An Axiomatic Approach to Congestion Control

Doron Zarchy
Hebrew University of Jerusalem
doronz@cs.huji.ac.il

Radhika Mittal
UC Berkeley
radhika@eecs.berkeley.edu

Michael Schapira
Hebrew University of Jerusalem
schapiram@huji.ac.il

Scott Shenker
UC Berkeley, ICSI
shenker@icsi.berkeley.edu

ABSTRACT

Recent years have witnessed a surge of interest in congestion control. Unfortunately, the overwhelmingly large design space along with the increasingly diverse range of application environments makes evaluating congestion control protocols a daunting task. Researchers often use simulation and experiments to examine the performance of designs in *specific* contexts, but this gives limited insight into the more general properties of these schemes and provides no information about the *inherent* limits of congestion control designs, e.g., which properties are *simultaneously* achievable. To complement simulation and experimentation, we advocate a principled framework for reasoning about congestion control protocols. We report on our initial steps in this direction, which was inspired by the axiomatic approach from social choice theory and game theory. We consider several natural requirements ("axioms") from congestion control protocols – e.g., efficient resource-utilization, loss-avoidance, fairness, stability, and TCP-friendliness – and investigate which combinations of these can be achieved within a single design. Thus, our framework allows us to investigate the fundamental tradeoffs between desiderata, and to identify where existing and new congestion control architectures fit within the space of possible outcomes. We believe that our results are but a first step in the axiomatic exploration of congestion control and leave the reader with exciting directions for future research.

1 INTRODUCTION

Recent years have witnessed a revival of interest in both industry and academia in improving congestion control on the Internet. The quest for better congestion control designs is complicated by the extreme diversity and range of (i) the design space (as exemplified by the stark conceptual and operational differences between recent proposals [8, 11, 29, 30]), (ii) the desired properties (ranging from high performance to fairness, to TCP-friendliness), (iii) the envisioned operational setting (inter- and intra-datacenter, wireless, the commercial Internet, satellite), and (iv) the application loads and requirements (small vs. large traffic demands, latency- vs. bandwidth-sensitive, etc.).

Rather than trying to make general statements that apply across the entire range of possibility, most congestion control research uses simulation and experiments of designs under a limited range of network conditions. This is extremely important for understanding the detailed performance of particular schemes in specific settings. In contrast, traditional theoretical methods – such as network-utility maximization [16, 26] and control-theoretic approaches [1, 22] – yield results that apply across a wider range of conditions, but typically cannot address important concerns such as the temporal aspects of dynamics of congestion control, TCP friendliness, or robustness to "non-congestion loss" [8].

We advocate an axiomatic approach to congestion control, which is complementary to the simulation/experiment and theoretical work currently pursued. Our approach, modeled on similar efforts in social choice theory and game theory [5], identifies a set of requirements ("axioms") and then identifies (i) which of its subsets of requirements can coexist (i.e., there are designs that achieve all of them) and which subsets cannot be met simultaneously (i.e., no design can *simultaneously* achieve all of them), and (ii) whether some requirements immediately follow from satisfying other requirements. Thus, the axiomatic approach can shed light on the *inherent* tradeoffs involved in congestion control protocol design, and can classify existing and proposed solutions according to the properties they satisfy.

The axiomatic approach has been applied to many computer science environments, including reputation systems [27], recommendation systems [4], link prediction [10], and networking environments [10, 18, 23]. To the best of our knowledge, ours is the first application of this approach to congestion control protocols (though [23] touches on the subject briefly).

To be sure, the axiomatic approach has its limitations, as it revolves around investigating these properties in a highly simplified model. However, we feel that the results, in terms

HotNets-XVI, November 30–December 1, 2017, Palo Alto, CA, USA

ACM ISBN 978-1-4503-5569-8/17/11...$15.00
https://doi.org/10.1145/3152434.3152445

of which axioms can coexist and which cannot, provide insights that apply far beyond the simple model (even if the detailed theoretical results do not). Thus, we contend that the axiomatic approach is a useful addition to the evaluatory arsenal that researchers should apply to congestion control.

In the following section, we introduce the simple network model where we can evaluate congestion control designs. We then, in Section 3, formulate several natural axioms (or requirements) for congestion control protocols, including efficient link-utilization, loss-avoidance, fairness, stability, and TCP-friendliness. In Section 4, we derive our basic results on the feasibility of achieving these requirements and then, in Section 5, we show that congestion control protocols can be regarded as points in a multidimensional space reflecting the extent to which they satisfy these requirements.

