Title	Does it hurt when others prosper?: Exploring the impact of heterogeneous reordering robustness of TCP
Author(s)	Lai, C; Leung, KC; Li, VOK
Citation	The 32nd IEEE Conference on Computer Communications (IEEE INFOCOM 2013), Turin, Italy, 14-19 April 2013. In IEEE Infocom Proceedings, 2013, p. 2958-2966
Issued Date	2013
URL	http://hdl.handle.net/10722/184993
Rights	IEEE Infocom Proceedings. Copyright © IEEE Computer Society.

Does It Hurt When Others Prosper?
Exploring the Impact of Heterogeneous Reordering Robustness of TCP

Chengdi Lai, Ka-Cheong Leung, and Victor O.K. Li
Department of Electrical and Electronic Engineering
The University of Hong Kong
Pokfulam Road, Hong Kong, China
E-mail: {laichengdi, kcleung, vli}@eee.hku.hk

***Abstract*— The congestion control mechanisms in the standardized Transmission Control Protocol (TCP) may misinterpret packet reordering as congestive loss, leading to spurious congestion response and under-utilization of network capacity. Therefore, many TCP enhancements have been proposed to better differentiate between packet reordering and congestive loss, in order to enhance the reordering robustness (RR) of TCP. Since such enhancements are incrementally deployed, it is important to study the interactions of TCP flows with heterogeneous RR. This paper presents the first systematic study of such interactions by exploring how changing RR of TCP flows influences the bandwidth sharing among these flows. We define the quantified RR (QRR) of a TCP flow as the probability that packet reordering causes congestion response. We analyze the variation of bandwidth sharing as QRR changes. This leads to the discovery of several interesting properties. Most notably, we discover the counter-intuitive result that changing one flow's QRR does not affect its competing flows in certain network topologies. We further characterize the deviation, from the ideal case of bandwidth sharing, as RR changes. We find that enhancing RR of a flow may increase, rather than decrease, the deviation in some typical network scenarios.**

I. Introduction

In the Internet, congestion control mechanisms in Transmission Control Protocol (TCP) become increasingly heterogeneous. The number of coexisting TCP variants increase and the differences among different variants are accentuated; see measurement results reported in [13], [17], [24]. The major driving force behind such heterogeneity is various emerging network environments and applications. These motivate new TCP enhancements, whose incremental deployment brings heterogeneity to TCP. In view of such heterogeneity, a crucial concern is whether flows of the existing TCP variants will experience performance degradation ("*does it hurt*") with the gradual deployment of new TCP enhancements ("*when others prosper*").

One major class of TCP enhancements is motivated by the poor performance of the standardized TCP congestion response algorithms [1] in the presence of packet reordering. Packet reordering refers to the disruption of the normal order of a packet flow, and can be caused by parallelism at multiple levels within the networks. For example, to match the processing speed with the increase in the link capacity, a modern router often employs multiple processing units to process packets arriving at a single incoming line card. Moreover, for interconnecting routers, a high-speed link is sometimes realized via multiple parallel links. Packets belonging to the same flow may take different physical paths within and across routers, regardless of whether they traverse the same logical route. This gives rise to the possibility that a packet transmitted later by an end system may arrive at the destination earlier than the preceding packets.

Other forms of parallelism that can cause packet reordering include, but are not limited to, the following: 1) a flow director in a multi-processor network interface card (NIC) assigns packets belonging to the same flow to different cores due to dynamic load balancing among the cores [23]; 2) quality-of-service (QoS) scheduling assigns packets belonging to the same flow to different priority queues; and 3) packet level multi-path routing.

With the prevalence of parallelism in various network components, we envision modern networks to be *reordering networks*, i.e., networks with packet reordering. According to some recent measurements on TCP traffic, packet reordering is path-dependent and, on average, around 1% of data packets experience reordering in a campus network and several backbone networks [14].

The standardized TCP relies on fast retransmit and fast recovery for prompt congestion response, and assumes in-order packet delivery. A receiver alarms the sender via a duplicate acknowledgement (*dupack*) if a received packet is not the next packet in sequence. When the number of consecutively issued *dupack* exceeds a certain fixed threshold value, known as *dupthresh*, the sender assumes the in-sequence packet to be lost due to congestion and backs off. Thus, if the number of packets by which a packet is reordered (known as *reordering length*) exceeds *dupthresh*, packet reordering will result in spurious backoff of TCP and under-utilization of network resources. In general, the reordering length increases as packet rate increases, making spurious backoff due to packet reordering particularly significant in high speed networks. At the data rate of 100 Mbps, up to 15 packet ordering events have been observed to have reordering lengths greater than *dupthresh* in one-minute measurements of backbone traffic [7].

Numerous *reordering TCP enhancements* have been proposed to attain more accurate differentiation between packet reordering and congestive loss, thereby enhancing the *reordering robustness* (RR) of TCP; please refer to [10] for a survey. While the standardized TCP has not incorporated these enhancements, the current releases of Linux have adopted a reordering TCP enhancement that can dynamically adjust *dupthresh* based on the recorded statistics of packet reordering [18]. The Internet is thus undergoing the incremental deployment of reordering TCP enhancements, resulting in the heterogeneity in RR of TCP flows.

A. Our Contributions

In the context of the heterogeneous RR of TCP flows, our thesis statement "*Does it hurt when others proper?*" poses the following basic question (BQ):

BQ: *How does enhancing RR of some TCP flows influence bandwidth sharing among all TCP flows?*

978-1-4673-5946-7/13/$31.00 ©2013 IEEE

In this paper, we present the first systematic study on the impact of heterogeneous RR of TCP flows by searching for an answer to our BQ. In the literature, the study is preliminary, largely constrained to simulating or emulating competition among flows of the standardized TCP and reordering TCP enhancements *over networks that guarantee orderly packet delivery*; see, for example, [2], [3], [25]. Over more general network settings that can introduce reordering, the interaction of TCP flows with heterogeneous RR is much more sophisticated. Our motivating examples will demonstrate, in some cases, that it is inevitable for a TCP flow to be adversely affected (negative impact) when RR of its competing flows is enhanced, whereas in other cases, it can actually benefit from such change (positive impact).

We characterize the variation of bandwidth sharing as RR changes in terms of the directional derivatives of flow rates with respect to the quantified RR (QRR). We develop two structural properties, known as symmetricity and offset. The symmetricity property states that changes in one flow's reordering robustness will affect the transmission of another flow in the same way (positive/negative/no impact) as it will be affected by the other. The counter-intuitive offset property suggests that, in a network where the number of routes taken by TCP flows over bottleneck links equals the number of bottleneck links, enhancing one flow's RR may not affect its competing flow as long as the two flows do not traverse exactly the same set of bottleneck links. We have validated these properties via simulations.

We further characterize the deviation from the optimal bandwidth sharing as RR changes. We find that enhancing RR of a flow may increase, rather than decrease, the deviation in some typical network scenarios.

The methodology employed in our analysis can potentially be extended to evaluate the impact of incrementally deploying other types of TCP enhancements.

The rest of the paper is organized as follows. Section II introduces two motivating examples. Our major results are presented in Section IV, following our system model in Section III. Section V describes our experimental validation. Section VI presents the related work. Section VII concludes the paper with some insights from our model and analysis, and discusses some possible extensions to our work.

II. Motivating Examples

We present two motivating examples on the interaction of TCP flows with heterogeneous RR over reordering networks.

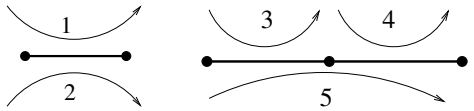
In the presence of packet reordering, if a TCP flow with poor RR fails to fully utilize the bandwidth, it is reasonable for flows of reordering TCP enhancements to attain higher throughput than this flow by grabbing the unused bandwidth. At the same time, it is highly desirable that the enhanced flows will *not* cause performance deterioration of their competing TCP flows with poor RR. However, in some typical scenarios, such adverse effect seems inevitable. This is illustrated in the following example.

Example 1: In Fig. 1(a), two TCP flows with poor RR, Flows 1 and 2, traverse a link that reorders packets. At the link buffer, a buffer-based active queue management algorithm (AQM) determines the packet dropping rate from the buffer. We impose a very mild assumption that the packet dropping rate is zero if the buffer is mostly empty, and non-zero if the buffer is persistently occupied. The aggregate transmission of the two flows fails to fully utilize the link capacity, leaving the link buffer empty most of the time. Thus, packets are rarely dropped.

