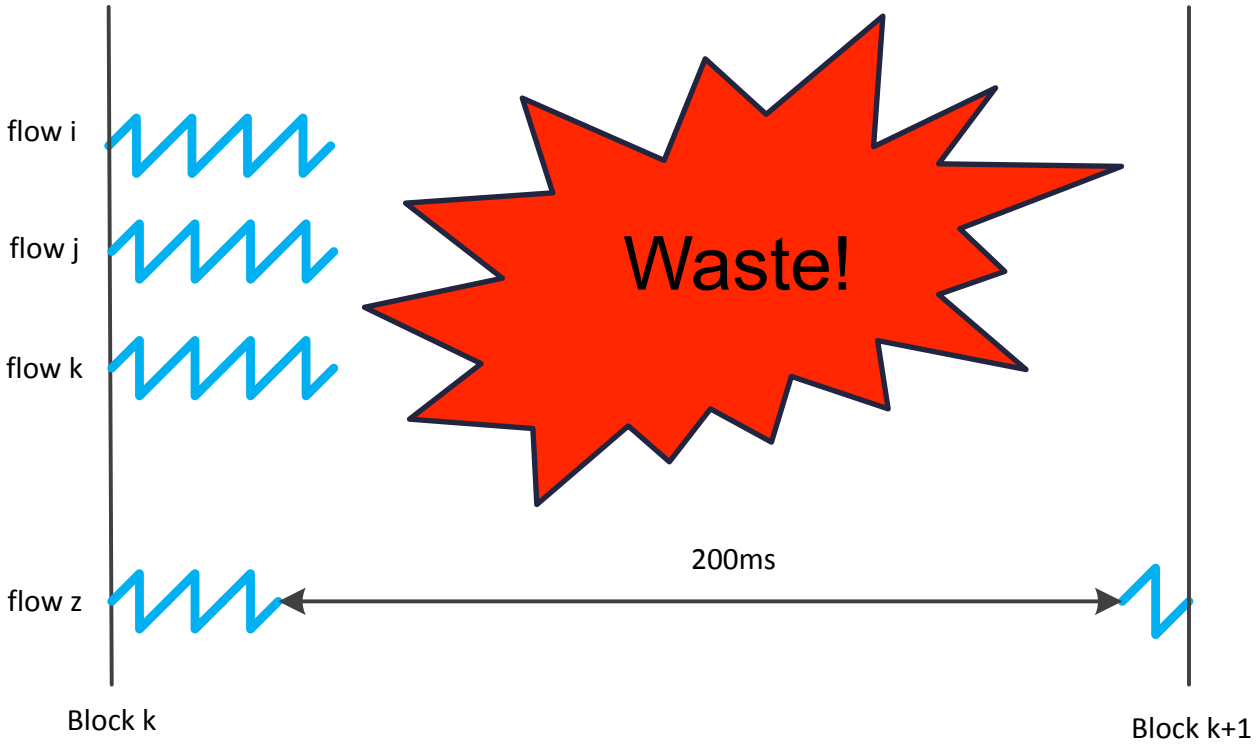


Courtesy Tsinghua University
Cisco/Tsinghua Joint Lab

© 2013-2014 Cisco and/or its affiliates. All rig

Effects of TCP Timeout

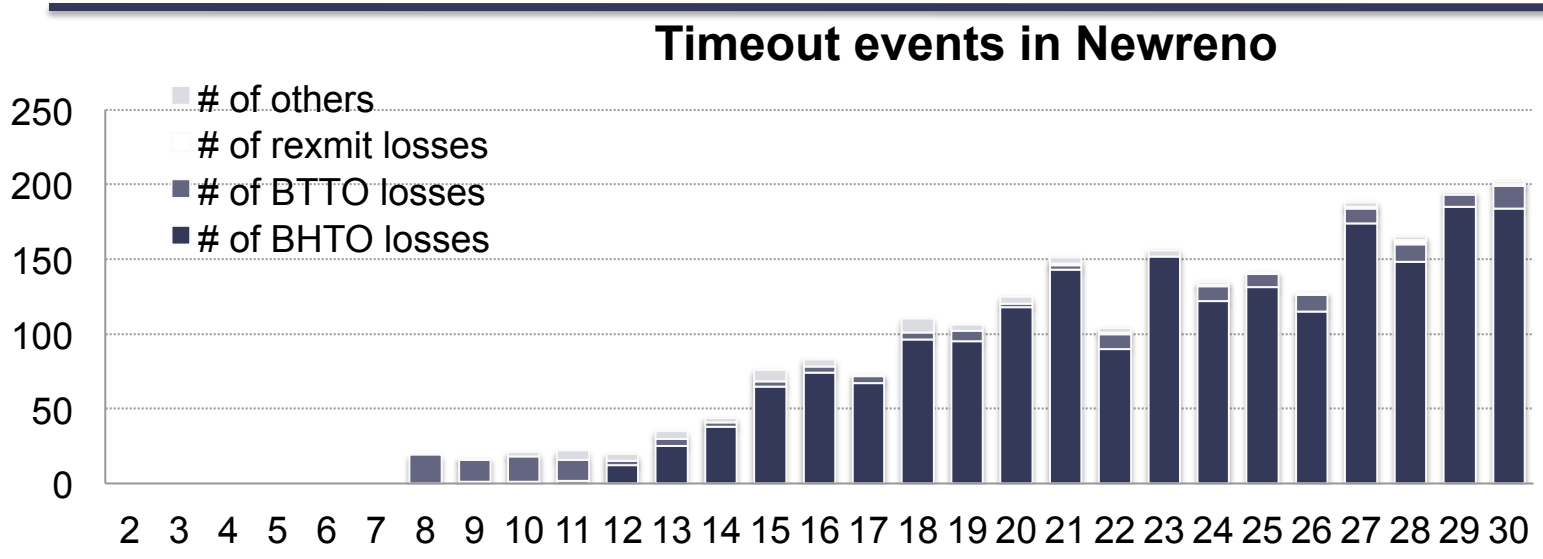
- The ultimate reason for throughput collapse in Incast is timeout.