We argue that from a protocol design perspective, desirable congestion control protocols are those that reside on the Pareto frontier in the multidimensional space induced by our axioms. To illustrate this point, we introduce a new class of protocols, termed "robust AIMD", which extends traditional TCP by borrowing ideas from the recently introduced PCC protocol [11]. We show that "robust-AIMD" protocols can be viewed as points on the Pareto frontier that outperform traditional TCP yet are significantly less aggressive to legacy TCP connections than PCC.

We believe that the above results are but a first step in the axiomatic exploration of congestion control and leave the reader with exciting directions for future research.

2 MODELING PROTOCOL DYNAMICS

To illustrate our approach, we consider a simple fluid-flow model of dynamics of congestion control protocols (which builds upon [6, 9, 21]). We focus on window-based protocols in congestion-avoidance mode interacting on a single (bottleneck) link with FIFO (droptail) queuing. We defer axiomatic explorations of richer models (e.g., more expressive queuing policies, network-level interactions), and of pacing-based protocols (e.g., the PCC and BBR) to future research. We show below that studying our simple (and tractable) model gives rise to interesting insights about the fundamental tradeoffs involved in congestion control protocol design.

Senders dynamically interacting on a link. *n senders* $1, \dots n$ send traffic on a link of bandwidth $B > 0$ (measured in units of MSS/s), propagation delay Θ, and buffer size τ (measured in units of MSS). Importantly, in our model B, Θ, and τ are all unknown to the senders, precluding the possibility of building into protocols a priori assumptions such as "the link is only shared by senders running protocol P", "the minimum latency is at least $2ms$", etc. We let $C = B \times 2\Theta$, i.e., C represents the minimum possible bandwidth-delay product for the link, or its "capacity". Senders experience *synchronized* feedback. Time in our model is regarded as an infinite sequence of discrete time steps $t = 0, 1, 2, \dots$, each of RTT duration, in which end-host (sender) decisions occur. At the beginning of each time step t, sender i can adjust its congestion window by selecting a value in the range $\{0, 1, \dots, M\}$, in units of MSS. We assume that $1 << M$. Let $x_i^{(t)}$ denote the size of the sender i's congestion window at time t, let $\bar{x}^{(t)} = (x_1^{(t)}, \dots, x_n^{(t)})$, and let $X^{(t)} = \sum_i x_i^{(t)}$.

The duration of time step t, $RTT(t)$, is a function of the link's propagation delay and the buffer's queuing delay. This is captured (as in [17, 19, 32]) as follows:

$$RTT(\bar{x}^{(t)}, C, \tau) = \begin{cases} \max(2\Theta, \frac{X^{(t)}-C}{B} + 2\Theta) \\ \text{if } X^{(t)} < C + \tau \\ \Delta \quad \text{otherwise} \end{cases} \tag{1}$$

where Δ represents a timeout-triggered cap on the RTT in the event of packet loss.

When traffic sent on the link at time t exceeds $C + \tau$, excess traffic is dropped according to the FIFO (droptail) queuing policy. The *loss rate* experienced by each sender at time step t as a function of the congestion window sizes of all senders and the link capacity and buffer size, is captured as:

$$L(\bar{x}^{(t)}, C, \tau) = \begin{cases} (1 - \frac{C+\tau}{X^{(t)}}) & \text{if } X^{(t)} > C + \tau \\ 0 & \text{otherwise} \end{cases}$$

To simplify exposition, we will denote the loss rate at time t simply by $L^{(t)}$.

Congestion control protocols and the induced dynamics. Each sender i's selection of congestion-window sizes is dictated by the *congestion control protocol P_i* employed by that sender. A congestion control protocol (deterministically) maps the history of congestion-window sizes of that sender, and of the RTTs and loss rates experienced by that sender, to the sender's next selection of congestion window size. A protocol is *loss-based* if its choice of window-sizes is invariant to the RTT values. Any choice of initial congestion windows $\bar{x}^{(0)} = (x_1^{(0)}, \dots, x_n^{(0)})$ and congestion control protocol P_i for each sender i *deterministically* induces a *dynamic* of congestion control as follows: senders start sending as in $\bar{x}^{(0)}$ and, at each time step, every sender i repeatedly reacts to its environment as prescribed by P_i. Different choices of initial congestion windows enable us to reason about, e.g., connections (with smaller window sizes) starting to send after other connections (with larger window sizes).

Protocols belonging to the families of Additive-Increase-Multiplicative-Decrease (**AIMD(a,b)** [13, 31]) and Multiplicative-Increase-Multiplicative-Decrease (**MIMD(a,b)** [2, 3]) are easy to model in our framework: AIMD(a,b) increases the window size $x_i^{(t)}$ additively by a (MSS) if the loss $L^{(t)}$ at time t is 0, whereas MIMD(a,b) increases the window size multiplicatively by a factor of a. Both protocols multiplicatively decrease the window size by a factor of b if $L^{(t)} > 0$. We formalize two more prominent families of protocols within our model: Binomial [6], and TCP Cubic [15].

