

Theoretical and Experimental Analysis of the SABUL Congestion Control Algorithm

Phoemphun Oothongsap, Yannis Viniotis, and Mladen Vouk
North Carolina State University, Raleigh NC 27606, USA

Abstract

Several new protocols such as RBUDP, User-Level UDP, Tsunami, and SABUL, have been proposed as alternatives to TCP for high speed data transfer. The purpose of this paper is to analyze the effects of SABUL congestion control algorithm on SABUL performance matrices such as bandwidth utilization, self-fairness, aggressiveness and average packet losses. We propose a deterministic model of SABUL congestion control algorithm and use the model to assess these metrics. Our results explain SABUL throughput oscillations, derive bounds on its aggressiveness/responsiveness, show that SABUL can be self-fair, and identify conditions under which SABUL may experience excessive packet losses. We also validate our model by doing experimental analysis.

I. INTRODUCTION

The high-performance networks being developed at present offer the promise of connectivity at speeds up to 40 Gbps or more. Such networks can enable new classes of high performance applications, such as remote data analysis/visualization and high performance grid-based computation. Although there is significant bandwidth available for such applications, the effective use of the available bandwidth is a challenge.

Several studies [7], [11], [2] (and references therein) have shown that, in practice, user-level distributed applications connected by high-speed networks (e.g., Abilene) cannot fully utilize the available bandwidth. The main reason for this subpar performance is the congestion control mechanism of the transport protocol (e.g., TCP). Thus, to improve bandwidth utilization, two alternatives are to (i) improve the performance of TCP, and, (ii) develop new transport protocols that are suitable for a high-delay but high-bandwidth environment.

Several studies have attempted to improve TCP performance by (i) adjusting TCP receiver buffer size [5], [19], (ii) using TCP with Selective Acknowledgment [12], (iii) using TCP parallel streams [18], (iv) increasing

TCP packet size [3], and (v) modifying TCP congestion control algorithm to adjust to highspeed environments [4], [11].

Examples of new hybrid protocols are RBUDP [10], User-level UDP [2], and SABUL [8], [9]. These protocols use UDP to transfer the data and TCP or UDP for signaling. RBUDP, User-level UDP and Tsunami [20] have no congestion control mechanism, while SABUL provides congestion control. SABUL has been evaluated empirically and by simulations [8], [9] quite extensively. Only limited theoretical studies related to SABUL performance are available [8].

The purpose of this paper is to study the effects of SABUL congestion control algorithm on performance matrices such as (instantaneous) bandwidth utilization, self-fairness, aggressiveness/responsiveness, and long-term average packet losses. The work presented here is built upon our earlier study of SABUL properties [13] [14] [15]. The study is based on the simplified model of SABUL's congestion control algorithm. The model is sufficient to capture the effects of the algorithm parameters and can be used in selecting parameter values in order to control the algorithm behavior (and in particular its oscillations and aggressiveness).

The remainder of this paper is organized as follows. We present a brief overview of the SABUL protocol in the next section. In section III we formulate a deterministic model of SABUL behavior. We summarize our results in section IV. For clarity of presentation, the proof of these results is presented in the appendix.

II. A BRIEF OVERVIEW OF THE SABUL CONGESTION CONTROL ALGORITHM

Unlike TCP, which uses a window-based algorithm, SABUL employs a rate-based algorithm to adjust its sending rate as a response to congestion and/or traffic losses. SABUL rate control algorithm adjusts the “inter-packet gap” (i.e., the amount of time before two packets are sent to the network) in rounds.

A “round” in SABUL is defined as a fixed time interval T . The inter-packet gap is calculated at the beginning of a round and kept constant during the entire round. Let $\alpha \geq 0$ denote a target packet loss rate, the idea being that losses up to α are acceptable to the application. During the n^{th} round, i.e., during the time interval $[(n-1)T, nT)$, SABUL uses feedback from the receiver to measure $\rho(n)$, the actual packet loss rate in the n^{th} round. Based on this measurement, SABUL increases or decreases $\delta(n)$, the inter-packet gap to be used in the next round, as Equations 1 through 4 specify

$$\delta(n+1) = \delta(n) * (1 + k_1(\rho(n) - \alpha)) + c, \text{ if } \rho(n) > \alpha \quad (1)$$

$$\delta(n+1) = \delta(n) * (1 + k_2(\rho(n) - \alpha)), \text{ if } \rho(n) < \alpha \quad (2)$$

k_1, k_2 and c are positive constants to be selected by the implementor. Equation 1 specifies how the inter-packet gap is increased, as a result of “high losses”. Equation 2 specifies how the inter-packet gap is decreased, as

a result of “low losses.” We focus on these two equations in the analysis presented in this paper.¹ Based on the above equations, $\lambda(n)$, the instantaneous sending rate or throughput, throughout the n^{th} round, is simply given by

$$\lambda(n) = \frac{L}{\frac{L}{C} + \delta(n)} \quad (5)$$

In Equation 5, L denotes the length of the transmitted packets (assumed constant for simplicity) and C denotes the capacity of the sender’s network interface.

III. DETERMINISTIC ANALYSIS

As we have seen in section 2, SABUL rate control algorithm depends on multiple parameters. In this section we propose a simple, deterministic model to study qualitatively the behavior of the sending rate in Equation 5 and gain insight into how various parameters affect properties of the algorithm, such as oscillations, aggressiveness, responsiveness, fairness, and packet losses. To the best of our knowledge, while oscillation has been observed via simulations and experimentation [9], [8], no explicit study of the remaining metrics of SABUL has been previously reported.

A. Modeling assumptions

Consider a SABUL sender, connected to a high-speed network through an interface of capacity C Mbps. The sender has an infinite supply of packets of constant length L bits, which it transmits to a SABUL receiver, through a “bottleneck router”, as shown in Figure 1. Let μ denotes the service rate (bit/sec) a single SABUL connection receives at the bottleneck router. In this model, we assume the μ is constant. This will be the case, for example, when guaranteed-rate schedulers, such as any variant of Weighted Fair Queueing, are employed at the bottleneck router [16]. Moreover, we assume that the SABUL connection has a dedicated buffer of size K packets inside the router.

These assumptions enable us to utilize a deterministic model in order to analyze Equations 1 and 2. Despite its simplicity, the model captures the essential features of actual operation (see [13]).

¹To be more precise, SABUL uses two more equations, in its rate control algorithm:

$$\delta(n+1) = \delta(n) + d, \quad \text{if } \rho(n) = \alpha \quad (3)$$

$$\delta(n+1) = \delta(n) + b, \quad \text{if excessive packet are lost} \quad (4)$$

Equation 4 specifies how the inter-packet gap is increased in the case of “excessive” packet losses. Both equations 3 and 4 lead to a behavior similar to the one affected by Equation 1; we omit them from the analysis for simplicity.

B. Results

For simplicity of presentation, in this section, we say that SABUL operates in “decrease mode” (resp. “increase mode”) if, during the n^{th} round, it uses Equation 1 (resp. Equation 2) to increase (resp. decrease) the sending rate by adjusting the inter-packet gap.

