**Data Center TCP (DCTCP)**

Mohammad Alizadeh, Aibar Parveen Patel, Balaji Priyanathan

**ABSTRACT**

Cloud data centers host diverse applications, making it essential to have low latency. However, real-life traffic (e.g., TCP) may require more than just simple packet loss or congestion. In this paper, we present pFabric, a protocol that provides near-optimal datacenter transport. pFabric reduces average flow completion time by a factor of 2 for flows that are also a single priority number set independent of TCP. Flows have very small buffers and implement a priority-based scheduling mechanism. As a consequence, flows start at a rate 20% higher than other flows, leading to improved performance.

**Categories and Subject Descriptors:** C.2.2 [Computer-Communication Networks]: Network Architecture and Design

**General Terms:** Design, Performance, Experimentation

**Keywords:** Datacenter network, TCP, datacenter transport, flow scheduling

## 1. INTRODUCTION

In recent years, data centers have become a large-scale consolidation of enterprise IT and data and with the emergence of cloud computing services. As a result, the design of data centers has been built to be highly available, to support stateful services, and to meet the requirements of multiple tenants. This trend has led to the development of new technologies, such as the use of software-defined networks (SDN) and the introduction of datacenter networks (DCNs).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

**SIGCOMM'10, August 23-27, 2010, New Delhi, India.**

(C) 2010 ACM 978-1-60558-778-8/10/0008...

---

**TIMELY: RTT-based Congestion Control for the Datacenter**

Rahdika Mittal (UC Berkeley), Vinh The Lam, Nandita Dukkipati, Emily Blem, Hassan Wassef, Monia Chrobadi (Microsoft), Amin Vahdat, Yangong Wang, David Wetherall, David Zats

Google Inc.

**ABSTRACT**

RTT-based congestion control is widely used in the Internet to ensure that end-to-end traffic does not exceed the available bandwidth. However, in datacenter networks, the round-trip time (RTT) can be significantly lower than in the Internet, leading to a mismatch between the RTT and the available bandwidth. This problem can be addressed by using a more accurate estimation of the RTT, which can be obtained by measuring the RTT locally within the datacenter network. In this paper, we present a new congestion control algorithm, called TIMELY, that uses a local RTT measurement to estimate the available bandwidth and dynamically adjust the congestion control parameters accordingly.

**Categories and Subject Descriptors:** C.2.2 [Computer-Communication Networks]: Network Architecture and Design

**General Terms:** Design, Performance, Experimentation

**Keywords:** Datacenter network, TCP, datacenter transport, flow scheduling

---

**1. INTRODUCTION**

Datacenter networks require tight integration with the cloud infrastructure. In this paper, we present a new congestion control algorithm, called TIMELY, that uses a local RTT measurement to estimate the available bandwidth and dynamically adjust the congestion control parameters accordingly. TIMELY is designed to work in datacenter networks, where the round-trip time (RTT) can be significantly lower than in the Internet, leading to a mismatch between the RTT and the available bandwidth. This problem can be addressed by using a more accurate estimation of the RTT, which can be obtained by measuring the RTT locally within the datacenter network.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

**SIGCOMM '15, August 17-21, 2015, London, United Kingdom**

(C) 2015 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-3542-6/V15/08...

DOI: http://dx.doi.org/10.1145/2785956.2787510

---

**Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

**SIGCOMM '11, August 12-16, 2011, Hong Kong, China**

Copyright 2011 ACM 978-1-4503-0559-4/11...

---

**Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

**SIGCOMM '09, August 17-21, 2009, London, United Kingdom**

Copyright 2009 ACM 978-1-60558-659-9/09...

---

**Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

**SIGCOMM '07, August 20-24, 2007, Vancouver, Canada**

Copyright 2007 ACM 978-1-59593-656-6/07...

---

**Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

**SIGCOMM '05, August 15-19, 2005, Chicago, Illinois**

Copyright 2005 ACM 978-1-58113-963-3/05...

---

**Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

**SIGCOMM '04, August 23-27, 2004, Paris, France**

Copyright 2004 ACM 978-1-58113-840-9/04...

---

**Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

**SIGCOMM '03, August 25-29, 2003, Vancouver, Canada**

Copyright 2003 ACM 978-1-58113-647-9/03...