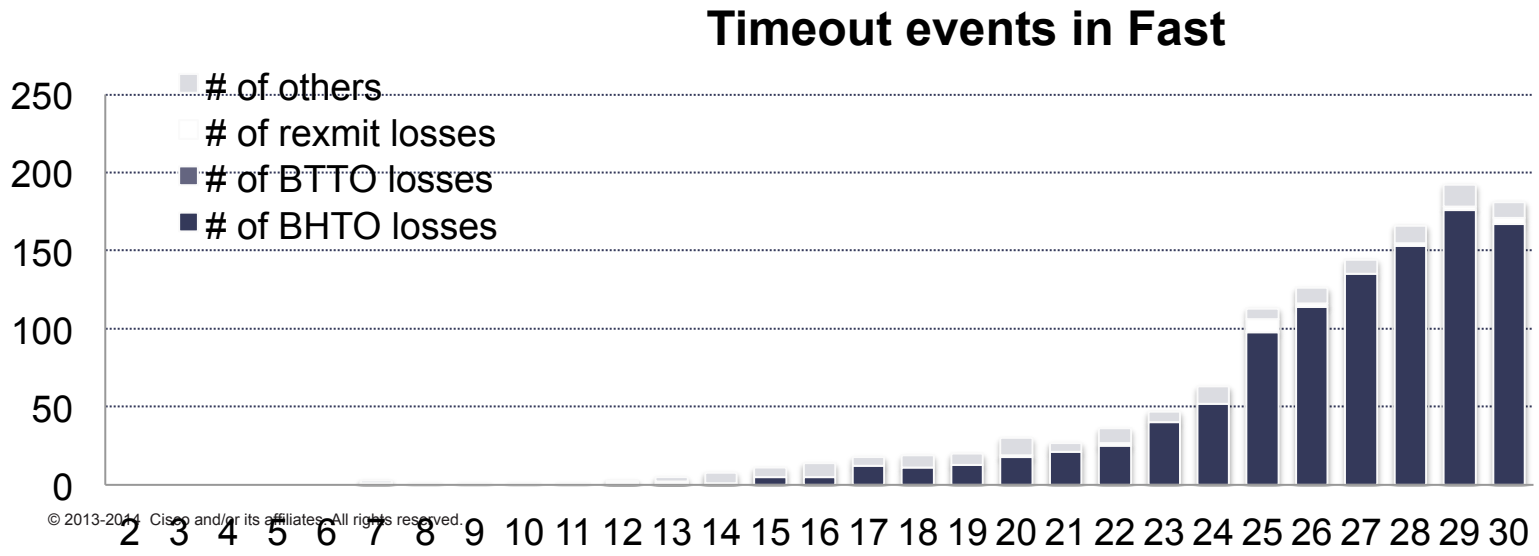


Courtesy Tsinghua University
Cisco/Tsinghua Joint Lab

© 2013-2014 Cisco and/or its affiliates. All rights reserved.

Prevalence of TCP Timeout





Courtesy Tsinghua University
Cisco/Tsinghua Joint Lab

© 2013-2014 Cisco and/or its affiliates. All rights reserved.

