社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2004－AL－93　(8)
2004／1／30

長距離・短距離通信が混在する環境での TCP/IP のデータ転送速度の理論的解析

伊藤 剛志　　稲葉 真理
東京大学

長距離と短距離の通信が混在する環境での TCP/IP の性能を理論的に解析する研究について、二つの方法による結果を述べる。どちらも TCP の輻輳制御アルゴリズムを簡略化した AIMD 輻輳制御アルゴリズムを対象とし、ボトルネックが 1 個だけある単純なネットワークモデルを採用する。一つは通信が時間とともに動的に発生・完了する環境での total flow time の competitive analysis で、距離が一定の場合より competitive ratio が距離を反映した加速度の最大と最小の比だけ悪くなることを示す。もう一つは single-drop モデルでの定常状態の解析で、加速度が一定の場合には定常状態での帯域利用率の合計は速度を落とす通信の選びかたによらないこと、2 本の通信が存在して加速度が異なる場合には定常状態での帯域利用率の合計は加速度が小さい通信を常に優先して通す場合に最大となることなどを示す。

Theoretical Analysis of Throughput of TCP/IP Congestion Control Algorithm with Different Distances

Tsuyoshi ITO　　Mary INABA
The University of Tokyo

Two results are presented on the theoretical analysis of the performance of TCP/IP in environments where both long- and short-distance communications exist. Both studies treat AIMD congestion control algorithm, a simplified version of the congestion control algorithm used in TCP, and assume single-bottleneck network model. One result is about the competitive analysis of the total flow time in environments where communications arrive and complete as time goes, and shows that the coexistence of communications with different distances results in worse competitive ratio than the cases all the communications have an equal distances by factor of ratio of maximum to minimum acceleration. The other result is about the analysis of stationary states in the single-drop model. We show that if all the connections have an equal acceleration, the total bandwidth utilization does not depend on which connection decreases its transmission rate, and that if there are exactly two connections with different accelerations, the maximum total utilization is achieved by passing the data of the connection with lower acceleration as much as possible.

1 Introduction

The Transmission Control Protocol (TCP) is used by most data transfer in the Internet. The congestion control in TCP makes a guess on the appropriate transmission rate by only using the data exchanged between the endpoints of the connection. The current congestion control algorithm increases the transmission rate at a constant rate while the transmission succeeds, and drops it to a half of the current rate when congestion is detected. This algorithm is called Additive Increase and Multiplicative Decrease (AIMD) [1].

These days, the backbone network over gigabits per second such as Abilene and GÉANT is rapidly constructed, and the bandwidth of the links in the Internet, especially of the long-distance ones, is increasing. The increase has revealed the problem the current TCP congestion control has: the current TCP results in very low throughput when used for long-distance data transfer [13], which is known as the performance problem with Long Fat Pipe Networks (LFNs). To tackle the LFN problem, many alternative congestion control algorithms for TCP have been proposed [5,8,10]. Currently their perfor-

mance is evaluated mainly by means of experiments and simulations. Theoretical analysis of the current TCP algorithm is now required for analytical comparison of different congestion control algorithms, which is the main theme of this paper. Among many causes pointed out for the LFN problem such as high transmission error rate, we focus on the coexistence of short-distance connections with very-long-distance ones.

At the same time, there are more and more needs for the transfer of various kinds of large data. For example, people will send e-mails with video images of tens or hundreds of megabytes length in near future. As an example where huge data is concerned, some research institutes currently receive data of terabytes or more produced by scientific measurement instruments by the physical transportation of Digital Linear Tapes (DLTs), but they can receive them online if the LFN problem is resolved [7]. This indicates the necessity of the analysis of the performance of the long-time transfer of very large data.

In this paper, in quest of the exact reason the current TCP does not perform well on LFNs, theoretical analyses are performed from various viewpoints on the most fundamental network model with a single bottleneck, as depicted in Figure 1. The performance is analyzed in the case that each of the connections with different distances transfers large data. As we focus on the transfer of large data, we consider only the AIMD congestion avoidance phase of TCP of sufficiently long period, ignoring the effect of the slow start phase which is relatively short period of time.

In the real world, the distance of a connection affects the behavior of the AIMD mainly in three ways. (1) Acceleration: In the AIMD algorithm, the transmission rate of a connection increases by $\alpha = c/T^2$ per unit time while the transmission succeeds, where c is Sender Maximum Segment Size (SMSS), which is a constant for usual case, and T is Round Trip Time (RTT), which reflects the distance of the connection. This α is called the *acceleration* of the connection. (2) Response time: After a node transmits its data, it takes the time amount of RTT to know whether the transmission has succeeded or failed. (3) The number of congestion points: Long-distance connections pass more congestion points such as routers and switches than short-distance connections.— We focus on the difference of (1) to isolate the effects of different distances of connections. We say the environment is *homogeneous* if all the connections have an equal acceleration, and *heterogeneous* otherwise.