---

**Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

**SIGCOMM '02, September 8-12, 2002, San Francisco, California**

Copyright 2002 ACM 978-1-58113-479-9/02...

---
Tenants choose Congestion Control

Multitenant Datacenters

Enterprise Datacenters
Can’t upgrade legacy applications
adapted from Judd, NSDI ‘15
New Congestion Control Algorithms

pFabric
Timely
DCTCP

Multitenant & Enterprise Datacenters
Updatable & Easy to reconfigure
Hypervisor

Overlay CC

Underlay CC

Datacenter Network
Guest Introspection

- VM
- Guest Memory
- VM
- Hypervisor

BIG BROTHER IS WATCHING YOU
TCP Header Modification

![Diagram showing TCP header modification between a VM, Hypervisor, and vSwitch with SYN and SACK values.]

- VM: SYN SACK: 0
- Hypervisor: SYN SACK: 1
- vSwitch: SYN SACK: 1
Buffering
Fake ACKs

SEQ: 42

VM

Hypervisor

ACK
SEQ: 41

vSwitch
TCP Proxy

- VM
- Hypervisor
- vSwitch
- ACK
- TCP Proxy
vCC Flow Modification Techniques

- Guest Introspection
- Buffering
  - TCP Header Modification
- Fake ACK generation
- TCP Proxy
A Binary Feedback Scheme for Congestion Avoidance in Computer Networks

K. K. RAMAKRISHNAN and RAJ JAIN
Digital Equipment Corporation

We propose a scheme for congestion avoidance in the network layer. The scheme uses a minimal amount of state information and a single feedback bit. It adjusts the amount of traffic allowed into the network by modifying the congestion indication bit in each packet. This indication is communicated back to the users. The scheme is distributed, adapts to the dynamic state of the network, and is simple to implement. It has low overhead and is also provided to multiple sources. This paper presents an analysis of the performance of the scheme when used in combination with a round-robin flow control algorithm. It demonstrates that the scheme is effective in reducing congestion and improving performance.

Categories and Subject Descriptors: C.2.1 [Computer Systems Organization]: Network architecture and design—network communication protocols and services, network [forwarding]; C.2.3 [Computer Communication Networks]: Network Operations and Management; C.4 [Computer Systems Organization]: Computer systems organization—general; C.7.1 [Computing Methodologies]: Artificial Intelligence—general

Additional Key Words and Phrases: Computer network, congestion avoidance, congestion control, congestion indications, performance, network power

1. INTRODUCTION

Congestion in computer networks is a major problem that affects network operations and user experience. It occurs when the demand for network resources exceeds the supply, resulting in degraded performance and increased latencies. Congestion control mechanisms are designed to prevent or mitigate congestion by adjusting the rate at which data is sent across the network.

The control mechanisms adopted to control congestion can be broadly divided into two categories: passive and active. Passive mechanisms are designed to adjust the rate of data transmission based on feedback from the network, while active mechanisms are designed to proactively control the rate of data transmission to prevent congestion.

1 Introduction

This paper proposes a new congestion control scheme for TCP (Transmission Control Protocol) networks. The scheme is based on the use of a binary feedback scheme that adjusts the amount of traffic allowed into the network by modifying the congestion indication bit in each packet. This indication is communicated back to the users and used to adjust the rate of data transmission.

The proposed scheme is distributed, adapts to the dynamic state of the network, and is simple to implement. It has low overhead and is also provided to multiple sources. This paper presents an analysis of the performance of the scheme when used in combination with a round-robin flow control algorithm. It demonstrates that the scheme is effective in reducing congestion and improving performance.

Abstract

This paper presents and analyzes an efficient mechanism for congestion control in computer networks. The mechanism is based on the use of a binary feedback scheme that adjusts the amount of traffic allowed into the network by modifying the congestion indication bit in each packet. This indication is communicated back to the users and used to adjust the rate of data transmission.

The proposed scheme is distributed, adapts to the dynamic state of the network, and is simple to implement. It has low overhead and is also provided to multiple sources. This paper presents an analysis of the performance of the scheme when used in combination with a round-robin flow control algorithm. It demonstrates that the scheme is effective in reducing congestion and improving performance.

