

Keeping an Eye on Congestion Control in the Wild with **Nebby**

Ayush Mishra¹, Lakshay Rastogi², Raj Joshi³, Ben Leong¹

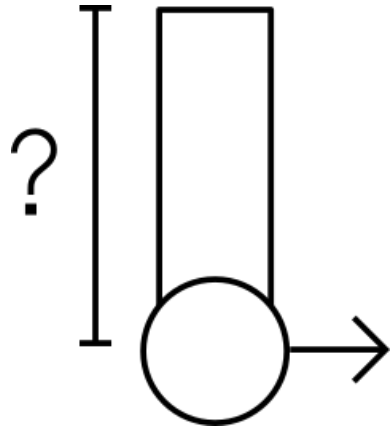
¹National University of Singapore

²IIT Kanpur

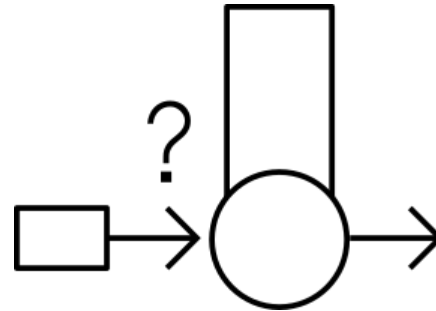
³Harvard University



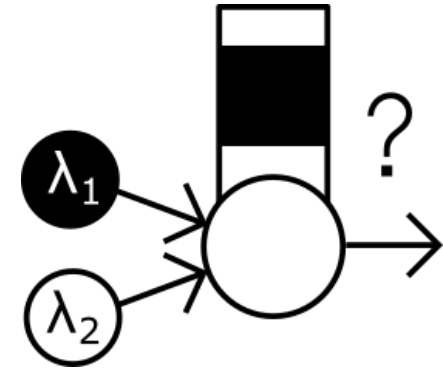
The make up of the Internet's Congestion Control Landscape influences how we think about



Buffer Sizing



AQMs and
In-network
policing

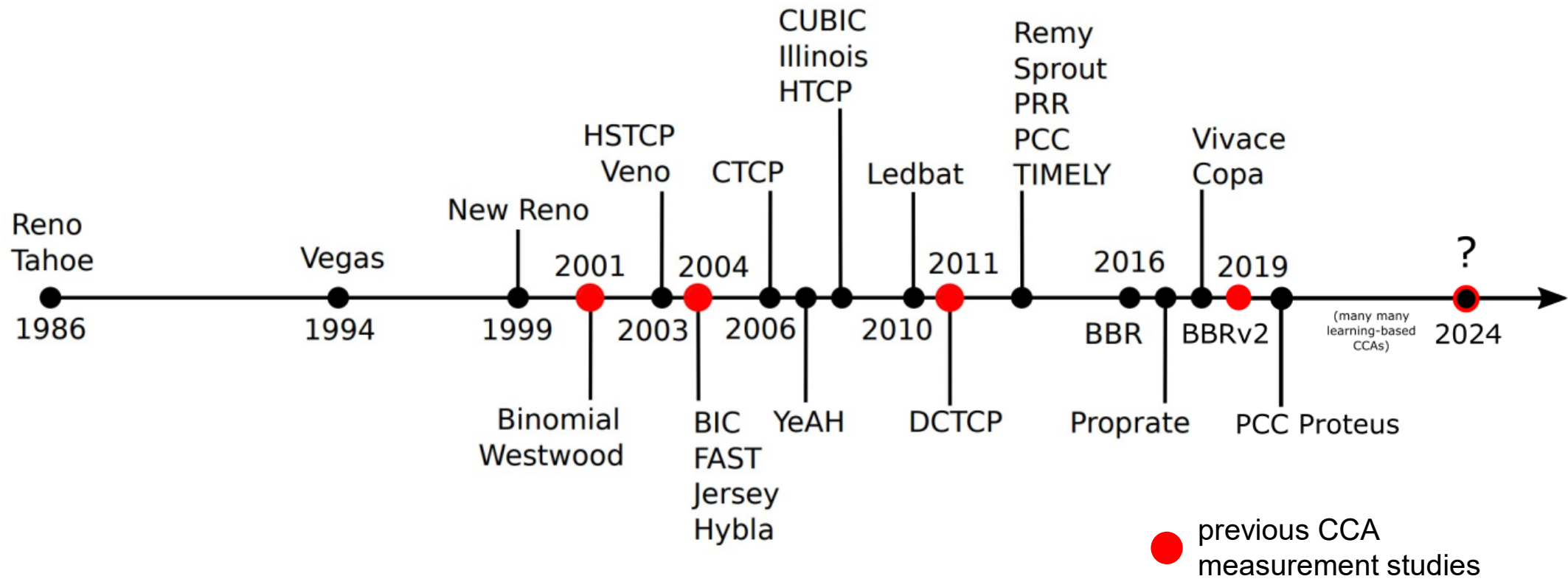


Fairness and
Deployability

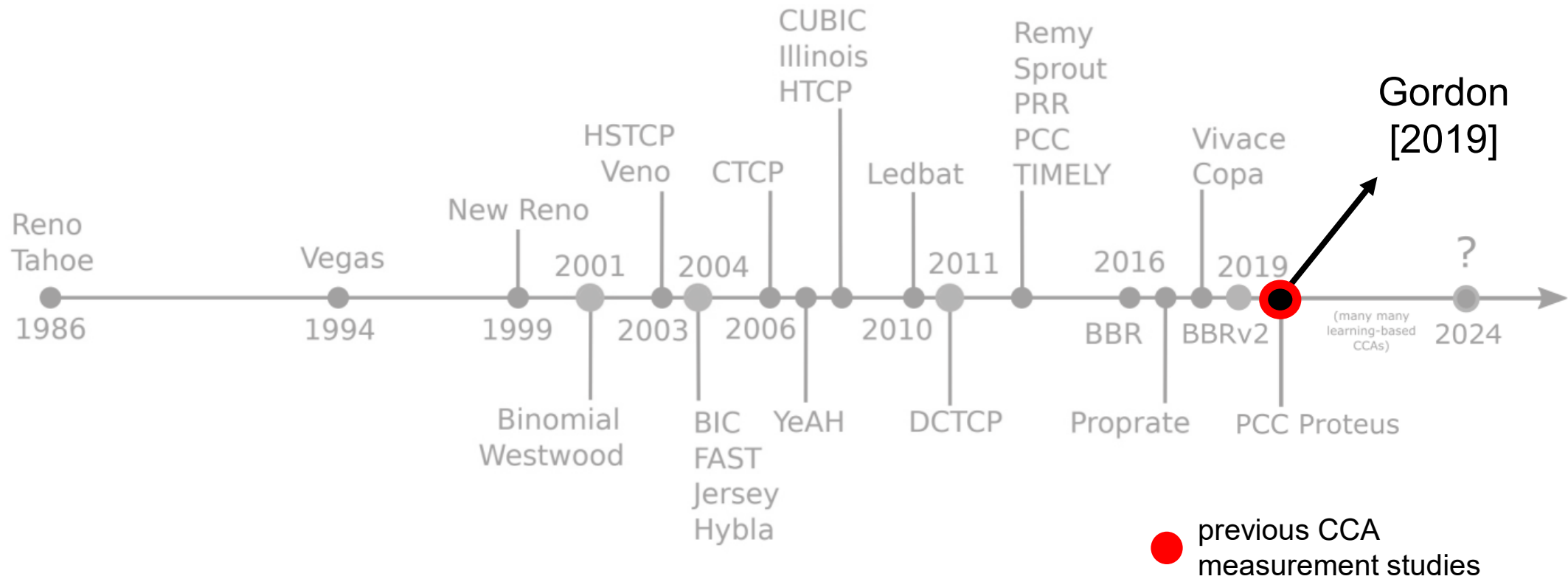
End-host Congestion Control is a unique design space where we expect users to meet their *selfish* goals without causing harm.

We also need to monitor the responsible deployment of CCAs on the Internet.

This is not a new problem.



This is not a new problem.



So what does the Internet's
current Congestion Control
Landscape look like?

So we decided to re-run the most recent of the measurement tools...

Gordon [2019]

The Great Internet TCP Congestion Control Census

Ayush Mishra
ayush@comp.nus.edu.sg
National University of Singapore
Singapore

Xiangpeng Sun
sun.xiangpeng@comp.nus.edu.sg
National University of Singapore
Singapore

Atishya Jain
atishya.jain.cs516@cse.iitd.ac.in
Indian Institute of Technology, Delhi
India

Sameer Pande
sameer.vivek.pande.cs117@cse.iitd.ac.in
Indian Institute of Technology, Delhi
India

Raj Joshi
rajjoshi@comp.nus.edu.sg
National University of Singapore
Singapore

Ben Leong
benleong@comp.nus.edu.sg
National University of Singapore
Singapore

ABSTRACT

In 2016, Google proposed and deployed a new TCP variant called BBR. BBR represents a major departure from traditional congestion control as it uses estimates of bandwidth and round-trip delays to regulate its sending rate. BBR has since been introduced in the upstream Linux kernel and deployed by Google across its data centers. Since the last major study to identify TCP congestion control variants on the Internet was done before BBR, it is timely to conduct a new census to give us a sense of the current distribution of congestion control variants on the Internet. To this end, we designed and implemented *Gordon*, a tool that allows us to measure the congestion window (cwnd) corresponding to each successive RTT in the TCP connection response of a congestion control algorithm. To compare a measured flow to the known variants, we created a localized bottleneck and introduced a variety of network changes like loss events, changes in bandwidth and delay, while normalizing all measurements by RTT. We built an offline classifier to identify the TCP variant based on the cwnd trace over time.

Our results suggest that CUBIC is currently the dominant TCP

KEYWORDS

congestion control; measurement study

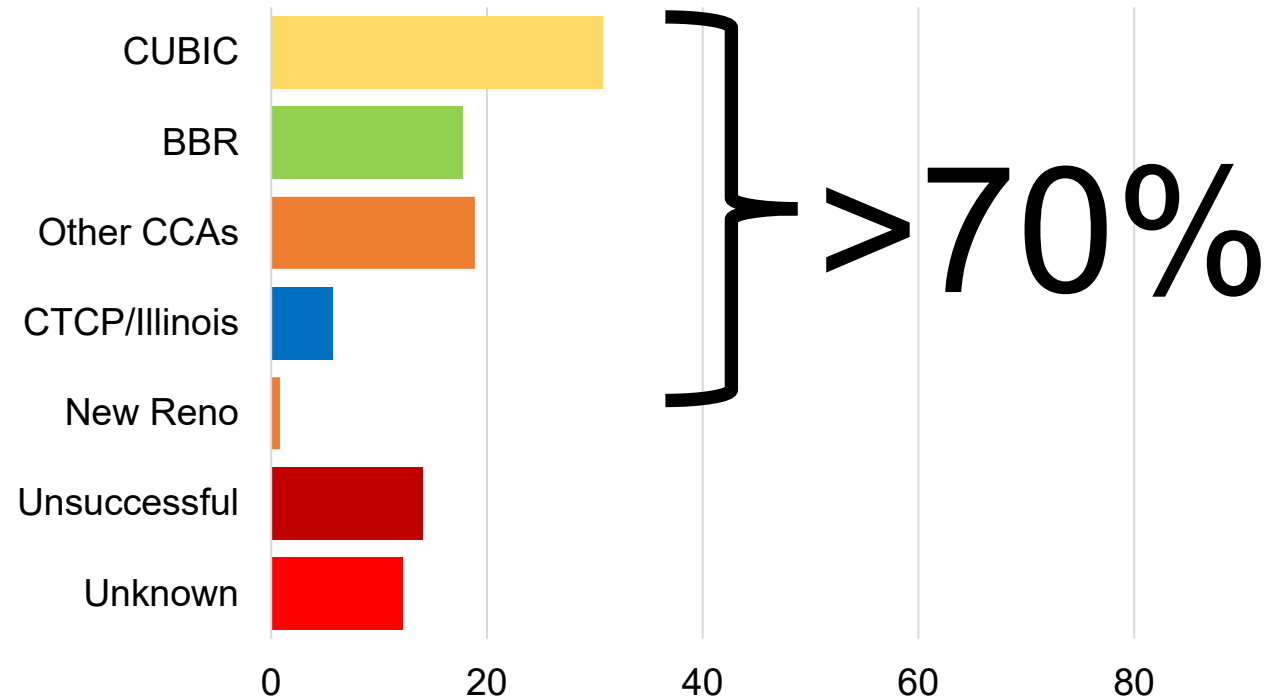
ACM Reference Format:

Ayush Mishra, Xiangpeng Sun, Atishya Jain, Sameer Pande, Raj Joshi, and Ben Leong. 2020. The Great Internet TCP Congestion Control Census. In *ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '20 Abstracts)*, June 8–12, 2020, Boston, MA, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3393691.3394221>

1 INTRODUCTION

Over the past 30 years, TCP congestion control has evolved to adapt to the changing needs of the users and to exploit improvements in the underlying network. Most recently, in 2016, Google proposed and deployed a new TCP variant called BBR [2]. BBR represents a major departure from traditional congestion-window-based congestion control. Instead of using packet loss as a congestion signal, BBR uses estimates of the bandwidth and round-trip delays to regulate

2019



Keeping an Eye on Congestion Control in the Wild with
Nebby, SIGCOMM '24

So we decided to re-run the most recent
of the measurement tools...
But we were **not successful**.