Edmonds et al. [4] consider the single-bottleneck network and prove by theoretical analysis that the AIMD algorithm performs well when all the connections have a common acceleration, that is, in the homogeneous case. In section 4, we extend their result to the heterogeneous case and show a result that suggests the AIMD does not perform well when connections have different accelerations, thus explaining the low throughputs under the coexistence of short- and very-long-distance communication.

In section 5, we further analyze the total bandwidth utilization and the share of the available bandwidth in the stationary state. Many existing results, including the result by Edmonds et al. and our extension to it, assume that when congestion occurs at the bottleneck, all the connections drop their transmission rate at the same time as depicted in Figure 2 (a). With this assumption, it is shown that the total utilization does not depend on the number of connections. To fill a gap between this assumption and the reality, we consider another model of the drop as shown in Figure 2 (b). In the new model, when congestion occurs, one connection is chosen as *victim* and only the victim drops its transmission rate and the transmission rate of the other connections does not change. We call this model the *single-drop model* and refer to the previous model as the *all-drop model*. Using the single-drop model, we show that under several different conditions, the total utilization increases as the number of connections increases. We prove that under some conditions in the single-drop model, the total utilization becomes worse with the heterogeneity increases. In addition, we prove that in the all-drop model and part of the single-drop model, the bandwidth is shared among the connections in proportion to their accelerations, hence in proportion to the inverse of the square of their RTTs.

2 Related works

TCP congestion control is an algorithm which works without knowledge about the bandwidth of links or information about other communication sharing the network. There are two approaches to the theoretical analysis of the performance of such incomplete-information algorithms. Probabilistic analysis is the analysis of the average case after assuming some probabilistic distribution of the unknown information, and competitive analysis is the analysis of the worst *competitive ratio* of the performance to the fictional case where the complete information were available to an algorithm.

Probabilistic analysis. Several papers [11, 12, 14] analyze how the throughput of homogeneous TCP connections is affected by random packet losses under the assumption that every packet is dropped independently with a constant probability.

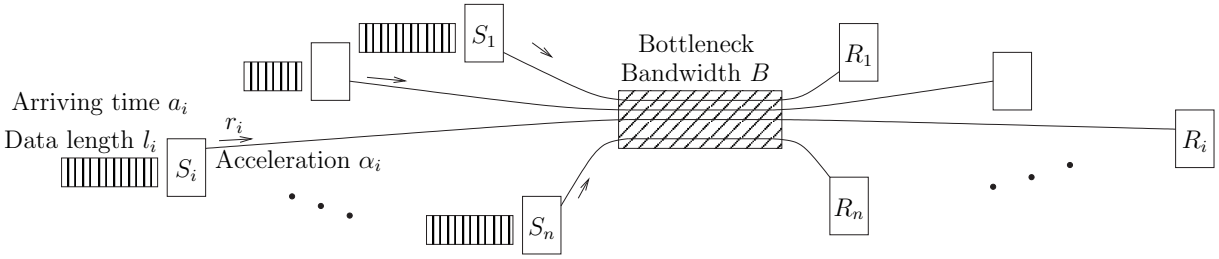

Figure 1: A single-bottleneck network consisting of a bottleneck with bandwidth B and n connections with different distances. Each connection C_i has the acceleration α_i which is inversely proportional to the square of its RTT.

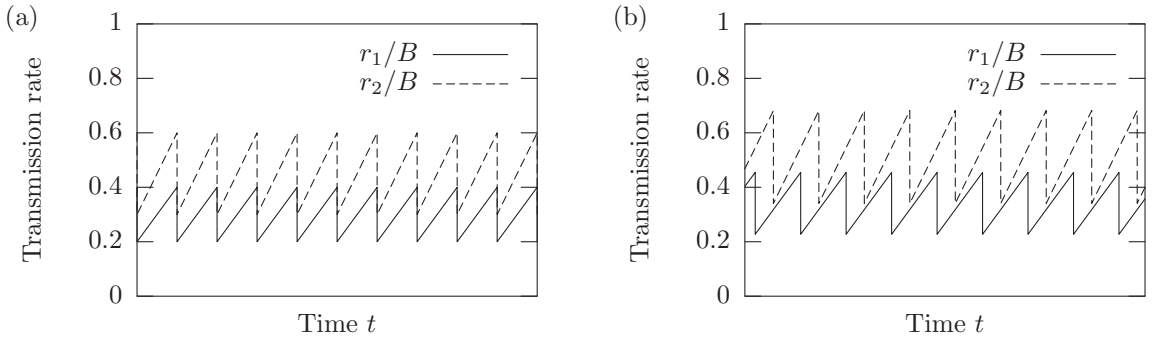

Figure 2: Time evolution of the transmission rates r_1 and r_2 of the two connections with the accelerations $\alpha_1 : \alpha_2 = 2 : 3$ and the drop factor $\beta = 1/2$. (a) uses the all-drop model, and (b) uses the single-drop model and Periodic victim policy.