TCP and Explicit Congestion Notification

Sally Floyd

Lawrence Berkeley Laboratory
One Cyclotron Road, Berkeley, CA 94704
floyd@ee.lbl.gov

This paper discusses the use of Explicit Congestion Notification (ECN) mechanisms in TCP/IP networks. The first part proposes a new mechanism for TCP's response to ECN mechanisms in the network layer. This mechanism is called RenD (Reference Design) and it modifies the ECN header for packets forwarding to other congested nodes. The second part presents simulation results that show the advantages of RenD compared to other existing mechanisms.

Future routers are likely to have more developed mechanisms for the detection of congestion. With the IETF BCP scheme, for example, routers detect congestion by monitoring the average queue size, and set the ECN bit in packet headers when the average queue size exceeds a certain threshold. Recently proposed Random Early Detection (RED) gateways have a similar behavior, but they differ from our scheme in terms of implementation. In RED gateways, mechanisms for detecting congestion are used before the queue overflows and they are not limited to packet drops as the method of informing sources of congestion.

TCP and Explicit Congestion Notification

Sally Floyd

Lawrence Berkeley Laboratory
One Cyclotron Road, Berkeley, CA 94704
floyd@ee.lbl.gov

This paper discusses the use of Explicit Congestion Notification (ECN) mechanisms in TCP/IP networks. The first part proposes a new mechanism for TCP's response to ECN mechanisms in the network layer. This mechanism is called RenD (Reference Design) and it modifies the ECN header for packets forwarding to other congested nodes. The second part presents simulation results that show the advantages of RenD compared to other existing mechanisms.

Future routers are likely to have more developed mechanisms for the detection of congestion. With the IETF BCP scheme, for example, routers detect congestion by monitoring the average queue size, and set the ECN bit in packet headers when the average queue size exceeds a certain threshold. Recently proposed Random Early Detection (RED) gateways have a similar behavior, but they differ from our scheme in terms of implementation. In RED gateways, mechanisms for detecting congestion are used before the queue overflows and they are not limited to packet drops as the method of informing sources of congestion.

TCP and Explicit Congestion Notification

Sally Floyd

Lawrence Berkeley Laboratory
One Cyclotron Road, Berkeley, CA 94704
floyd@ee.lbl.gov

This paper discusses the use of Explicit Congestion Notification (ECN) mechanisms in TCP/IP networks. The first part proposes a new mechanism for TCP's response to ECN mechanisms in the network layer. This mechanism is called RenD (Reference Design) and it modifies the ECN header for packets forwarding to other congested nodes. The second part presents simulation results that show the advantages of RenD compared to other existing mechanisms.

Future routers are likely to have more developed mechanisms for the detection of congestion. With the IETF BCP scheme, for example, routers detect congestion by monitoring the average queue size, and set the ECN bit in packet headers when the average queue size exceeds a certain threshold. Recently proposed Random Early Detection (RED) gateways have a similar behavior, but they differ from our scheme in terms of implementation. In RED gateways, mechanisms for detecting congestion are used before the queue overflows and they are not limited to packet drops as the method of informing sources of congestion.

TCP and Explicit Congestion Notification

Sally Floyd

Lawrence Berkeley Laboratory
One Cyclotron Road, Berkeley, CA 94704
floyd@ee.lbl.gov

This paper discusses the use of Explicit Congestion Notification (ECN) mechanisms in TCP/IP networks. The first part proposes a new mechanism for TCP's response to ECN mechanisms in the network layer. This mechanism is called RenD (Reference Design) and it modifies the ECN header for packets forwarding to other congested nodes. The second part presents simulation results that show the advantages of RenD compared to other existing mechanisms.

Future routers are likely to have more developed mechanisms for the detection of congestion. With the IETF BCP scheme, for example, routers detect congestion by monitoring the average queue size, and set the ECN bit in packet headers when the average queue size exceeds a certain threshold. Recently proposed Random Early Detection (RED) gateways have a similar behavior, but they differ from our scheme in terms of implementation. In RED gateways, mechanisms for detecting congestion are used before the queue overflows and they are not limited to packet drops as the method of informing sources of congestion.

TCP and Explicit Congestion Notification

Sally Floyd

Lawrence Berkeley Laboratory
One Cyclotron Road, Berkeley, CA 94704
floyd@ee.lbl.gov

This paper discusses the use of Explicit Congestion Notification (ECN) mechanisms in TCP/IP networks. The first part proposes a new mechanism for TCP's response to ECN mechanisms in the network layer. This mechanism is called RenD (Reference Design) and it modifies the ECN header for packets forwarding to other congested nodes. The second part presents simulation results that show the advantages of RenD compared to other existing mechanisms.