A binomial protocol **BIN**(a, b, k, l) for $a > 0,\ 0 < b \leq 1,\ k \geq 0,\ l \in [0, 1]$ is defined as follows:

$$x_i^{(t+1)} = \begin{cases} x_i^{(t)} + \frac{a}{(x_i^{(t)})^k} & \text{if } L^{(t)} = 0 \\ x_i^{(t)} - b(x_i^{(t)})^l & \text{if } L^{(t)} > 0 \end{cases}$$

TCP Cubic, **CUBIC**(c, b), can be captured as follows:

$$x_i^{(t+1)} = \begin{cases} x_i^{max} + c(T - (\frac{x_i^{max}(1-b)}{c})^{\frac{1}{3}})^3 & \text{if } L^{(t)} = 0 \\ x_i^{max} b & \text{if } L^{(t)} > 0 \end{cases}$$

where x_i^{max} is the window size of sender i when the last packet loss was experienced, T is the number of time steps elapsed from the last packet loss, $b \in (0, 1)$ is the rate-decrease factor, and $c > 0$ is a scaling factor.

3 AXIOMATIC APPROACH

We consider eight natural requirements ("axioms") from congestion control protocols. Intuitively, these requirements capture the desiderata of well-utilizing spare network resources, avoiding excessive loss, converging to a stable rate-configuration, achieving fairness amongst competing flows, attaining high-performance in the presence of non-congestion loss [8], being friendly towards legacy TCP connections, and not suffering excessive latency. However, what precisely is the definition of "well-utilizing a link"? Is it utilizing 90% of the link from some point onwards? 95%? What does it mean to be friendly to TCP? Is it not to exceed by over 1.5x the throughput of a TCP connection sharing the same link? 3x?

Different choices of values to plug into our requirements can have significant implications, to the extent of opposite answers to questions regarding whether two (or more) requirements can be satisfied simultaneously. To allow for a nuanced discussion of the interplay between axioms, we *parameterize* the requirements ("metrics"). A congestion control protocol can be regarded as a point in the 8-dimensional space induced by the following metrics, according to its "score" in each metric.

Importantly, the metrics listed below constitute a first attempt at formulating an axiomatic framework for congestion control. As discussed in Section 6, **proposing and investigating other metrics, and refining our metrics, is an important direction for future research**. See [12] for a related discussion.

Metric I: link-utilization. We say that a congestion-control protocol P is *α-efficient* if when all senders employ P, for any initial configuration of senders' window sizes, there is some time step T such that from T onwards $X^{(t)} \geq \alpha C$.

Metric II: fast-utilization. Our next metric is intended to exclude protocols that take an unreasonable amount of time to utilize spare bandwidth (e.g., that increase the window size by a single MSS every 1,000 RTTs). Intuitively, the following definition of *α-fast-utilization* captures the property that, for any "long enough" time period, the protocol consumes link capacity at least as fast as a protocol that increases the congestion window by α MSS in each RTT. A congestion-control protocol P is *α-fast-utilizing* if there exists $T > 0$ such that if a P-sender i's window size is $x_i^{(t_1)}$ at time step t_1 and by time step $t_1 + \Delta_t$, for any $\Delta_t \geq T$, does not experience loss, nor increased RTT (if not loss-based), then $\Sigma_{t=t_1}^{t_1+\Delta_t}(x_i^{(t)} - x_i^{(t_1)}) \geq \frac{\alpha \Delta_t^2}{2}$.

Metric III: loss-avoidance. We say that a congestion-control protocol P is *α-loss-avoiding* if when all senders employ P, for any initial configuration of senders' window sizes, there is some time step T such that from T onwards the loss rate $L^{(t)}$ is bounded by α (e.g., $\alpha = 0.01$ translates to not exceeding loss rate of 1%). We refer to protocols that are 0-loss-avoiding, i.e., protocols that, from some point onwards, do not incur loss, as "0-loss".

Metric IV: fairness. We say that a congestion-control protocol P is *α-fair* if when all senders use P and for any configuration of senders' window sizes, from some time $T > 0$ onwards, the average window size of each sender i is at least an α-fraction that of any other sender j.

Metric V: convergence. We say that a congestion-control protocol P is *α-convergent*, for $\alpha \in [0, 1]$, if there is a configuration of window sizes $(x_1^*, \ldots, x_n^*) \in [0, M]^n$ and time step T such that for any $t > T$ and sender $i \in N$, $\alpha x_i^* \leq x_i^{(t)} \leq (2 - \alpha)x_i^*$ (e.g., $\alpha = 0.9$ means that from some point onwards the window sizes are within 10% from a fixed point).