Gordon
[2019]

The Great Internet TCP Congestion Control Census

Ayush Mishra
ayush@comp.nus.edu.sg
National University of Singapore
Singapore

Xiangpeng Sun
sun.xiangpeng@comp.nus.edu.sg
National University of Singapore
Singapore

Atishya Jain
atishya.jain.cs516@cse.iitd.ac.in
Indian Institute of Technology, Delhi
India

Sameer Pande
sameer.vivek.pande.cs117@cse.iitd.
ac.in
Indian Institute of Technology, Delhi
India

Raj Joshi
rajjoshi@comp.nus.edu.sg
National University of Singapore
Singapore

Ben Leong
benleong@comp.nus.edu.sg
National University of Singapore
Singapore

ABSTRACT

In 2016, Google proposed and deployed a new TCP variant called BBR. BBR represents a major departure from traditional congestion control as it uses estimates of bandwidth and round-trip delays to regulate its sending rate. BBR has since been introduced in the upstream Linux kernel and deployed by Google across its data centers. Since the last major study to identify TCP congestion control variants on the Internet was done before BBR, it is timely to conduct a new census to give us a sense of the current distribution of congestion control variants on the Internet. To this end, we designed and implemented *Gordon*, a tool that allows us to measure the congestion window (cwnd) corresponding to each successive RTT in the TCP connection response of a congestion control algorithm. To compare a measured flow to the known variants, we created a localized bottleneck and introduced a variety of network changes like loss events, changes in bandwidth and delay, while normalizing all measurements by RTT. We built an offline classifier to identify the TCP variant based on the cwnd trace over time.

Our results suggest that CUBIC is currently the dominant TCP

KEYWORDS

congestion control; measurement study

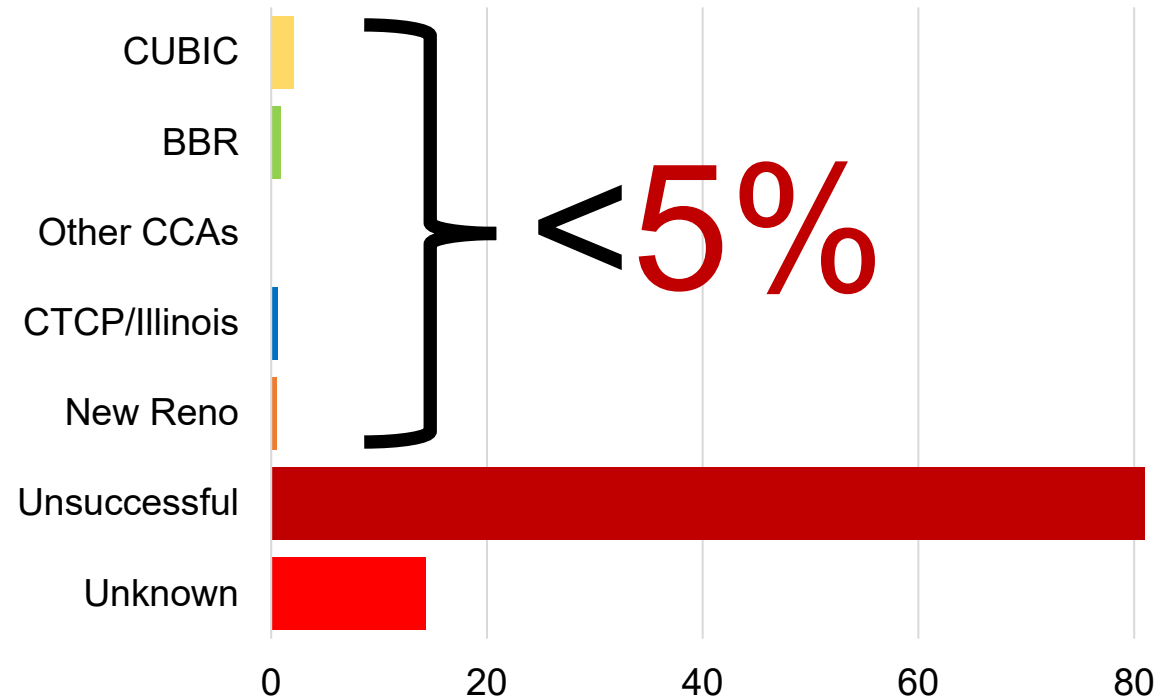
ACM Reference Format:

Ayush Mishra, Xiangpeng Sun, Atishya Jain, Sameer Pande, Raj Joshi, and Ben Leong. 2020. The Great Internet TCP Congestion Control Census. In *ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '20 Abstracts)*, June 8–12, 2020, Boston, MA, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3393691.3394221>

1 INTRODUCTION

Over the past 30 years, TCP congestion control has evolved to adapt to the changing needs of the users and to exploit improvements in the underlying network. Most recently, in 2016, Google proposed and deployed a new TCP variant called BBR [2]. BBR represents a major departure from traditional congestion-window-based congestion control. Instead of using packet loss as a congestion signal, BBR uses estimates of the bandwidth and round-trip delays to regulate

2023



Keeping an Eye on Congestion Control in the Wild with
Nebby, SIGCOMM '24

This is has been a trend in all previous CCA classification tools.

Let's identify CCAS

...but **TBIT** doesn't work

...but **CAAI** doesn't work

TBIT
[2001,2005]

CAAI
[2011]

IG, Gordon
[2019]



On Inferring TCP Behavior

Jhosita Padgug and Saily Pajot
AT&T Center for Internet Research 6-02 (2019)
padgug@atc.uzh.ch, saily@atc.uzh.ch

ABSTRACT

Inferring the behavior of TCP is a challenging task. In this paper, we propose the Inferring TCP Behavior (ITB) framework, which is designed to infer the behavior of TCP by analyzing the network traffic. ITB is designed to be a general framework that can be applied to any network traffic. ITB is designed to be a general framework that can be applied to any network traffic. ITB is designed to be a general framework that can be applied to any network traffic.

Measuring the Evolution of Transport Protocols in the Internet

Mark Allman, Saily Pajot
ICIR Center for Internet Research
allman@cs.cmu.edu, saily@atc.uzh.ch

To appear in Computer Communications Review, April 2005.

ABSTRACT

In this paper we explore the evolution of the Internet's transport protocols. We focus on the evolution of TCP, which is the most widely used transport protocol. We analyze the evolution of TCP by analyzing the network traffic. We analyze the evolution of TCP by analyzing the network traffic. We analyze the evolution of TCP by analyzing the network traffic.

Mark Allman, Saily Pajot
ICIR Center for Internet Research
allman@cs.cmu.edu, saily@atc.uzh.ch

RESEARCH TRANSACTIONS ON NETWORKING, VOL. 20, NO. 4, APRIL 2012
Pages: 1311 – 1324, DOI: 10.1109/RTN.2012.2278271

TCP Congestion Avoidance Algorithm Identification

Peng Yang, Member, IEEE, Juan Shao, Wen Luo, Liang Xu, Member, IEEE, Jitender Dogra, Member, IEEE, and Ying Li, Member, IEEE

Abstract—The Internet has recently been evolving from homogeneous congestion control to heterogeneous congestion control. Several years ago, Internet traffic was mainly controlled by the traditional TCP algorithms, such as Reno, Cubic, and Compound TCP (CTCP). However, there is a very high risk of the performance and stability of the Internet with heterogeneous congestion control. One fundamental reason is the lack of the deployment information of different TCP algorithms. In this paper, we first propose a novel TCP Congestion Avoidance Algorithm Identification (CAAI) for actively identifying the TCP algorithm of a remote Web server. CAAI can identify all default TCP algorithms, including Reno, Cubic, and CTCP, and most non-default TCP algorithms of major operating system families. We then present the CAAI measurement result of other major Web servers. We found that 46.2% of the Web servers use Reno, 40.8% use Cubic, and 13.1% use CTCP. The measurement results show that the majority of TCP servers are not controlled by Reno anymore, and a strong sign that the Internet congestion control has changed from homogeneous to heterogeneous.

Index Terms—Heterogeneous congestion control, Internet measurement, TCP congestion control.

Inspector Gadget: A Framework for Inferring TCP Congestion Control Algorithms

Subhadra Ching
Purdue University

Abstract—Over the last decade, in an attempt to improve performance, the community has proposed a multitude of TCP algorithms. These algorithms have been proposed in the context of the development of new congestion control algorithms. However, the development of new congestion control algorithms has not been performed in isolation. There are a number of factors that influence the development of new congestion control algorithms. In this paper, we propose a framework for inferring TCP congestion control algorithms. The framework is designed to be a general framework that can be applied to any network traffic. The framework is designed to be a general framework that can be applied to any network traffic.

Society's growing dependence on better Quality of Experience (QoE) and the need for better network management tools, such as network management, is creating a demand for a framework for inferring TCP congestion control algorithms. In this paper, we propose a framework for inferring TCP congestion control algorithms. The framework is designed to be a general framework that can be applied to any network traffic. The framework is designed to be a general framework that can be applied to any network traffic.

The Great Internet TCP Congestion Control Census

Ayush Mishra
ayush@comp.nyu.edu
National University of Singapore

Samuel Pande
samuel.pande@nyu.edu
National University of Singapore

Xiangpeng Sun
sun.xiangpeng@nyu.edu
National University of Singapore

Raj Jishi
rajishi@comp.nyu.edu
National University of Singapore

Atishya Jain
atishya.jain@nyu.edu
National University of Singapore

Ben Leong
leongben@comp.nyu.edu
National University of Singapore

ABSTRACT

In 2019, Google proposed and deployed a new TCP variant called BBR. BBR represents a major departure from traditional congestion control as it uses estimates of bandwidth and round trip delays to regulate its sending rate. BBR has since been introduced in the operating system kernel and deployed in Google servers in data centers. Since the last major study to identify TCP congestion control variants in the Internet was the Internet BBR, it is timely to conduct a new census to give us a sense of the current distribution of congestion control variants in the Internet. To this end, we designed and implemented Gadget, a tool that allows us to measure the congestion window (cwnd) corresponding to each successful RTT in the TCP connection response of a congestion control algorithm. To capture a measured flow in the Internet, we created a localized backbone and introduced a variety of network changes like low latency, changes in bandwidth and delay, while maintaining all measurements by ITT. We built an offline classifier to identify the TCP variant based on the cwnd over time. We then used the classifier to identify the TCP variant in the Internet. Our results suggest that BBR is currently the dominant TCP variant in the Internet.

This is has been a trend in all previous CCA classification tools.

Let's identify CCAS

...but **TBIT** doesn't work

...but **CAAI** doesn't work

but **IG** and **Gordon** don't work

TBIT
[2001,2005]

CAAI
[2011]

IG, Gordon
[2019]

Nebby
[2024]



On Inferring TCP Behavior

Josiah Padgug and Sally Floyd
AT&T Center for Internet Research 6-020 (2005)
padgug@att.net, sfloyd@att.net

ABSTRACT

Inferring the behavior of TCP flows is a difficult task. The difficulty arises from the performance of a single packet capture. This paper presents a methodology for inferring TCP behavior from a single packet capture. The methodology is based on the observation that the behavior of a TCP flow is determined by the sequence of events that occur in the flow. The methodology is based on the observation that the behavior of a TCP flow is determined by the sequence of events that occur in the flow. The methodology is based on the observation that the behavior of a TCP flow is determined by the sequence of events that occur in the flow.

Measuring the Evolution of Transport Protocols in the Internet

Alberto Mellardo
IBM Research
amellardo@ibm.com

Mark Allman, Sally Floyd
IBM Center for Internet Research
amellardo@att.net, sfloyd@att.net

To appear in Computer Communications Review, April 2005.

In this paper we explore the evolution of the Internet's transport protocols. We focus on the evolution of TCP, which is the most widely used transport protocol. We present a methodology for inferring TCP behavior from a single packet capture. The methodology is based on the observation that the behavior of a TCP flow is determined by the sequence of events that occur in the flow. The methodology is based on the observation that the behavior of a TCP flow is determined by the sequence of events that occur in the flow. The methodology is based on the observation that the behavior of a TCP flow is determined by the sequence of events that occur in the flow.

Inferring the behavior of TCP flows is a difficult task. The difficulty arises from the performance of a single packet capture. This paper presents a methodology for inferring TCP behavior from a single packet capture. The methodology is based on the observation that the behavior of a TCP flow is determined by the sequence of events that occur in the flow. The methodology is based on the observation that the behavior of a TCP flow is determined by the sequence of events that occur in the flow. The methodology is based on the observation that the behavior of a TCP flow is determined by the sequence of events that occur in the flow.

TCP Congestion Avoidance Algorithm Identification

Peng Yang, Member IEEE, Juan Shao, Wen Luo, Liang Xu, Member IEEE, Hender Denig, Member IEEE, and Ying Lu, Member IEEE

Abstract—The Internet has recently been evolving from heterogeneous congestion control to heterogeneous congestion control. Several years ago, Internet traffic was mainly controlled by the traditional TCP algorithms, such as Reno, Cubic, and Compound TCP (CTCP). However, there is very little information about the performance and stability of the Internet with heterogeneous congestion control. One fundamental reason is the lack of the deployment information of different TCP algorithms. In this paper, we first propose a novel called TCP Congestion Avoidance Algorithm Identification (CAAI) for actively identifying the TCP algorithm of a remote Web server. CAAI can identify all different TCP algorithms, including Reno, Cubic, and CTCP, and their non-standard TCP algorithms of engineering system families. We then present the CAAI measurement result of about 3400 Web servers. We found that only 4.31% ~ 4.47% of the Web servers still use Reno, 46.02% of the Web servers use CUBIC, 4.43% of the Web servers use CTCP. Our measurement results show a strong sign that the majority of TCP flows are now controlled by Reno servers, and a strong sign that the Internet congestion control has changed from homogeneous to heterogeneous.

Index Terms—Heterogeneous congestion control, Internet measurement, TCP congestion control.

Inspector Gadget: A Framework for Inferring TCP Congestion Control Algorithm

Shihui Ding
Purdue University

Di

Abstract—Over the last decade, in an attempt to improve on performance, the community has proposed a multitude of TCP congestion control algorithms. However, the lack of information about the performance of these algorithms has led to the development of new congestion control algorithms. While the performance of these algorithms is well understood, their behavior in the presence of congestion is not. In this paper, we present a framework for inferring the congestion control algorithm of a remote Web server. The framework is based on the observation that the behavior of a TCP flow is determined by the sequence of events that occur in the flow. The framework is based on the observation that the behavior of a TCP flow is determined by the sequence of events that occur in the flow. The framework is based on the observation that the behavior of a TCP flow is determined by the sequence of events that occur in the flow.