The performance metrics of interest are grouped into two areas: transient and long-term average behavior. Transient behavior metrics include oscillations and aggressiveness. Long-term metrics include average packet losses and fairness. In Lemmas 1 through 6 we provide conditions on the algorithm parameters under which the algorithm oscillates for ever between the two modes or “locks” into the increase mode. The duration of oscillations is shown to be bounded in Lemmas 4 and 5. In Lemma 7, we provide conditions on the algorithm parameters under which the algorithm can allow excessive average packet losses. In Lemma 8, we provide conditions on the algorithm parameters under which the algorithm exhibits fairness properties. For a complete proof of these results, see [13].

Transient behavior of the algorithm Lemmas 1 to 5 describe conditions under which the SABUL sending rate will oscillate, with oscillation periods that are of bounded duration.

Lemma 1: Suppose that

$$C > \frac{\mu}{1 - \alpha} \quad (6)$$

and $k_2\alpha > 1$. As long as SABUL algorithm operates in the “increase mode”, the sequence of sending rates $\lambda(n), n = 1, 2, \dots$, is a *monotonically increasing* sequence; the sequence of losses $\rho(n), n = 1, 2, \dots$, is a *nondecreasing sequence*. Moreover, as

$$\lambda(n) \rightarrow_{n \rightarrow \infty} \frac{\mu}{1 - \alpha} \text{ and as } \rho(n) \rightarrow_{n \rightarrow \infty} \alpha. \quad (7)$$

Intuitively, condition 6 of Lemma 1 says that any sender interface capacity, C , greater than $\frac{\mu}{(1-\alpha)}$, is “high” enough to eventually lead to losses. The limiting value for the sending rate implies that SABUL will “capture” the available bottleneck link capacity and that losses *will* occur.

Lemma 2: Suppose that

$$C > \frac{\mu}{1 - \alpha} \quad (8)$$

As long as the SABUL algorithm operates in the “decrease mode”, the sequence of sending rates $\{\lambda(n), n = 1, 2, \dots\}$, is a *monotonically decreasing* sequence; the sequence of losses $\{\rho(n), n = 1, 2, \dots\}$, is also a *monotonically decreasing* sequence. Moreover, in the limit, we have

$$\lambda(n) \rightarrow_{n \rightarrow \infty} 0 \text{ and as } \rho(n) \rightarrow_{n \rightarrow \infty} 0. \quad (9)$$

The monotonicity property and the zero limit for losses stated in Lemma 2 imply that SABUL will not “lock” into the “decrease mode”.

Lemma 3: Suppose that the parameters of the rate control algorithm satisfy the conditions:

$$C > \frac{\mu}{1 - \alpha}, \quad k_2 \alpha < 1, \quad k_2 > \frac{C}{C - \mu} \quad (10)$$

Then, SABUL algorithm will *oscillate* between the “increase” and the “decrease” modes *forever*.

The condition $C > \mu/(1 - \alpha)$ allows the source to increase the sending rate as high as C . However, as the Lemma 3 states, SABUL will not be *overly aggressive*, since a period of increase will always be followed by a period of decrease and vice versa.

We next show that the oscillations between increase and decrease modes have upper bounds; in other words, the time intervals between the oscillations are *finite* bounded. Let $T_{2k+1}, k = 0, 1, 2, 3, \dots$, be the sequence of intervals in which SABUL operates under Equation 2. Let $T_{2k}, k = 1, 2, 3, \dots$, be the sequence of (finite) intervals in which SABUL operates under Equation 1. Let λ_{min} be the minimum value of the sending rate when SABUL switches from Equation 1 to Equation 2. Let T' be a (finite) interval of SABUL sending rate sequence when SABUL operates under Equation 2 with the initial value of the sending rate being equal to λ_{min} .

The following two Lemmas provide bounds for the duration of the increase and decrease modes of operation.

Lemma 4: $T_{2k+1} \leq \max(T_1, T')$; $\forall k = 0, 1, 2, 3, \dots$.

Lemma 5: $T_{2k} \leq T_2$; $\forall k = 1, 2, 3, \dots$.

The last Lemma in this section provides a sufficient condition for the algorithm to lock into a single mode of operation. Intuitively, the condition of the Lemma states that the interface capacity at the senders is “small” (or equivalently, the loss tolerance α is set very high).

Lemma 6: Suppose that

$$C < \frac{\mu}{1 - \alpha} \text{ or, equivalently, } \alpha > \frac{C - \mu}{C} \quad (11)$$

If SABUL starts in the “increase mode”, it will operate in the “increase mode” forever.

Long-term behavior The long-term performance metrics of interest are average packet loss and fairness.

We define the average packet loss γ as

$$\gamma = \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{i=1}^k \rho(i) \quad (12)$$

We study fairness between SABUL and connections only, i.e., the TCP-friendliness of the protocol is not analyzed.

Lemma 7 states that average losses can be higher than the parameter α . This result may be surprising at a first glance. Note, however, that the control equations use the instantaneous losses only. The average loss is not directly controllable by SABUL.

Lemma 7: Suppose that the parameters of the rate control algorithm satisfy the inequalities

$$c \leq (1 + k_1)k_2\alpha\left(\frac{L}{\mu} - \frac{L}{C}\right) - k_1\left(\frac{L}{\mu} - \frac{L}{C}\right)$$

$$(1 + k_1)k_2\alpha > k_1, k_2\alpha < 1, k_2 > 2, k_2\left(1 - \frac{\mu}{C}\right) \geq 2$$

Then the average packet loss rate γ in Equation 12 exceeds α .

We next provide (sufficient) conditions under which two SABUL connections are fair to each other.

Lemma 8: Consider two SABUL connections that have the same bottleneck link. Suppose that each connection has the same rate control interval, T , and the same parameters k_1, k_2, α, c . Then both connections will eventually achieve the same throughput, i.e.,

$$\lim_{n \rightarrow \infty} \lambda_1(n) = \lim_{n \rightarrow \infty} \lambda_2(n)$$

IV. EXPERIMENTS

A. Experimental Setup

To validate the mathematical model and understand the general behaviors of SABUL, experiments were performed in two environments: (i) a private local area network where the round trip time (RTT) is in microseconds (a short-haul network), (ii) Abilene network where the round trip time (RTT) is in milliseconds (a long-haul network). The preliminary results are shown in [15].

Figure 3 represents the experimental setup for the private local area network. In a local private network, we distinguish "slow" and "fast" end-clients (machines). The model of the "slow" machines is IBM Netfinity 4000R server (CPU speed of 700 MHz, 1 GB of memory, 1 Gbps Ethernet fiber adapter). The model of the "fast" machines is IBM eServer x335 (CPU speed is 1 GHz, 1.5 GB of memory, 1Gbps Ethernet fiber adapter). All machines are running Linux. They were interconnected via an Extreme Networks 6800 series Blackdiamond, 10Gb switch.