Metric VI: robustness to non-congestion loss. A natural demand from congestion control protocols is to be *robust* to loss that does not result from congestion [8, 11]. Clearly, formulating this requirement in all possible contexts is challenging. We focus on a simple, yet enlightening, scenario (used in [11] to motivate PCC): Suppose that a single sender i sends on a link of infinite capacity (so as to remove from consideration congestion-based loss). We say that a protocol P is α-robust if when the sender experiences constant *random* packet loss rate of at most $\alpha \in [0, 1]$, then, for any choice of initial senders' window sizes and value $\beta > 0$, there is some $T > 0$ such that for every $t > T$, $x_i^{(t)} \geq \beta$ (i.e., non-congestion loss of rate at most α does not prevent utilization of spare capacity).

Metric VII: TCP-friendliness. We say that a protocol P is *α-friendly* to another protocol Q if, for any combination of sender-protocols such that some senders use P and others use Q, for every initial configuration of senders' window sizes, and for every P-sender i and Q-sender j, from some point in time $T > 0$ onwards j's average window size is at least an α-fraction of i's average window size. Observe that friendliness, as defined above, is closely related to fairness (Metric IV), but fairness is with respect to many instantiations of the *same* protocol, whereas friendliness captures interactions between *different* protocols. We say that a protocol P is *α-TCP-friendly* if P is α-friendly towards AIMD(1,0.5) (i.e., TCP Reno).

Metric VIII: latency-avoidance. We say that protocol P is α-latency-avoiding if for sufficiently large link capacity C and buffer size τ, and regardless of sender's initial window sizes, when all senders on the link employ P, there is some time step T such that from T onwards $RTT(t) < (1 + \alpha)2\Theta$. The term 2Θ captures the minimum possible RTT (twice the link's propagation delay)

4 AXIOMATIC DERIVATIONS

We present below theoretical results highlighting the intricate connections between our metrics. Our results establish that some properties of a congestion control scheme are immediate consequences of other properties or, conversely, cannot be satisfied in parallel to satisfying other properties. The latter form of results exposes fundamental tensions between metrics, implying that attaining a higher "score" in one metric inevitably comes at the expense lowering the score in another. We leverage this insight in Section 5 to reason about the Pareto frontier for protocol design in the 8-dimensional metric space induced by our metrics.

We begin with the following simple observation:

CLAIM 1. *Any loss-based protocol that is* 0*-loss is not* α*-fast-utilizing for any* $\alpha > 0$*.*

To see why the above is not obvious, consider a protocol P that slowly increases its rate until encountering loss for the first time and then slightly decreases the rate so as to not exceed the link's capacity. While both 0-loss (from some point in time no loss occurs) and almost fully-utilizing the link, this protocol is not α-fast-utilizing for any $\alpha > 0$. The reason is that, for loss-based protocols, α-fast-utilization implies after sufficiently long time without packet loss, the protocol must increase its rate until encountering loss yet again.

We present below other, more subtle, connections between our metrics. All of our bounds apply across all possible network parameters (link capacity, buffer size) and number of senders, with the exception of Theorem 3 (where the reliance on C and τ is explicit). We start with the following result, which relates convergence, fast-utilization, and efficiency.

THEOREM 1. *Any protocol that is* α*-convergent and* β*-fast-utilizing, for some* $\beta > 0$*, is at least* $\frac{\alpha}{2-\alpha}$*-efficient.*

We next turn our attention to TCP-friendliness. Intuitively, attaining good performance and being TCP-friendly are at odds. Our next two results formalize this intuition by upper bounding the level of TCP friendliness attainable by a protocol according to the extent to which that protocol is fast-utilizing, efficient, and robust to non-congestion loss.

THEOREM 2. *Any loss-based protocol that is* α*-fast-utilizing and* β*-efficient is at most* $\frac{3(1-\beta)}{\alpha(1+\beta)}$*-TCP-friendly*

The above upper bound on TCP-friendliness is tight, in the sense that $AIMD(\alpha, \beta)$ satisfies the requirements of Theorem 2 and is $\frac{3(1-\beta)}{\alpha(1+\beta)}$-TCP-friendly [7].

THEOREM 3. *Any loss-based protocol that is* α*-fast-utilizing,* β*-efficient, and* ϵ*-robust, for* $\epsilon > 0$*, is at most* $\frac{3(1-\beta)}{(4(\frac{C+\tau}{1-\epsilon})-\alpha)(1+\beta)}$*-TCP friendly.* [1]

On establishing TCP-friendliness. Often, to establish that a congestion control protocol is TCP-friendly, simulation or experimental results are presented to demonstrate that it does not harm TCP by "too much". What, though, guarantees that friendliness towards a certain TCP variant implies friendliness towards other TCP variants? Our next result suggests a principled approach to establishing TCP-friendliness: prove that the protocol is friendly to a specific TCP variant (say, TCP Reno) and deduce that it is at least as friendly to any "more aggressive" TCP variant.