Future routers are likely to have more developed mechanisms for the detection of congestion. With the IETF BCP scheme, for example, routers detect congestion by monitoring the average queue size, and set the ECN bit in packet headers when the average queue size exceeds a certain threshold. Recently proposed Random Early Detection (RED) gateways have a similar behavior, but they differ from our scheme in terms of implementation. In RED gateways, mechanisms for detecting congestion are used before the queue overflows and they are not limited to packet drops as the method of informing sources of congestion.

TCP and Explicit Congestion Notification

Sally Floyd

Lawrence Berkeley Laboratory
One Cyclotron Road, Berkeley, CA 94704
floyd@ee.lbl.gov

This paper discusses the use of Explicit Congestion Notification (ECN) mechanisms in TCP/IP networks. The first part proposes a new mechanism for TCP's response to ECN mechanisms in the network layer. This mechanism is called RenD (Reference Design) and it modifies the ECN header for packets forwarding to other congested nodes. The second part presents simulation results that show the advantages of RenD compared to other existing mechanisms.

Future routers are likely to have more developed mechanisms for the detection of congestion. With the IETF BCP scheme, for example, routers detect congestion by monitoring the average queue size, and set the ECN bit in packet headers when the average queue size exceeds a certain threshold. Recently proposed Random Early Detection (RED) gateways have a similar behavior, but they differ from our scheme in terms of implementation. In RED gateways, mechanisms for detecting congestion are used before the queue overflows and they are not limited to packet drops as the method of informing sources of congestion.

TCP and Explicit Congestion Notification

Sally Floyd

Lawrence Berkeley Laboratory
One Cyclotron Road, Berkeley, CA 94704
floyd@ee.lbl.gov

This paper discusses the use of Explicit Congestion Notification (ECN) mechanisms in TCP/IP networks. The first part proposes a new mechanism for TCP's response to ECN mechanisms in the network layer. This mechanism is called RenD (Reference Design) and it modifies the ECN header for packets forwarding to other congested nodes. The second part presents simulation results that show the advantages of RenD compared to other existing mechanisms.

Future routers are likely to have more developed mechanisms for the detection of congestion. With the IETF BCP scheme, for example, routers detect congestion by monitoring the average queue size, and set the ECN bit in packet headers when the average queue size exceeds a certain threshold. Recently proposed Random Early Detection (RED) gateways have a similar behavior, but they differ from our scheme in terms of implementation. In RED gateways, mechanisms for detecting congestion are used before the queue overflows and they are not limited to packet drops as the method of informing sources of congestion.
3 Dropped
2 Marked
1 Queued
All Packets Queued

ECN Marked & non-ECN Dropped

All Packets Dropped
Bottleneck Link

Sender

ECN
ECN
ECN
translated to ECN

Receiver
vCC

Virtualized Congestion Control
SSTHRESH: Slow Start Threshold
CWND: Congestion Window
SND_WND: Send Window
VM

SSTHRESH: 10
CWND: 30
SND_WND: 15

Hypervisor

CWR

SSTHRESH: 10
CWND: 15
SND_WND: 15
VM

SSTHRESH: 10
CWND: 30
SND_WND: 16

Hypervisor

RWIN: 16

SSTHRESH: 10
CWND: 16
SND_WND: 16

10 30 16
16 16 31 16
10 30 15 15 16 16 33
ECN Experiments in this Presentation:
Linux 3.19 mininet testbed
vCC implemented as a kernel patch

Public vCC Code:
Linux patch and mininet test suite available on Github
Public Code and Extended Paper:
http://webee.technion.ac.il/~isaac/vcc/

Our Paper:
ECN unfairness, virtual-ECN vCC, ESX app-level throttling vCC, technique survey, future implementation analysis
Thank You!
vCC

Virtualized Congestion Control

Translate between Congestion Control Algorithms in the Hypervisor

in order to

• Achieve uniform congestion control in multitenant and enterprise datacenters

• Programatically assign congestion control to fine-grained flow signatures

• Simplify rollout of new congestion control algorithms