Figure 4 shows the network setup for the long-haul experiments. End hosts are located at three different Abilene end-point locations: North Carolina State University (NCSU), Georgia Institute of Technology (GT) and University of Washington (UW). At NCSU, there is one machine named localhost. At GT, there are three machines named fast1, fast2, and fast3. At UW, there is one machine named fasttcp. These machines run Linux operating system (Kernel version 2.4.18) with CPU speeds of 3 GHz and 2 GB of memory. Each machine is equipped with 1-gigabit Ethernet card. These machines are connected to each other through Abilene backbone network. The effective point-to-point capacity of each link between Abilene end-points used in these experiments is 2.4 Gb/s. Then the bottleneck link in the path is 1Gb/s, to the end-host.

B. Experimental Results

The purpose of this set of experiments was to study SABUL self-fairness, bandwidth utilization, and factors affecting these properties. Multiple SABUL connections were studied in both short- and long-haul networks. We emphasize the long-haul network part because the main purpose of SABUL is to aid file transfer in high-speed long RTT networks.²

Table I shows bandwidth utilization of two SABUL connections in different network environments. The first and second columns represent the RTT from $Source_i$ to destination. The third and fourth columns represent the initial sending rate of each connection in Mb/s. The fifth and sixth columns represent the rate control interval (round length) of each connection in msec. The seventh and eighth columns represent the first and second set of congestion control parameters. The first set (Pr_0) of congestion control parameters is the default value of SABUL protocol parameters ($k_1 = 0.1, k_2 = 10, c = 0.5 * 10^{-6}, d = 0.1 * 10^{-6}, b = 2 * 10^{-6}$). The second set (Pr_1) of congestion control parameters is set according to the conditions in Lemma 7 ($k_1 = 0.00101, k_2 = 100, c = 0.002 * 10^{-6}, d = 0.1 * 10^{-6}, b = 0.002 * 10^{-6}$). The ninth and tenth columns represent the average sending rate of each connection in Mb/s and the last column represents the figure showing the instantaneous sending rate of each experiment. The results in this table are from memory to memory transfers. The sum of the average sending rate of each experiment is less than 1 Gb/s, since each experiment has been done on the public network and there are two connections. Thus, each connection will experience packet losses and they need to reduce the sending rate. Then SABUL instantaneous sending rate will show a saw tooth pattern as shown in Figures 5 and 6. Therefore, the sum of the average sending rate will be less than 1 Gb/s.

Table II shows bandwidth utilization of three SABUL connections in various network environment. The results in this table are also from memory to memory transfers.

Table I shows the two connections compete on the same bottleneck link. The results show that SABUL connections may or may not be fair to each other. In table I, we can categorize the experiments into four cases: (i) same RTT and rate control interval, (ii) different RTT and same rate control interval, (iii) different RTT and different rate control interval, and (iv) different congestion control parameters. We notice that both connections get the similar average sending rate when rate control interval is the same regardless of RTT and initial sending rate. Both connections show an unfairness behavior when the rate control interval of both connections are different. This behavior can be explained as follows. SABUL sender recalculates a new sending rate every time it receives a SYN packet from the receiver and the receiver generates a SYN packet every

²The results shown in this paper express the general behavior of the SABUL congestion control algorithm. However, the results may vary if network environments are not the same.

TABLE I
AVERAGE SENDING RATE OF TWO SABUL CONNECTIONS

RTT_1	RTT_2	$Init_1$	$Init_2$	T_1	T_2	$Para_1$	$Para_2$	$Rate_1$	$Rate_2$	Figure
57.5	23.3	280	360	200	200	Pr_0	Pr_0	440	435	5
57.5	81	280	365	200	200	Pr_0	Pr_0	425	425	6
23.3	23.2	363	280	400	200	Pr_0	Pr_0	400	500	7
57	23	382	288	200	400	Pr_0	Pr_0	500	350	8
57.5	81	292	388	400	200	Pr_0	Pr_0	390	500	9
57.5	57.5	380	275	200	200	Pr_0	Pr_0	425	425	10
57.5	81	300	120	200	200	Pr_0	Pr_1	50	480	11
0.204	0.204	60	275	200	200	Pr_0	Pr_1	10	560	12

TABLE II
AVERAGE SENDING RATE OF THREE SABUL CONNECTIONS

RTT_1	RTT_2	RTT_3	$Init_1$	$Init_2$	$Init_3$	T_1	T_2	T_3	$Rate_1$	$Rate_2$	$Rate_3$	Figure
0.204	0.204	0.203	320	260	210	200	200	200	320	320	320	13
0.204	0.204	0.203	320	260	210	600	400	200	295	310	361	14
23.3	23.2	23.3	293	291	279	200	200	200	270	270	270	15
57.0	57.5	57	298	293	258	200	200	200	298	310	295	16
23.3	23.2	57.5	273	282	286	200	200	200	220	280	300	17

constant rate control interval. For the connection having a short rate control interval, the sender will receive a signal to increase a sending rate more often than the connection having a longer rate control interval. Moreover, SABUL congestion control is a variant of Multiplicative Increase and Multiplicative Decrease algorithm. The sender increases sending rate aggressively. Then the connection with a short rate control interval increases the sending rate more aggressively than the connection with a longer rate control interval, causing unfairness. Even though the connection with a short rate control interval sees the number of packet losses larger than the one with a long rate control interval (thus the one with a short rate control interval will reduce the sending rate severely), with multiplicative increase algorithm, the connection with a short rate control interval will increase its sending rate aggressively. Then on the average sending rate, the connection with a short rate control interval will get higher throughput than the one with a long rate control interval as we show in Figures 8 and 9. Moreover, the initial sending rate of each connection has no impact on SABUL fairness as long as both

connections have the same rate control interval as we show in Figures 5 and 6. This is because the connection having the high sending rate may experience more packet losses than the one having the low sending rate. Thus the connection having the high sending rate will reduce the sending rate more aggressively than another connection. With this behavior, the low sending rate connection will have more room to increase its sending rate and eventually, both connections will converge to same sending rate. Moreover, another source of SABUL unfairness is protocol congestion control parameters. The last two experiments (as shown in Figures 11 and 12) in this set show that the connection with the second set of protocol parameters achieve higher sending rate the one with default value of protocol parameters (during the experiments, we run UDP with rate 420 Mb/s as a background traffic to help expressing the unfairness of both connections). This is because the connection with parameter set Pr_1 causes a SABUL sender to decrease its sending rate less aggressively than the one with parameter set Pr_0 . Moreover, Pr_1 does not reduce the sending rate to a value that is small enough for the bottleneck queue to be drained, while the connection with Pr_0 suffers a large amount of losses and most of its packets are dropped at the bottleneck queue (this leads to throughput losses).

Figures 5 to 12 show the instantaneous sending rate of each experiment. The x-axis represents the experimental time in seconds and the y-axis represents SABUL instantaneous throughput in Mbits/sec. In figures 5 to 10, we notice that SABUL still maintains the oscillation property. And also, we notice the “synchronized” behavior, i.e., sources oscillate in phase. Synchronized behavior is an unpleasant behavior since it can reduce the overall throughput of the system. Synchronized behavior occurs due to the drop-tail operations at the router, and the round trip time (RTT) effect. With drop-tail routers, each congestion period introduces global synchronization in the network as noted in [6]. When the queues overflow, packets from several connections are dropped and these connections decrease their sending rate at the same time. The consequence is loss of throughput at the router. The effect of the RTT on the send rate fluctuation was already mentioned.