We introduce the following terminology: a protocol P is *more aggressive* than a protocol Q if for any combination of P- and Q-senders, and initial sending rates, from some point in time onwards, the average goodput of *any* P-sender is higher than that of *any* Q-sender.

THEOREM 4. *Let* P *and* Q *be two protocols such that (1) each protocol is either AIMD, BIN, or MIMD, (2)* P *is* α*-TCP-friendly, and (3)* Q *is more aggressive than Reno. Then,* P *is* α*-friendly to* Q*.*

We leave the extension of Theorem 4 to other families of congestion control protocols to future research.

Loss-based vs. latency-avoiding protocols. Mo et al. show that the loss-based TCP Reno is very aggressive towards the latency-avoiding TCP Vegas [20]. Intuitively, this is a consequence of Vegas backing off upon exceeding some latency bound while Reno continues to increase its rate since it is oblivious to latency. The following result shows that any "reasonable" loss-based protocol is extremely unfriendly towards *any* latency-avoiding protocol.

THEOREM 5. *A loss-based protocol that is* α*-efficient, for any* $\alpha > 0$*, is not* β*-friendly, for any* $\beta > 0$*, with respect to any protocol that is* γ*-latency avoiding, for any* $\gamma > 0$*.*

5 PROTOCOL ANALYSIS AND DESIGN

Our theoretical framework, as presented in Section 2, allows us to associate each congestion control protocol with a 8-tuple of real numbers, representing its scores in the eight metrics. We present below our results in this direction and explain the implications for protocol design.

5.1 Mapping Protocols to Points

Table 1 presents our theoretical results mapping protocols to points in our 8-dimensional metric space for the families of protocols described in Section 2 (i.e., AIMD, MIMD, BIN, CUBIC), and also for ROBUST-AIMD (to be discussed in Section 5.2).

We present in angle brackets worst-case bounds across all choices of network parameters (e.g., very shallow buffer, very high number of senders, etc.). To gain insights into how the specific link parameters (capacity C, buffer size τ) and number of senders n affect efficiency, stability, etc., we complement some of the worst-case bounds with more nuanced results reflecting the dependence on these parameters.

The results regarding the fairness and TCP-friendliness of BIN are derived from [6] and the fairness of CUBIC is derived from [15]. Observe that MIMD is ∞-fast-utilizing as its rate increases superlinearly. As all protocols considered are

[1] We assume $(C + \tau) > \frac{\alpha}{2}$.

Protocol	Efficiency	Loss-Avoiding	Fast-Utilizing	TCP-Friendly	Fair	Conv
AIMD(a,b)	$\min(1, b(1+\frac{\tau}{C}))$ $<b>$	$1-\frac{C+\tau}{C+\tau+na}$ $<1>$	$<a>$	$<\frac{3(1-b)}{a(1+b)}>$	$<1>$	$<\frac{2b}{1+b}>$
MIMD(a,b)	$\min(1, b(1+\frac{\tau}{C}))$ $<b>$	$<\frac{a}{1+a}>$	$<\infty>$	$\frac{2log_a\frac{1}{b}}{C+\tau-2log_a\frac{1}{b}}$ $<0>$	$<0>$	$<\frac{2b}{1+b}>$
BIN(a,b,l,k)	$\min(1, (1-b)(1+\frac{\tau}{C}))$ $<(1 - b)>$	$1-\frac{C+\tau}{C+\tau+a(\frac{C+\tau}{n})^k}$ $<1>$	$<a>$ if k=0 $<0>$ if k>0	$<\sqrt{\frac{3}{2}}(\frac{b}{a})^{\frac{1}{1+l+k}}>$ if $l+k\geq 1$ $<0>$ otherwise	$<1>$	$<\frac{2-2b}{2-b}>$
CUBIC(c,b)	$\min(1, b(1+\frac{\tau}{C}))$ $<b>$	$1-\frac{C+\tau}{C+\tau+nc}$ $<1>$	$<c>$	$\sqrt{\frac{3}{2}}\sqrt[4]{\frac{4(1-b)}{c(3+b)(C+\tau)}}$ $<0>$	$<1>$	$<\frac{2b}{1+b}>$
Robust-AIMD(a,b,k)	$\min(1, \frac{b(1+\frac{\tau}{C})}{1-k})$ $<\frac{b}{1-k}>$	$\frac{(C+\tau)k+na(1-k)}{(C+\tau)+na(1-k)}$ $<1>$	$<a>$	$\frac{3(1-b)}{(4(\frac{C+\tau}{1-k})-a)(1+b)}$ $<0>$	$<1>$	$<\frac{2b}{1+b}>$

Table 1: Protocol Characterization. Worst-case bounds shown within angle-brackets.

loss-based, their scores for latency avoidance are unbounded and so omitted from the table. Also omitted are our results for robustness: all protocols are 0-robust, with the exception of $Robust-AIMD(a,b,k)$, which is k-robust.