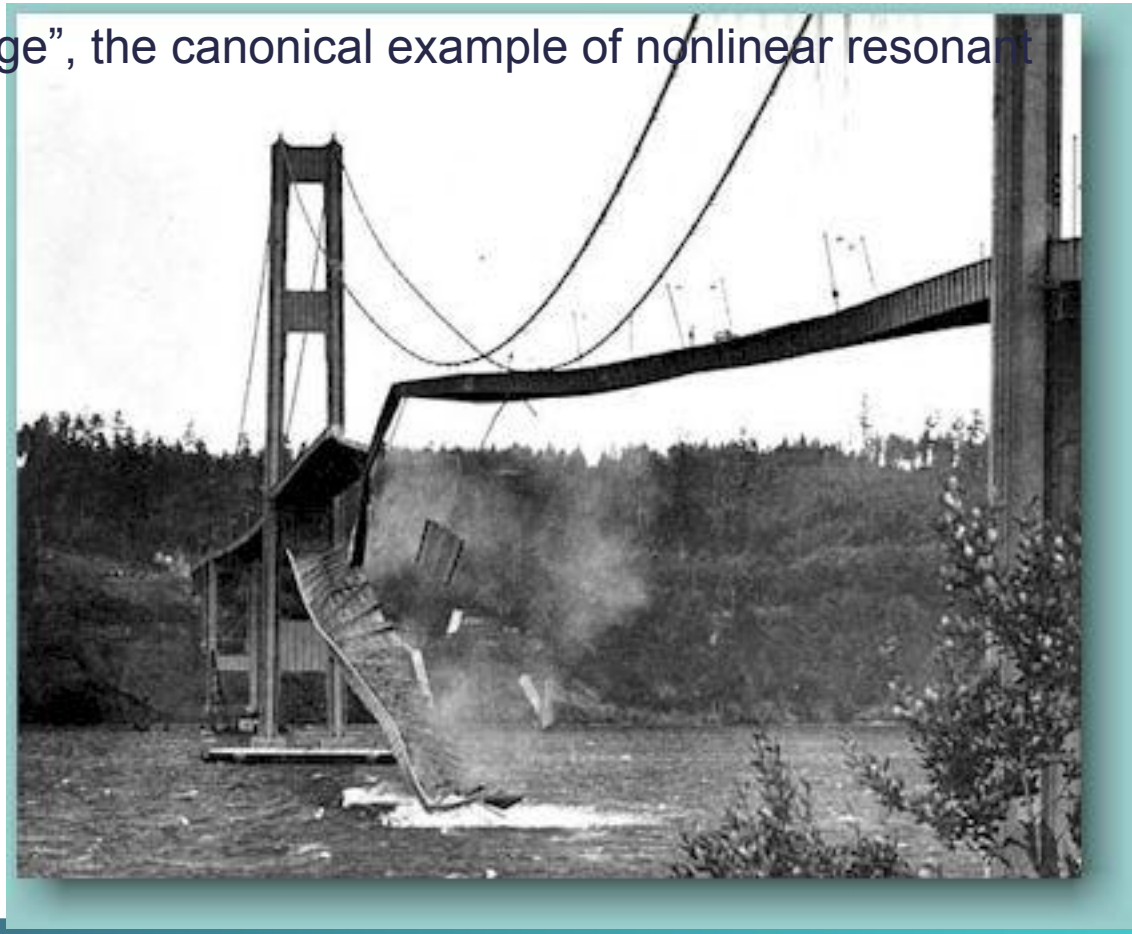
Tsinghua conclusions

- Using a Delay-based procedure helped quite a bit, but didn’t solve incast cold.
  - It did, however, significantly increase TCP’s capability to maximize throughput, minimize latency, and improve reliability on short timescales.
- We also need something else to fix the incast problem, probably at the application layer in terms of how many VMs are required

© 2013-2014 Cisco and/or its affiliates. All rights reserved.

What's the other half of the incast problem?

- In two words, amplification and coupling.
- Amplification Principle
  - Non-linearities occur at large scale which do not occur at small to medium scale.
  - Think "Tocoma Narrows Bridge", the canonical example of nonlinear resonant amplification in physics



  - RFC 3439

© 2013-2014 Cisco and/or its affiliates. All rights reserved.

What's the other half of the incast problem?

- Coupling Principle
  - As things get larger, they often exhibit increased interdependence between components.

- When a request is sent to $O(10^4)$ other machines and they all respond
  - Bad things happen…

© 2013-2014 Cisco and/or its affiliates. All rights reserved.

Large scale shared-nothing analytic engine

- *Time to start looking at next generation analytics*
- UCSD CNS – moving away from rotating storage to solid-state drives dramatically improves Tritonsort while reducing VM count.
- Facebook: uses Memcache as basic storage medium

Google Dumps MapReduce in Favor of New Hyper-Scale Analytics System

BY YEVGENIY SVERDLIK ON JUNE 25, 2014 2 COMMENTS

Tweet

Google has abandoned MapReduce, the system for running data analytics jobs spread across many servers the company developed and later open sourced, in favor of a new cloud analytics system it has built called Cloud Dataflow.