The results in Table II are explained in a fashion similar to the results in Table I, except that we did not show the the effect of congestion control parameters due to space limitations. Once again, this behavior shows that SABUL average sending rate depends primarily on the employed rate control interval and not RTT. RTT has no effect on the self-fairness of this protocol, but rate control interval does. As would be expected, the synchronization behavior still occurs The behavior is clearly shown in figures 15 to 17. Even though it is not apparent in figures 13 and 14 due to the log scale, the behavior is still the same.

V. CONCLUSION

In this paper, we have analyzed transient and long-term average SABUL performance by using a simple deterministic model for the rate control algorithm. The metrics of interest include oscillatory behavior, aggressiveness, long-term average packet losses and fairness. The results provide conditions on the algorithm

parameters under which the algorithm can exhibit oscillatory/locking behavior, long-term losses above the desired tolerance and fairness. The model can be used as a guideline for selecting values for the algorithm's parameters in practice.

Moreover, we validated our model by doing the experimental analysis. The results show how bandwidth utilization is affected by round trip time (RTT), rate control interval (T), and protocol parameters. In addition, the experimental results reveal that SABUL self-fairness property depends heavily on rate control interval (T), and protocol parameters, while RTT has no effect on SABUL self-fairness.

VI. APPENDIX

In this appendix, we present proofs of lemmas 1, 4, 7, and 8. Proofs for the other lemmas are similar and presented in [13].

Proof of Lemma 1. Consider the n^{th} round. Since $\rho(n) - \alpha < 0$, we have that: $\delta(n+1) = \delta(n)(1 + k_2(\rho(n) - \alpha)) < \delta(n)$,

and thus from Equation 5, $\lambda(n+1) = \frac{L}{\frac{L}{C} + \delta(n+1)} > \frac{L}{\frac{L}{C} + \delta(n)} = \lambda(n)$, establishing the monotonically increasing property of the sequence $\{\lambda(n), n = 1, 2, \dots\}$.

Suppose next that SABUL *always* operates under Equation 2. Then the sequence $\{\lambda(n), n = 1, 2, \dots\}$ will converge (from below) to a rate $\lambda(\infty) \triangleq \lambda$.

We will show first that the limiting rate exceeds the capacity μ of the bottleneck link, i.e., that $\lambda = \lim_{n \rightarrow \infty} \lambda(n) > \mu$.

Suppose, by contradiction, that $\lambda \leq \mu$. Then, $\forall n$, we have that $\lambda(n) \leq \mu$, and thus $\rho(n) = 0$. From Equation 2, we have that

$$\delta(n) = \delta(n-1)(1 + k_2(\rho(n-1) - \alpha)) = \delta(0)(1 - k_2\alpha)^n.$$

Therefore, $\lim_{n \rightarrow \infty} \delta(n) \rightarrow 0$ and thus from Equation 5, $\lambda = C$, which exceeds the capacity μ of the bottleneck link by virtue of inequality 6. We will show next that the limiting rate λ is exactly equal to $\frac{\mu}{1-\alpha}$. Suppose, first, by contradiction, that $\lambda > \frac{\mu}{1-\alpha}$; this implies that

$$1 - \frac{\mu}{\lambda} > \alpha. \quad (13)$$

We can easily show that, $\forall n$ $\rho(n) = \max(0, 1 - \frac{\mu}{\lambda(n)})$. Therefore, the sequence $\{\rho(n), n = 1, 2, \dots\}$ is also monotonically increasing and thus

$$\rho \triangleq \lim_{n \rightarrow \infty} \rho(n) = 1 - \frac{\mu}{\lambda}.$$

From Equation 13, $\rho > \alpha$, and thus for sufficiently large n , $\rho(n) > \alpha$, which contradicts Equation 2.

Suppose, next, by contradiction, that $\lambda < \frac{\mu}{1-\alpha}$. Then $1 - \frac{\mu}{\lambda} = \bar{\alpha} < \alpha$, and thus

$$\rho = \bar{\alpha} < \alpha. \quad (14)$$

Inequality 14 implies that there exists a sufficiently large integer n_0 such that for all $n > n_0$, we have that

$$|\rho(n) - \alpha| > \epsilon,$$

where $\epsilon > 0$ does not depend on n . In other words, the above equation and Equation 2 imply that for all $n > n_0$,

$$\delta(n+1) = \delta(n)(1 + k_2(\rho(n) - \alpha)) < \delta(n)(1 - k_2\epsilon)$$

and thus

$$\lim_{k \rightarrow \infty} \delta(k + n_0) \leq \delta(n_0) \lim_{k \rightarrow \infty} (1 - k_2\epsilon)^k = 0,$$

which in turn implies that $\lambda = C > \frac{\mu}{1-\alpha}$, again a contradiction, by virtue of inequality 6. ■

Proof of Lemma 4. To avoid trivialities, suppose that $\lambda(0) < \mu$ and $\rho(0) < \alpha$, with the sender algorithm starting in increase mode. The sequence of sending rates is a monotonically increasing sequence. Then

$$\lambda(0) = \frac{L}{\frac{L}{C} + \delta(0)} < \lambda(1) = \frac{L}{\frac{L}{C} + \delta(0)(1 - k_2\alpha)}, < \dots < \lambda(n) = \frac{L}{\frac{L}{C} + \delta(0)(1 - k_2\alpha)^n}$$

This implies that it will take n intervals to change from $\lambda(0) < \mu$ to $\lambda(n) > \mu$.

This implies that it will take a finite number, n , of rounds to increase the sending rate from $\lambda(0) < \mu$ to $\lambda(n) > \mu$. The value of n can be determined from the equation:

$$\frac{L}{\frac{L}{C} + \delta(0)(1 - k_2\alpha)^n} \geq \mu$$

or

$$n \geq \left\lceil \frac{\log \frac{1}{\delta(0)} \left(\frac{L}{\mu} - \frac{L}{C} \right)}{\log(1 - k_2\alpha)} \right\rceil \quad (15)$$

Up until this n^{th} round, the connection's queue at the bottleneck has no effect in it. Next, we determine a lower bound to the number of rounds it will take SABUL to fill up the bottleneck queue. Toward this end, assume that the initial queue size is zero. Then, beginning with the $(n+1)^{\text{st}}$ round, the queue will start building up. It will take a finite number, l , of additional rounds, to fill up the queue and have losses greater than α .