Now, suppose that Flow 1 becomes more robust against packet reordering by being upgraded with reordering TCP enhancement. Without frequent spurious backoffs due to packet reordering, Flow 1 fills the previously empty link buffer, leading to a nonzero packet dropping rate by the AQM. Consequently, Flow 2, which backs off upon both congestive loss and packet reordering, yields backoff more frequently. Flow 2 is thus adversely affected by the enhanced RR of Flow 1. Moreover, we see that such adverse effect is inevitable. This is because as long as the TCP enhancements are more robust against reordering, they tend to fill at least part of the link buffer.

On the other hand, it is possible for a TCP flow to be *positively* impacted by such reordering TCP enhancements. This is illustrated in the following example.

Example 2: In Fig. 1(b), two TCP flows with poor RR, Flows 3 and 4, are competing with a flow of a reordering TCP enhancement, Flow 5, over a two-link network. The left link reorders packets and the right link does not. Now, suppose that Flow 3 becomes more robust against reordering. Its more aggressive probing for the available bandwidth will increase the active dropping rate over the left link by the AQM. This causes Flow 5 to back off more frequently. This in turn yields more available bandwidth to Flow 4 over the right link. In this case, Flow 4 benefits from the improved RR of Flow 3.



(a) Example 1. (b) Example 2.

Fig. 1. Two motivating examples.

Therefore, in the presence of packet reordering, the interaction of reordering TCP enhancements with TCP with poor RR is complicated. However, a fundamental understanding on how TCP flows with heterogeneous RR interact is lacking. We now present a systematic approach to such understanding.

III. Reordering Network Model

Notation conventions: We use bold fonts to denote matrices and vectors, as in "$\mathbf{x}$". $\mathbf{A}_{*i}$ and $\mathbf{A}_{j*}$ refer to the ith column and the jth row of Matrix $\mathbf{A}$, respectively. $\mathbf{A}_{ji}$ refers to the entry on the jth row, ith column. $\mathbf{e}_i$ refers to the unit vector in the standard basis that has one in the ith entry and zeroes in all other entries. A summary of notations is presented in Table I.

We begin by considering a generic communication network with a set of TCP flows F and a set of links L. We study the equilibrium of the network, namely, the status of the network where the transmissions of end systems match with the feedback signal provided by the network.

To avoid unnecessary complexity, we consider uni-path routing in our model. Suppose that a flow $f \in F$ traverses a route, which consists of a set of links $L_f \subset L$. Accordingly, the routing matrix $\mathbf{R}_{|L|\times|F|}$ is defined as:[1]

$$\mathbf{R}_{lf} \triangleq \begin{cases} 1 & l \in L_f \\ 0 & l \notin L_f \end{cases} \quad (1)$$

[1]In the case of multi-path routing, our results continue to hold by redefining $\mathbf{R}_{lf}$ as the fraction of data traffic belonging to Flow f which traverses Link l.

TABLE I

NOTATIONS

α_f, κ_f, σ_f	protocol-specific constant for Flow f; see (3)
c_l	transmission capacity of Link l
d_f	round-trip time (RTT) of Flow f
F	set of flows
L	set of links
$L(\boldsymbol{\theta})$	set of bottleneck links associated with a given QRR $\boldsymbol{\theta}$
L_f	set of links traversed by Flow f
$L_{\mathbf{z}}$	$L(\boldsymbol{\eta}(T))$ for $T \in (0, \overline{T})$; see Theorem 3
p_l^c	equilibrium probability that a packet is dropped by a queue management algorithm at Link l
$\mathbf{p^c}$	$(p_l^c : l \in L)$
$\mathbf{p}_{\mathbf{z}}^{\mathbf{c}}$	$(p_l^c : l \in L_{\mathbf{z}})$
$\mathbf{p^c}(\boldsymbol{\theta})$	value of $\mathbf{p^c}$ associated with a given QRR $\boldsymbol{\theta}$
q_f	equilibrium probability that a packet belonging to Flow f is inferred lost
q_f^c	equilibrium probability that a packet belonging to Flow f is dropped by a queue management algorithm
q_f^w	probability that a packet belonging to Flow f is reordered
$\mathbf{R}$	routing matrix for all links
$\mathbf{R}(\boldsymbol{\theta})$	routing matrix for links in $L(\boldsymbol{\theta})$
$\mathbf{R}_{\mathbf{z}}$	routing matrix for links in $L_{\mathbf{z}}$; see Theorem 3
ρ_f	probability that a packet belonging to Flow f is reordered and inferred lost falsely
θ_f	QRR of Flow f
$\boldsymbol{\theta}$	$(\theta_f : f \in F)$
x_f	equilibrium data rate of Flow f
$\mathbf{x}$	$(x_f : f \in F)$
$\mathbf{x}(\boldsymbol{\theta})$	value of $\mathbf{x}$ associated with a given QRR $\boldsymbol{\theta}$
y_l	equilibrium incoming data rate at Link l

Over Link $l \in L$, y_l is determined by:

$$y_l = \sum_{r \in R} \mathbf{R}_{lf} x_f \tag{2}$$

We further assume that:

A1. $c_l \geq y_l$. Otherwise, data persistently arrive faster than being delivered, leading to overflow of a router buffer.

We refer to a link $l \in L$ as a *bottleneck link* if $c_l = y_l$ and a *non-bottleneck link* otherwise. Now, a queue management algorithm determines p_l^c based on the backlog and/or the incoming data rate, and we assume:

A2. $p_l^c = 0$ *over non-bottleneck links.* In these links, the backlog is almost zero most of the time, and thus a queue management algorithm rarely drops packets.

Most queue management algorithms observe Assumption A2, except that a convenient model for drop-tail is lacking at the moment [11].

Each flow f determines x_f via congestion control. Due to the versatile approaches in congestion control, it is very difficult to model all possible approaches in a tractable and concise manner. We thus make the following simplifying assumption on modelling congestion control.

A3. The congestion avoidance (CA) algorithm, namely, the part of a congestion control mechanism that determines the evolution of the size of the congestion window (cwnd), is loss-based. cwnd is incremented if all data packets are successfully delivered and acknowledged, and is reduced to a fraction of its current value upon inferring a packet loss.[2] In the Internet, loss-based CA is dominant.[3]

[2]The other popular approach for CA is *delay-based*, which infers the network load from the estimated queueing delay at intermediate routers and adjusts *cwnd* accordingly.

TABLE II

VALUES OF α_f, σ_f, AND κ_f ASSUMED BY POPULAR TCP VARIANTS

	α_f	σ_f	κ_f
TCP AIMD	2	2	2
CUBIC TCP	1.23	1.33	0.33
Compound TCP	20.41	1.25	1.25

Following Assumption A3, for Flow f, we hypothesize that x_f, d_f, and q_f are related via the following empirical formula:

$$\alpha_f = x_f^{\sigma_f} d_f^{\kappa_f} q_f \tag{3}$$

where α_f, κ_f, and σ_f are positive constants. We further assume $\sigma_f \geq 1$. Otherwise, x_f drops faster than q_f increases, suggesting poor robustness against heavy losses. It has been shown that the three most popular TCP variants, namely, TCP AIMD, CUBIC TCP, and Compound TCP, satisfy (3) in [11], [8], and [20], respectively. A summary of the values of α_f, σ_f, and κ_f assumed by these variants under recommended parameter settings is presented in Table II.

In reordering networks, when a packet is *inferred* lost by TCP, it can actually be dropped due to congestion, or not lost but reordered. Analytically, this is:

$$q_f = q_f^c + (1 - q_f^c)\rho_f \approx q_f^c + \rho_f \tag{4}$$

Now, it is time to formally define a reordered packet as below.

Definition 1: A packet arriving at the receiver is reordered if it arrives later than its subsequent packets.

In practice, q_f^w can be estimated as the percentage of reordered packets among all packets of Flow f. This is valid even when the intensity of packet reordering fluctuates throughout a TCP session.

The reordering length and RR of a TCP flow jointly determine whether a reordered packet can cause a spurious backoff in TCP. Given the reordering length distribution, the probability that a reordered packet causes a spurious backoff resembles RR. Thus, we quantify RR as below.

Definition 2: The quantified RR (QRR) of a flow is the conditional probability that a reordered packet belonging to that flow is falsely inferred lost.