The Great Internet TCP Congestion Control Census

Ayush Mishra
mishra@comp.nyu.edu

National University of Singapore

Xiangpeng Sun
sunxp@comp.nyu.edu

National University of Singapore

Atishay Jain
atishay.jain@nyu.edu

National University of Singapore

Samuel P. D. Costa
samuel.p.d.costa@nyu.edu

National University of Singapore

Raj Jishi
rajishi@comp.nyu.edu

National University of Singapore

Ben Leong
leongben@comp.nyu.edu

National University of Singapore

ABSTRACT

In 2019, Google proposed and deployed a new TCP variant called BBR. BBR represents a major departure from traditional congestion control as it uses estimates of bandwidth and round trip delays to regulate its sending rate. BBR has since been implemented in the operating system kernel and deployed to Google servers in data centers. Since the last major study to identify TCP congestion control variants in the Internet was done before BBR, it is timely to conduct a new census to give us a sense of the current distribution of congestion control variants in the Internet. To this end, we designed and implemented Inspector, a tool that allows us to measure the congestion window (cwnd) corresponding to each successful RTT in the TCP connection response of a congestion control algorithm. To capture a measured flow in the Internet, we created a localized backbone and introduced a variety of network changes like low events, changes in bandwidth and delay, while maintaining all measurements by ITT. We built an offline classifier to identify the TCP variant based on the cwnd over time. Our results show that CUBIC is currently the dominant TCP variant in the Internet.

Keeping an Eye on Congestion Control in the Wild with Nebby

Ayush Mishra[†], Lakshay Rastogi[‡], Raj Jishi[†], and Ben Leong[†]

[†]National University of Singapore [‡]Indian Institute of Technology, Karpur

ABSTRACT

The Internet congestion control landscape is rapidly evolving. Since the introduction of BBR and the deployment of QUIC, it has become increasingly commonplace for companies to modify and implement their own congestion control algorithms (CCAs). To respond effectively to these developments, it is crucial to understand the state of CCA deployments in the wild. Unfortunately, existing CCA identification tools are not future-proof and do not work well with modern CCAs and encrypted protocols like QUIC. In this paper, we articulate the challenges in designing a future-proof CCA identification tool and propose a measurement methodology that directly addresses these challenges. The resulting measurement tool, called Nebby, can identify all the CCAs currently available in the Linux kernel and BBRv2 with an average accuracy of 96.7%. We found that among the Alexa Top 20k websites, the share of BBR has shrunk since 2019 and that only 8% of them responded to QUIC requests. Among these QUIC servers, CUBIC and BBR seem equally popular. We show that Nebby is extensible by extending it for Copas and an undocumented family of CCAs that is deployed by 6% of the measured websites, including major corporations like Hulu and Apple.

However, recent developments suggest that CCAs on the Internet are evolving faster than ever before. The deployment of BBR and its variants is a perfect example of this rapid evolution. While BBR was first introduced back in 2016, the algorithm has continued to evolve over the years. At the time of writing, Google alone is known to have deployed three different versions of BBR [13, 22, 33]. Outside of Google, operators have also been found to deploy modified versions of BBR according to their own needs [48]. The adoption of QUIC [39] on the Internet is another catalyst that has influenced the evolution of the Internet's CCA landscape in recent years. While the QUIC standard itself does not introduce any new CCAs, QUIC congestion control is implemented in the user space and thus makes it significantly easier to implement new CCAs and to deploy modified versions of existing CCAs. There is evidence that operators are already deploying their own variants of CCAs like CUBIC and BBR in their QUIC stacks [47]. These variants can behave very differently from their kernel counterparts. Given that these developments have major consequences for the Internet's congestion control landscape, it is crucial to keep an eye on CCAs in the wild. Unfortunately, existing CCA identification tools [24, 31, 46, 50, 54, 63] do not work well with modern CCAs

Crucial design goal: let's also be



Future-proof



Let's identify
CCAS

...but **TBIT**
doesn't work

...but **CAAI**
doesn't work

but **IG** and **Gordon**
don't work

Nebby

[2024]

TBIT
[2001,2005]

CAAI
[2011]

IG, Gordon
[2019]



On Inferring TCP Behavior

Jessica Fridman and Sally Padgug
AT&T Center for Internet Research 6-020 (JCFR)
jessica@att.net, sally@att.net

ABSTRACT

Since the widely available Internet's history has been marked by TCP, the network protocol that governs the flow of data, it is natural to expect that the network's behavior is largely determined by TCP. In this paper, we explore the question of whether the Internet's behavior is indeed largely determined by TCP, and if so, how we can infer TCP's behavior from the network's behavior.

Measuring the Evolution of Transport Protocols in the Internet

Alberto Melloni
IBM Research
amelloni@ibm.com

Mark Allman, Sally Padgug
ICIR Center for Internet Research
amelloni@ibm.com

To appear in Computer Communications Review, April 2005.

ABSTRACT

In this paper we explore the question of whether the Internet's behavior is indeed largely determined by TCP, and if so, how we can infer TCP's behavior from the network's behavior. We present a methodology for measuring the evolution of transport protocols in the Internet, and we apply it to the case of TCP. Our results show that TCP is indeed the dominant protocol in the Internet, and that its behavior is largely determined by the network's behavior.

Internet of transport protocols in the Internet's history. In the past, the network's behavior was largely determined by TCP, and it is natural to expect that the network's behavior is still largely determined by TCP. In this paper, we explore the question of whether the Internet's behavior is indeed largely determined by TCP, and if so, how we can infer TCP's behavior from the network's behavior.

RESEARCH TRANSACTIONS ON COMPUTING SYSTEMS, VOL. 52, NO. 4, APRIL 2013
Page 1111–1124, 2013, DOI: 10.1109/RTCS.2013.2278271

TCP Congestion Avoidance Algorithm Identification

Peng Ying, Member IEEE, Juan Shao, Wen Luo, Liang Xu, Member IEEE, Jintao Dong, Member IEEE, and Ying Li, Member IEEE