Suppose SABUL sender is at the “increase mode”, $\rho(n) < \alpha$. Then

$$\delta(n+1) = \delta(n)(1 + k_2(\rho(n) - \alpha))$$

and

$$\lambda(n+1) = \frac{L}{\frac{L}{C} + \delta(n+1)}$$

Suppose $\mu \leq \lambda(n) < \frac{\mu}{1-\alpha}$, $\forall n$. Then $\rho(n) = 0$ and

$$\delta(n+1) = \delta(n)(1 - k_2\alpha) \quad \text{and} \quad \lambda(n+1) = \frac{L}{\frac{L}{C} + \delta(n)(1 - k_2\alpha)}$$

Under the conditions in Lemma 3, $k_2 > \frac{C}{C-\mu}$, and $k_2\alpha < 1$, then we have

$$\begin{aligned} \delta(n+1) &< \delta(n)\left(1 - \frac{C\alpha}{C-\mu}\right) \\ \lambda(n+1) &> \frac{L}{\frac{L}{C} + \delta(n)\left(1 - \frac{C\alpha}{C-\mu}\right)} \end{aligned} \quad (16)$$

Since $\lambda(n) \geq \mu$ then

$$\delta(n) \leq \frac{L}{\mu} - \frac{L}{C} \quad (17)$$

Substitute Equation 17 to 16, then

$$\begin{aligned} \lambda(n+1) &\geq \frac{L}{\frac{L}{C} + \left(\frac{L}{\mu} - \frac{L}{C}\right)\left(1 - \frac{C\alpha}{C-\mu}\right)} \\ &= \frac{1}{\frac{1}{C} + \left(\frac{C-\mu}{C\mu}\right)\left(\frac{C-\mu-C\alpha}{C-\mu}\right)} \\ &= \frac{\mu}{1-\alpha} \end{aligned} \quad (18)$$

However, $\rho(n+1)$ may or may not be greater than α . This depends on the number of bits left in the bottleneck queue. Suppose the number of bits left in the queue at round $(n+1)$ is less than KL , where K is the queue size. Then $\rho(n+1) < \alpha$. Thus

$$\delta(n+2) = \delta(n+1)(1 - k_2\alpha) < \delta(n+1)$$

and

$$\lambda(n+2) = \frac{L}{\frac{L}{C} + \delta(n+1)(1 - k_2\alpha)} > \lambda(n+1)$$

By induction, $\lambda(n+1) < \lambda(n+2) < \dots < \lambda(n+l) < \dots$. Since the buffer size is finite, then it will take finite interval to fill up the bottleneck queue. Suppose it will take l intervals to fill up the bottleneck queue. Then the number of bits left in the bottleneck queue at round $(n+l)$ is the following

$$l * T(\lambda(n+1) - \mu) > KL$$

We know that at the $(n+l)^{th}$ round, we must have $\lambda(n+l) > \mu$. This implies that

$$\frac{L}{C} + \delta(n+l) < \frac{L}{\mu}$$

Then we can calculate the loss rate in the following

$$\rho(n+l) = 1 - \frac{\mu}{\lambda(n+l)}$$

where $\lambda(n+l) > \frac{\mu}{1-\alpha}$. Then

$$\rho(n+l) > 1 - \frac{\mu}{\frac{\mu}{1-\alpha}} = \alpha$$

We can determine l as the smallest integer that satisfies the inequality:

$$l * T(\lambda(n+1) - \mu) > KL$$

(We can easily derive this inequality from a D/D/1/K queue with arrival rate $\lambda(n+1)$ and service rate μ .)

Since $\lambda(n+1) > \frac{\frac{L}{C} + (\frac{L}{\mu} - \frac{L}{C})(1-k_2\alpha)}{C - Ck_2\alpha + \mu k_2\alpha}$, we have

$$l * T\left(\frac{C\mu}{C - Ck_2\alpha + k_2\mu\alpha} - \mu\right) > KL$$

$$l \geq \frac{KL(C - Ck_2\alpha + k_2\mu\alpha)}{T(Ck_2\alpha\mu - k_2\alpha\mu^2)}$$

Thus the duration T_1 of the first increase mode (i.e., the time to switch from Equation 2 to 1 cannot exceed $n+l$ rounds.

Note that T_1 is not an upper bound to the duration of subsequent rounds, since the starting rate at such rounds can be smaller than $\lambda(n_0)$, the initial rate in round 1. In order to bound the duration of subsequent increase modes, we will estimate next the value of λ_{min} . From Equation 1, we have

$$\delta(n+1) = \delta(n)(1 + k_1(\rho(n) - \alpha)) + c$$

and $\delta(n+1)$ will attain its maximum value if $\rho(n) = 1 - \frac{\mu}{C}$. Let λ_{max} denote the maximum value that the send rate can achieve during and increase period. Clearly, $\lambda_{max} \leq C$. Then λ_{min} can be calculated in two cases: (i) when the value of λ_{max} is less than the link interface capacity C but greater than μ , and (ii) when the value of λ_{max} is equal to the interface link capacity C .

If the value of λ_{max} is less than the link interface capacity C , then for the next round SABUL will reduce the sending rate with $c > \frac{L\alpha}{\mu} - \frac{Lk_1}{\mu} + \frac{Lk_1\alpha}{\mu} + \frac{Lk_1}{C}$ where $k_1 > 0$. Then we can rewrite Equation 1 as follow

$$\delta_{max} = \delta(n+1) = \delta(n)(1 + k_1(\rho(n) - \alpha)) + \frac{L\alpha}{\mu} - \frac{Lk_1}{\mu} + \frac{Lk_1\alpha}{\mu} + \frac{Lk_1}{C}$$

However, if $\lambda(n) > \frac{\mu}{1-\alpha}$, SABUL will switch the operation from Equation 1 to Equation 2. Then minimum value of $\lambda(n)$ for SABUL to switch the operation is equal to $\frac{\mu}{1-\alpha}$. Then

$$\lambda_{min} = \frac{L}{\frac{L}{C} + \delta(n)(1 + k_1(1 - \frac{\mu}{C} - \alpha)) + \frac{L\alpha}{\mu} - \frac{Lk_1}{\mu} + \frac{Lk_1\alpha}{\mu} + \frac{Lk_1}{C}}$$

Substitute $\delta(n)$ with $\frac{L(1-\alpha)}{\mu} - \frac{L}{C}$. Then

$$\lambda_{min_1} = \frac{C^2\mu}{C^2 - k_1C\mu + k_1C\mu\alpha + k_1\mu\alpha - k_1C^2\alpha + k_1\alpha^2C^2 + k_1C\mu\alpha}$$

If the value of λ_{max} is equal to the link interface capacity C , then for the next interval SABUL will reduce the sending rate with $c \geq \frac{L}{\mu} - \frac{L}{C}$ where $k_1 > 0$. Then

$$\lambda_{min_2} = \frac{C^2\mu}{(C^2(1-\alpha) - C\mu)(C + k_1(C(1-\alpha) - \mu)) + C^2}$$

and thus we can take $\lambda_{min} = \min(\lambda_{min_1}, \lambda_{min_2})$. Since by assumption $\lambda_{min} < \mu$, then SABUL will take n' rounds to increase to send rate from λ_{min} to μ , where n' can be determined from the equality

$$n' = \lceil \frac{\log \frac{1}{\delta_{(min)}} (\frac{L}{\mu} - \frac{L}{C})}{\log(1 - k_2\alpha)} \rceil$$

and $\delta_{min} = \frac{L}{\lambda_{min}} - \frac{L}{C}$. Then the duration T' of this round is bounded by $n' + l$. Thus, in general, $T_{2k+1} \leq \max(T_1, T')$. ■

