The zero-bit alternative

Jana Iyengar

"efficient use of the often volatile radio bearer"

When the radio link is the bottleneck on the end-to-end path:

- Queue build-up occurs at (or near) the radio link
- Packets from flows in queue need to be scheduled

This is a scheduling problem.

Also want queues to not be persistently large.

This is a queue management problem.

FQ-CoDel

- Hybrid packet scheduler and Active Queue Management algorithm
- Provides traffic isolation between competing entities (flows, users)
- AQM provides low delay
- No ECN marking required → immediately deployable

FQ-CoDel

A bit more detail:

- New / sparse flows scheduled immediately
- Flows with continuous backlog scheduled after sparse flows
- No explicit classification needed

draft-ietf-aqm-fq-codel-06

TCP does not perform well on cellular networks

Evolving congestion control landscape

2012

Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks

Keith Winstein, Anirudh Sivaraman, and Hari Balakrishnan
MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, Mass.
`{keithw,anirudh,hari}@mit.edu`

Abstract

Sprout is an end-to-end transport protocol for interactive applications that desire high throughput and low delay. Sprout works well over cellular wireless networks, where link speeds change dramatically with time, and current protocols build up multi-second queues in network gateways. Sprout does not use TCP-style reactive congestion control; instead the receiver observes the packet arrival times to infer the uncertain dynamics of the network path. This inference is used to forecast how many bytes may be sent by the sender, while bounding the risk that packets will be delayed inside the network for too long.

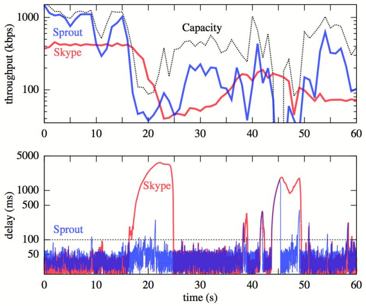
In evaluations on traces from four commercial LTE and 3G networks, Sprout, compared with Skype, reduced self-inflicted end-to-end delay by a factor of 7.9 and achieved 2.2× the transmitted bit rate on average. Compared with Google's Hangout, Sprout reduced delay by a factor of 7.2 while achieving 4.4× the bit rate, and compared with Apple's Facetime, Sprout reduced delay by a factor of 8.7 with 1.9× the bit rate.

Although it is end-to-end, Sprout matched or outperformed TCP Cubic running over the CoDel active queue management algorithm, which requires changes to cellular carrier equipment to deploy. We also tested Sprout as a tunnel to carry competing interactive and bulk traffic (Skype and TCP Cubic), and found that Sprout was able to isolate client application flows from one another.

1 Introduction

Cellular wireless networks have become a dominant mode of Internet access. These mobile networks, which include LTE and 3G (UMTS and 1xEV-DO) services, present new challenges for network applications, because they behave differently from wireless LANs and from the Internet's traditional wired infrastructure.

Figure 1: Skype and Sprout on the Verizon LTE downlink trace. For Skype, overshoots in throughput lead to large standing queues. Sprout tries to keep each packet's delay less than 100 ms with 95% probability.



For an interactive application such as a videoconferencing program that requires both high throughput and low delay, these conditions are challenging. If the application sends at too low a rate, it will waste the opportunity for higher-quality service when the link is doing well. But when the application sends too aggressively, it accumulates a queue of packets inside the network waiting to be transmitted across the cellular link, delaying subsequent packets. Such a queue can take several seconds to drain, destroying interactivity (see Figure 1).

Our experiments with Microsoft's Skype, Google's Hangout, and Apple's Facetime running over traces from

Evolving congestion control landscape

2012

Stochastic Forecasts Achieve

Keith Winste
MIT Computer Science
{keithw,

Abstract

Sprout is an end-to-end transport protocol for inte
applications that desire high throughput and low
Sprout works well over cellular wireless networks,
link speeds change dramatically with time, and
protocols build up multi-second queues in networ
ways. Sprout does not use TCP-style reactive c
tion control; instead the receiver observes the pac
rival times to infer the uncertain dynamics of the n
path. This inference is used to forecast how many
may be sent by the sender, while bounding the ri
packets will be delayed inside the network for too

In evaluations on traces from four commercia
and 3G networks, Sprout, compared with Skype, r
self-inflicted end-to-end delay by a factor of 7
achieved 2.2× the transmitted bit rate on average.
pared with Google's Hangout, Sprout reduced dela
factor of 7.2 while achieving 4.4× the bit rate, an
pared with Apple's Facetime, Sprout reduced dela
factor of 8.7 with 1.9× the bit rate.