TABLE I TCP Algorithms Available in Major Operating System Families	
Operating System	TCP Algorithms
Windows (NT, 2000, XP, Vista, 7, 8, 10)	RENO, ILTP, CUBIC, BBR, Vegas, DCTCP, HSTCP, HTCP, HYPERCONG, HTCP, ILTP, ILTPV2, ILTPV3, ILTPV4, ILTPV5, ILTPV6, ILTPV7, ILTPV8, ILTPV9, ILTPV10, ILTPV11, ILTPV12, ILTPV13, ILTPV14, ILTPV15, ILTPV16, ILTPV17, ILTPV18, ILTPV19, ILTPV20, ILTPV21, ILTPV22, ILTPV23, ILTPV24, ILTPV25, ILTPV26, ILTPV27, ILTPV28, ILTPV29, ILTPV30, ILTPV31, ILTPV32, ILTPV33, ILTPV34, ILTPV35, ILTPV36, ILTPV37, ILTPV38, ILTPV39, ILTPV40, ILTPV41, ILTPV42, ILTPV43, ILTPV44, ILTPV45, ILTPV46, ILTPV47, ILTPV48, ILTPV49, ILTPV50, ILTPV51, ILTPV52, ILTPV53, ILTPV54, ILTPV55, ILTPV56, ILTPV57, ILTPV58, ILTPV59, ILTPV60, ILTPV61, ILTPV62, ILTPV63, ILTPV64, ILTPV65, ILTPV66, ILTPV67, ILTPV68, ILTPV69, ILTPV70, ILTPV71, ILTPV72, ILTPV73, ILTPV74, ILTPV75, ILTPV76, ILTPV77, ILTPV78, ILTPV79, ILTPV80, ILTPV81, ILTPV82, ILTPV83, ILTPV84, ILTPV85, ILTPV86, ILTPV87, ILTPV88, ILTPV89, ILTPV90, ILTPV91, ILTPV92, ILTPV93, ILTPV94, ILTPV95, ILTPV96, ILTPV97, ILTPV98, ILTPV99, ILTPV100, ILTPV101, ILTPV102, ILTPV103, ILTPV104, ILTPV105, ILTPV106, ILTPV107, ILTPV108, ILTPV109, ILTPV110, ILTPV111, ILTPV112, ILTPV113, ILTPV114, ILTPV115, ILTPV116, ILTPV117, ILTPV118, ILTPV119, ILTPV120, ILTPV121, ILTPV122, ILTPV123, ILTPV124, ILTPV125, ILTPV126, ILTPV127, ILTPV128, ILTPV129, ILTPV130, ILTPV131, ILTPV132, ILTPV133, ILTPV134, ILTPV135, ILTPV136, ILTPV137, ILTPV138, ILTPV139, ILTPV140, ILTPV141, ILTPV142, ILTPV143, ILTPV144, ILTPV145, ILTPV146, ILTPV147, ILTPV148, ILTPV149, ILTPV150, ILTPV151, ILTPV152, ILTPV153, ILTPV154, ILTPV155, ILTPV156, ILTPV157, ILTPV158, ILTPV159, ILTPV160, ILTPV161, ILTPV162, ILTPV163, ILTPV164, ILTPV165, ILTPV166, ILTPV167, ILTPV168, ILTPV169, ILTPV170, ILTPV171, ILTPV172, ILTPV173, ILTPV174, ILTPV175, ILTPV176, ILTPV177, ILTPV178, ILTPV179, ILTPV180, ILTPV181, ILTPV182, ILTPV183, ILTPV184, ILTPV185, ILTPV186, ILTPV187, ILTPV188, ILTPV189, ILTPV190, ILTPV191, ILTPV192, ILTPV193, ILTPV194, ILTPV195, ILTPV196, ILTPV197, ILTPV198, ILTPV199, ILTPV200, ILTPV201, ILTPV202, ILTPV203, ILTPV204, ILTPV205, ILTPV206, ILTPV207, ILTPV208, ILTPV209, ILTPV210, ILTPV211, ILTPV212, ILTPV213, ILTPV214, ILTPV215, ILTPV216, ILTPV217, ILTPV218, ILTPV219, ILTPV220, ILTPV221, ILTPV222, ILTPV223, ILTPV224, ILTPV225, ILTPV226, ILTPV227, ILTPV228, ILTPV229, ILTPV230, ILTPV231, ILTPV232, ILTPV233, ILTPV234, ILTPV235, ILTPV236, ILTPV237, ILTPV238, ILTPV239, ILTPV240, ILTPV241, ILTPV242, ILTPV243, ILTPV244, ILTPV245, ILTPV246, ILTPV247, ILTPV248, ILTPV249, ILTPV250, ILTPV251, ILTPV252, ILTPV253, ILTPV254, ILTPV255, ILTPV256, ILTPV257, ILTPV258, ILTPV259, ILTPV260, ILTPV261, ILTPV262, ILTPV263, ILTPV264, ILTPV265, ILTPV266, ILTPV267, ILTPV268, ILTPV269, ILTPV270, ILTPV271, ILTPV272, ILTPV273, ILTPV274, ILTPV275, ILTPV276, ILTPV277, ILTPV278, ILTPV279, ILTPV280, ILTPV281, ILTPV282, ILTPV283, ILTPV284, ILTPV285, ILTPV286, ILTPV287, ILTPV288, ILTPV289, ILTPV290, ILTPV291, ILTPV292, ILTPV293, ILTPV294, ILTPV295, ILTPV296, ILTPV297, ILTPV298, ILTPV299, ILTPV300, ILTPV301, ILTPV302, ILTPV303, ILTPV304, ILTPV305, ILTPV306, ILTPV307, ILTPV308, ILTPV309, ILTPV310, ILTPV311, ILTPV312, ILTPV313, ILTPV314, ILTPV315, ILTPV316, ILTPV317, ILTPV318, ILTPV319, ILTPV320, ILTPV321, ILTPV322, ILTPV323, ILTPV324, ILTPV325, ILTPV326, ILTPV327, ILTPV328, ILTPV329, ILTPV330, ILTPV331, ILTPV332, ILTPV333, ILTPV334, ILTPV335, ILTPV336, ILTPV337, ILTPV338, ILTPV339, ILTPV340, ILTPV341, ILTPV342, ILTPV343, ILTPV344, ILTPV345, ILTPV346, ILTPV347, ILTPV348, ILTPV349, ILTPV350, ILTPV351, ILTPV352, ILTPV353, ILTPV354, ILTPV355, ILTPV356, ILTPV357, ILTPV358, ILTPV359, ILTPV360, ILTPV361, ILTPV362, ILTPV363, ILTPV364, ILTPV365, ILTPV366, ILTPV367, ILTPV368, ILTPV369, ILTPV370, ILTPV371, ILTPV372, ILTPV373, ILTPV374, ILTPV375, ILTPV376, ILTPV377, ILTPV378, ILTPV379, ILTPV380, ILTPV381, ILTPV382, ILTPV383, ILTPV384, ILTPV385, ILTPV386, ILTPV387, ILTPV388, ILTPV389, ILTPV390, ILTPV391, ILTPV392, ILTPV393, ILTPV394, ILTPV395, ILTPV396, ILTPV397, ILTPV398, ILTPV399, ILTPV400, ILTPV401, ILTPV402, ILTPV403, ILTPV404, ILTPV405, ILTPV406, ILTPV407, ILTPV408, ILTPV409, ILTPV410, ILTPV411, ILTPV412, ILTPV413, ILTPV414, ILTPV415, ILTPV416, ILTPV417, ILTPV418, ILTPV419, ILTPV420, ILTPV421, ILTPV422, ILTPV423, ILTPV424, ILTPV425, ILTPV426, ILTPV427, ILTPV428, ILTPV429, ILTPV430, ILTPV431, ILTPV432, ILTPV433, ILTPV434, ILTPV435, ILTPV436, ILTPV437, ILTPV438, ILTPV439, ILTPV440, ILTPV441, ILTPV442, ILTPV443, ILTPV444, ILTPV445, ILTPV446, ILTPV447, ILTPV448, ILTPV449, ILTPV450, ILTPV451, ILTPV452, ILTPV453, ILTPV454, ILTPV455, ILTPV456, ILTPV457, ILTPV458, ILTPV459, ILTPV460, ILTPV461, ILTPV462, ILTPV463, ILTPV464, ILTPV465, ILTPV466, ILTPV467, ILTPV468, ILTPV469, ILTPV470, ILTPV471, ILTPV472, ILTPV473, ILTPV474, ILTPV475, ILTPV476, ILTPV477, ILTPV478, ILTPV479, ILTPV480, ILTPV481, ILTPV482, ILTPV483, ILTPV484, ILTPV485, ILTPV486, ILTPV487, ILTPV488, ILTPV489, ILTPV490, ILTPV491, ILTPV492, ILTPV493, ILTPV494, ILTPV495, ILTPV496, ILTPV497, ILTPV498, ILTPV499, ILTPV500, ILTPV501, ILTPV502, ILTPV503, ILTPV504, ILTPV505, ILTPV506, ILTPV507, ILTPV508, ILTPV509, ILTPV510, ILTPV511, ILTPV512, ILTPV513, ILTPV514, ILTPV515, ILTPV516, ILTPV517, ILTPV518, ILTPV519, ILTPV520, ILTPV521, ILTPV522, ILTPV523, ILTPV524, ILTPV525, ILTPV526, ILTPV527, ILTPV528, ILTPV529, ILTPV530, ILTPV531, ILTPV532, ILTPV533, ILTPV534, ILTPV535, ILTPV536, ILTPV537, ILTPV538, ILTPV539, ILTPV540, ILTPV541, ILTPV542, ILTPV543, ILTPV544, ILTPV545, ILTPV546, ILTPV547, ILTPV548, ILTPV549, ILTPV550, ILTPV551, ILTPV552, ILTPV553, ILTPV554, ILTPV555, ILTPV556, ILTPV557, ILTPV558, ILTPV559, ILTPV560, ILTPV561, ILTPV562, ILTPV563, ILTPV564, ILTPV565, ILTPV566, ILTPV567, ILTPV568, ILTPV569, ILTPV570, ILTPV571, ILTPV572, ILTPV573, ILTPV574, ILTPV575, ILTPV576, ILTPV577, ILTPV578, ILTPV579, ILTPV580, ILTPV581, ILTPV582, ILTPV583, ILTPV584, ILTPV585, ILTPV586, ILTPV587, ILTPV588, ILTPV589, ILTPV590, ILTPV591, ILTPV592, ILTPV593, ILTPV594, ILTPV595, ILTPV596, ILTPV597, ILTPV598, ILTPV599, ILTPV600, ILTPV601, ILTPV602, ILTPV603, ILTPV604, ILTPV605, ILTPV606, ILTPV607, ILTPV608, ILTPV609, ILTPV610, ILTPV611, ILTPV612, ILTPV613, ILTPV614, ILTPV615, ILTPV616, ILTPV617, ILTPV618, ILTPV619, ILTPV620, ILTPV621, ILTPV622, ILTPV623, ILTPV624, ILTPV625, ILTPV626, ILTPV627, ILTPV628, ILTPV629, ILTPV630, ILTPV631, ILTPV632, ILTPV633, ILTPV634, ILTPV635, ILTPV636, ILTPV637, ILTPV638, ILTPV639, ILTPV640, ILTPV641, ILTPV642, ILTPV643, ILTPV644, ILTPV645, ILTPV646, ILTPV647, ILTPV648, ILTPV649, ILTPV650, ILTPV651, ILTPV652, ILTPV653, ILTPV654, ILTPV655, ILTPV656, ILTPV657, ILTPV658, ILTPV659, ILTPV660, ILTPV661, ILTPV662, ILTPV663, ILTPV664, ILTPV665, ILTPV666, ILTPV667, ILTPV668, ILTPV669, ILTPV670, ILTPV671, ILTPV672, ILTPV673, ILTPV674, ILTPV675, ILTPV676, ILTPV677, ILTPV678, ILTPV679, ILTPV680, ILTPV681, ILTPV682, ILTPV683, ILTPV684, ILTPV685, ILTPV686, ILTPV687, ILTPV688, ILTPV689, ILTPV690, ILTPV691, ILTPV692, ILTPV693, ILTPV694, ILTPV695, ILTPV696, ILTPV697, ILTPV698, ILTPV699, ILTPV700, ILTPV701, ILTPV702, ILTPV703, ILTPV704, ILTPV705, ILTPV706, ILTPV707, ILTPV708, ILTPV709, ILTPV710, ILTPV711, ILTPV712, ILTPV713, ILTPV714, ILTPV715, ILTPV716, ILTPV717, ILTPV718, ILTPV719, ILTPV720, ILTPV721, ILTPV722, ILTPV723, ILTPV724, ILTPV725, ILTPV726, ILTPV727, ILTPV728, ILTPV729, ILTPV730, ILTPV731, ILTPV732, ILTPV733, ILTPV734, ILTPV735, ILTPV736, ILTPV737, ILTPV738, ILTPV739, ILTPV740, ILTPV741, ILTPV742, ILTPV743, ILTPV744, ILTPV745, ILTPV746, ILTPV747, ILTPV748, ILTPV749, ILTPV750, ILTPV751, ILTPV752, ILTPV753, ILTPV754, ILTPV755, ILTPV756, ILTPV757, ILTPV758, ILTPV759, ILTPV760, ILTPV761, ILTPV762, ILTPV763, ILTPV764, ILTPV765, ILTPV766, ILTPV767, ILTPV768, ILTPV769, ILTPV770, ILTPV771, ILTPV772, ILTPV773, ILTPV774, ILTPV775, ILTPV776, ILTPV777, ILTPV778, ILTPV779, ILTPV780, ILTPV781, ILTPV782, ILTPV783, ILTPV784, ILTPV785, ILTPV786, ILTPV787, ILTPV788, ILTPV789, ILTPV790, ILTPV791, ILTPV792, ILTPV793, ILTPV794, ILTPV795, ILTPV796, ILTPV797, ILTPV798, ILTPV799, ILTPV800, ILTPV801, ILTPV802, ILTPV803, ILTPV804, ILTPV805, ILTPV806, ILTPV807, ILTPV808, ILTPV809, ILTPV810, ILTPV811, ILTPV812, ILTPV813, ILTPV814, ILTPV815, ILTPV816, ILTPV817, ILTPV818, ILTPV819, ILTPV820, ILTPV821, ILTPV822, ILTPV823, ILTPV824, ILTPV825, ILTPV826, ILTPV827, ILTPV828, ILTPV829, ILTPV830, ILTPV831, ILTPV832, ILTPV833, ILTPV834, ILTPV835, ILTPV836, ILTPV837, ILTPV838, ILTPV839, ILTPV840, ILTPV841, ILTPV842, ILTPV843, ILTPV844, ILTPV845, ILTPV846, ILTPV847, ILTPV848, ILTPV849, ILTPV850, ILTPV851, ILTPV852, ILTPV853, ILTPV854, ILTPV855, ILTPV856, ILTPV857, ILTPV858, ILTPV859, ILTPV860, ILTPV861, ILTPV862, ILTPV863, ILTPV864, ILTPV865, ILTPV866, ILTPV867, ILTPV868, ILTPV869, ILTPV870, ILTPV871, ILTPV872, ILTPV873, ILTPV874, ILTPV875, ILTPV876, ILTPV877, ILTPV878, ILTPV879, ILTPV880, ILTPV881, ILTPV882, ILTPV883, ILTPV884, ILTPV885, ILTPV886, ILTPV887, ILTPV888, ILTPV889, ILTPV890, ILTPV891, ILTPV892, ILTPV893, ILTPV894, ILTPV895, ILTPV896, ILTPV897, ILTPV898, ILTPV899, ILTPV900, ILTPV901, ILTPV902, ILTPV903, ILTPV904, ILTPV905, ILTPV906, ILTPV907, ILTPV908, ILTPV909, ILTPV910, ILTPV911, ILTPV912, ILTPV913, ILTPV914, ILTPV915, ILTPV916, ILTPV917, ILTPV918, ILTPV919, ILTPV920, ILTPV921, ILTPV922, ILTPV923, ILTPV924, ILTPV925, ILTPV926, ILTPV927, ILTPV928, ILTPV929, ILTPV930, ILTPV931, ILTPV932, ILTPV933, ILTPV934, ILTPV935, ILTPV936, ILTPV937, ILTPV938, ILTPV939, ILTPV940, ILTPV941, ILTPV942, ILTPV943, ILTPV944, ILTPV945, ILTPV946, ILTPV947, ILTPV948, ILTPV949, ILTPV950, ILTPV951, ILTPV952, ILTPV953, ILTPV954, ILTPV955, ILTPV956, ILTPV957, ILTPV958, ILTPV959, ILTPV960, ILTPV961, ILTPV962, ILTPV963, ILTPV964, ILTPV965, ILTPV966, ILTPV967, ILTPV968, ILTPV969, ILTPV970, ILTPV971, ILTPV972, ILTPV973, ILTPV974, ILTPV975, ILTPV976, ILTPV977, ILTPV978, ILTPV979, ILTPV980, ILTPV981, ILTPV982, ILTPV983, ILTPV984, ILTPV985, ILTPV986, ILTPV987, ILTPV988, ILTPV989, ILTPV990, ILTPV991, ILTPV992, ILTPV993, ILTPV994, ILTPV995, ILTPV996, ILTPV997, ILTPV998, ILTPV999, ILTPV1000, ILTPV1001, ILTPV1002, ILTPV1003, ILTPV1004, ILTPV1005, ILTPV1006, ILTPV1007, ILTPV1008, ILTPV1009, ILTPV1010, ILTPV1011, ILTPV1012, ILTPV1013, ILTPV1014, ILTPV1015, ILTPV1016, ILTPV1017, ILTPV1018, ILTPV1019, ILTPV1020, ILTPV1021, ILTPV1022, ILTPV1023, ILTPV1024, ILTPV1025, ILTPV1026, ILTPV1027, ILTPV1028, ILTPV1029, ILTPV1030, ILTPV1031, ILTPV1032, ILTPV1033, ILTPV1034, ILTPV1035, ILTPV1036, ILTPV1037, ILTPV1038, ILTPV1039, ILTPV1040, ILTPV1041, ILTPV1042, ILTPV1043, ILTPV1044, ILTPV1045, ILTPV1046, ILTPV1047, ILTPV1048, ILTPV1049, ILTPV1050, ILTPV1051, ILTPV1052, ILTPV1053, ILTPV1054, ILTPV1055, ILTPV1056, ILTPV1057, ILTPV1058, ILTPV1059, ILTPV1060, ILTPV1061, ILTPV1062, ILTPV1063, ILTPV1064, ILTPV1065, ILTPV1066, ILTPV1067, ILTPV1068, ILTPV1069, ILTPV1070, ILTPV1071, ILTPV1072, ILTPV1073, ILTPV1074, ILTPV1075, ILTPV1076, ILTPV1077, ILTPV1078, ILTPV1079, ILTPV1080, ILTPV1081, ILTPV1082, ILTPV1083, ILTPV1084, ILTPV1085, ILTPV1086, ILTPV1087, ILTPV1088, ILTPV1089, ILTPV1090, ILTPV1091, ILTPV1092, ILTPV1093, ILTPV1094, ILTPV1095, ILTPV1096, ILTPV1097, ILTPV1098, ILTPV1099, ILTPV1100, ILTPV1101, ILTPV1102, ILTPV1103, ILTPV1104, ILTPV1105, ILTPV1106, ILTPV1107, ILTPV1108, ILTPV1109, ILTPV1110, ILTPV1111, ILTPV1112, ILTPV1113, ILTPV1114, ILTPV1115, ILTPV1116, ILTPV1117, ILTPV1118, ILTPV1119, ILTPV1120, ILTPV1121, ILTPV1122, ILTPV1123, ILTPV1124, ILTPV1125, ILTPV1126, ILTPV1127, ILTPV1128, ILTPV1129, ILTPV1130, ILTPV1131, ILTPV1132, ILTPV1133, ILTPV1134, ILTPV1135, ILTPV1136, ILTPV1137, ILTPV1138, ILTPV1139, ILTPV1140, ILTPV1141, ILTPV1142, ILTPV1143, ILTPV1144, ILTPV1145, ILTPV1146, ILTPV1147, ILTPV1148, ILTPV1149, ILTPV1150, ILTPV1151, ILTPV1152, ILTPV1153, ILTPV1154, ILTPV1155, ILTPV1156, ILTPV1157, ILTPV1158, ILTPV1159, ILTPV1160, ILTPV1161, ILTPV1162, ILTPV1163, ILTPV1164, ILTPV1165, ILTPV1166, ILTPV1167, ILTPV1168, ILTPV1169, ILTPV1170, ILTPV1171, ILTPV1172, ILTPV1173, ILTPV1174, ILTPV1175, ILTPV1176, ILTPV1177, ILTPV1178, ILTPV1179, ILTPV1180, ILTPV1181, ILTPV1182, ILTPV1183, ILTPV1184, ILTPV1185, ILTPV1186, ILTPV1187, ILTPV1188, ILTPV1189, ILTPV1190, ILTPV1191, ILTPV1192, ILTPV1193, ILTPV1194, ILTPV1195, ILTPV1196, ILTPV1197, ILTPV1198, ILTPV1199, ILTPV1200, ILTPV1201, ILTPV1202, ILTPV1203, ILTPV1204, ILTPV1205, ILTPV1206, ILTPV1207, ILTPV1208, ILTPV1209, ILTPV1210, ILTPV1211, ILTPV1212, ILTPV1213, ILTPV1214, ILTPV1215, ILTPV1216, ILTPV1217, ILTPV1218, ILTPV1219, ILTPV1220, ILTPV1221, ILTPV1222, ILTPV1223, ILTPV1224, ILTPV1225, ILTPV1226, ILTPV1227, ILTPV1228, ILTPV1229, ILTPV1230, ILTPV1231, ILTPV1232, ILTPV1233, ILTPV1234, ILTPV1235, ILTPV1236, ILTPV1237, ILTPV1238, ILTPV1239, ILTPV1240, ILTPV1241, ILTPV1242, ILTPV1243, ILTPV1244, ILTPV1245, ILTPV1246, ILTPV1247, ILTPV1248, ILTPV1249, ILTPV1250, ILTPV1251, ILTPV1252, ILTPV1253, ILTPV1254, ILTPV1255, ILTPV1256, ILTPV1257, ILTPV1258, ILTPV1259, ILTPV1260, ILTPV1261, ILTPV1262, ILTPV1263, ILTPV1264, ILTPV1265, ILTPV1266, ILTPV1267, ILTPV1268, ILTPV1269, ILTPV1270, ILTPV1271, ILTPV1272, ILTPV1273, ILTPV1274, ILTPV1275, ILTPV1276, ILTPV1277, ILTPV1278, ILTPV1279, ILTPV1280, ILTPV1281, ILTPV1282, ILTPV1283, ILTPV1284, ILTPV1285, ILTPV1286, ILTPV1287, ILTPV1288, ILTPV1289, ILTPV1290, ILTPV1291, ILTPV1292, ILTPV1293, ILTPV1294, ILTPV1295, ILTPV1296, ILTPV1297, ILTPV1298, ILTPV1299, ILTPV1300, ILTPV1301, ILTPV1302, ILTPV1303, ILTPV1304, ILTPV1305, ILTPV1306, ILTPV1307, ILTPV1308, ILTPV1309, ILTPV1310, ILTPV1311, ILTPV1312, ILTPV1313, ILTPV1314, ILTPV1315, ILTPV1316, ILTPV1317, ILTPV1318, ILTPV1319, ILTPV1320, ILTPV1321, ILTPV1322, ILTPV1323, ILTPV1324, ILTPV1325, ILTPV1326, ILTPV1327, ILTPV1328, ILTPV1329, ILTPV1330, ILTPV1331, ILTPV1332, ILTPV1333, ILTPV1334, ILTPV1335, ILTPV1336, ILTPV1337, ILTPV1338, ILTPV1339, ILTPV1340, ILTPV1341, ILTPV1342, ILTPV1343, ILTPV1344, ILTPV1345, ILTPV1346, ILTPV1347, ILTPV1348, ILTPV1349, ILTPV1350, ILTPV1351, ILTPV1352, ILTPV1353, ILTPV1354, ILTPV1355, ILTPV1356, ILTPV1357, ILTPV1358, ILTPV1359, ILTPV1360, ILTPV1361, ILTPV1362, ILTPV1363, ILTPV1364, ILTPV1365, ILTPV1366, ILTPV1367, ILTPV1368, ILTPV1369, ILTPV1370, ILTPV1371, ILTPV1372, ILTPV1373, ILTPV1374, ILTPV1375, ILTPV1376, ILTPV1377, ILTPV1378, ILTPV1379, ILTPV1380, ILTPV1381, ILTPV1382, ILTPV1383, ILTPV1384, ILTPV1385, ILTPV1386, ILTPV1387, ILTPV1388, ILTPV1389, ILTPV1390, ILTPV1391, ILTPV1392, ILTPV1393, ILTPV1394, ILTPV1395, ILTPV1396, ILTPV1397, ILTPV1398, ILTPV1399, ILTPV1400, ILTPV1401, ILTPV1402, ILTPV1403, ILTPV1404, ILTPV1405, ILTPV1406, ILTPV1407, ILTPV1408, ILTPV1409, ILTPV1410, ILTPV1411, ILTPV1412, ILTPV1413, ILTPV1414, ILTPV1415, ILTPV1416, ILTPV1417, ILTPV1418, ILTPV1419, ILTPV1420, ILTPV1421, ILTPV1422, ILTPV1423, ILTPV1424, ILTPV1425, ILTPV1426, ILTPV1427, ILTPV1428, ILTPV1429, ILTPV1430, ILTPV1431, ILTPV1432, ILTPV1433, ILTPV1434, ILTPV1435, ILTPV1436, ILTPV1437, ILTPV1438, ILTPV1439, ILTPV1440, ILTPV1441, ILTPV1442, ILTPV1443, ILTPV1444, ILTPV1445, ILTPV1446, ILTPV1447, ILTPV1448, ILTPV1449, ILTPV1450, ILTPV1451, ILTPV1452, ILTPV1453, ILTPV1454, ILTPV1455, ILTPV1456, ILTPV1457, ILTPV1458, ILTPV1459, ILTPV1460, ILTPV1461, ILTPV1462, ILTPV1463, ILTPV1464, ILTPV1465, ILTPV1466, ILTPV1467, ILTPV1468, ILTPV1469, ILTPV1470, ILTPV1471, ILTPV1472, ILTPV1473, ILTPV1474, ILTPV1475, ILTPV1476, ILTPV1477, ILTPV1478, ILTPV1479, ILTPV1480, ILTPV1481, ILTPV1482, ILTPV1483, ILTPV1484, ILTPV1485, IL