Proof of Lemma 7. Suppose SABUL performs under “increase mode”. Then there exists a sufficiently large integer n_0 such that during the n_0^{th} round, we have $\lambda(n_0) > \mu$, $\rho(n_0) > \alpha$, the bottleneck queue is full, and $\delta(n_0) \leq (\frac{L}{\mu} - \frac{L}{C})(1 - k_2\alpha)$. During the next round, the new value of $\delta(n_0 + 1)$ is calculated via the equation:

$$\delta(n_0 + 1) = \delta(n_0)(1 + k_1(\rho(n_0) - \alpha)) + c \quad (19)$$

$$\leq \delta(n_0)(1 + k_1) + c \quad (20)$$

$$\leq (\frac{L}{\mu} - \frac{L}{C})(1 - k_2\alpha)(1 + k_1) + c$$

Then

$$\lambda(n_0 + 1) \geq \frac{L}{\frac{L}{C} + (\frac{L}{\mu} - \frac{L}{C})(1 - k_2\alpha)(1 + k_1) + c} \quad (21)$$

Since $c \leq (1 + k_1)k_2\alpha(\frac{L}{\mu} - \frac{L}{C}) - k_1(\frac{L}{\mu} - \frac{L}{C})$, then

$$\lambda(n_0 + 1) \geq \frac{L}{\frac{L}{C} + (\frac{L}{\mu} - \frac{L}{C})(1 - k_2\alpha)(1 + k_1) + (1 + k_1)k_2\alpha(\frac{L}{\mu} - \frac{L}{C}) - k_1(\frac{L}{\mu} - \frac{L}{C})} \quad (22)$$

$$\begin{aligned} &= \frac{L}{\frac{L}{C} + \frac{L}{\mu} - \frac{L}{C}} \\ &= \mu \end{aligned} \quad (23)$$

Suppose $\lambda(n_0 + 1) = \mu$; then $\rho(n_0 + 1) = 0$ and $\delta(n_0 + 1) = \frac{L}{\mu} - \frac{L}{C}$. In the next round, $(n_0 + 2)$, the sender will increase the sending rate. Then

$$\delta(n_0 + 2) = \delta(n_0 + 1)(1 - k_2\alpha) = (\frac{L}{\mu} - \frac{L}{C})(1 - k_2\alpha) \quad (24)$$

$$\lambda(n_0 + 2) = \frac{L}{\frac{L}{C} + (\frac{L}{\mu} - \frac{L}{C})(1 - k_2\alpha)} = \frac{\mu C}{C - Ck_2\alpha + k_2\mu\alpha}$$

Note that the bottleneck queue is still full, then

$$\rho(n_0 + 2) = 1 - \frac{\mu}{\lambda(n_0 + 2)} = 1 - \frac{\mu}{C - Ck_2\alpha + k_2\mu\alpha} = k_2\alpha - \frac{\mu}{C}k_2\alpha = k_2\alpha(1 - \frac{\mu}{C}) \quad (25)$$

since $k_2(1 - \frac{\mu}{C}) \geq 2$, then

$$\rho(n_0 + 2) \geq k_2\alpha > \alpha \quad (26)$$

At the $(n_0 + 3)^{th}$ round, SABUL sender will decrease the sending rate. Then

$$\delta(n_0 + 3) = \delta(n_0 + 2)(1 + k_1(\rho(n_0 + 2) - \alpha)) + c \leq \delta(n_0 + 2)(1 + k_1) + c$$

Since $\delta(n_0 + 2) = (\frac{L}{\mu} - \frac{L}{C})(1 - k_2\alpha)$, we can easily see that

$$\lambda(n_0 + 3) \geq \mu$$

If $\lambda(n_0 + 3) = \mu$, the steps from Equation 24 to 26 are repeated. Then by induction, we then have

$$\rho(n_0 + i) < \alpha \quad \forall i, i = 0, 2, 4, \dots$$

$$\rho(n_0 + j) > \alpha \quad \forall j, j = 1, 3, 5, \dots$$

and

$$\gamma = \lim_{k \rightarrow \infty} \frac{1}{k} \left(\sum_{i=1}^{n_0-1} \rho(i) + \sum_{i=n_0}^{\infty} \rho(i) \right) > \frac{\rho(n_0 + 1) + \rho(n_0 + 2)}{2} \geq \frac{2\alpha + 0}{2} = \alpha$$

We then have that $\gamma > \alpha$.

If $\mu < \lambda(n_0 + 1) \leq \frac{\mu}{1-\alpha}$ and $\rho(n_0 + 1) < \alpha$, the previous steps are repeated. The case when $\lambda(n_0 + 1) > \frac{\mu}{1-\alpha}$ and $\rho(n_0 + 1) > \alpha$ is similar and thus will not be discussed here. ■

Proof of Lemma 8. We will demonstrate the fairness property by using a “phase space” technique similar to that in [1]. Consider the $(\delta_1(n), \delta_2(n))$ plane shown in figure 2. A point in this plane represents the “state” of the two senders at the beginning of round n . All points on the $\delta_1(n) = \delta_2(n)$ line represent fair systems, since the rates of the two senders are equal (hence this line is labeled the fairness line). Points “above” the fair line represent systems in which sender 1 has a larger sending rate than sender 2. The (nonlinear) curve represented by the equation $\frac{\delta_1(n) * \delta_2(n)}{\delta_1(n) + \delta_2(n)} = constant$ separates the state space into an “overload” and an “underload” region.

Under the conditions of the lemma, both senders receive the same congestion indication, in a synchronized manner. They both adjust their inter-packet gaps according to the control equations in 1 and 2, leading to

typical sample paths in this phase space as exemplified in figure 2. We can summarize how the control equations behave as follows:

(1). AIAD (Additive Increase and Additive Decrease): the phase plot will move parallel to the $\delta_1 = \delta_2$ line, since both senders increase or decrease their $\delta_i(n)$ by the same amount.

(2). MIMD (Multiplication Increase and Multiplicative Decrease): the phase plot will move along the line joining (δ_1, δ_2) to the origin, since both senders increase or decrease their $\delta_i(n)$ by the same amount.

Suppose both senders start in the “increase mode” with an “unfair” initial condition $\lambda_1(0) > \lambda_2(0)$ (which implies that $\delta_1(0) < \delta_2(0)$) and $\sum_{i=1}^2 \lambda_i(0) < \frac{\mu}{1-\alpha}$ (i.e, point A in the uploaded region). Then both senders will increase their sending rate for a finite number of rounds, until $\sum_{i=1}^2 \lambda_i(n) > \frac{\mu}{1-\alpha}$ (i.e, until point A moves into point B in the overloaded region).