We validated the theoretical results in Table 1 via experiments with the Emulab [28]. The purpose of our experiments was *not* to show that the achieved throughputs, loss rates, etc., *exactly* match the results in Table 1, but only to verify the *trends* arising from our theoretical analysis. We experimented with protocols implemented in the Linux kernel, namely, TCP Reno (AIMD(1,0.5)), TCP Cubic (CUBIC(0.4,0.8)), and TCP Scalable (MIMD(1.01,0.875) in some environments and AIMD(1,0.875) in others). Our experiments investigated the interaction of a varying number of connections (2-4) on a single link, for varying bandwidths (20Mbps, 30Mbps, 60Mbps, and 100Mbps) and buffer sizes (10 MSS / 100 MSS), and a fixed RTT of 42ms. Our preliminary findings establish, for each metric, the same hierarchy over protocols (from "worst" to "best") as induced by the theoretical results. We defer the exposition of these experimental results and the examination of other experimental settings to the full version of this paper.

5.2 Seeking Points on the Pareto Frontier

The Pareto frontier for protocol design. Not every point in the 8-dimensional space induced by our metrics is *feasible*, in the sense that there are some points such that no protocol can attain their associated scores. Indeed, by Theorem 2, no protocol can *simultaneously* achieve near-perfect scores for fast-utilization, efficiency, *and* TCP-friendliness. We refer to the region of *feasible points* in our 8-dimensional space as the *feasibility region for protocol design*. Our focus is on the *Pareto frontier* of this feasibility region. A feasible point is on the Pareto frontier if no other feasible point is strictly better in terms of one of our metrics without being strictly worse in terms of another metric. Any point on the Pareto frontier thus captures scores that are both attainable by a congestion control protocol and cannot be strictly improved upon. Different points on this frontier capture different tradeoffs between our metrics.

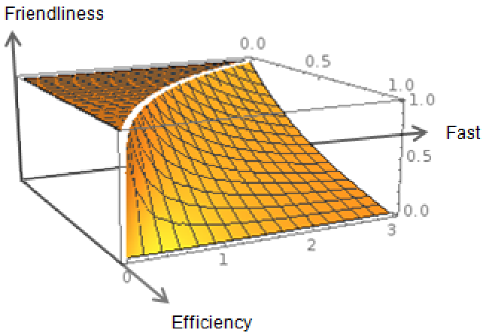

Figure 1: Pareto frontier of Efficiency, Friendliness, and Fast-utilization

Robust-AIMD as an illustration. To illustrate the above, let us restrict our attention to the efficiency, fast-utilization, and TCP-friendliness metrics and consider the tension between these metrics, as captured by Theorem 2. Figure 1 describes the Pareto frontier in the 3-dimensional subspace spanned by these three metrics. Points on this Pareto frontier are of the form $(\alpha, \beta, \frac{3(1-\beta)}{\alpha(1+\beta)})$ (corresponding to fast-utilization, efficiency, and TCP-friendliness scores, respectively). Observe that each of these points is indeed feasible as AIMD(α,β) attains these scores (see Table 1). We now add the requirement of robustness to non-congestion loss to the mix. We introduce a robust variant of AIMD, called Robust-AIMD, designed to achieve *robustly* good performance without being overly aggressive to TCP. Robust-AIMD (by the results in Table 1 and Theorem 3) cannot be improved upon, in terms of one of the four metrics under consideration, without lowering its score in another (and thus lies on the Pareto frontier of their induced 4-dimensional space).

Robust-AIMD can be regarded as a hybrid of traditional AIMD and PCC [11]. Under Robust-AIMD, time is divided

(n,BW)	(2,20)	(2,30)	(2,60)	(2,100)	(3,20)	(3,30)	(3,60)	(3,100)	(4,20)	(4,30)	(4,60)	(4,100)
R-AIMD	2.48x	1.69x	1.19x	1.94x	1.76x	1.35x	2.75x	1.92x	1.30x	2.47x	2.24x	2.00x