De Vendictis et al. [2] consider the environment with two connections where one connection uses the current TCP and the other uses a different congestion control algorithm called TCP Vegas, and analyze the throughputs of the connections in the stationary state.

Competitive analysis. At the top of our knowledge, the application of the competitive analysis to the performance evaluation of TCP congestion control was first proposed and performed by Karp et al. [9]. They formalized the congestion control as the algorithm to guess a secret available bandwidth which changes little by little over time. Edmonds et al. [4] consider the setting where multiple homogeneous connection jobs arrive and complete over time. They regard TCP as an online and distributed algorithm to share the available bandwidth among ongoing connections and compare it to scheduling algorithms which share the available processors among ongoing jobs in the centralized manner. They show that TCP achieves a constant competitive ratio independent of the number of connections by the competitive analysis against the optimal offline scheduling algorithm. However, the result holds only for the homogeneous case.

3 Definitions

Figure 1 illustrates the *single-bottleneck network* we consider. The network consists of one bottleneck with bandwidth B, n *senders* $S_1, \ldots, S_n$ on one side of the bottleneck, and n corresponding *receivers* $R_1, \ldots, R_n$ on the other side. Sender S_i sends its data to receiver R_i, together making a *connection* C_i. S_i sends data at the rate of r_i per unit time, where r_i, called the *transmission rate* of C_i, changes as time goes on. Any algorithm must control the transmission rates so that their sum $\sum_{i=1}^{n} r_i$ never exceed B. Here we use *fluid model*: r_i can be any nonnegative real value and the data can be sent as if it does not have the minimum unit such as a packet, an octet or even a bit.

Each connection C_i is associated with three constants: the *arriving time* a_i, the *data length* $l_i > 0$, and the *acceleration* $\alpha_i > 0$. The connection C_i starts at time a_i to send l_i amount of data. We consider both the case of $l_i < \infty$ and the case of $l_i = \infty$. The acceleration α_i is used by the AIMD algorithm as described later.

In this paper, the behavior of the AIMD congestion control algorithm is formalized as follows. A constant $0 < \beta \leq 1$ fixed. β is called *drop factor* and common to all the connections. Each C_i

maintains its transmission rate $r_i \geq 0$ as follows. While $\sum_{i=1}^{n} r_i < B$, in other words, the sum of transmission rates of the n connections is less than the bottleneck bandwidth, each C_i transmits an infinitesimally small amount $r_i\,dt$ of data for an infinitesimally short time dt and increases r_i by $\alpha_i\,dt$. When $\sum_{i=1}^{n} r_i = B$, meaning that the sum of transmission rates hits the bandwidth, what happens depends on which *drop model* we adopt. (1) *All-drop model*: All the r_i's are multiplied by $(1-\beta)$ instantly at the same time, as shown in Figure 2 (a). (2) *Single-drop model*: One connection C_i is chosen as *victim* and its transmission rate r_i is multiplied by $(1-\beta)$ instantly. Note that in the single-drop model, the choice of victim is not unique, and we will discuss about *victim policies* in section 5. For example, Periodic victim policy defined in section 5.3 chooses every connection as victim in turn as shown in Figure 2 (b).

When $l_i < \infty$ and $\int_{a_i}^{t} r_i\,dt = l_i$, meaning that connection C_i has sent all of its data, then connection C_i terminates. In this case, the time elapsed since the arriving time a_i until the termination of C_i is called the *flow time* f_i of connection C_i, and the sum $F = \sum_{i=1}^{n} f_i$ is called the *total flow time*. The connection C_i is *alive* since its arrival until its termination.

In the current TCP congestion control algorithm, β is fixed to $1/2$, and α_i is inversely proportional to the square of RTT of connection C_i. The case that α_i's are equal for all the connections is called *homogeneous* case, and the other case *heterogeneous* case.

4 Competitive analysis of total flow time in heterogeneous environments

In this section, we assume the all-drop model and we consider the case that $l_i < \infty$ for all i, that is, each sender sends a finite amount of data. In this setting, we consider the optimization problem of minimizing the total flow time.

Now consider the arriving time a_i is not known until the request of data transfer of C_i arrives at time a_i. Similarly, consider the data length l_i is not known until the sender sends l_i amount of data, reaching the end of data. This situation is common, because it corresponds to the case that the congestion control algorithm is implemented as a protocol stack independent of the application which decides when and which data to send. The AIMD algorithm works without any problem in this situation, because it does not use any information given in future to work. In this sense, the AIMD algorithm is called an *online algorithm*.

Besides, the AIMD is a *distributed algorithm* in the following sense. Each connection C_i only requires the information about its own parameters, a_i, α_i and l_i, and does not need to know the bottleneck bandwidth B or the parameters of the other connections, provided the sender knows whether $\sum r_i < B$ or $\sum r_i = B$. In TCP, this last additional information is supplied by the presence or the absence of acknowledgment from the receiver.