Why is CCA identification hard?

People keep deploying *new* CCAs – we **can't be ad hoc**

Work well with a wider range of applications and application traffic – be **client-agnostic**

We can't appear hostile – we need to be as **passive** as possible

Why is CCA identification hard?

People keep deploying *new* CCAs – we **can't be ad hoc**

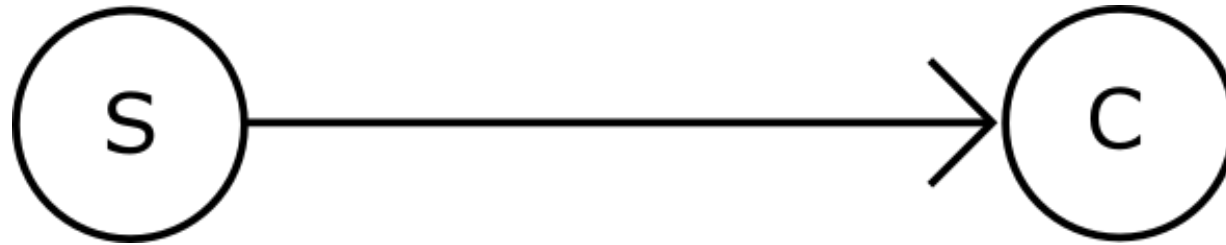
Work well with a wider range of applications and application traffic – be **client-agnostic**

We can't appear hostile – we need to be as **passive** as possible

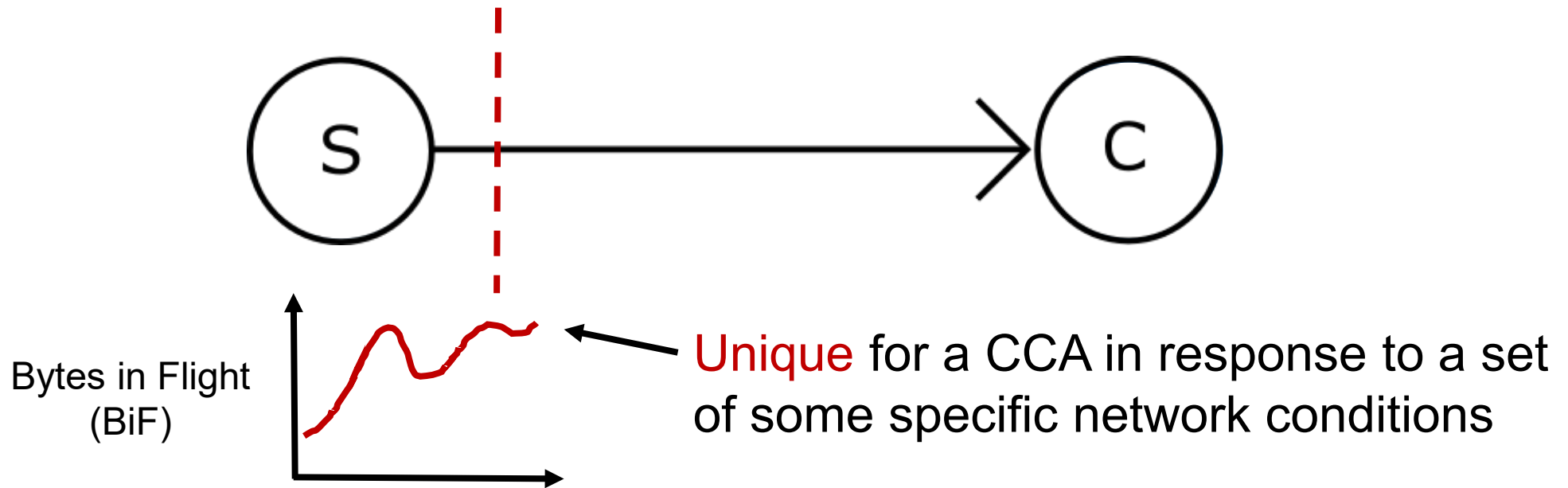
There *is* an obvious way to identify CCAs while meeting all these criteria, albeit in the controlled setting.

Let's review the task we have at hand:

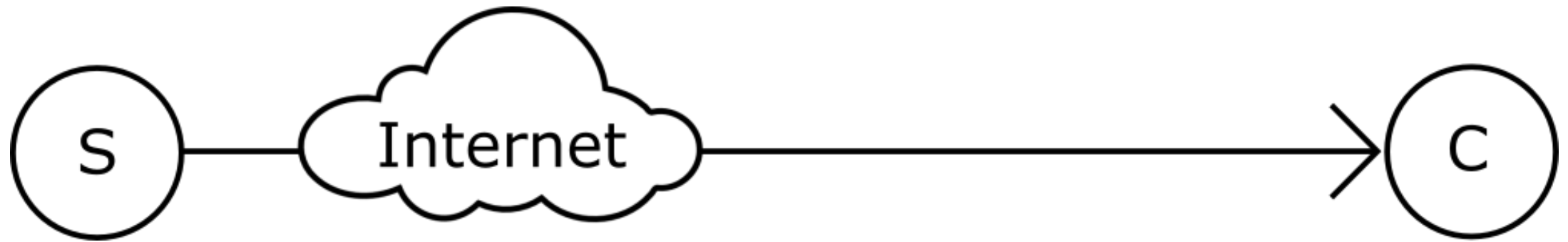
Identify the Congestion Control
Algorithm run by a server



Identify the Congestion Control Algorithm run by a server

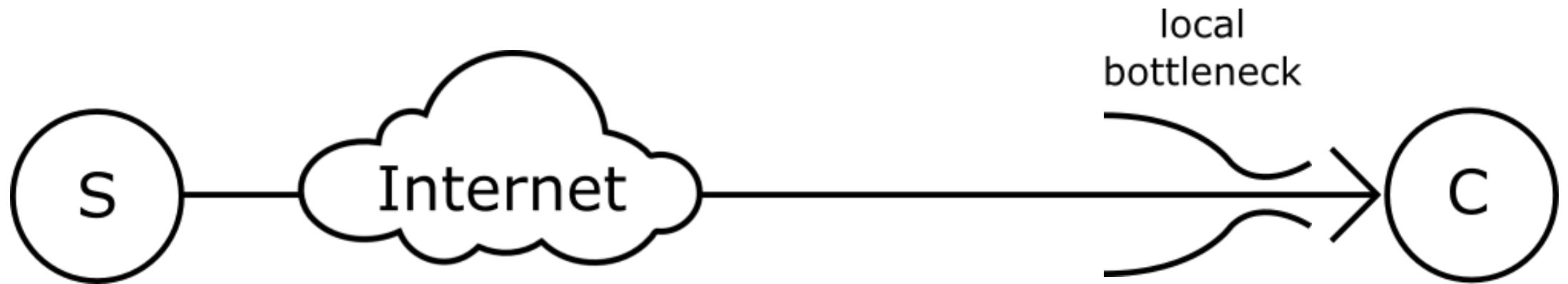


Identify the Congestion Control Algorithm run by a **remote** server on the **Internet**

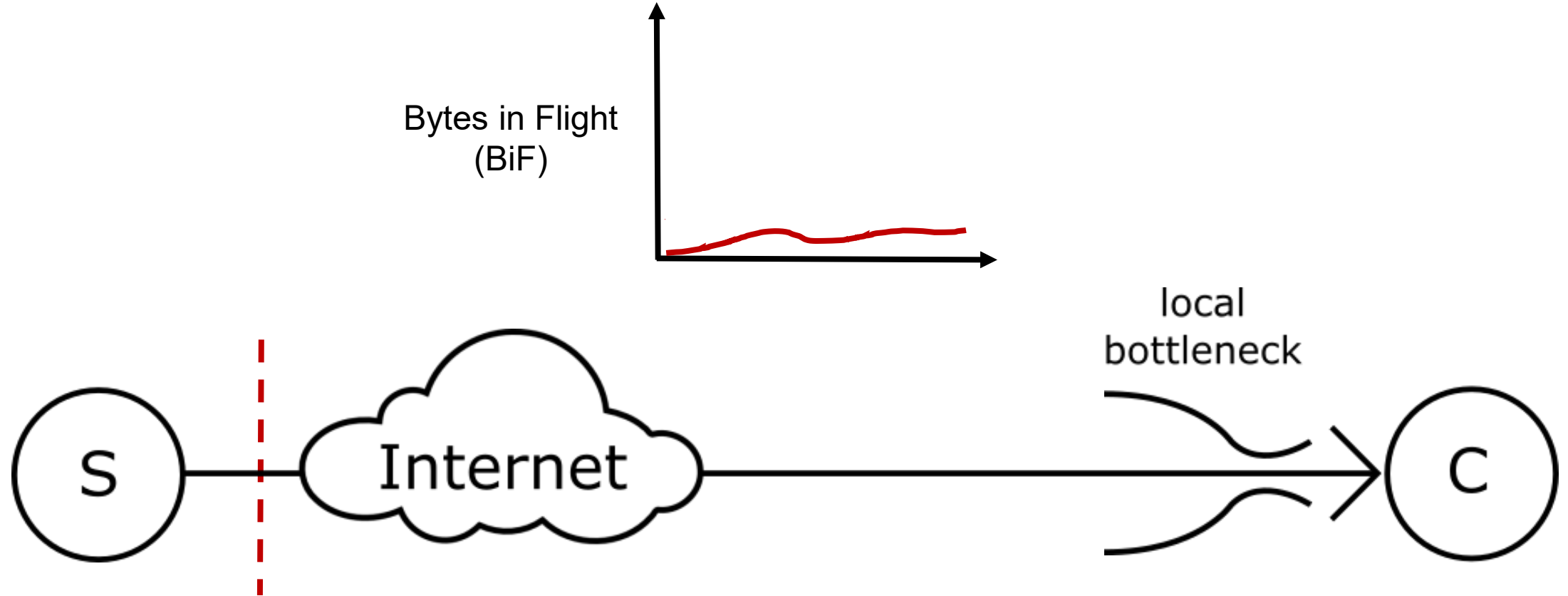


We no more have **control** over the network conditions

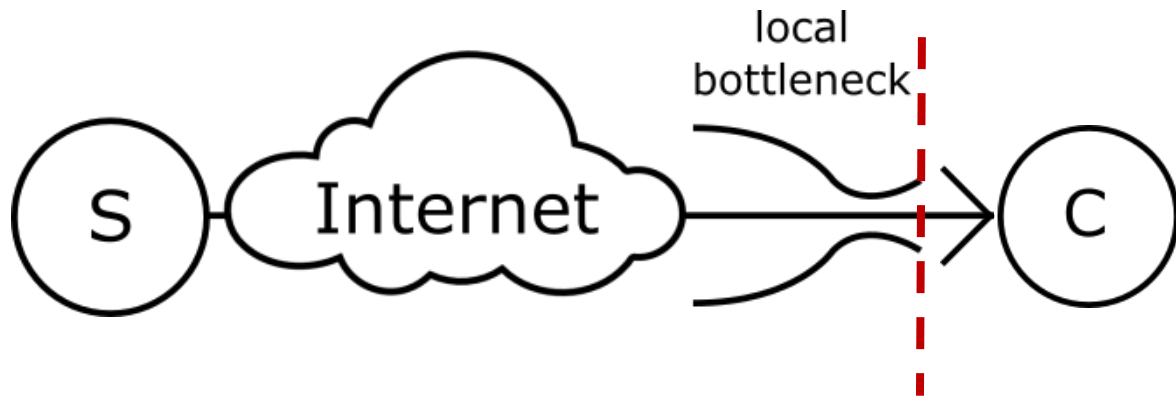
Identify the Congestion Control Algorithm run by a **remote** server on the **Internet**