In general, a decrease in θ_f suggests that RR of Flow f is enhanced. Following the definitions, we have:

$$\rho_f = \theta_f q_f^w \tag{5}$$

On the other hand, q_f^c is determined as:

$$q_f^c \approx \sum_{l \in L} \mathbf{R}_{lf} p_l^c \tag{6}$$

We now characterize the network equilibrium specified in (2)-(6) as the optimal solution to a utility maximization problem, subject to capacity constraints.

[3]TCP AIMD, CUBIC TCP [8], and Compound TCP [20] are default TCP in major operating systems, namely, Microsoft Windows Desktop, Linux, and Microsoft Windows Server, respectively. A recent census of the most popular 5000 web servers in the Internet estimate their aggregate share to be over 90% [24]. TCP AIMD and CUBIC TCP employ loss-based CA. When spurious backoff (say, due to packet reordering) is frequent, the CA in Compound TCP becomes loss-based. Otherwise, it further incorporates delay information in determining the growth of *cwnd*. We leave the latter scenario as part of future work.

Theorem 1: For the given network topology and reordering probabilities, $\mathbf{z} = \mathbf{x}$ uniquely solves:

$$NUM(\boldsymbol{\theta}) \begin{cases} \max_{\mathbf{z}} \sum_{f \in F} U_f(z_f, \theta_f) \\ s.t.\ \mathbf{R}\mathbf{z} \preceq \mathbf{c} \end{cases} \tag{7}$$

where $U_f : \mathbb{R}_{++} \times (0,1) \to \mathbb{R}$ is defined as:

$$U_f(z_f, \theta_f) \triangleq -\frac{\alpha_f}{(\sigma_f - 1) d_f^{\kappa_f} z_f^{\sigma_f - 1}} - \theta_f q_f^w z_f \tag{8}$$

Proof: The proof follows a similar line of reasoning as [11]. ∎

Remark 1: From the perspective of microeconomics, U_f can be viewed as the utility function of Flow f, since Flow f in effect maximizes U_f subject to the capacity constraints in equilibrium. The form of U_f is a reverse-engineered result. It is determined by the relationship between x_f and q_f^c (which economically correspond to the amount of consumption and the price in equilibrium, respectively), i.e., (3) and (4).

Remark 2: In (8), if $\theta_f q_f^w = 0$, U_f monotonically increases with x_f . Thus, Flow f will fully utilize the available network capacity. Otherwise, U_f decreases with x_f when x_f reaches beyond a certain point. A significantly large $\theta_f q_f$ tends to "discourage" Flow f from fully utilizing the network capacity.

As a result of the theorem, $\mathbf{x}$ can be viewed as a vector-valued function of $\boldsymbol{\theta}$. Formally, we can write $\mathbf{x} = \mathbf{x}(\boldsymbol{\theta})$. For the given network topology and link capacities, the set of bottleneck links is uniquely determined by the data rate vector $\mathbf{x}$, and is thus indirectly uniquely determined by $\boldsymbol{\theta}$. We therefore denote the set of bottleneck links for a given $\boldsymbol{\theta}$ as $L(\boldsymbol{\theta})$, and the matrix obtained by preserving all rows of $\mathbf{R}$ that correspond to the bottleneck links only as $\mathbf{R}(\boldsymbol{\theta})$.

The following corollary relates $\boldsymbol{\theta}$ to $\mathbf{p^c}$.

Corollary 1: Suppose further that $\mathbf{R}(\boldsymbol{\theta})$ is of full row rank, $\mathbf{p^c}$ is uniquely determined by $\boldsymbol{\theta}$. Formally, we can write $\mathbf{p^c} = \mathbf{p^c}(\boldsymbol{\theta})$.

We shall assume that the assumption of Corollary 1 holds for the rest of the paper, as is common in the literature [19].

By now, it is clear that $\mathbf{x}(\boldsymbol{\theta})$ reflects directly the bandwidth sharing of TCP flows resulting from the given QRR of TCP flows, which is the focus of our BQ. A systematic answer to our BQ can thus be obtained by characterizing how changing $\boldsymbol{\theta}$ influences $\mathbf{x}(\boldsymbol{\theta})$ and performance metrics calibrating $\mathbf{x}(\boldsymbol{\theta})$. We are going to do so in the next section.

IV. How Enhancing RR Influences Bandwidth Sharing

In this section, we answer our BQ raised in Section I:

1. First and foremost, in Section IV-A, we provide an analytical answer to our BQ by finding the derivatives of $\mathbf{x}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

2. In Section IV-B, we further develop the symmetricity and offset properties based on the structure of the directional derivatives of $\mathbf{x}(\boldsymbol{\theta})$. These enable us to gain further insight on how and when enhancing RR of one flow influences other flows without computing the derivatives numerically.

3. It is important to characterize how "good" a resulting bandwidth sharing is. In Section IV-C, we propose a metric $\mathcal{G}(\mathbf{x})$ for measuring the "goodness" of $\mathbf{x}$. We study the variation of $\mathcal{G}(\mathbf{x}(\boldsymbol{\theta}))$ with respect to changes of $\boldsymbol{\theta}$ in terms of the directional derivatives, and show that bandwidth sharing can get worse when RR is enhanced.

A. Variation of Bandwidth Sharing

Example 3: (Example 1 revisited) We revisit Example 1 illustrated in Fig. 1(a), where two flows share a reordering link. Fig. 2 plots the numerically computed equilibrium data rates of Flow i, where $i = 1, 2$, against θ_2, with RTT of 0.1 seconds, link capacity of 350 packets/second, reordering probability of 0.01, and $\theta_1 = 1$. The data rates are computed as $\frac{\sqrt{2}}{d_i\sqrt{q_i}}$, where $i = 1, 2$, by (3) with TCP AIMD for CA. At $\theta_2 = 0.46$, the data rates appear directionally differentiable [16] in either directions of increasing and decreasing θ_2. Yet, the two directional derivatives seem unequal.

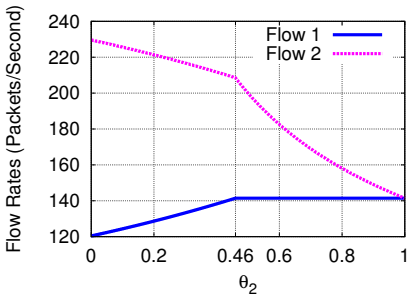


Fig. 2. Equilibrium flow rates versus QRR.

Therefore, for the general case, we conjecture that the set of the bottleneck links $L(\boldsymbol{\theta})$ may vary as $\boldsymbol{\theta}$ changes. The differentiability of $\mathbf{x}(\boldsymbol{\theta})$ and $\mathbf{p^c}(\boldsymbol{\theta})$ at an arbitrary $\boldsymbol{\theta}$ is not guaranteed, and we cannot directly dive into computing the derivatives. We need to establish the differentiability and other important functional properties first. These are attained in the following theorem. The proof applies sensitivity analysis for nonlinear programming (NLP). Interested readers can refer to Appendix A for a brief introduction to sensitivity analysis.

Theorem 2: For $\boldsymbol{\theta} \in (0,1)^{|F|}$, $\mathbf{x}(\boldsymbol{\theta})$ and $\mathbf{p^c}(\boldsymbol{\theta})$ are locally Lipschitz continuous and directionally differentiable.

Proof: Please refer to Appendix B. ∎

On this basis, we provide a complete characterization of how bandwidth sharing among TCP flows changes as RR of TCP flows changes in terms of the directional derivatives of $\mathbf{x}(\boldsymbol{\theta})$. The characterization applies to any $\boldsymbol{\theta} \in (0,1)^{|F|}$ even though the set of bottleneck links can vary as $\boldsymbol{\theta}$ changes.

Theorem 3: For $\boldsymbol{\theta} = \overline{\boldsymbol{\theta}} \in (0,1)^{|F|}$ and a given direction $\mathbf{z}$, let $\boldsymbol{\eta}(T) \triangleq \overline{\boldsymbol{\theta}} + T\mathbf{z}$ $(T \geq 0)$. Suppose that, for all links in the set $\{l \in L(\overline{\boldsymbol{\theta}}) : p_l^c(\overline{\boldsymbol{\theta}}) = 0$ and $\inf_{y_l(\boldsymbol{\eta}(T)) < c_l} T = 0\}$, there exists $T_l' > 0$ such that:

$$y_l(\boldsymbol{\eta}(T)) < c_l \text{ for } T \in (0, T_l') \tag{9}$$

Then:

1. There exists $\overline{T} > 0$ such that $L(\boldsymbol{\eta}(T))$ is invariantly a fixed subset of L, denoted as $L_{\mathbf{z}}$, for $T \in (0, \overline{T})$.