In contrast to the online and distributed AIMD algorithm, we can consider fictional *offline* and *centralized* algorithms. This kind of algorithms know B, and a_i and l_i of all the n connections before any request arrives, and controls all the r_i's simultaneously. Because offline and centralized algorithms have more access to knowledge than online and distributed algorithms like the AIMD, the optimal offline and centralized algorithm achieves no longer total flow time than the AIMD.

For the homogeneous case where $\alpha_1 = \cdots = \alpha_n = \alpha$, Edmonds et al. prove the following.

Theorem 1 ([4]). *The AIMD is competitive to the optimal offline and centralized algorithm with a limited bottleneck bandwidth in the following sense. Let $q \geq 1$ be an integer, $\varepsilon > 0$,*

$$s = (2+\varepsilon) \cdot \frac{1}{1-(1-\beta)^{q-1}} \cdot \frac{2}{2-\beta}\Big(1+\frac{1}{q}\Big), \quad (1)$$

and suppose we compare the total flow time $F(\mathrm{AIMD}(\mathcal{C}, B))$ of the set $\mathcal{C} = \{C_1, \ldots, C_n\}$ of connections achieved by the AIMD with bottleneck bandwidth B and that achieved $F(\mathrm{OPT}(\mathcal{C}, B/s))$ by the optimal offline and centralized algorithm with bottleneck bandwidth B/s. Then, for $D = 2(q+1)nB/\alpha$, it holds that

$$\frac{F(\mathrm{AIMD}(\mathcal{C}, B))}{F(\mathrm{OPT}(\mathcal{C}, B/s)) + D} \leq 2 + \frac{4}{\varepsilon}.$$

To prove this, they compare AIMD with an online centralized algorithm called Equi-partition, or EQUI. At time t, EQUI allocates $B/|\mathrm{Alive}(t)|$ bandwidth to each $C_i \in \mathrm{Alive}(t)$, where $\mathrm{Alive}(t)$ is the set of connections alive at time t.

Theorem 2 ([4]). *Let t be the time when the connection C_i drops for the jth time in AIMD. Then, $r_i[t-0] \geq (1-(1-\beta)^{j-1})B/|\mathrm{Alive}(t)|$.*

From Theorem 2, it can be shown that $F(\mathrm{AIMD}(\mathcal{C}, B)) \leq F(\mathrm{EQUI}(\tilde{\mathcal{C}}, B')) + D$, where $F(\mathrm{EQUI}(\tilde{\mathcal{C}}, B'))$ means the total flow time achieved by EQUI with bottleneck $B' = B/(\frac{1}{1-(1-\beta)^{q-1}} \cdot \frac{2}{2-\beta}(1+\frac{1}{q}))$, and $\tilde{\mathcal{C}}$ is a slightly modified set of connections which may include *sequencial phases*. A

sequencial phase takes a constant amount of time to complete, regardless of how much bandwidth is allocated to the connection. By combining this result and the following theorem, the proof of Theorem 1 is completed.

Theorem 3 ([3]). *EQUI with bottleneck B' is $(2+\frac{4}{\varepsilon})$-competitive to the optimal offline and centralized algorithm with bottleneck $B'/(2+\varepsilon)$ even if connections have sequencial phases. Formally, $F(\mathrm{EQUI}(\tilde{\mathcal{C}}, B')) \le (2+\frac{4}{\varepsilon})F(\mathrm{OPT}(\tilde{\mathcal{C}}, B'/(2+\varepsilon)))$.*

Now we consider the heterogeneous case where each connection has an acceleration within a range $\alpha_{\min} \le \alpha_i \le \alpha_{\max}$. We consider Proportional Partition (PROP) instead of EQUI in heterogeneous case. PROP allocates the bandwidth to connections in proportion to their accelerations. Formally, PROP allocates $B\alpha_i / \sum_{C_{i'} \in \mathrm{Alive}(t)} \alpha_{i'}$ bandwidth to $C_i \in \mathrm{Alive}(t)$ at time t.

Theorem 2 is extended to heterogeneous case naturally.

Theorem 4. *Let t be the time when the connection C_i drops for the jth time in AIMD. Then, $r_i[t-0] \ge (1-(1-\beta)^{j-1})B\alpha_i / \sum_{C_{i'} \in \mathrm{Alive}(t)} \alpha_{i'}$.*

To prove an upper bound corresponding to Theorem 1 in heterogeneous case, the competitive analysis of PROP is required.