Table 2: TCP-friendliness of Robust-AIMD(1,0.8,0.01) vs. PCC

into short (roughly 1 RTT) "monitor intervals". In each monitor interval, the sender sends at a certain rate and uses selective ACKs from the receiver to learn the resulting loss rate. Robust-AIMD uses an AIMD-like rule for adjusting transmission rate: the sender has a congestion window (similarly to TCP and unlike PCC), that is additively increased by a predetermined constant a (MSS) if the experienced loss rate is lower than a fixed constant $\epsilon > 0$, and multiplicatively decreased by a predetermined constant b if the loss rate exceeds ϵ. Thus, Robust-AIMD is modeled as follows. Robust-AIMD(a,b,ϵ):

$$x_i^{(t+1)} = \begin{cases} x_i^{(t)} + a & \text{if } L^{(t)} < \epsilon \\ x_i^{(t)} b & \text{if } L^{(t)} \geq \epsilon \end{cases}$$

Our theoretical results for Robust-AIMD suggest that it constitutes a point in the design space of congestion control protocols that provides performance comparable to AIMD in a significantly more robust manner, while being more friendly to legacy TCP connections than PCC (whose behavior is strictly more aggressive than MIMD(1.01,0.99)). In fact, the proof of our theoretical result regarding the TCP-friendliness of Robust-AIMD shows that its TCP-friendliness is *monotone* in the number of Robust-AIMD connections, in the sense that the more Robust-AIMD connections share a link the better its friendliness to TCP connections on the link becomes.

Evaluating Robust-AIMD under real-world network conditions, and identifying the "right" parameter settings for such protocols, lies well beyond the scope of this study. Yet, to validate the above theoretical findings regarding Robust-AIMD, we experimentally evaluated Robust-AIMD(1,0.8, ϵ), for ϵ values of 0.005, 0.007, and 0.01 (corresponding to loss rates of 0.5%, 0.7%, and 1%, respectively). As expected, Robust-AIMD(1,0.8)'s outperformed the evaluated AIMD and MIMD protocols (specifically, Reno, Cubic, Scalable) in terms of robustness and efficiency, and was outperformed by PCC. Our experimental results comparing Robust-AIMD's TCP friendliness to PCC appear in Table 2. Each entry in the table specifies the improvement in % of Robust-AIMD(1,0.8) over PCC for different choices of number of senders on the link (n) and link bandwidth, constant RTT of 42ms and buffer size of 100 MSS. Observe that Robust-AIMD consistently attains >1.5x TCP-friendliness than PCC (1.92x improvement on average). (Our results for Robust-AIMD's scores in other metrics, omitted, also qualitatively comply with our theoretical results.)

We view Robust-AIMD as an example of how the axiomatic approach can be leveraged to identify points in the design space that lie on the Pareto frontier, and guide the theoretical investigation and experimental exploration of such points.

6 CONCLUSION

We have put forth an additional approach for analyzing congestion control designs, which complements the current simulation/experimentation and control-theoretic/network-utility-maximization theoretical approaches. Our axiomatic approach yields insight, in a simple model, into which properties can co-exist, and which follow from other properties. However, we view our treatment here as merely the first step in this direction, with many promising directions for future research:

Other congestion control protocols. One important future task is applying our axiomatic approach to other congestion control schemes (including the recently proposed PCC [11], Sprout [30] and BBR [8]). This would require us to extend our model to deal with pacing-based congestion control (which is needed for PCC and BBR).

More realistic network model. Generalizing our model to capture network-wide protocol interaction (see [14]), stochastic effects of queuing at routers, and unsynchronized network feedback [24], is also a natural research direction.

Other axioms, other derivations. What other metrics of performance, fairness, etc., should be incorporated into our axiomatic approach (see [12] for a discussion of evaluation metrics)? What other results regarding the relations between metrics can be proven? We believe that further investigation along these lines could shed light on the inherent tradeoffs involved in protocol design.

Better experimental evaluations. As discussed in Section 5, to provide preliminary validation of theoretical results, we experimented with different protocols. Conducting larger-scale, more realistic experiments to validate our findings, and also to guide the search for new theoretical results, is important for establishing the usefulness of our approach.

Applying the axiomatic approach to other networking contexts. Thus far, axiomatic approaches have been applied to a few, very specific, networking environments [10, 18, 23]. We believe that applying the axiomatic approach to other contexts (e.g., intradomain [18] and interdomain routing, traffic engineering, in-network queueing [25], network security) is another important direction for future research, and that results along these lines could contribute to more principled discussions about these contexts.

ACKNOWLEDGEMENTS

We thank Akshay Narayan and the rest of the MIT Networks and Mobile Systems (NMS) group for valuable feedback. We also thank the anonymous HotNets reviewers for their helpful comments. Michael Schapira thanks Moshe Tennenholtz for many insightful conversations about the axiomatic approach.

REFERENCES

[1] M. Alizadeh, A. Kabbani, B. Atikoglu, and B. Prabhakar. Stability analysis of QCN: the averaging principle. In *SIGMETRICS*, 2011.