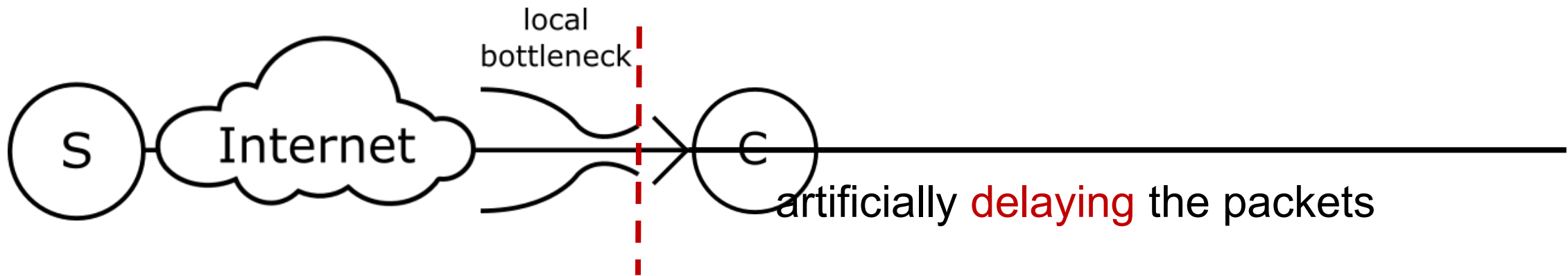
We can solve this how we have solved it before: via a **localized bottleneck**

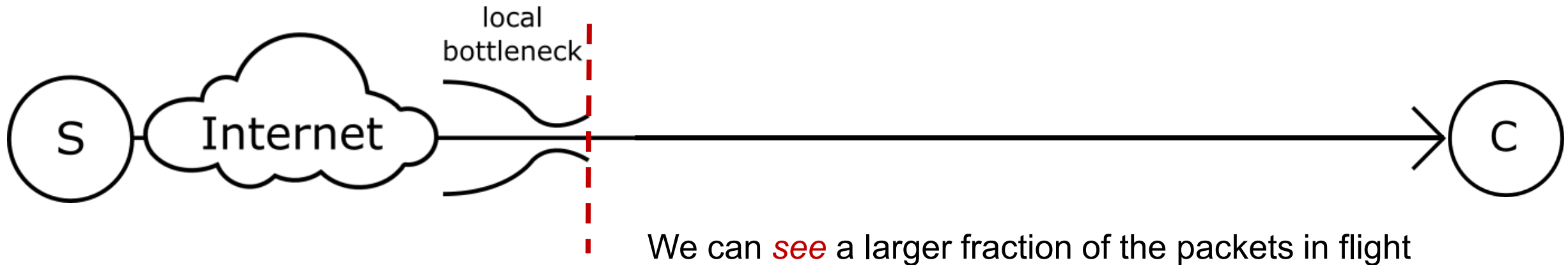
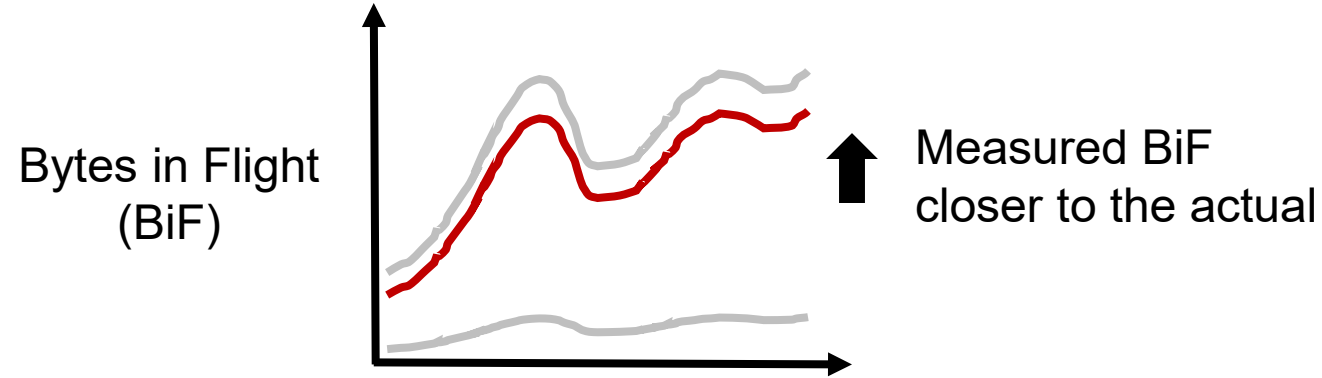


What's the solution here?
We can't go any **nearer** to the server...



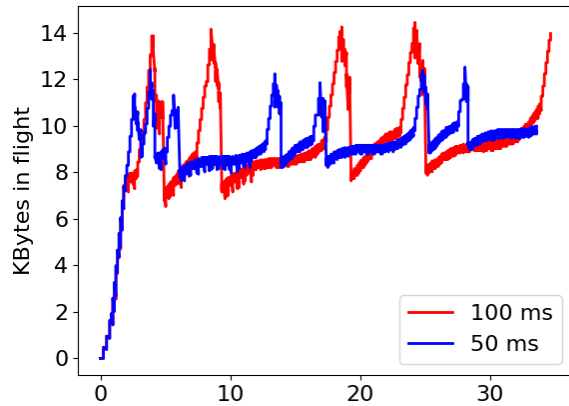
...but we can get **further** from the client



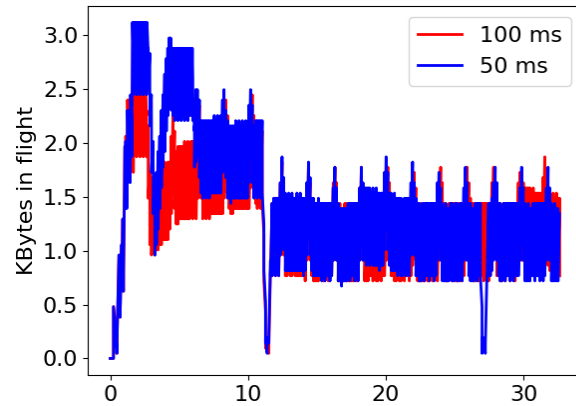


Nebby's measurement methodology is built on this **key insight**

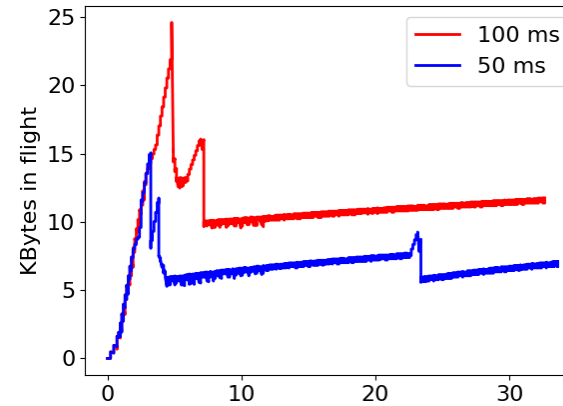
This simple strategy is good enough to capture distinct BiF traces for most CCAs



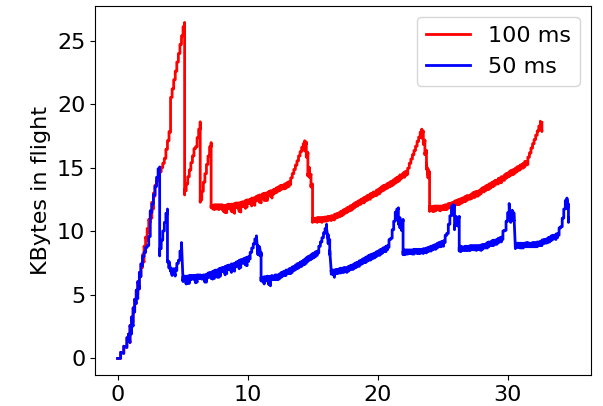
TCP CUBIC



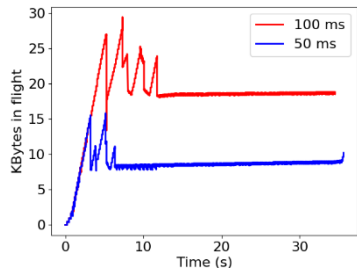
TCP BBR



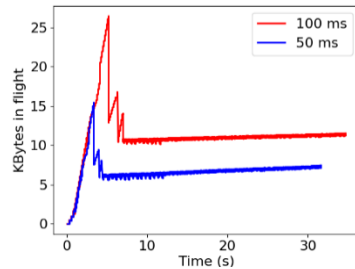
TCP New Reno



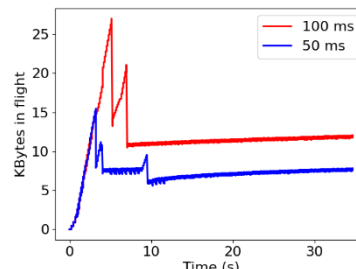
HTCP



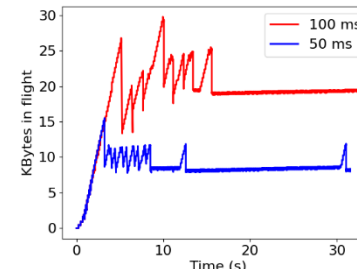
TCP BIC



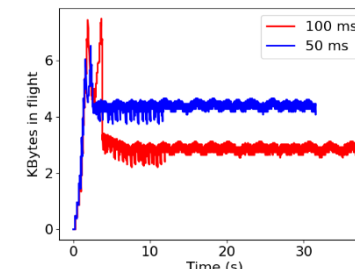
TCP Highspeed



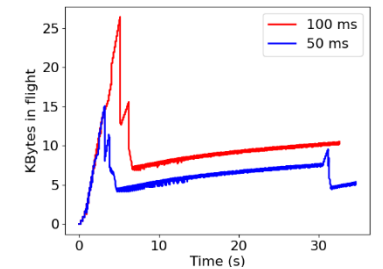
TCP Illinois



TCP Scalable

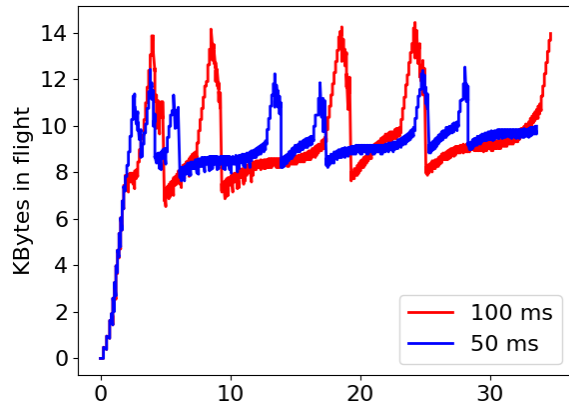


TCP Vegas

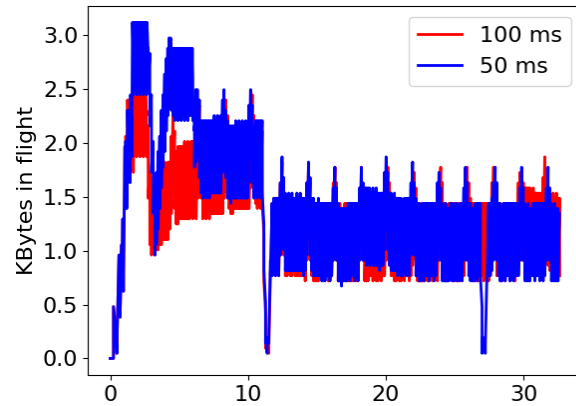


TCP Westwood

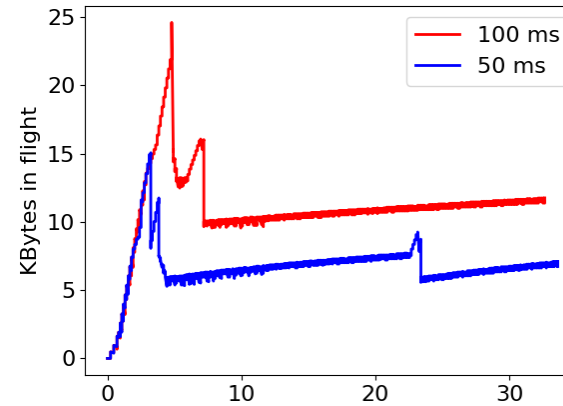
This simple strategy is good enough to capture **distinct BiF traces** for most CCAs