Theorem 5. *Let $s > 1$. Assume that there exists a function $\mathcal{R}(s')$ such that for any $s' < s$ closer enough to s and any set $\mathcal{C}$ of connections, $F\big(\mathrm{EQUI}(\mathcal{C}, B')\big) \le \mathcal{R}(s') F\big(\mathrm{OPT}(\mathcal{C}, B'/s')\big)$. Then for any set $\mathcal{C}$ of connections, $F\big(\mathrm{PROP}(\mathcal{C}, B')\big) \le \lambda\Big(\lim_{s' \to s-0} \mathcal{R}(s')\Big) F\big(\mathrm{OPT}(\mathcal{C}, B'/s)\big)$, where $\lambda = \alpha_{\max}/\alpha_{\min}$.*

Proof. Let $\mathcal{C}$ be a set of connections. For any $\alpha > 0$, we define a homogeneous set $\mathcal{C}_\alpha$ from $\mathcal{C}$ by dividing each connection $C_i \in \mathcal{C}$ into $\lceil \alpha_i/\alpha \rceil$ equal connections with arrival time a_i and data length $l_i/\lceil \alpha_i/\alpha \rceil$.

For $k = 1, 2, \ldots$, consider the set $\mathcal{C}_{\alpha_{\min}/k}$. Then $k \le \lceil k\alpha_i/\alpha_{\min} \rceil \le \lceil k\lambda \rceil$, thus $F\big(\mathrm{EQUI}(\mathcal{C}_{\alpha_{\min}/k}, kB'/(k+1))\big) \ge k\,F\big(\mathrm{PROP}(\mathcal{C}, B')\big)$ and $F\big(\mathrm{OPT}(\mathcal{C}_{\alpha_{\min}/k}, B')\big) \le \lceil k\lambda \rceil F\big(\mathrm{OPT}(\mathcal{C}, B')\big)$. They give

$$
\begin{aligned}
& F\big(\mathrm{PROP}(\mathcal{C}, B')\big) \\
&\le \frac{1}{k} F\left(\mathrm{EQUI}\Big(\mathcal{C}_{\alpha_{\min}/k}, \frac{k}{k+1} B'\Big)\right) \\
&\le \frac{1}{k} \mathcal{R}\big(sk/(k+1)\big) F\big(\mathrm{OPT}(\mathcal{C}_{\alpha_{\min}/k}, B'/s)\big) \\
&\le \frac{1}{k} \lceil k\lambda \rceil \mathcal{R}\big(sk/(k+1)\big) F\big(\mathrm{OPT}(\mathcal{C}, B'/s)\big) \\
&\le \Big(\lambda + \frac{1}{k}\Big) \mathcal{R}\big(sk/(k+1)\big) F\big(\mathrm{OPT}(\mathcal{C}, B'/s)\big).
\end{aligned}
$$

By considering the limit as $k \to \infty$, $F\big(\mathrm{PROP}(\mathcal{C}, B')\big) \le \lambda\Big(\lim_{s' \to s-0} \mathcal{R}(s')\Big) F\big(\mathrm{OPT}(\mathcal{C}, B'/s)\big)$. □

Theorem 5 states that any upper bound of the competitive ratio of EQUI is also applicable to PROP with an extra factor of λ. Combining Theorems 2 and 5 gives the following corollary.

Corollary 6. *Let $\varepsilon > 0$. Then for any $\mathcal{C}$, $F\big(\mathrm{PROP}(\mathcal{C}, B')\big) \le \lambda(2+\frac{4}{\varepsilon}) F\Big(\mathrm{OPT}\big(\mathcal{C}, B'/(2+\varepsilon)\big)\Big)$, where $\lambda = \alpha_{\max}/\alpha_{\min}$.*

Theorem 4 and Corollary 6 give the following extension of 1 to the heterogeneous case.

Theorem 7. *Let $\mathcal{C} = \{C_1, \ldots, C_n\}$ be a set of connections whose accelerations satisfy $\alpha_{\min} \le \alpha_i \le \alpha_{\max}$ for all i. Let $\varepsilon > 0$, and define s as in equation (1). Let $F(\mathrm{AIMD}(\mathcal{C}, B))$ be the total flow time achieved by the AIMD with bottleneck bandwidth B and $F(\mathrm{OPT}(\mathcal{C}, B/s))$ be that achieved by the optimal offline and centralized algorithm with bottleneck bandwidth B/s. Then, it holds that $\frac{F(\mathcal{C})}{F_{\mathrm{OPT}}(\mathcal{C})+D} \le \lambda\big(2+\frac{4}{\varepsilon}\big)$, where $D = 2(q+1)nB/\alpha_{\min}$.*

On the other hand, no lower bound has been proven for AIMD. Because Theorem 5 states AIMD behaves like PROP for long-lasting connections, here we assume that the total flow time of AIMD is at least as long as that of PROP with the same resource. [3] proves that for $s > 2$, the competitive ratio of EQUI with resource B' to OPT with resource B'/s is at least $2/s$. The example given there can be modified to give a lower bound for the competitive ratio of PROP, which is $2\lambda/s$ where $\lambda = \alpha_{\max}/\alpha_{\min}$. This lower bound is also λ times as large as the homogeneous case.