Although it is end-to-end, Sprout matched or
formed TCP Cubic running over the CoDel active
management algorithm, which requires changes to
lar carrier equipment to deploy. We also tested Sp
a tunnel to carry competing interactive and bulk
(Skype and TCP Cubic), and found that Sprout w
to isolate client application flows from one anothe

1 Introduction

Cellular wireless networks have become a do
mode of Internet access. These mobile networks,
include LTE and 3G (UMTS and 1xEV-DO) se
present new challenges for network applications, b
they behave differently from wireless LANs and fr
Internet's traditional wired infrastructure.

2015

PCC: Re-architecting Congestion Control for Consistent High Performance

Mo Dong*, Qingxi Li*, Doron Zarchy**, P. Brighten Godfrey*, and Michael Schapira**

*University of Illinois at Urbana-Champaign
**Hebrew University of Jerusalem

Abstract

TCP and its variants have suffered from surprisingly poor performance for decades. We argue the TCP family has little hope of achieving consistent high performance due to a fundamental architectural deficiency: hardwiring packet-level events to control responses. We propose Performance-oriented Congestion Control (PCC), a new congestion control architecture in which each sender continuously observes the connection between its *actions* and *empirically experienced performance*, enabling it to consistently adopt actions that result in high performance. We prove that PCC converges to a stable and fair equilibrium. Across many real-world and challenging environments, PCC shows consistent and often 10× performance improvement, with better fairness and stability than TCP. PCC requires no router hardware support or new packet format.

1 Introduction

In the roughly 25 years since its deployment, TCP's congestion control architecture has been notorious for degraded performance. TCP performs poorly on lossy links, penalizes high-RTT flows, underutilizes high bandwidth-delay product (BDP) connections, cannot handle rapidly changing networks, can collapse under data center incast [24] and incurs very high latency with bufferbloat [28] in the network.

As severe performance problems have accumulated over time, protocol "patches" have addressed problems in specific network conditions such as high BDP links [31,52], satellite links [23,42], data center [18,55], a very difficult task within TCP's rate control architecture, which we refer to as **hardwired mapping**: certain predefined packet-level events are hardwired to certain predefined control responses. TCP reacts to events that can be as simple as "one packet loss" (TCP New Reno) or can involve multiple signals like "one packet loss and RTT increased by *x*%" (TCP Illinois). Similarly, the control response might be "halve the rate" (New Reno) or a more complex action like "reduce the window size *w* to $f(\Delta RTT)w$" (Illinois). The defining feature is that the control action is a direct function of packet-level events.

A hardwired mapping has to make *assumptions* about the network. Take a textbook event-control pair: a packet loss halves the congestion window. TCP *assumes* that the loss indicates congestion in the network. When the assumption is violated, halving the window size can severely degrade performance (e.g. if loss is random, rate should stay the same or increase). It is fundamentally hard to formulate an "always optimal" hardwired mapping in a complex real-world network because the actual optimal response to an event like a loss (i.e. decrease rate or increase? by how much?) is sensitive to network conditions. And modern networks have an immense diversity of conditions: random loss and zero loss, shallow queues and bufferbloat, RTTs of competing flows varying by more than 1000×, dynamics due to mobile wireless or path changes, links from Kbps to Gbps, AQMs, software routers, rate shaping at gateways, virtualization layers and middleboxes like firewalls, packet inspectors and load balancers. These factors add complexity far beyond what can be summarized by the relatively simplistic assumptions embedded in a hardwired mapping. Most