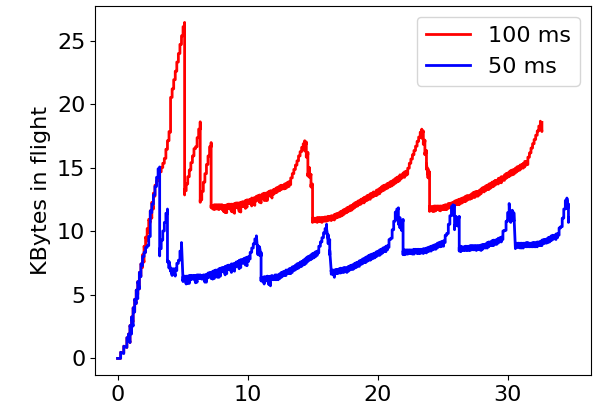
TCP CUBIC



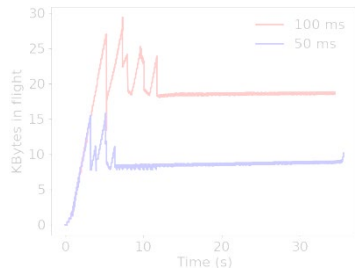
TCP BBR



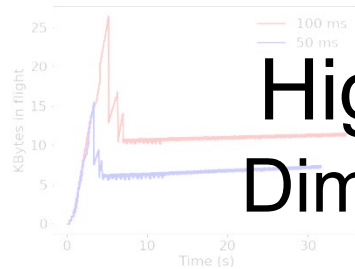
TCP New Reno



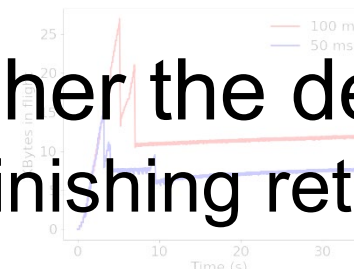
HTCP



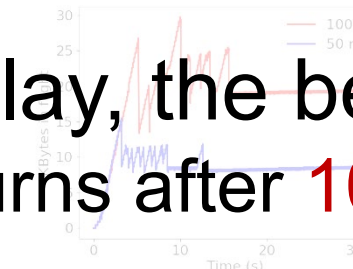
TCP BIC



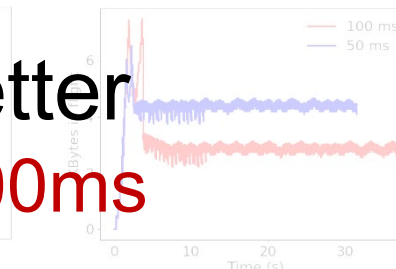
TCP Highspeed



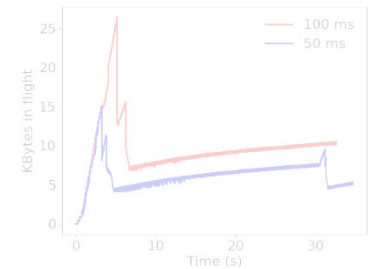
TCP Illinois



TCP Scalable



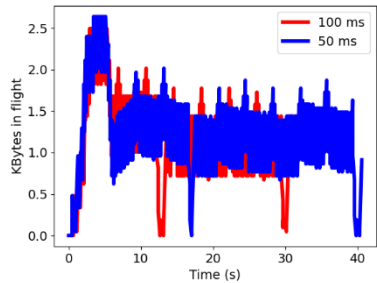
TCP Vegas



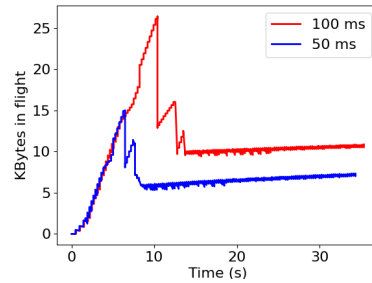
TCP Westwood

Higher the delay, the better
Diminishing returns after **100ms**

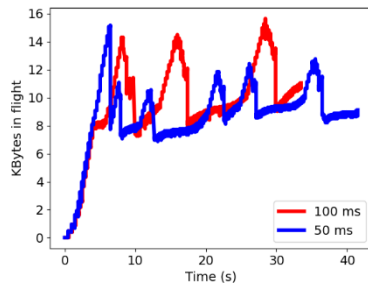
Moreover, since Nebby is mostly passive, it can work with **QUIC** and **real browser** traffic too



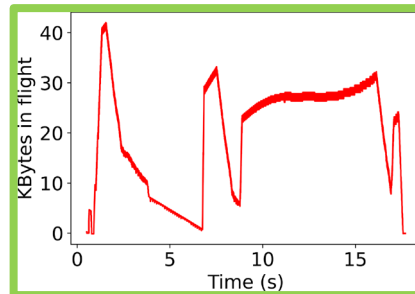
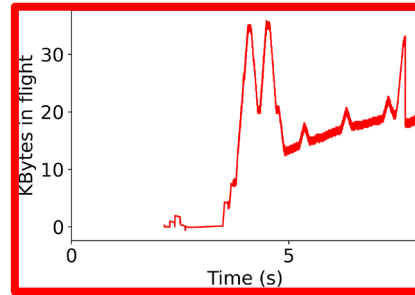
mvfst BBR



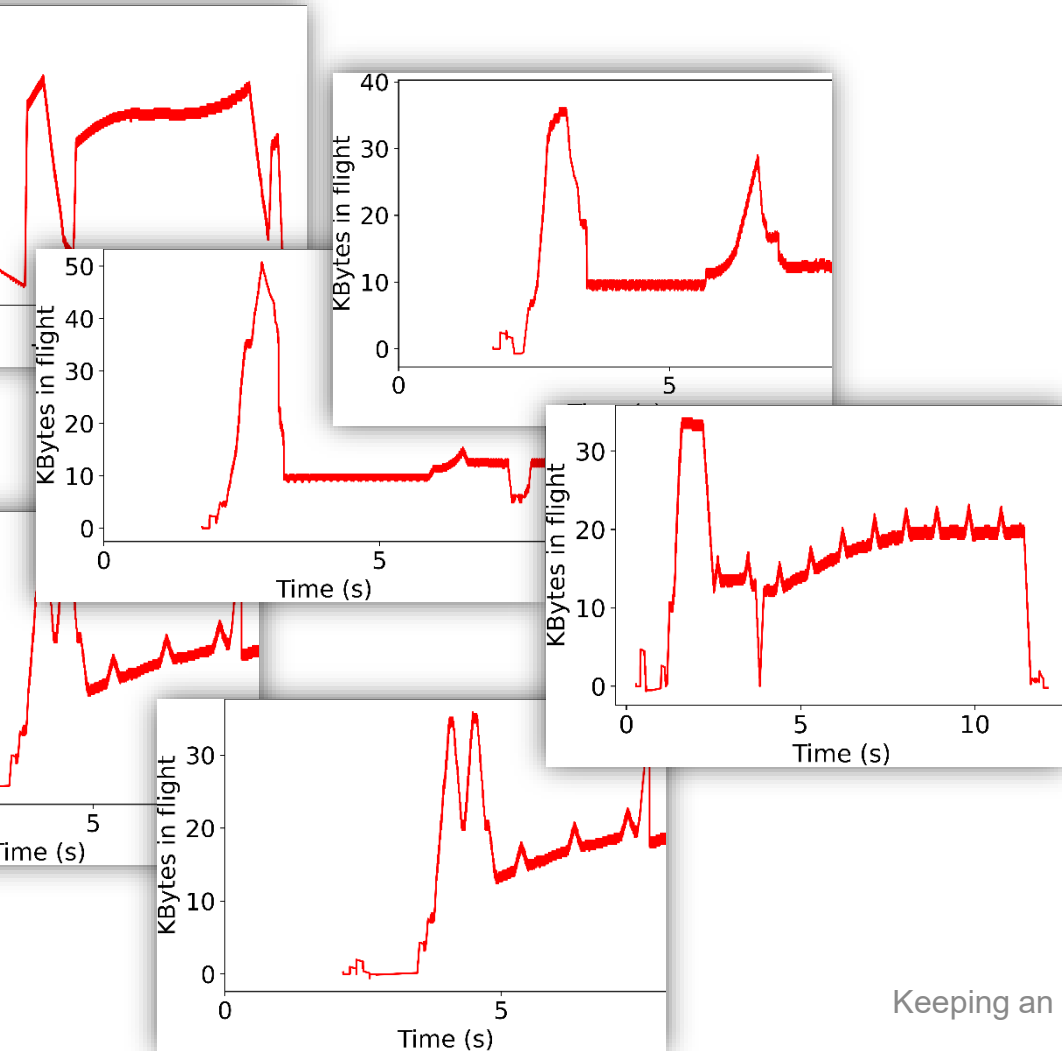
quicgo BBR



quiche CUBIC



How do we build a **classifier** for all these traces?

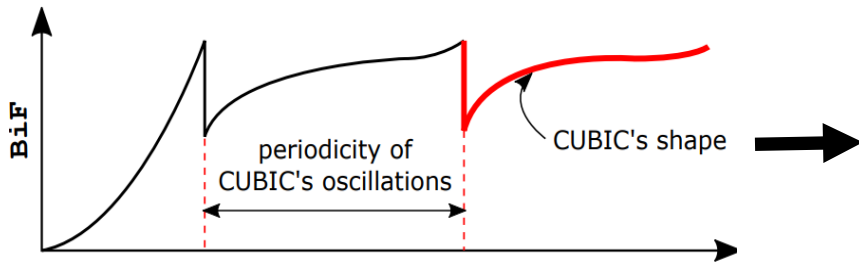


Avoiding a common pitfall:
ML-based classifier

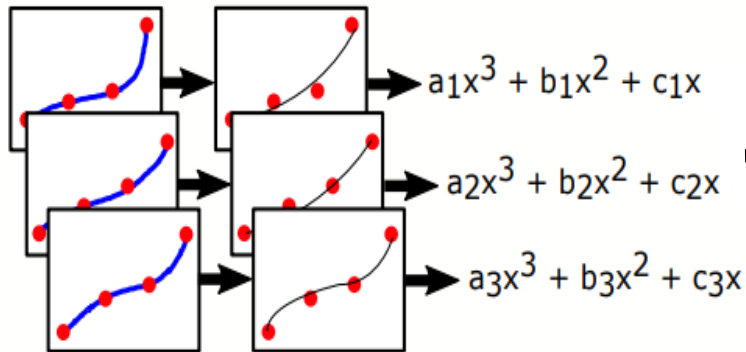
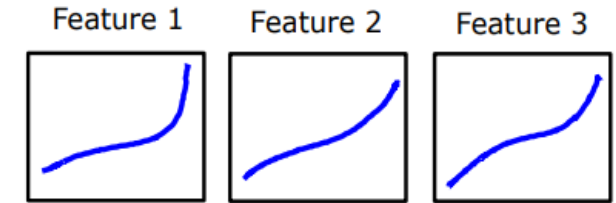
Key insight:

All reasonable CCAs are feedback loops with periodic probes and oscillations in the congestion avoidance phase.

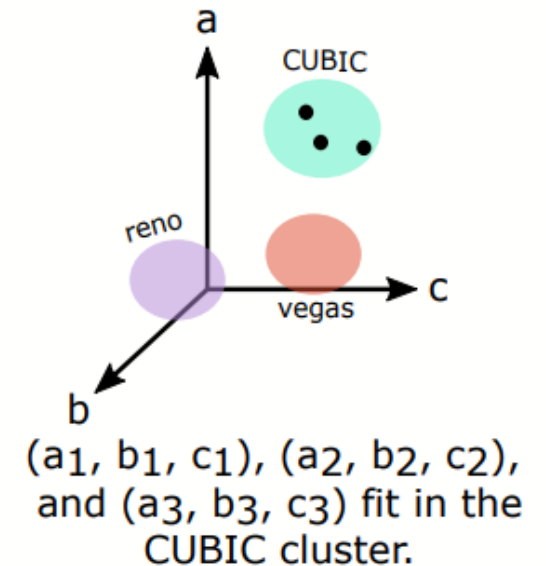
Using a shape-based Classifier



Extract these
periodic regions:
duration and
frequency



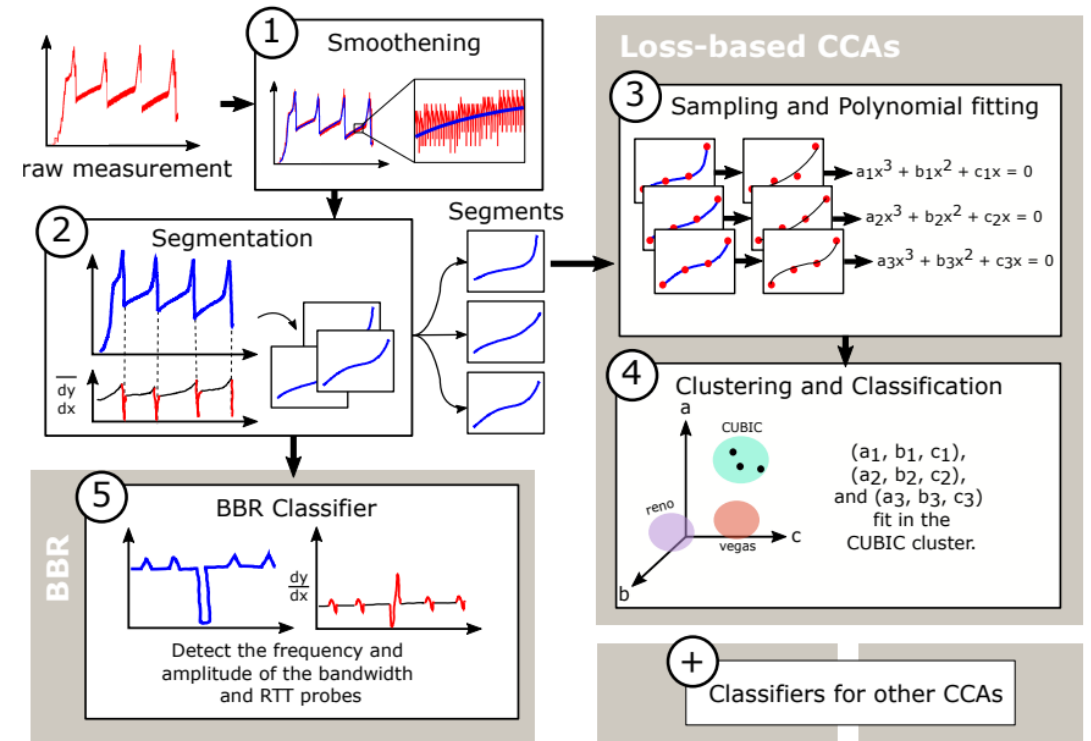
Fit Polynomials and
compare them with
the polynomials for
oscillations of
known CCAs



Using a shape-based Classifier

Shape-based classifier can successfully classify all CCAs in the Linux kernel and BBRv2 with an average **accuracy of 96%**

Nebby is **demonstrably extensible**, with support for CCAs like Copa and PCC. Can detect unknown CCAs deployed by popular websites.