5 Analysis of asymptotic bandwidth utilization

In this section, we consider the case that $l_i = \infty$ for all i, that is, all the senders have infinite data to transmit and the connections never terminate. As discussed in the introduction, this is an approximation of the case that all the connections continue for a long time. Under this assumption, we analyze the asymptotic bandwidth utilization.

Let us introduce some notations. Let $A = \sum_{i=1}^n \alpha_i$. The transmission rate at time t is denoted by $r_i[t]$. In case a drop occurs at time t, we distinguish the transmission rate just before time t and just after time t as $r_i[t-0]$ and $r_i[t+0]$.

Let $\boldsymbol{r}[t] = (r_1[t], \ldots, r_n[t])^{\mathrm{T}}$ and $\boldsymbol{r}[t \pm 0] = (r_1[t \pm 0], \ldots, r_n[t \pm 0])^{\mathrm{T}}$.

For $t_1 \leq t_2$, the amount $W_i[t_1, t_2]$ of data transmitted in connection C_i between time t_1 and t_2 is

$$W_i[t_1, t_2] = \int_{t_1}^{t_2} r_i[t]\, dt,$$

and we let $W[t_1, t_2]$ be the total amount of data transmitted in n connections between the same period,

$$W[t_1, t_2] = \sum_{i=1}^{n} W_i[t_1, t_2] = \int_{t_1}^{t_2} (r_1[t] + \cdots + r_n[t])\, dt.$$

The (asymptotic) bandwidth utilization U_i of connection C_i and the (asymptotic) total utilization U are defined as the limit of time average of the proportion of transmission rate in available bandwidth[1]:

$$U_i = \frac{1}{B} \lim_{T\to\infty} \frac{W_i[0,T]}{T} \quad \text{and} \quad U = \frac{1}{B} \lim_{T\to\infty} \frac{W[0,T]}{T}.$$

A larger total utilization means the algorithm makes use of much bandwidth and that it is efficient. Besides, a small variation in the values of U_i means the algorithm is fair.

5.1 Total and individual utilizations in all-drop case

Theorem 8. *In the all-drop model, the total and individual utilizations are* $U = 1 - \frac{\beta}{2}$ *and* $U_i = \frac{\alpha_i}{A} U$.

This is proven by representing the utilization of every connection by n-vector and calculating the eigenvector of the matrix which represents the transformation from the utilization vector just before one drop to that just before the next drop.

Theorem 8 says that in the all-drop model, the total utilization does not depend on the number of connections. This is different than the empirical fact. In the following sections, we consider the single-drop model.

5.2 Total utilization in homogeneous single-drop case

In this section we consider the homogeneous single-drop case where $\alpha_1 = \cdots = \alpha_n = \alpha$.

Theorem 9. *In the homogeneous single-drop model, total bandwidth utilization* U *is* $U = \frac{(2-\beta)n}{(2-\beta)n+\beta}$ *regardless of how we choose victim of each drop.*

[1] U_i and U may not have limit values depending on the choice of victims. In such cases, U_i and U are not defined.

The proof uses a potential function

$$\varphi(\boldsymbol{r}) = \frac{1}{2\alpha} \cdot \frac{(2-\beta)\{B^2 - (B - \sum r_i)^2\} - \beta \sum r_i^2}{(2-\beta)n + \beta},$$

with which $W[0,T] + \varphi(\boldsymbol{r}[t]) = \frac{(2-\beta)n}{(2-\beta)n+\beta} BT$ can be proven.

Theorem 9 shows that in the single-drop model, the total utilization U increases as n increases, which means dividing data into multiple streams gives better total throughput. This is different from the case of the all-drop model.

5.3 Total and individual utilizations under Periodic victim policy

In this section, we consider Periodic victim policy as a typical example of a deterministic policy. This policy is similar to the all-drop model in that it chooses every connection C_i equal times.

Definition 1. *Periodic victim policy* is the policy where connection C_1 is chosen as victim of the first drop, C_2 of the next drop, then $C_3, \ldots, C_n$, and this process is repeated infinitely. An example is shown in Figure 2 (b).

Theorem 10. *Under Periodic victim policy, it holds*

$$U = \frac{2-\beta}{2 - \beta(1 - \sum_{i=1}^{n} (\alpha_i/A)^2)}, \quad U_i = U \cdot \frac{\alpha_i}{A}.$$

The proof is similar to that of Theorem 8.

Theorem 10 implies that under Periodic victim policy, the bandwidth is shared in proportion to α_i like the all-drop model, and α_i's with small deviation give better total utilization.

5.4 Upper and lower bounds of total utilization in heterogeneous single-drop case

In this section, we consider Priority victim policy, which is the most unfair policy in some sense. Figure 3 (a) illustrates this policy. Intuitively, Priority victim policy chooses the connection C_i with the largest i that has nonzero transmission rate as victim. However, this informal definition is not accurate because the transmission rates are always nonzero. Instead, we define Priority victim policy as follows.