2. Denote the routing matrix over the set of links $L_{\mathbf{z}}$ as $\mathbf{R_z}$ and $\mathbf{p_z^c} = (p_l^c : l \in L_{\mathbf{z}})$:

$$D_{\mathbf{z}}\mathbf{x}(\overline{\boldsymbol{\theta}}) = (\mathbf{J}\mathbf{R}_{\mathbf{z}}^T(\mathbf{R_z}\,\mathbf{J}\,\mathbf{R}_{\mathbf{z}}^T)^{-1}\mathbf{R_z}\,\mathbf{J} - \mathbf{J})\mathbf{Q}\mathbf{z} \tag{10}$$

$$D_{\mathbf{z}}\mathbf{p}_{\mathbf{z}}^{\mathbf{c}}(\overline{\boldsymbol{\theta}}) = (\mathbf{R_z}\,\mathbf{J}\,\mathbf{R}_{\mathbf{z}}^T)^{-1}\mathbf{R_z}\,\mathbf{J}\,\mathbf{Q}\,\mathbf{z} \tag{11}$$

where

$$\mathbf{J} \triangleq \text{diag}\,\{j_f\}, \quad j_f \triangleq \frac{d_f^{\kappa_f} x_f^{\sigma_f + 1}}{\alpha_f \sigma_f}, \quad \mathbf{Q} \triangleq \text{diag}\{q_f^w\} \tag{12}$$

Proof: Please refer to Appendix C. ∎

B. Symmetricity and Offset Properties

We now develop two structural properties, known as symmetricity and offset, based on the derived directional derivatives of $\mathbf{x}(\boldsymbol{\theta})$ and $\overline{\mathbf{p}}^{\mathbf{c}}(\boldsymbol{\theta})$, which enable us to predict *how* and *when* changing RR of a TCP flow will affect other TCP flows, respectively, without the need to actually compute the derivatives.

Proposition 1: For all $f, g \in F$ and $f \neq g$, $D_{-\mathbf{e}_g} x_f$ and $D_{-\mathbf{e}_f} x_g$ have the same sign if $L_{-\mathbf{e}_f} = L_{-\mathbf{e}_g}$, $q_f^w > 0$ and $q_g^w > 0$.

$D_{\mathbf{e}_g} x_f$ reflects the change in x_f when θ_g is perturbed, while $D_{\mathbf{e}_f} x_g$ reflects the change in x_g when θ_f is perturbed. Therefore, we can interpret the results qualitatively in terms of the following symmetricity property.

Symmetricity property: The change in QRR of one flow affects the data rate of another flow in the same manner (positively/negatively/no impact) as it will be affected by the other.

While one may well expect such symmetricity property, it is often hard to justify it in complicated network topologies.

In reality, for two flows A and B, it may be easier to evaluate A's impact on B than to evaluate B's impact on A. Leveraging on the symmetricity property, we can simply perform the easier task to determine the impact in the other direction.

Proposition 2: Suppose $\mathbf{R_z}$ assumes the form:

$$\begin{matrix} v_1 \text{ columns} & v_2 \text{ columns} & & v_{|L_\mathbf{z}|} \text{ columns} \end{matrix}$$
$$\left[\overbrace{\mathbf{r}^1 \cdots \mathbf{r}^1} \quad \overbrace{\mathbf{r}^2 \cdots \mathbf{r}^2} \quad \cdots \overbrace{\mathbf{r}^{|L_\mathbf{z}|} \cdots \mathbf{r}^{|L_\mathbf{z}|}} \right] \tag{13}$$

where $\mathbf{r}^1, \mathbf{r}^2, \cdots, \mathbf{r}^{|L_\mathbf{z}|}$ are linearly independent column vectors representing different routes over bottleneck links and $\sum_{i=1}^{|L_\mathbf{z}|} v_i = |F|$. Then, we have:

$$D_{-\mathbf{e}_f} x_g \begin{cases} < 0 & f \text{ and } g \text{ traverse the route over} \\ & \text{bottleneck links, } \mathbf{r}^i \text{, for some } i \\ = 0 & \text{otherwise} \end{cases} \tag{14}$$

Proof: (sketch) $\mathbf{R_z}$ can be rewritten as the product of two matrices. The first matrix is $[\mathbf{r}^1, \mathbf{r}^2, \cdots, \mathbf{r}^{|L_\mathbf{z}|}]$, which is invertible. The second matrix has all its elements being 1 or 0. Substituting this decomposition into (10) and rearranging leads to the result of the proposition. ∎

Observe that $\mathbf{R_z}$ will assume the form (13) if the total number of different routes taken by TCP flows over all bottleneck links equals $|L_\mathbf{z}|$, the number of the bottleneck links. One can easily conceive such topologies. Three typical topologies having this characteristic are illustrated in Fig. 3, where only the bottleneck links are exhibited. For example, in Fig. 3 (a), there are two different routes over bottleneck links, namely, a route consisting of only the left link traversed by Flow 1, and a route consisting of both bottleneck links traversed by Flows 2 and 3.

Thus, we have the following offset property.

Offset property: In networks where the number of different routes taken by TCP flows over all bottleneck links equals the total number of the bottleneck links, enhancing RR of one TCP flow will not affect the data rate of another flow if and only if the two flows do not traverse exactly the same subset of bottleneck links. Otherwise, they will affect each other negatively.

The result is surprising at first because it imposes a seemingly loose condition for two TCP flows to have no impact on each other. In particular, the condition permits the possibility that the transmission paths of these two TCP flows share some bottleneck links (as long as one of them traverses a bottleneck link not traversed by the other). When two flows share a bottleneck link, the mutual impacts due to enhancing RR seem inevitable.

We can show that $D_{-\mathbf{e}_g} q_f^c = 0$ if f and g do not traverse the same set of bottleneck links. This counter-intuitive property occurs because the network *offsets* the change induced by enhancing RR of Flow f so that it does not alter the aggregate congestive loss rate seen by Flow g. This ensures the transmission rate of Flow g is unchanged.

Taking it one step further, we can conjecture two possible reasons why changing θ_g does not change q_f^c:

1. The decrease in θ_g (which corresponds to enhancing RR of Flow g) will not alter the congestive loss rates over the shared bottleneck links of f and g.

2. The decrease in θ_g does change the congestive loss rate over some shared bottleneck links of Flows f and g. In this case, the congestive loss rates over other bottleneck links traversed by Flow f will change in a way that the aggregate congestive loss rate, q_f^c, remains the same.

It turns out that both cases can occur in reality, as we will demonstrate in our experimental validation presented in Section V. Together, they lead to the occurrence of the offset property.

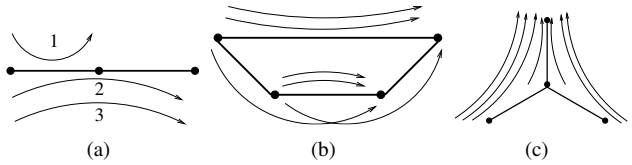


Fig. 3. Examples of networks satisfying (13).

C. "Goodness" of Bandwidth Sharing

We propose to measure the goodness of bandwidth sharing $\mathbf{x}$ as its distance from the optimal flow rate vector $\mathbf{x}^*$, and define:

$$\mathcal{G}(\mathbf{x}) \triangleq \|\mathbf{x} - \mathbf{x}^*\| \tag{15}$$

where $\|\cdot\|$ denotes the Euclidean norm. $\mathbf{x}^*$ can be assigned arbitrarily within the capacity constraint. For example, it can be a good tradeoff between fairness and efficient utilization of the available network capacity [21].

Obviously, a smaller value of $\mathcal{G}(\mathbf{x}(\boldsymbol{\theta}))$ means that bandwidth sharing becomes closer to being optimal and is thus intuitively "better". Following this definition, the "goodness" of bandwidth sharing resulting from the given RR of the flows is:

$$\mathcal{G}(\mathbf{x}(\boldsymbol{\theta})) = \|\mathbf{x}(\boldsymbol{\theta}) - \mathbf{x}^*\| \tag{16}$$

It is useful to differentiate "goodness" with the much used notion of optimality in the literature. Optimality concerns the maximization of utility gained by end-users, and thus has interesting economic interpretations. Theorem 1 provides an optimality result by specifying the form of the utility function being maximized in the equilibrium. In contrast, our defined "goodness" focuses more on calibrating the deviation of the resulting bandwidth allocation from the optimal sharing $\mathbf{x}^*$.