After losses occur, both senders will decrease their sending rate (in a typical sample path of this behavior, the phase plot will move to point C). As we can see, the fairness of both connections has been improved when the sending rate is decreased. For the decreasing algorithm, the additive portion is the key to “pushing” the system closer to fairness line. Note that, under the conditions of Lemma 4, the system will oscillate between the over- and underload region forever. Therefore, eventually, both senders will achieve equal sending rates. ■

REFERENCES

- [1] C. Chiu, R. Jain, Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks, *Computer Networks and ISDN systems*, vol. 17, pp. 1-14, 1989.
- [2] P. Dickens, W. Gropp, P. Woodward, High Performance Wide Area Data Transfers Over High Performance Networks, *Proceedings of the International Parallel and Distributed Processing Symposium, (IPDPS'02)*.
- [3] P. Dykstra, Gigabit Ethernet Jumbo Frames and why you should care, <http://sd.wareonearth.com/phil/jumbo.html>, download on February 2003.
- [4] S. Floyd, S. Ratnasamy, S. Shenker, Modifying TCP's Congestion Control for High Speeds <http://www.icir.org/floyd/hstcp.html>, download on February 2003.
- [5] M. Fisk, W. Feng, Dynamic Right-Sizing in TCP, *Proc. of the Los Alamos Computer Science Institute symposium*, October 2001.
- [6] S. Floyd, V. Jacobson, Random Early Detection Gateways for Congestion Avoidance, *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397-413, August 1993.
- [7] W. Feng, P. Tinnakornsriruphap, The Failure of TCP in High Performance Computational Grids, *Proceedings of the Super Computing 2000, (SC2000)*.
- [8] Y. Gu, M. Mazzucco, X. Hong, R. Grossman, Rate Based Congestion Control over High Bandwidth/Delay Links, Submitted to IEEE/ACM Transaction on Networking, <http://www.rgrossman.com/faq/sabul-faq-03.htm>, download on February 2003.
- [9] Y. Gu, X. Hong, M. Mazzucco, R. Grossman, SABUL: A High Performance Data Transfer Protocol, Submitted for Publications, <http://www.rgrossman.com/faq/sabul-faq-03.htm>, download on February 2003.

- [10] E. He, J. Leigh, O. Yu, Reliable Blast UDP: Predictable High Performance Bulk Data transfer, *IEEE Cluster Computing 2002*, Chicago, Illinois, September 2002.
- [11] V. Jacobson, R. Braden, D. Borman, TCP Extensions for High Performance, RFC 1323, May 1992.
- [12] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, TCP Selective Acknowledgement Options, RFC 2018, October 1996.
- [13] P. Oothongsap, Analysis of High-Speed Data Transfer Protocol Algorithms, North Carolina State University, Ph.D. Dissertation, 2004.
- [14] P. Oothongsap, M. Vouk, Y. Viniotis, A simulation analysis of SABUL Data Transfer Protocol, CIIT 2003, Scottsdale, AZ, November 2003.
- [15] P. Oothongsap, Y. Viniotis, M. Vouk, Experimental Analysis of the SABUL Congestion Control Algorithm, IFIP Networking 2004, Athens, Greece, May 2004.
- [16] A.K. Parekh, R.G. Gallager, Processor Sharing Approach to Flow Control in Integrated Service Networks: The Single Node Case *IEEE/ACM Transactions on Networking*, vol.1, pp. 344-357, 1993.
- [17] SABUL Source Code, <http://sourceforge.net/projects/dataspace>, download on February 03.
- [18] H. Sivakumar, S. Bailey, R.L. Grossman, Pockets: The Case for Application-level Network Stripping for Data Intensive Applications using High Speed Wide Area Networks., *Super Computing 2000*, find page and volume.
- [19] J. Semke, J. Mahdavi, M. Mathis, Automatic TCP Buffer Tuning, *Proc. ACM SIGCOMM*, vol. 28, pp. 315-323, October 1998.
- [20] S. Wallace, Tsunami File Transfer Protocol, <http://datatag.web.cern.ch/datatag/pfldnet2003/program.html>, download on February 03.

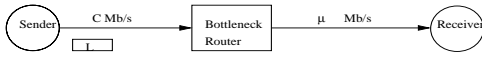


Fig. 1. A single SABUL connection

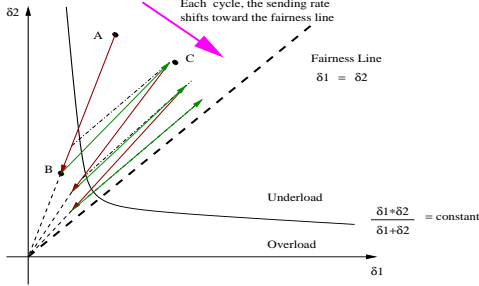


Fig. 2. Sample path showing the convergence to fairness for a SABUL rate control algorithm