Definition 2. Let $0 < \varepsilon < 1/n$. *ε-Priority victim policy* is the policy where on every drop, connection C_i with the largest i that satisfies $r_i \geq \varepsilon B$ is chosen as victim. *Priority victim policy* is the limit of ε-Priority policy as $\varepsilon \to 0$.

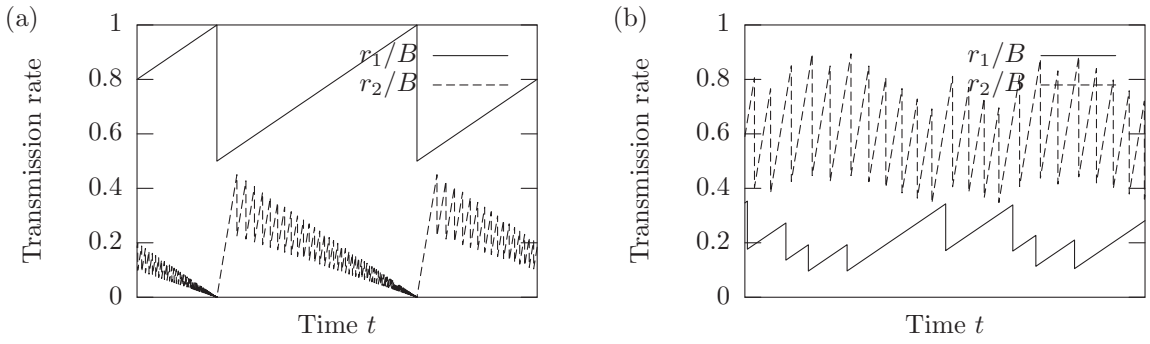

Figure 3: Time evolution of the transmission rates r_1 and r_2 of the two connections with the accelerations $\alpha_1 : \alpha_2 = 1 : 9$ and the drop factor $\beta = 1/2$, under (a) Priority victim policy and (b) Share-Random victim policy.

Theorem 11. *Let $A_0 = 0$ and $A_i = \alpha_1 + \cdots + \alpha_i$. Under Priority victim policy,*

$$U = 1 - \prod_{i=1}^{n} \frac{(2-\beta)A_{i-1} + \beta\alpha_i}{(2-\beta)A_{i-1} + 2\alpha_i},$$

$$U_i = \frac{(2-\beta)\alpha_i}{(2-\beta)A_{i-1} + 2\alpha_i} \prod_{j=1}^{i-1} \frac{(2-\beta)A_{j-1} + \beta\alpha_j}{(2-\beta)A_{j-1} + 2\alpha_j}.$$

When $n = 2$, Priority victim policy gives the maximum and the minimum of the total utilization as the following theorem implies.

Theorem 12. *Let $n = 2$ and $\alpha_1 \le \alpha_2$. If the total utilization U converges to some value, it holds*

$$1 - \frac{\beta}{2} \cdot \frac{\beta\alpha_1 + (2-\beta)\alpha_2}{2\alpha_1 + (2-\beta)\alpha_2} \le U \le 1 - \frac{\beta}{2} \cdot \frac{(2-\beta)\alpha_1 + \beta\alpha_2}{(2-\beta)\alpha_1 + 2\alpha_2}.$$

The proof is obtained by potential function method similar to that used in the proof of Theorem 9.

This theorem indicates an interesting fact that as long as the total utilization is concerned, the router should discard the packet from the connection with the higher acceleration upon congestion. This strategy may also be useful to discourage the use of high acceleration by selfish connection, thus achieving high total utilization and penalty to selfish connection at the same time.

5.5 Simulation of two heterogeneous connections under Share-Random victim policy

Definition 3. *Share-Random victim policy* is the policy where on every drop, each C_i is chosen as victim with probability r_i/B, as shown in Figure 3 (b).

Share-Random victim policy is the policy which is most easily implemented by a router placed at the bottleneck. Provided all the packets are infinitesimally short and the same length, the number of packets received by the router for each connection C_i at some moment is in proportion to the transmission rate r_i. When the sum $\sum r_i$ exceeds the capacity B of the router, the router will discard one packet received at the moment, which is for the connection C_i with the probability r_i/B. This scenario assumed the drop-tail behavior of the router, but the same thing happens if the router uses the Random Early Detection (RED) [6] given the buffer in the router is small enough.

We performed the numerical simulation of the utilizations by two connections under Share-Random policy, with B and $\beta = 1/2$ fixed and α_1 and α_2 altered while maintaining $A = \alpha_1 + \alpha_2 = 1$. Figure 4 shows the total and individual utilizations in this case. From Figure 4 (b) and the results with other values of β, we conjecture the following.