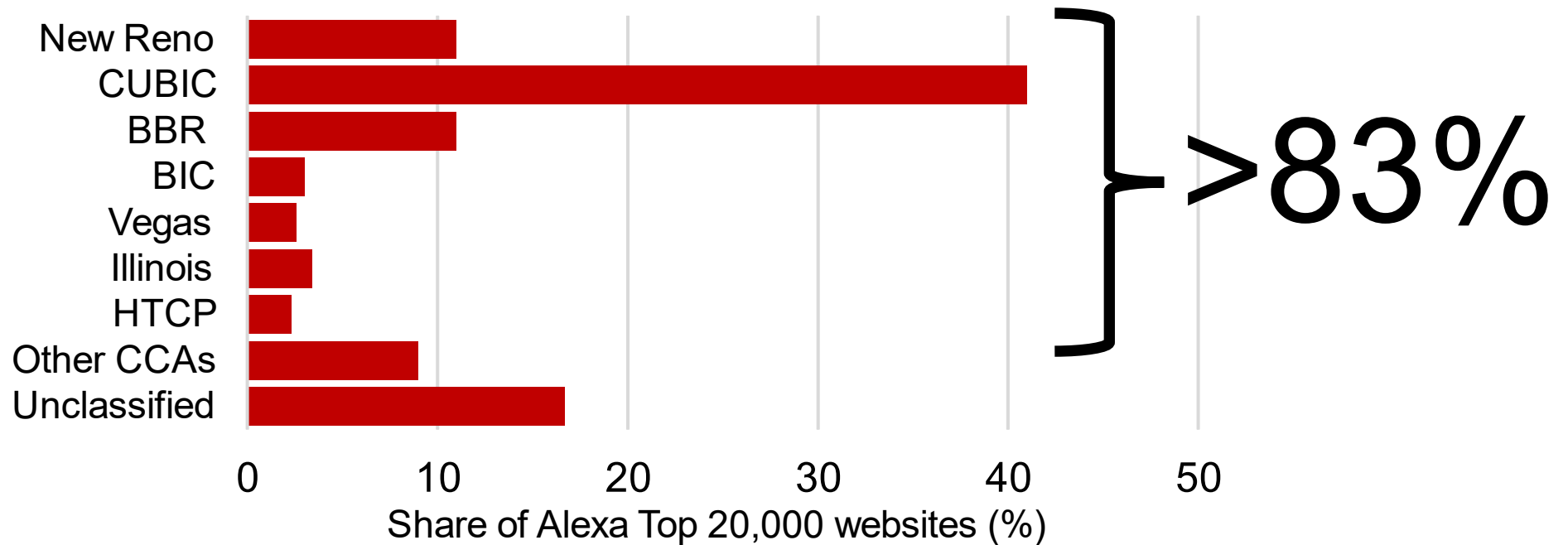
Internet Measurement Results

We made our measurements from `aws` instances in Ohio, Paris, Mumbai, Singapore, and Sao Paulo

We measured the **Alexa Top 20,000 websites** over a single **TCP** connection (`wget`) and a single **QUIC** connection (`quiche`)

We also measured a selection of websites that stream video, audio, and other dynamic content via a **chromium web browser**.

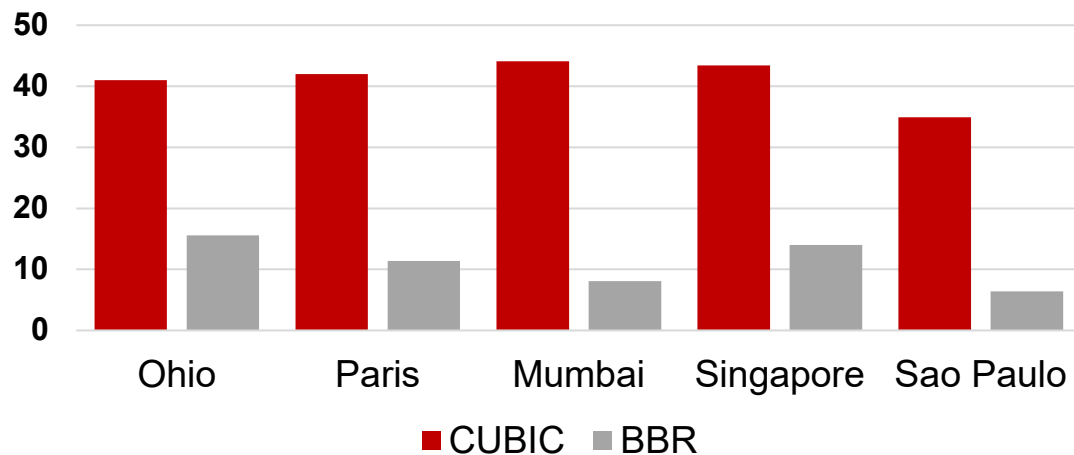
Internet Measurement Results



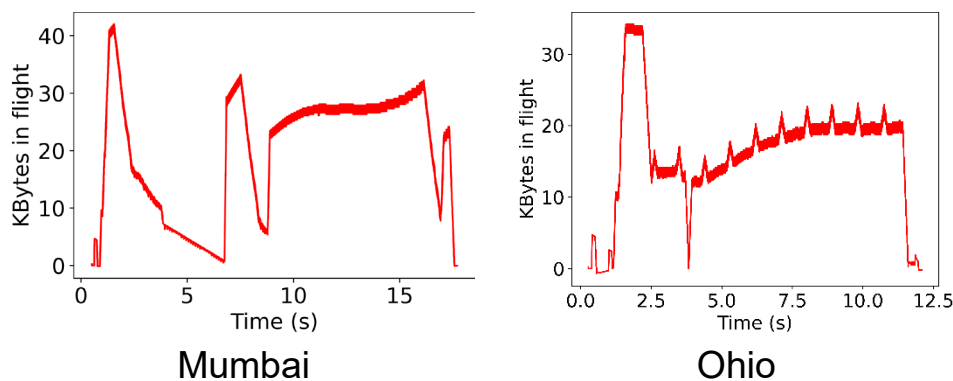
5 key findings

...and many more nuggets covered in the paper

#1 CCA Deployment differs by region



While CUBIC and BBR remain the two most dominant CCAs on the Internet, their deployment differs by region



Some websites deploy different CCAs in different regions. For example, `amazon.in` uses CUBIC in Mumbai and BBRv1 in Ohio

#2 Slow Migration to BBRv2

Since the last measurement study, Google has proposed BBRv2, a more fairness-conscious alternative to BBRv1

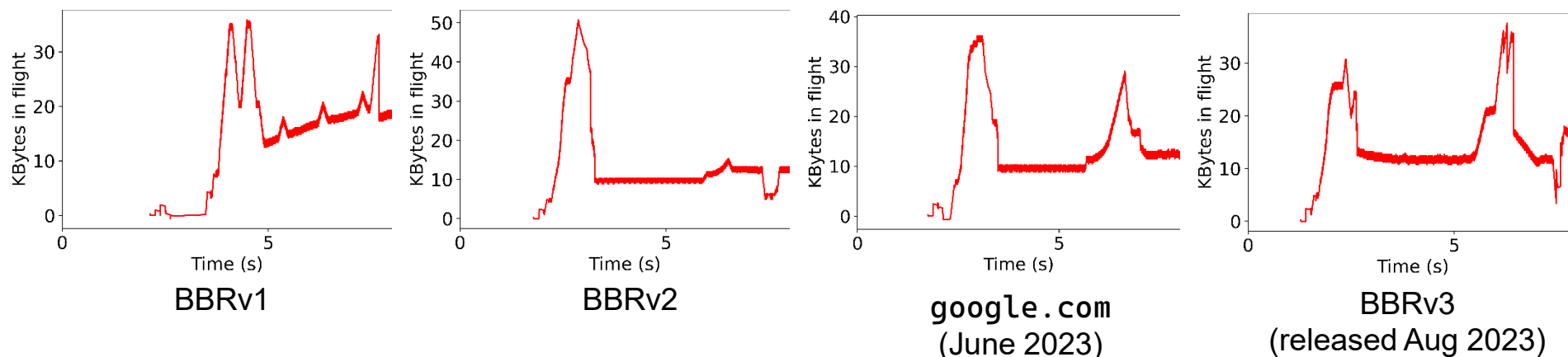
However, despite this, about 98% of websites that deployed BBRv1 in 2019 have either stuck to BBRv1 or switched to CUBIC

Most websites that deploy BBRv2 are new adopters of BBR

#3 Unknown CCAs on the Internet

Nebby found about **24% of websites** deploy CCAs that did not resemble any CCAs in the Linux kernel or BBRv2

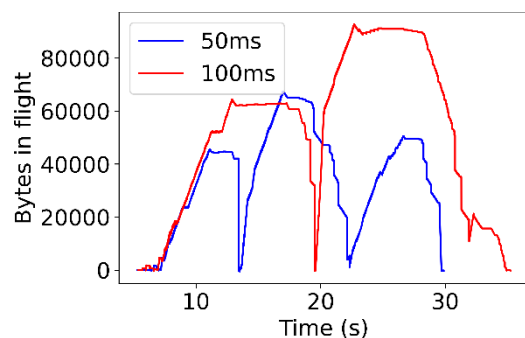
About 200 of these websites were Google domains(!)



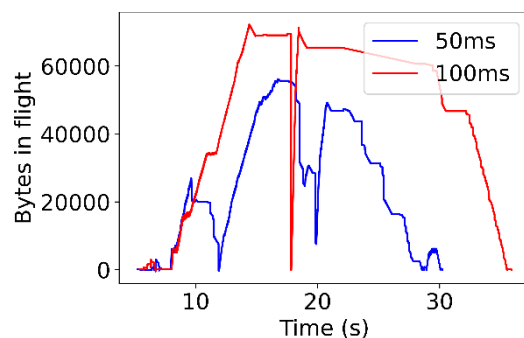
#3 Unknown CCAs on the Internet

Nebby found about **24% of websites** deploy CCAs that did not resemble any CCAs in the Linux kernel or BBRv2

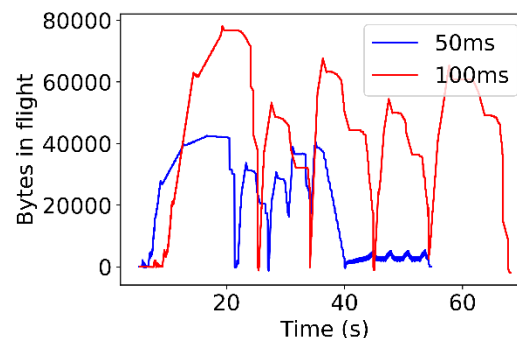
A large proportion of these websites hosted on Akamai were also deploying their own CCAs



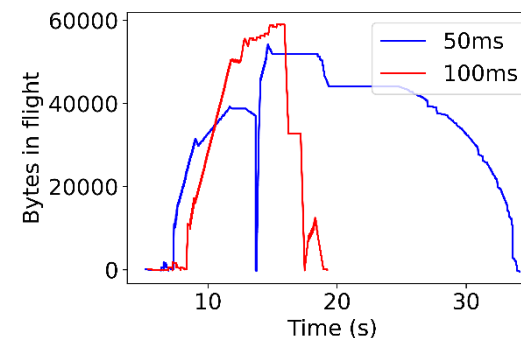
apple.com



hulu.com



pornhub.com



tiktok.com

#4 QUIC still has a long way to go

We saw a surprisingly small number of websites in the Alexa Top 20,000 websites respond to QUIC

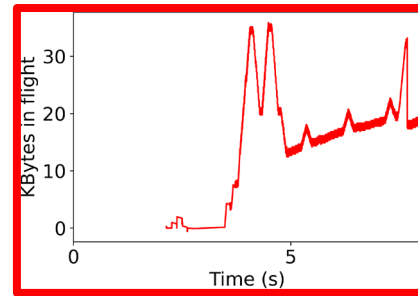
About only 8% of measured websites deployed QUIC

CUBIC and BBR were equally popular amongst websites deploying QUIC

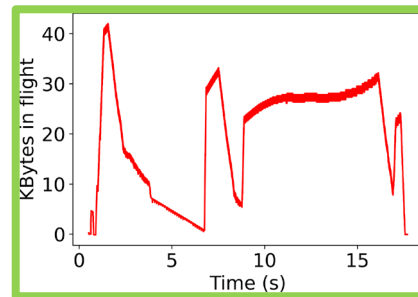
#5 Deployment differs by asset type

Websites like `twitch.tv` and `apple.tv.com` chose to deploy different CCAs for serving different content

In general, BBR is **more popular with websites serving video**



BBR



CUBIC



Nebby's Limitations

The current weakest link is the classifier – even though it is modular, it requires **a new module for new CCAs**

The classifier is also only as good as the selected network profile. However, there is scope for **generating specific network profiles for new CCAs**



In Summary...

We introduce a **fresh methodology** for studying and identifying CCAs on the Internet for **TCP**, **QUIC**, and **live clients**.

We show that while BBR's adoption has slowed down, most **bandwidth intensive applications** still opt for **BBR**.

BBRv3 and **AkamaiCC** are case studies in using Nebby to catch the deployment of unknown congestion control algorithms.

Nebby is **open source** and available on GitHub!