[2] E. Altman, K. Avrachenkov, and B. Prabhu. Fairness in MIMD congestion control algorithms. In *INFOCOM 2005*, 2005.

[3] E. Altman, R. El-Azouzi, Y. Hayel, and H. Tembine. The evolution of transport protocols: An evolutionary game perspective. *Computer Networks*, 2009.

[4] R. Andersen, C. Borgs, J. Chayes, U. Feige, A. Flaxman, A. Kalai, V. Mirrokni, and M. Tennenholtz. Trust-based recommendation systems: an axiomatic approach. In *World Wide Web*, 2008.

[5] K. J. Arrow. A difficulty in the concept of social welfare. *Political Economy*.

[6] D. Bansal and H. Balakrishnan. Binomial congestion control algorithms. In *INFOCOM*, 2001.

[7] L. Cai, X. Shen, J. Pan, and J. W. Mark. Performance analysis of TCP-friendly AIMD algorithms for multimedia applications. *Transactions on Multimedia*, 2005.

[8] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson. BBR: Congestion-based congestion control. *Queue*, 2016.

[9] D.-M. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN systems*, 1989.

[10] S. Cohen and A. Zohar. An axiomatic approach to link prediction. In *AAAI*. Citeseer, 2015.

[11] M. Dong, Q. Li, D. Zarchy, P. B. Godfrey, and M. Schapira. PCC: Re-architecting congestion control for consistent high performance. In *NSDI 15*, 2015.

[12] S. Floyd. Metrics for the Evaluation of Congestion Control Mechanisms. RFC 5166, Mar. 2008.

[13] S. Floyd, M. Handley, and J. Padhye. A comparison of equation-based and AIMD congestion control. 2000.

[14] P. Godfrey, M. Schapira, A. Zohar, and S. Shenker. Incentive compatibility and dynamics of congestion control. In *SIGMETRICS*, 2010.

[15] S. Ha, I. Rhee, and L. Xu. CUBIC: a new TCP-friendly high-speed TCP variant. *ACM SIGOPS Operating Systems Review*, 2008.

[16] F. P. Kelly, L. Massouli, and N. S. Walton. Resource pooling in congested networks: proportional fairness and product form. *Queueing Systems*, 2009.

[17] T. Lakshman and U. Madhow. The performance of TCP/IP for networks with high bandwidth-delay products and random loss. *Transactions on Networking (ToN)*, 1997.

[18] O. Lev, M. Tennenholtz, and A. Zohar. An axiomatic approach to routing. In *INFOCOM 2015*, 2015.

[19] R. Mittal, N. Dukkipati, E. Blem, H. Wassel, M. Ghobadi, A. Vahdat, Y. Wang, D. Wetherall, D. Zats, et al. TIMELY: RTT-based congestion control for the datacenter. In *SIGCOMM*, 2015.

[20] J. Mo, R. J. La, V. Anantharam, and J. Walrand. Analysis and comparison of TCP reno and vegas. In *INFOCOM'99*, 1999.

[21] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: A simple model and its empirical validation. *SIGCOMM*, 1998.

[22] F. Paganini, J. Doyle, and S. H. Low. A control theoretical look at internet congestion control. *Lecture notes in control and information sciences*, 2003.

[23] S. Shenker. A theoretical analysis of feedback flow control. In *SIGCOMM*, 1990.

[24] R. Shorten, F. Wirth, and D. Leith. A positive systems model of TCP-like congestion control: asymptotic results. *Transactions on Networking (TON)*, 2006.

[25] A. Sivaraman, K. Winstein, S. Subramanian, and H. Balakrishnan. No Silver Bullet: Extending SDN to the Data Plane. In *Twelfth ACM Workshop on Hot Topics in Networks (HotNets-XII)*, College Park, MD, November 2013.

[26] R. Srikant. *The mathematics of Internet congestion control*. 2012.

[27] M. Tennenholtz. Reputation systems: An axiomatic approach. In *Uncertainty in artificial intelligence*, 2004.

[28] B. White, J. Lepreau, L. Stoller, R. Ricci, G. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An integrated experimental environment for distributed systems and networks. 2002.

[29] K. Winstein and H. Balakrishnan. TCP ex machina: Computer-generated congestion control. In *SIGCOMM*, 2013.

[30] K. Winstein, A. Sivaraman, H. Balakrishnan, et al. Stochastic forecasts achieve high throughput and low delay over cellular networks. In *NSDI*, 2013.

[31] Y. R. Yang and S. S. Lam. General AIMD congestion control. In *Network Protocols, 2000*, 2000.

[32] L. Zhang, S. Shenker, and D. D. Clark. Observations on the dynamics of a congestion control algorithm: The effects of two-way traffic. *SIGCOMM*, 1991.