MapReduce has been a highly popular infrastructure and programming model for doing parallelized distributed computing on server clusters. It is the basis of Apache Hadoop, the Big Data infrastructure platform that has enjoyed widespread deployment and become core of many companies' commercial products.

The technology is unable to handle the amounts of data Google wants to analyze these days, however. Urs Hölzle, senior vice president of technical infrastructure at the Mountain View, California-based giant, said it got too cumbersome once the size of the data reached a few petabytes.

"We don't really use MapReduce anymore," Hölzle said in his keynote presentation at the Google I/O conference in San Francisco Wednesday. The company stopped using the system "years ago."

© 2013-2014 Cisco and/or its affiliates. All rights reserved.

My view

- TCP and related protocols should use a delay-based or jitter-based procedure such as FAST or CDG. This demonstrably helps maximize throughput while minimizing latency, and does better than loss-based procedures on short timescales.
  - What other timescales? There are known issues with TCP Congestion Control on long delay links.
  - Note that Akamai owns the CalTech FAST technology, presumably with the intent to use it on some timescales, and Amazon appears to use it within data centers.
- Ongoing work to fix CDG in FreeBSD 10.0.
- What do we need to do to move away from Map/Reduce applications or limit their VM count besides using solid-state storage and shared-nothing architectures?

© 2013-2014 Cisco and/or its affiliates. All rights reserved.

Thank you.