We now show how changing $\boldsymbol{\theta}$ influences the "goodness" of $\mathcal{G}(\mathbf{x}(\boldsymbol{\theta}))$. The following theorem relates the variation of $\mathcal{G}(\mathbf{x}(\boldsymbol{\theta}))$ to the variation of $\mathbf{x}(\boldsymbol{\theta})$.

Theorem 4: For $\boldsymbol{\theta} = \overline{\boldsymbol{\theta}} \in (0,1)^{|F|}$ and direction $\mathbf{z}$:

$$D_\mathbf{z}(\mathcal{G}(\mathbf{x}(\overline{\boldsymbol{\theta}}))) = \frac{1}{\mathcal{G}(\mathbf{x}(\overline{\boldsymbol{\theta}}))} (\mathbf{x}(\overline{\boldsymbol{\theta}}) - \mathbf{x}^*)^T D_\mathbf{z} \mathbf{x}(\overline{\boldsymbol{\theta}}) \tag{17}$$

TABLE III
CONFIGURATIONS FOR SIMULATION STUDY

Topology	Configuration	Connection 1	2	3	4	5	6
Fig. 4(a)	S-1	D	D	R	R	R	R
	S-2	D	D	D	R	R	R
	S-3	D	D	R	R	R	D
Fig. 4(b)	O-1	R	D	D	D	-	-
	O-2	D	D	D	R	-	-
	O-3	D	D	D	D	-	-

Notations: D: TCP-DCR; R: TCP NewReno

Proof: The proof makes use of the chain rule in computing derivatives [16] and is omitted due to constraints in space. ■

Example 4: (Example 2 revisited) In Fig. 1(b), suppose the left and right links have capacities $c_1 = 125$ packets/second and $c_2 = 250$ packets/second, respectively. The left link reorders packets with probability $p_1^w = 0.01$ and the right link does not reorder packets. Flow i has RTT d_i ms, data rate x_i packets/second, and QRR θ_i, where $i = 3, 4, 5$. We set $(d_3, d_4, d_5) = (50, 50, 100)$. We designate the desirable rate allocation $\mathbf{x}^*$ as the proportional fair allocation:[4]

$$(x_3^*, x_4^*, x_5^*) = (72, 197, 53) \tag{18}$$

Suppose for (θ_3, θ_5) in a neighborhood of $(\overline{\theta}_3, \overline{\theta}_5)$, both left and right links are bottleneck links. The routing matrix over the bottleneck links $\mathbf{R_z}$ for all directions $\mathbf{z}$ is therefore:

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \tag{19}$$

By Theorems 3 and 4, we can compute $D_{\mathbf{z}}(\mathcal{G}(\mathbf{x}(\overline{\boldsymbol{\theta}})))$ as:

$$\frac{1}{\mathcal{G}(\mathbf{x}(\overline{\boldsymbol{\theta}}))\sum_{f\in F}\frac{\sigma_f\kappa_f}{x_f^{\sigma_f+1}d_f^{\kappa_f}}}\begin{bmatrix} 3p_1^w x^d & 0 & -3p_1^w x^d \end{bmatrix}\mathbf{z} \tag{20}$$

where $x^d \triangleq (x_5(\overline{\boldsymbol{\theta}}) - x_5^*) = -(x_3(\overline{\boldsymbol{\theta}}) - x_3^*) = -(x_4(\overline{\boldsymbol{\theta}}) - x_4^*)$.

If $x^d > 0$, we have from (20) that $D_{-\mathbf{e_3}}\mathcal{G} > 0$, where $-\mathbf{e}_3 = (0, 0, -1)$ corresponds to the direction that θ_5 is reduced, or, equivalently, RR of Flow 5 is enhanced. In this case, $\mathcal{G}$ increases, or, equivalently, the bandwidth sharing among Flows 3, 4, and 5 becomes "worse". Similarly, if $x^d < 0$, we can show that $D_{-\mathbf{e_1}}\mathcal{G} > 0$. Therefore, we can conclude that the bandwidth sharing can become worse when RR of a flow is enhanced. Remarkably, the scenario giving rise to this counter-intuitive result is very common.

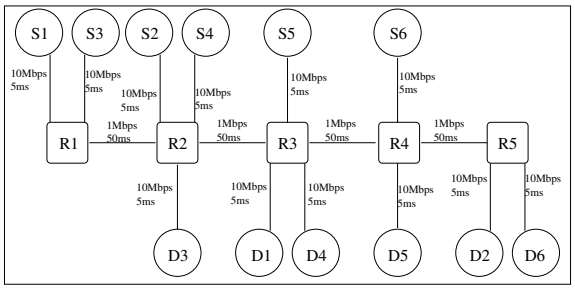
V. PERFORMANCE VALIDATION

In this section, we present experimental validation of our analytical results using Network Simulator (ns) Version 2.29.

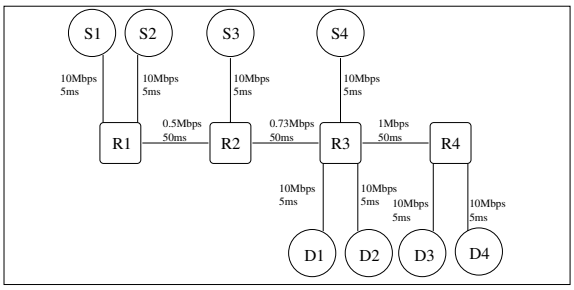
Our analytical results have been built on the model of a reordering network with heterogeneous TCP flows, which extends directly from the popular duality model of TCP flows [11]. Thus, instead of validating the analytical results (namely, the directional derivatives), we choose to validate the innovative insights derived, namely, the symmetricity and offset properties, that are central to answering our BQ qualitatively.

Section V-A explains the simulation setup. Section V-B validates the offset property by comparing its derived predictions with our simulation results.

[4]An allocation, $\mathbf{x}^*$, is proportional fair [9] if, for any other feasible rate allocation $\mathbf{x}$, $\sum_{r\in R}\frac{x_f - x_f^*}{x_f^*} < 0$. It can be obtained by solving the program $\max_{\mathbf{x}\succeq 0}\sum_{r\in R}\log x_f$ subject to $\mathbf{Rx} \preceq \mathbf{c}$.



(a) Simulation topology corresponding to routing matrix in (21).



(b) Simulation topology corresponding to routing matrix in (22).

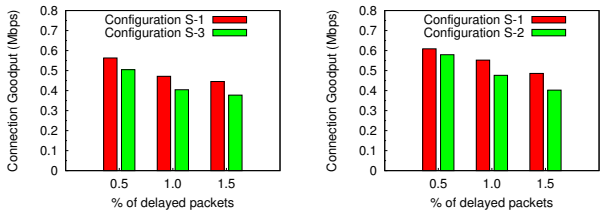
Fig. 4. Simulation topologies.

A. Simulation Setup

Two simulation topologies are employed, as illustrated in Fig. 4. In Fig. 4(a), long-live FTP traffic flows are from Sf to Df via TCP Connections f, where $f = 1, 2, 3, 4, 5, 6$. The TCP connections are established with different TCP variants. We examine three different configurations, Configurations S-1, S-2, and S-3, as summarized in Table III. TCP NewReno [1] is the standardized TCP. It employs AIMD for CA and sets *dupthresh* to be three. TCP-DCR [2] is a reordering TCP enhancement. It employs AIMD for CA and adaptively adjusts *dupthresh* based on the observed occurrences of packet reordering. It has demonstrated a good RR in our previous comparison study [10]. Nevertheless, the reported result depends on RR of the reordering TCP enhancement rather than the specific TCP enhancement chosen. We have observed similar results when TCP-DCR is replaced by two other TCP reordering enhancements with good RR, namely, RR-TCP [25] and TCP-PR [3]. (21) represents the routing matrix for the TCP connections, with the fth column corresponding to Connection f and the lth row corresponding to Link Rl-R$(l+1)$, where $l = 1, 2, 3, 4$. Similar to [25], we introduce packet reordering by delaying a configurable percentage of packets at Rl for a constant amount of time, where $l = 1, 2, 3, 4$.