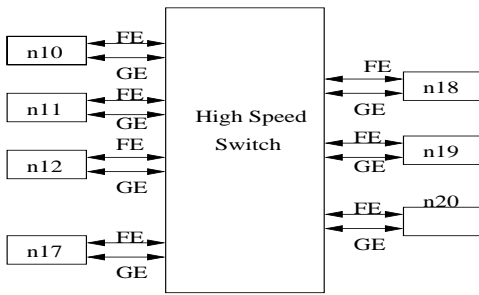


Fig. 3. Network Topology for a short-haul Network

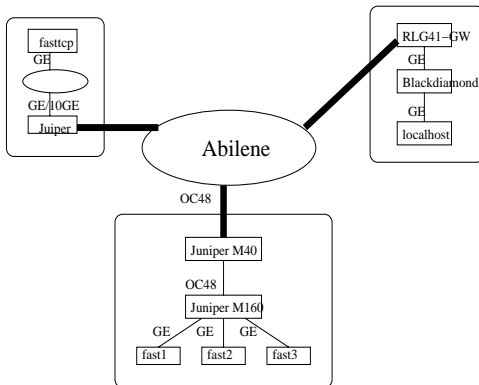


Fig. 4. Network Topology for a long-haul Network

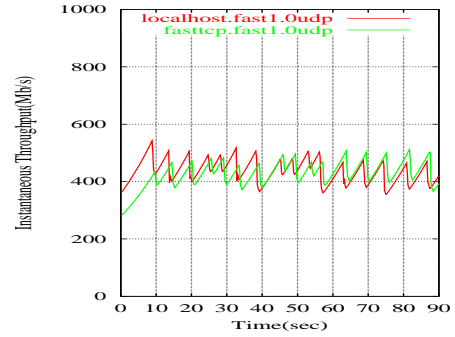


Fig. 5. Instantaneous sending rate from fasttcp and localhost to fast1

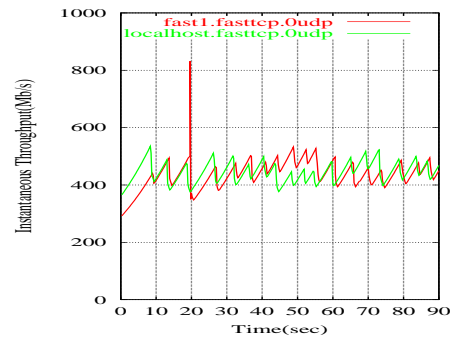


Fig. 6. Instantaneous sending rate from fast1 and localhost to fasttcp

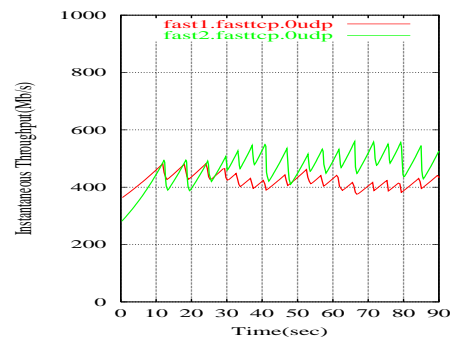


Fig. 7. Instantaneous sending rate from fast1 and fast2 to fasttcp

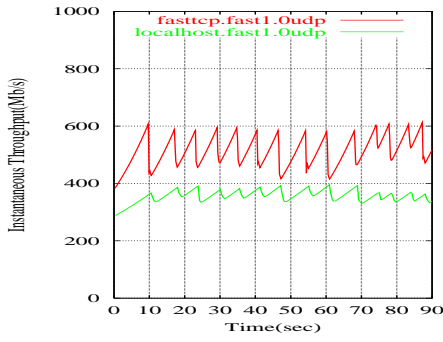


Fig. 8. Instantaneous sending rate from fasttcp and localhost to fast1

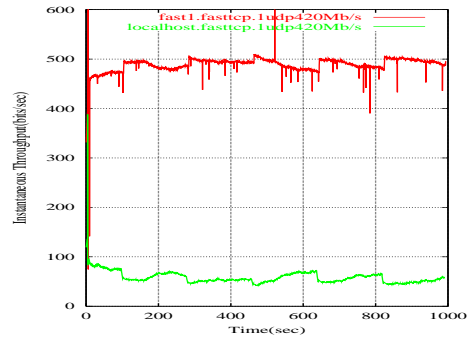


Fig. 11. Instantaneous sending rate from fast1 and localhost to fasttcp with different congestion control parameters

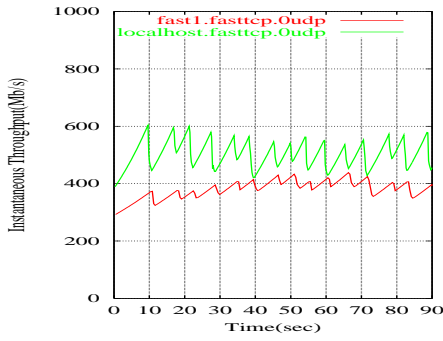


Fig. 9. Instantaneous sending rate from fast1 and localhost to fasttcp with different initial sending rate, rate control interval, and RTT

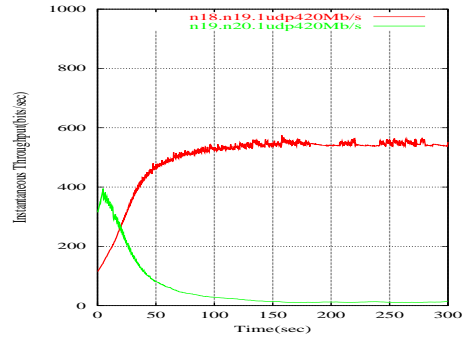


Fig. 12. Instantaneous sending rate from n18 and n19 to n20 with different congestion control parameters

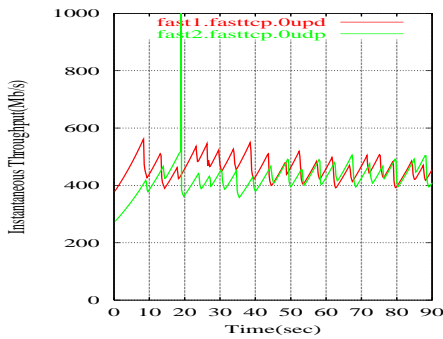


Fig. 10. Instantaneous sending rate from fast1 and localhost to fasttcp

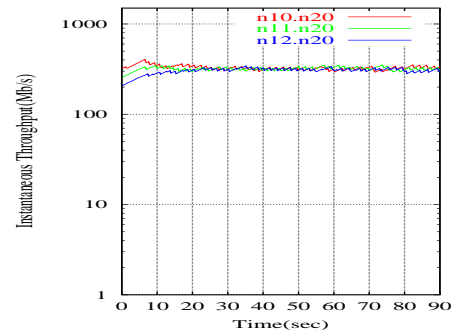


Fig. 13. Instantaneous sending rate from n10, n11, and n12 to n20

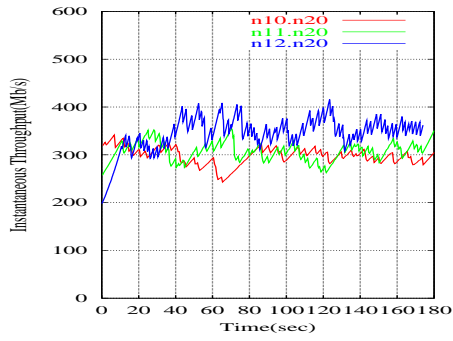


Fig. 14. Instantaneous sending rate from n10, n11, and n12 to n20

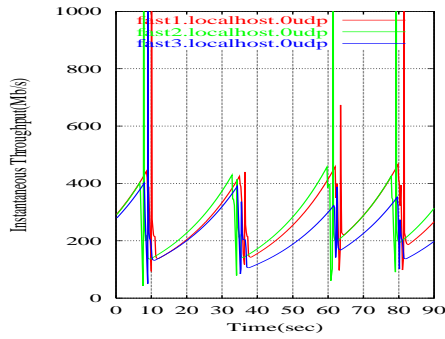


Fig. 15. Instantaneous sending rate from fast1, fast2 and fast3 to localhost

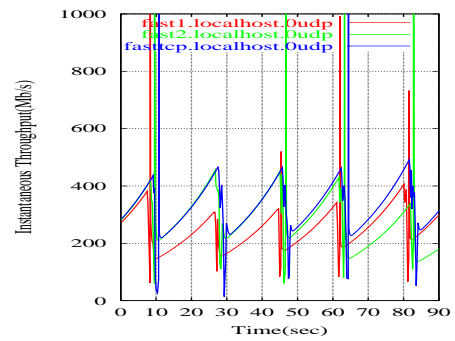


Fig. 17. Instantaneous sending rate from fast1, fast3 and fasttcp to localhost

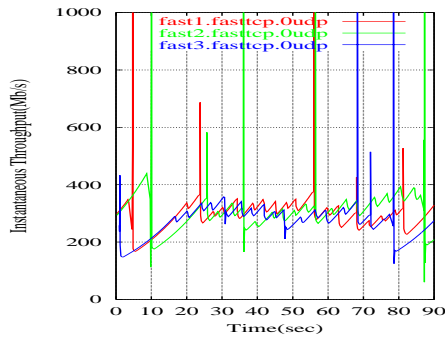


Fig. 16. Instantaneous sending rate from fast1, fast2 and fast3 to fasttcp