Evolving congestion control landscape

2012

Stochastic Forecasts Achieve I

Keith Winste
MIT Computer Science
{keithw,

Abstract

Sprout is an end-to-end transport protocol for inte
applications that desire high throughput and low
Sprout works well over cellular wireless networks,
link speeds change dramatically with time, and
protocols build up multi-second queues in networ
ways. Sprout does not use TCP-style reactive c
tion control; instead the receiver observes the pac
rival times to infer the uncertain dynamics of the n
path. This inference is used to forecast how many
may be sent by the sender, while bounding the ri
packets will be delayed inside the network for too
In evaluations on traces from four commercia
and 3G networks, Sprout, compared with Skype, r
self-inflicted end-to-end delay by a factor of 7
achieved 2.2× the transmitted bit rate on average.
pared with Google's Hangout, Sprout reduced dela
factor of 7.2 while achieving 4.4× the bit rate, an
pared with Apple's Facetime, Sprout reduced dela
factor of 8.7 with 1.9× the bit rate.
Although it is end-to-end, Sprout matched or
formed TCP Cubic running over the CoDel active
management algorithm, which requires changes to
lar carrier equipment to deploy. We also tested Sp
a tunnel to carry competing interactive and bulk
(Skype and TCP Cubic), and found that Sprout w
to isolate client application flows from one anothe

1 Introduction

Cellular wireless networks have become a do
mode of Internet access. These mobile networks,
include LTE and 3G (UMTS and 1xEV-DO) se
present new challenges for network applications, b
they behave differently from wireless LANs and fr
Internet's traditional wired infrastructure.

2015

PCC: Re-architecting

Mo Dong*, Qingxi Li
*Univ

Abstract

TCP and its variants have suffered fro
performance for decades. We argue th
little hope of achieving consistent hig
to a fundamental architectural defic
packet-level events to control respo
Performance-oriented Congestion Co
congestion control architecture in w
continuously observes the connectio
tions and *empirically experienced perf*
it to consistently adopt actions that re
mance. We prove that PCC converg
fair equilibrium. Across many real-w
ing environments, PCC shows consist
performance improvement, with bette
bility than TCP. PCC requires no route
or new packet format.

1 Introduction

In the roughly 25 years since its d
congestion control architecture has b
degraded performance. TCP perform
links, penalizes high-RTT flows, u
bandwidth-delay product (BDP) co
handle rapidly changing networks, c
data center incast [24] and incurs very
bufferbloat [28] in the network.
As severe performance problems
over time, protocol "patches" have
lems in specific network conditions
links [31,52], satellite links [23,42], d

2015

Adaptive Congestion Control for Unpredictable Cellular Networks

Yasir Zaki
NYU Abu Dhabi
Abu Dhabi, UAE
yasir.zaki@nyu.edu

Thomas Pötsch
University of Bremen
Bremen, Germany
tpoetsch@uni-bremen.de

Jay Chen
NYU Abu Dhabi
Abu Dhabi, UAE
jchen@cs.nyu.edu

Lakshminarayanan Subramanian
NYU and CTED
New York, USA
lakshmi@cs.nyu.edu

Carmelita Görg
University of Bremen
Bremen, Germany
cg@comnets.uni-bremen.de

ABSTRACT

Legacy congestion controls including TCP and its variants are known to perform poorly over cellular networks due to highly variable capacities over short time scales, self-inflicted packet delays, and packet losses unrelated to congestion. To cope with these challenges, we present Verus, an end-to-end congestion control protocol that uses delay measurements to react quickly to the capacity changes in cellular networks without explicitly attempting to predict the cellular channel dynamics. The key idea of Verus is to continuously learn a delay profile that captures the relationship between end-to-end packet delay and outstanding window size over short epochs and uses this relationship to increment or decrement the window size based on the observed short-term packet delay variations. While the delay-based control is primarily for congestion avoidance, Verus uses standard TCP features including multiplicative decrease upon packet loss and slow start.

times marginally higher). In comparison to Sprout, Verus achieves up to 30% higher throughput in rapidly changing cellular networks.

CCS Concepts

•Networks → Network protocol design; Transport protocols; Network performance analysis;

Keywords

Congestion control, Cellular network, Transport protocol, Delay-based

1. INTRODUCTION

Cellular network channels are highly variable and users often experience fluctuations in their radio link rates over short time scales due to scarce radio resources making these channels hard to predict [26, 20, 7]. TCP and its variants are known to perform poorly over cellular

Zero-bit alternative

Content- and protocol-agnostic modern packet scheduling and AQM

Modern congestion control