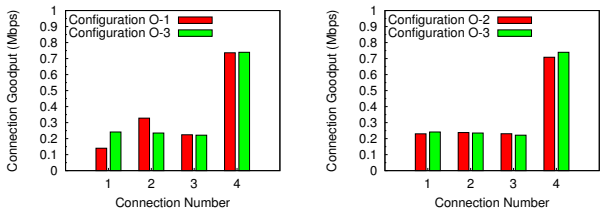
In Fig. 4(b), long-live FTP traffic flows are from Sf to Df via TCP Connections f, where $f = 1, 2, 3, 4$. We examine three different TCP configurations, Configurations O-1, O-2, and O-3, as shown in Table III. (22) represents the routing matrix for the TCP connections, with the fth column corresponding to Connection f and the lth row corresponding to Link Rl-R$(l+1)$, where $l = 1, 2, 3$. Again, we introduce packet reordering by delaying a certain percentage of packets at Rl, where $l = 1, 2, 3$.

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{21}$$



(a) Connection 3. (b) Connection 6.

Fig. 5. Connection goodput comparison for various TCP configurations.



(a) Configurations O-1 and O-3. (b) Configurations O-2 and O-3.

Fig. 6. Connection goodput comparison for various configurations.

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \tag{22}$$

The packet size is 1000 bytes. Each simulation run lasts 1100 seconds. Connection goodput is the performance metric used in this study. The statistics are collected after the first 100 simulated seconds.

B. Simulation Results

1) Validation of Symmetricity Property: We perform the validation by tests over the topology illustrated in Fig. 4(a).

We examine three different settings on the percentage of delayed packets in Rl, where $l = 1, 2, 3, 4$, namely, 0.5%, 1.5%, and 2%. Fig. 5 exhibits the connection goodputs of Connections 3 and 6 under various tested TCP configurations.

Configuration S-3 differs from Configuration S-1 in that Connection 6 becomes more reordering robust by being upgraded from TCP Reno to TCP-DCR. In Fig. 5(a), it is observed that the goodput of Connection 3 is reduced from Configuration S-1 to Configuration S-3 for different percentages of delayed packets. On the other hand, Configuration S-2 differs from Configuration S-1 in that Connection 3 becomes more reordering robust by being upgraded from TCP Reno to TCP-DCR. In Fig. 5(b), it is observed that the goodput of Connection 6 is reduced from Configuration S-1 to Configuration S-2. Therefore, enhancing RR of Connection 3 affects the transmission of Connection 6 in the same way as Connection 3 is affected by Connection 6. The symmetricity property is observed.

2) Validation of Offset Property: We perform the validation by tests over the topology illustrated in Fig. 4(b). We set the percentage of delayed packets at Rl, where $l = 1, 2, 3$ to 0.1%. In the tests, Link Rl-R$(l+1)$, where $l = 1, 2, 3$, is observed to be backlogged throughout the simulation period so that these links are considered as the bottleneck links in our analytical study. The routing matrix over the bottleneck links, $\mathbf{R}(\boldsymbol{\theta})$, is therefore (22).[5]

Now, (22) satisfies the network structure that the number of different routes over all bottleneck links equals the total number of the bottleneck links. We can thus use it to verify the offset property. Fig. 6 plots and compares the connection goodputs under Configurations O-1, O-2, and O-3.

Configuration O-3 differs from Configuration O-1 in that Connection 1 becomes more reordering robust by being upgraded from TCP Reno to TCP-DCR. The set of bottleneck links traversed by Connection 1 is the same as those traversed by Connection 2, but it differs from those traversed by Connections 3 and 4. According to the offset property, enhancing RR of Connection 1 affects Connection 2 adversely but does not affect Connections 3 and 4. Fig. 6(a) compares the goodputs of all connections in Configurations O-1 and O-3. It shows the same trend as we expect. Thus, the offset property is observed. Moreover, we observe in Fig. 6(a) that the aggregate transmission of Connections 1 and 2 remains approximately unchanged in both configurations. This brings the minimal impact to Links R2-R3 and R3-R4, which are the bottleneck links traversed by Connections 3 and 4. This is why enhancing RR of Connection 1 has little impact on the goodputs of Connections 3 and 4.

Configuration O-3 differs from Configuration O-2 in that Connection 4 becomes more reordering robust by being upgraded from TCP Reno to TCP-DCR. As none of the other connections traverse the same set of bottleneck links as that for Connection 4, we would expect that the goodputs of Connections 1, 2, and 3 remain more or less the same in Configurations O-2 and O-3 according to the offset property. Fig. 6(b) compares the goodputs of all connections in Configurations O-2 and O-3. It shows the same trend as we expect. Thus, the offset property is observed.

To derive further insights, we have recorded the queueing dynamics over Link Rl-R$(l+1)$, where $l = 1, 2, 3$, with Configurations O-2 and O-3. Our results report that there are increases in the average queue lengths over Links R1-R2 and R3-R4, and there is a decrease in the average queue length over Link R2-R3, when the configuration switches from O-2 to O-3. As RED sets the packet dropping rate proportional to the running average of the queue length, the congestive loss rate over a link changes in the same way as that of the average queue length. We can thus uncover the series of automatic adjustments of the congestive loss rates that offset the changes in RR of Connection 4 as follows:

1. Link R3-R4 is the only bottleneck link traversed by Connection 4. Enhancing RR of this connection (i.e. when the configuration switches from O-2 to O-3) induces an increase in the congestive loss rate over Link R3-R4.

2. Links R2-R3 and R3-R4 are the two bottleneck links traversed by Connection 3. There is a decrease in the congestive loss rate over Link R2-R3, thereby offsetting the increase in the congestive loss rate over Link R3-R4. This keeps the goodput of Connection 3 roughly unchanged.

3. Links R1-R2 and R2-R3 are the two bottleneck links traversed by Connections 1 and 2. There is an increase in the congestive loss rate over Link R1-R2, thereby offsetting the decrease in the congestive loss rate over Link R2-R3. This keeps the goodputs of Connections 1 and 2 roughly unchanged.

VI. Related Work

To the best of our knowledge, this work is the first systematic study of the interaction of TCP flows with heteroge-

[5]We also observe similar results when the percentage of delayed packets is increased up to 0.2%, beyond which some Link Rl-R$(l+1)$ is no longer always backlogged and hence (22) fails to represent $\mathbf{R}(\boldsymbol{\theta})$.

neous reordering robustness. Recent efforts [5], [15], [22], [24] have explored the interaction of high-speed TCP enhancements with Reno-like TCP. The high-speed TCP enhancements adapt TCP to attain better utilization of high-speed links, via more aggressive congestion avoidance algorithms; see [20] and the references therein. They are thus different from reordering TCP enhancements that improve the accuracy of inferring packet loss in reordering networks, investigated in this paper.

Our approach is to first establish the equilibrium data rates of the traversing traffic flows as a solution to a network utility maximization (NUM) problem parameterized by the reordering robustness of these flows. The NUM formulation for studying congestion control was first proposed in [9], [11]. Yet, these and their followup work do not take packet reordering and the RR of TCP flows, the focus of our work, into account.

In [21], the level of fairness was considered as a parameter of the NUM problem. The focus has been on how the aggregate data rate varies as the fairness parameter changes, and thus distinctly differs from the objective of our study. Besides, the variation of the fairness parameter is constrained to the range in which the set of bottleneck links do not change. By leveraging on the sensitivity analysis, we are able to accommodate cases in which the set of the bottleneck links can change on different QRRs, thereby eliminating this constraint.

VII. Conclusions

The interaction of TCP flows with heterogeneous RR in the presence of packet reordering is complex and unexplored. This hinders evaluating the impact of deploying reordering TCP enhancements on the existing networks, which philosophically corresponds to the question "*Does it hurt when others prosper?*" It turns out that a TCP flow can benefit or suffer when other flows become more robust against reordering, or "*prosper*". Our developed directional derivatives of $\mathbf{x}(\boldsymbol{\theta})$ expose how such effects take place. The resulting offset property reveals counter-intuitively that, in networks where the number of routes over all bottleneck links equals the number of the bottleneck links, a flow is not affected when its competing flow prosper if the two flows do not traverse the same set of bottleneck links. On the other hand, from the perspective of global welfare, the whole network can sometimes suffer when some of the flows "prosper".

There are several possible extensions to our work, including:

1. verifying the prediction of our model using live Internet measurements. In particular, it is interesting to find out the characteristics of network topologies that would give rise to the offset property, and

2. extending the model to study bandwidth sharing among TCP flows that respond heterogeneously to other network events, such as wireless losses. We remark that the major challenge is to seek a general yet tractable model of the versatile approaches for enhancing TCP.

Acknowledgements

This research is supported in part by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Grant No. HKU 714510E.