Conjecture. In heterogeneous two-connection case under Share-Random victim policy, it holds

$$\mathrm{E}[U] = \left(1 - \frac{\beta}{2}\right)\left(1 + \frac{2\beta}{4-\beta} \cdot \frac{\sqrt{\alpha_1\alpha_2}}{A}\right).$$

In addition, Figure 4 (a) suggests that in Share-Random case, the sharing of bandwidth among the connections is closer to the fair sharing than the all-drop case and the single-drop Periodic case. It is nearly proportional to the square root of the acceleration, or inversely proportional to RTT. This can be interpreted that the Share-Random victim policy mitigates the unfairness caused by different accelerations by choosing the connection with higher throughput more often than the other connection.

References

[1] D.-M. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion

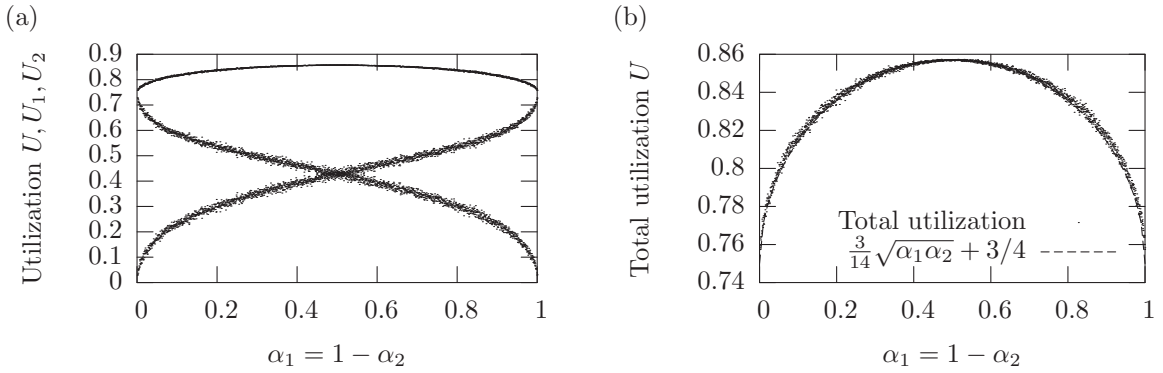

Figure 4: Total utilization U and the utilizations U_i by each connection C_i of two connections under Share-Random victim policy with $\beta = 1/2$, $B = 1$ and different values for α_1 and α_2, while keeping $\alpha_1 + \alpha_2 = 1$.

avoidance in computer networks. *Computer Networks and ISDN Systems*, 17(1):1–14, June 1989.

[2] A. De Vendictis and A. Baiocchi. Modeling a mixed TCP Vegas and TCP Reno scenario. In *Networking 2002: Proceedings of 2nd International IFIP-TC6 Networking Conference*, volume 2345 of *Lecture Notes in Computer Science*, pages 612–623, May 2002.

[3] J. Edmonds. Scheduling in the dark. *Theoretical Computer Science*, 235(1):109–141, Mar. 2000.

[4] J. Edmonds, S. Datta, and P. W. Dymond. TCP is competitive against a limited adversary. In *Proceedings of the Fifteenth Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 174–183, June 2003.

[5] S. Floyd. HighSpeed TCP for large congestion windows. Internet Draft (work in progress), Aug. 2003. `http://www.ietf.org/internet-drafts/draft-ietf-tsvwghighspeed-01.txt`.

[6] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, 1993.

[7] K. Hiraki, M. Inaba, J. Tamatsukuri, R. Kurusu, Y. Ikuta, H. Koga, and A. Zinzaki. Data Reservoir: Utilization of multi-gigabit backbone network for data-intensive research. In *Proceedings of the IEEE/ACM SC2002 Conference*, Nov. 2002.

[8] C. Jin, D. X. Wei, and S. H. Low. FAST TCP for high-speed long-distance networks. Internet Draft (work in progress), June 2003. `http://netlab.caltech.edu/pub/papers/draft-jwl-tcp-fast-01.txt`.

[9] R. M. Karp, E. Koutsoupias, C. H. Papadimitriou, and S. Shenker. Optimization problems in congestion control. In *41st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 66–74. IEEE Computer Society, 2000.

[10] T. Kelly. Scalable TCP: Improving performance in highspeed wide area networks. PFLDnet 2003, Feb. 2003. `http://datatag.web.cern.ch/datatag/pfldnet2003/papers/kelly.pdf`.

[11] T. V. Lakshman and U. Madhow. The performance of networks with high bandwidth-delay products and random loss. *IEEE/ACM Transactions on Networking*, 5(3):336–350, June 1997.

[12] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communication Review*, 27(3), July 1997.

[13] M. Nakamura, M. Inaba, and K. Hiraki. Fast Ethernet is sometimes faster than Gigabit Ethernet on LFN — observation of congestion control of TCP streams. In *Proceedings of the 15th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS)*. ACTA Press, Nov. 2003. To appear.

[14] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: A simple model and its empirical validation. In *Proceedings of the ACM SIGCOMM '98*, pages 303–314, 1988.