Appendix

A. Sensitivity Analysis in NLP

The study of sensitivity in NLP is mainly concerned with how an optimal solution and the optimal value vary when the parameters of an NLP problem are perturbed [6]. Consider the optimization problem parameterized by $\boldsymbol{\epsilon}$:

$$P(\boldsymbol{\epsilon}) \begin{cases} \max_{\mathbf{z}} h(\mathbf{z}, \boldsymbol{\epsilon}) \\ \text{s.t. } m_i(\mathbf{z}, \boldsymbol{\epsilon}) \leq 0 \quad \text{where } i = 1, 2, ..., n \end{cases} \tag{23}$$

Define the Lagrangian:

$$La(\mathbf{z}, \mathbf{u}, \boldsymbol{\epsilon}) \triangleq h(\mathbf{z}, \boldsymbol{\epsilon}) + \sum_{i=1}^{m} u_i m_i(\mathbf{z}, \boldsymbol{\epsilon}) \tag{24}$$

where u_i, for $i = 1, 2, ..., m$, are Lagrangian multipliers.

$(\mathbf{z}, \mathbf{u}, \boldsymbol{\epsilon})$ is said to satisfy the Karush-Kuhn-Tucker (KKT) condition if:

$$\begin{cases} \nabla_{\mathbf{z}} La(\mathbf{z}, \mathbf{u}, \boldsymbol{\epsilon}) = 0 \\ m_i(\mathbf{z}, \boldsymbol{\epsilon}) \leq 0, u_i \geq 0 \quad \text{where } i = 1, 2, ..., n \\ u_i m_i(\mathbf{z}, \boldsymbol{\epsilon}) = 0 \quad \text{where } i = 1, 2, ..., n \end{cases} \tag{25}$$

Now, we confine our scope to consider the scenario that, for each ϵ, $P(\boldsymbol{\epsilon})$ has a unique solution vector $\mathbf{z}(\boldsymbol{\theta})$ and a unique vector of the Lagrangian multipliers $\mathbf{u}(\boldsymbol{\theta})$ such that $(\mathbf{z}(\boldsymbol{\theta}), \mathbf{u}(\boldsymbol{\theta}), \boldsymbol{\epsilon})$ satisfies the KKT condition.

Theorem 5: Denote $\overline{\mathbf{z}} \triangleq \mathbf{z}(\overline{\boldsymbol{\epsilon}})$ *and* $\overline{\mathbf{u}} \triangleq \mathbf{u}(\overline{\boldsymbol{\epsilon}})$. *At* $(\overline{\mathbf{z}}, \overline{\mathbf{u}}, \overline{\boldsymbol{\epsilon}})$, *suppose:*

1. $\nabla^2_{\mathbf{z}} La(\overline{\mathbf{z}}, \overline{\mathbf{u}}, \overline{\boldsymbol{\epsilon}}) \prec 0$

2. *The gradient vectors* $\nabla_{\mathbf{z}} m_i(\overline{\mathbf{z}}, \overline{\boldsymbol{\epsilon}})$ *for* i *corresponding to the active constraints (namely,* $m_i(\overline{\mathbf{z}}, \overline{\boldsymbol{\epsilon}}) = 0$*) are linearly independent.*

Then, $(\mathbf{z}(\boldsymbol{\epsilon}), \mathbf{u}(\boldsymbol{\epsilon}))$ *is Lipschitz continuous and directionally differentiable in a neighbourhood* $N(\overline{\boldsymbol{\epsilon}})$ *of* $\overline{\boldsymbol{\epsilon}}$.

B. Proof of Theorem 2

Consider the nonlinear program NUM($\boldsymbol{\theta}$), which has $\mathbf{x}(\boldsymbol{\theta})$ as the unique solution and $\mathbf{p}^{\mathbf{c}}(\boldsymbol{\theta})$ as the unique vector of the Lagrangian multipliers satisfying the KKT condition. At a given $\boldsymbol{\theta} = \overline{\boldsymbol{\theta}} \in (0,1)^{|F|}$:

1. $\nabla^2_{\mathbf{x}} La(\mathbf{x}(\overline{\boldsymbol{\theta}}), \mathbf{p}^{\mathbf{c}}(\overline{\boldsymbol{\theta}}), \overline{\boldsymbol{\theta}}) = -diag\{\frac{\alpha_f \sigma_f}{d_f^{\kappa_f} x_f^{\sigma_f + 1}}\} \prec 0$

2. There are $|L(\overline{\boldsymbol{\theta}})|$ active constraints, i.e. $\mathbf{R}_{l*} x - c_l = 0, l \in L(\overline{\boldsymbol{\theta}})$. Their gradients $\mathbf{R}_{l*}$, $l \in L(\overline{\boldsymbol{\theta}})$ are linearly independent by our prior assumption that $\mathbf{R}(\boldsymbol{\theta})$ is of full row rank.

By Theorem 5, $(\mathbf{x}(\boldsymbol{\theta}), \mathbf{p}^{\mathbf{c}}(\boldsymbol{\theta}))$ is Lipschitz continuous and a directionally differentiable function of $\boldsymbol{\theta}$ in $N(\overline{\boldsymbol{\theta}})$. Since $\overline{\boldsymbol{\theta}}$ is arbitrarily chosen from $(0,1)^{|F|}$, we conclude that $\mathbf{x}(\boldsymbol{\theta})$ and $\mathbf{p}^{\mathbf{c}}(\boldsymbol{\theta})$ are locally Lipschitz continuous and directionally differentiable in $\boldsymbol{\theta} \in (0,1)^{|F|}$.

C. Proof of Theorem 3

1. Consider $l \in L \backslash L(\overline{\boldsymbol{\theta}})$. By definition, we have $y_l(\overline{\boldsymbol{\theta}}) < c_l$. By the continuity of $y_l(\boldsymbol{\theta})$, $y_l(\boldsymbol{\theta}) < c_l$ will continue to hold for $\boldsymbol{\theta}$ in a neighborhood of $\overline{\boldsymbol{\theta}}$. It follows that $L(\boldsymbol{\theta}) \subseteq L(\overline{\boldsymbol{\theta}})$ for $\boldsymbol{\theta}$ in this neighbourhood. Thus, there exists $\overline{T}_1 > 0$ such that $L(\boldsymbol{\eta}(T)) \subseteq L(\overline{\boldsymbol{\theta}})$ for $T \in (0, \overline{T}_1)$.

On the other hand, consider the set:

$$L_a(\overline{\boldsymbol{\theta}}) \triangleq \{l \in L(\overline{\boldsymbol{\theta}}) : p_l^c(\overline{\boldsymbol{\theta}}) > 0\} \tag{26}$$

By applying Assumptions A1 and A2, we can show that there exists $\overline{T}_2 > 0$ such that $L_a(\overline{\boldsymbol{\theta}}) \subseteq L(\boldsymbol{\eta}(T))$ for $T \in (0, \overline{T}_2)$. Therefore, for $T \in (0, \min(\overline{T}_1, \overline{T}_2))$:

$$L_a(\overline{\boldsymbol{\theta}}) \subseteq L(\boldsymbol{\eta}(T)) \subseteq L(\overline{\boldsymbol{\theta}}) \tag{27}$$

If $L_a(\overline{\boldsymbol{\theta}}) = L(\overline{\boldsymbol{\theta}})$, we have $L(\boldsymbol{\eta}(T)) = L(\overline{\boldsymbol{\theta}})$ for $T \in (0, \min(\overline{T}_1, \overline{T}_2))$. Otherwise, consider $l \in L(\overline{\boldsymbol{\theta}}) \backslash L_a(\overline{\boldsymbol{\theta}})$ and define:

$$T_l \triangleq \inf_{y_l(\boldsymbol{\eta}(T)) < c_l} T \tag{28}$$

If $T_l > 0$, it follows that $y_l(\boldsymbol{\eta}(T)) = c_l$ and thus $l \in L(\boldsymbol{\eta}(T))$ for $T \in (0, T_l)$. If $T_l = 0$, by (9), there exists $T_l' > 0$ such that $l \notin L(\boldsymbol{\eta}(T))$ for $T \in (0, T_l')$. Define $\overline{T}$ as:

$$\begin{aligned}\overline{T} \triangleq \min\big(&\overline{T}_1, \overline{T}_2, \min\{T_l : l \in L(\overline{\boldsymbol{\theta}}) \backslash L_a(\overline{\boldsymbol{\theta}}), T_l > 0\}, \\ &\min\{T_l' : l \in L(\overline{\boldsymbol{\theta}}) \backslash L_a(\overline{\boldsymbol{\theta}}), T_l = 0\}\big)\end{aligned} \tag{29}$$

which is greater than zero. Hence, for $T \in (0, \overline{T})$, $L(\boldsymbol{\eta}(T))$ is invariantly L_s, where:

$$L_s \triangleq L_a(\overline{\boldsymbol{\theta}}) \cup \{l \in L(\overline{\boldsymbol{\theta}}) \backslash L_a(\overline{\boldsymbol{\theta}}) : T_l > 0\} \tag{30}$$

2. Denote:

$$\mathbf{w}(T) \triangleq (\mathbf{x}(\boldsymbol{\eta}(T)), \overline{\mathbf{p}}^c(\boldsymbol{\eta}(T)), \boldsymbol{\eta}(T)) \tag{31}$$

and

$$\begin{aligned}&\mathbf{G}(\mathbf{x}, \mathbf{p}_{\mathbf{z}}^{\mathbf{c}}, \boldsymbol{\theta}) \\ \triangleq\ &((\frac{\alpha_f}{d_f^{\kappa_f} x_f^{\sigma_f}} - q_f) : f \in F; p_l^c(y_l - c_l) : l \in L_{\mathbf{z}})\end{aligned} \tag{32}$$

By Theorem 2, $\mathbf{x}(\boldsymbol{\theta})$ and $\overline{\mathbf{p}}^{\mathbf{c}}(\boldsymbol{\theta})$ are directionally differentiable at $\overline{\boldsymbol{\theta}}$. Thus, the right derivative of $\mathbf{G}(\mathbf{w}(T))$ at $T = 0$ exists and can be computed by the chain rule as:

$$\begin{aligned}&\lim_{T \to 0^+} \frac{\mathbf{G}(\mathbf{w}(T)) - \mathbf{G}(\mathbf{w}(0))}{T} \\ =\ & D\mathbf{G}(\mathbf{w}(0)) \cdot \lim_{T \to 0^+} \frac{\mathbf{w}(T) - \mathbf{w}(0)}{T}\end{aligned} \tag{33}$$

where

$$D\mathbf{G}(\mathbf{w}(0)) = \begin{pmatrix} -\mathbf{J}^{-1} & \overline{\mathbf{R}}^T & -\mathrm{diag}\{q_r^w\} \\ \overline{\mathbf{R}} & \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{34}$$

and

$$\lim_{T \to 0^+} \frac{\mathbf{w}(T) - \mathbf{w}(0)}{T} = \begin{pmatrix} D_{\mathbf{z}}\mathbf{x}(\overline{\boldsymbol{\theta}}) \\ D_{\mathbf{z}}\overline{\mathbf{p}}^c(\overline{\boldsymbol{\theta}}) \\ \mathbf{z} \end{pmatrix} \tag{35}$$

On the other hand, we have $\mathbf{G}(\mathbf{w}(T)) = \mathbf{0}$ for $T \in [0, \overline{T})$. Thus, its right derivative at $T = 0$ is zero. Rearranging the terms, we obtain (10) and (11).

REFERENCES

[1] M. Allman, V. Paxson, and E. Blanton. TCP Congestion Control. *IETF RFC 5681*, Sep. 2009.

[2] S. Bhandarkar and A.L.N. Reddy. TCP-DCR: Making TCP Robust to Non-Congestion Events. *Lect. Notes Comp. Sc.*, Vol. 3042, pp. 712-724, May 2004.

[3] S. Bohacek, J.P. Hespanha, J. Lee, C. Lim, and K. Obraczka. A New TCP for Persistent Packet Reordering. *IEEE/ACM Trans. Netw.*, Vol. 14, No. 2, pp. 369-382, Apr. 2006.

[4] B. Braden, D. Clark, J. Crowsoft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Patridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang. Recommendations on Queue Management and Congestion Avoidance in the Internet. *IETF RFC 2309*, Apr. 1998.

[5] L. Budzisz and R. Stanojevic, A. Schlote, F. Baker, and R. Shorten. On the Fair Coexistence of Loss- and Delay-Based TCP. *IEEE/ACM Trans. Netw.*, Vol. 19, No. 6, pp. 1811-1824, Dec. 2011.

[6] A.V. Fiacco. Sensitivity and Stability in NLP: Continuity and Differential Stability. *Encyclopedia of Optimization, Springer*, pp. 3467-3471, 2009.

[7] L. Gharai, C. Perkins, and T. Lehman. Packet Reordering, High Speed Networks and Transport Protocol Performance. *Proc. IEEE ICCCN 2004*, Oct. 2004.

[8] S. Ha, I. Rhee, and L. Xu. CUBIC: A New TCP-Friendly High-Speed TCP Variant. *ACM SIGOPS Operating System Review*, Vol. 42, No. 5, pp. 64-74, Jul. 2008.

[9] F. Kelly, A. Maulloo, and D. Tan. Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability. *J. Oper. Res. Soc.*, Vol. 49, No. 3, pp. 237-252, Mar. 1998.

[10] K.-C. Leung, V.O.K. Li, and D. Yang. An Overview of Packet Reordering in Transmission Control Protocol (TCP): Problems, Solutions, and Challenges. *IEEE Trans. Parall. and Distrib. Syst.*, Vol. 18, No. 4, pp. 522-535, Apr. 2007.

[11] S.H. Low. A Duality Model of TCP and Queue Management Algorithms. *IEEE/ACM Trans. Netw.*, Vol. 11, No. 4. pp. 525-536, Aug. 2003.

[12] R. Ludwig and R.H. Katz. The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions. *ACM SIGCOMM Comp. Comm. Rev.*, Vol. 30, No. 1, pp. 30-36, Jan. 2000.

[13] A. Medina, M. Allman, and S. Floyd. Measuring the Evolution of Transport Protocols in the Internet. *ACM SIGCOMM Comp. Comm. Rev.*, Vol. 35, No. 2, pp. 37-52, Apr. 2005.

[14] M. Mellian, M. Meo, L. Muscariello, and D. Rossi. Passive Analysis of TCP Anomalies. *Comput. Netw.*, Vol. 52, No. 14, pp. 2663-2676, Oct. 2008.

[15] K. Mills, J. Filliben, D. Cho, E. Schwartz, and D. Genin. Study of Proposed Internet Congestion Control Algorithms. *NIST Special Publication 500-282*, May 2010.

[16] J.R. Munkres. *Analysis on Manifolds*. Addison-Wesley Pub. Co., 1991.

[17] J. Padhye and S. Floyd. On Inferring TCP Behavior. *ACM SIGCOMM Comp. Comm. Rev.*, Vol. 31, No. 4, pp. 287-298, Oct. 2001.

[18] P. Sarolahti and A. Kuznetsov. Congestion Control in Linux TCP. *Proc. USENIX 2002*, pp. 49-62, Jun. 2002.

[19] R. Srikant. The Mathematics of Internet Congestion Control. *Cambridge, MA:Birkhauser*, 2004.

[20] K. Tan, J. Song, Q. Zhang, and M. Sridharan. A Compound TCP Approach for High-speed and Long Distance Networks. *Proc. IEEE INFOCOM 2006*, Apr. 2006.

[21] A. Tang, J. Wang, and S.H. Low. Counter-Intuitive Throughput Behaviors in Networks Under End-to-End Control. *IEEE/ACM Trans. Netw.*, Vol. 14, No. 2, pp. 355-368, Apr. 2006.

[22] A. Tang, J. Wang, S.H. Low, and M. Chiang. Equilibrium of Heterogeneous Congestion Control: Existence and Uniqueness. *IEEE/ACM Trans. Netw.*, Vol. 15, No. 4, pp. 824-837, Aug. 2007.

[23] W. Wu, P. Demar, and M. Crawford. Why Can Some Advanced Ethernet NICs Cause Packet Reordering. *IEEE Comm. Lett.*, Vol. 15, No. 2, pp. 253-255, Feb. 2011.

[24] P. Yang, W. Luo, L. Xu, J. Deogun, and Y. Lu. TCP Congestion Avoidance Algorithm Identification. *Proc. IEEE ICDCS 2011*, pp. 310-321, Jun. 2011.

[25] M. Zhang, B. Karp, S. Floyd, and L. Peterson. RR-TCP: A Reordering-Robust TCP with DSACK. *Proc. IEEE ICNP 2003*, pp. 95-106, Nov. 2003.