Amplification Effects of the Send Rate of TCP Connection through an Optical Burst Switching Network

Andrea Detti (*), Marco Listanti (**)

(*) Dept. of Electronic, University of Rome, “Tor Vergata”, Italy, e-mail: andrea.detti@uniroma2.it
(**) INFOCOM Dept., University of Rome, “La Sapienza”, Italy, e-mail: marco@infocom.uniroma1.it

ABSTRACT

This paper investigates support for TCP RENO flows in an Optical Burst Switching (OBS) network. In particular we evaluate the TCP send-rate, i.e. the amount of data sent per time unit taking into account the burst assembly mechanism at the edge nodes of the OBS network and burst loss events inside the network. The analysis demonstrates an interesting phenomenon, that we call *correlation benefit*. This phenomenon is introduced by the aggregation mechanism and can give rise, in some conditions, to a significant increase in the TCP send-rate. These results are obtained by means of an analytical model, based on a Markovian approach, and have been validated via an intensive simulation campaign.

***Index Terms*— Wavelength division multiplexing (WDM) networks, Optical Burst Switching (OBS), Transmission Control Protocol (TCP), Markovian Modelling.**

A. INTRODUCTION

The progressive increase in Internet traffic and the maturation of Wavelength Division Multiplexing (WDM) technology suggest that *All Optical Networks* are a candidate technology, at least in a medium to long term perspective, for the high-speed IP backbone [1,2]. A possible future network scenario is depicted in Fig. 1. In this scenario, a WDM-based, all optical backbone offers a transparent transport service to adjoining electronic IP networks. The interface between the electronic and optical worlds is implemented by *Edge Nodes* (ENs), whereas *Transit Nodes* (TNs) perform switching exclusively in the optical domain. Two developments can be foreseen for the future: in the short term, the optical backbone will be circuit-oriented, offering high capacity circuit-switched services through WDM end-to-end optical paths. In a longer-term perspective, higher bandwidth will be attained via Optical Packet Switching (OPS) or Burst Switching (OBS) [3,4,5,6,7,8,9].

In this paper, we focus on an OBS backbone[1]. The OBS technique allows asynchronous node operations and uses different wavelengths to transfer the optical packet payload – the *burst*– and its header: the *Burst Control Packet* (BCP). Buffering, deflection routing and wavelength conversion techniques, or a mix of these, have been proposed for burst contention resolution [7].

Although several issues in OBS networks have been investigated and the results published [6-19], the impact of the mechanism used for aggregation- *burstification* - on the tunneled protocols still has to be investigated in depth. It is known that data flows in the Internet primarily use the Transmission Control Protocol (TCP). Moreover, even if it is expected that in the future, real-time non-TCP traffic will grow rapidly, recent studies [20] have forecasted that, in the long term, a significant proportion of Internet traffic will be due to the progressive penetration of peer-to peer and server-to-server applications (e.g., GRID Computing, Content Delivery Networks, etc.), which are mostly supported by TCP.

Assuming that this is a realistic scenario, an interesting issue arises: the analysis of the effect of the burstification process on TCP flows across OBS networks.

From a general point of view, it can be argued that burstification can cause *delay in* TCP flows. Once an ingress EN has been reached, a TCP data unit (segment or ACK) has to wait for the end of the burstification process before it can be transmitted - imbedded within a burst. This extra delay causes a decrease in the send-rate for the TCP connection. Burstification may also introduce a degree of *correlation* between loss events for TCP segments. This can interfere with TCP recovery mechanisms [21]: several consecutive data units of the same TCP connection may be contained in the same burst; hence, the loss of this burst may yield the instantaneous loss of a whole sequence of

---

([1]) Although this paper focuses on OBS techniques, the main results of this paper can be extended, at least qualitatively, to Optical Packet Switching networks segments (or ACKs). Obviously, the correlation effect is greater as the number of data units from the same source contained within a burst increases.

In this paper, we investigate the correlation effect introduced by the burstification process on a TCP Reno connection in a bufferless OBS network. We show that as the number of segments per burst increases, the TCP send-rate increases as well. We call this effect *correlation benefit*. To achieve this goal, we adopt two methods: a modeling analysis and a simulation campaign in a showcase network scenario. The former method uses a simplified representation of network impairments to make the problem analytically tractable; the latter provides evidence supporting the practical relevance of the analytical approach. It is worth noting that the results shown in this paper are qualitative rather than quantitative. In other words, the study demonstrates the existence of correlation benefit, but only gives an approximate numerical evaluation of its importance.

A preliminary discussion on correlation benefit and delay penalties has already been presented in [22]; in this study, the authors report the effects of variation in the burstification period on the TCP send-rate. The main conclusions are: an increase in the burstification period has two counter-balanced effects on the TCP data rate: on the one hand, there is an increase in the rate due to the increase in the number of segments per burst (i.e., correlation benefit); on the other, there is a decrease due to the increase in round-trip-time (i.e., delay penalties). In this paper, we focus on the correlation effect: the analysis provides a detailed representation of TCP behavior, taking into account the effects of ACK loss. A specific section of the paper is dedicated to the practical significance of these results.

The paper is organized as follows: Section B discusses the analytical approach developed to evaluate TCP performance and to prove the existence of correlation benefit; afterwards; in Section C, we analyze the practical significance of these results. The main conclusions of the study are summarized in Section D.

B. CORRELATION BENEFIT MODELING ANALYSIS

This section presents an analytical model for the evaluation of the TCP send-rate and correlation benefit. The reference TCP connection model is discussed in Section B.1; a qualitative discussion of the effect of burstification on the TCP connection is presented in Section B.2; the analytical model for the TCP send-rate is developed in Section B.3; finally, Section B.4 describes the validation of the analytical model by means of simulations, presenting and analyzing the concept of correlation benefit .

B.1 TCP Connection Model

With reference to the network scenario in Fig. 1, Fig. 2 shows the model assumed for a TCP connection. The endpoints are named *source* and *receiver* and are supposed to support the "TCP Reno" [21]. We assume that the source only transmits TCP segments and that the receiver sends back ACKs.

The burstification process, i.e. the aggregation of incoming TCP segments, is carried out by the ingress EN by means of a device called the *burstifier*. Once a burst is ready; the EN immediately transmits a *Burst Control Packet* in order to reserve the bandwidth needed for burst transmission; as soon as the *offset time* [7] expires, it transmits the burst through the OBS network. The Egress EN disassembles the incoming burst by means of a *deburstifier* and routes the received TCP segments through the relevant destination access network. Analogous operations take place on the ACKs in the reverse direction.

As in [14], the burstifier is modeled as a FIFO packet queue (Fig. 3) into which the TCP segments flow. The queue is emptied (i.e., all the stored segments are sent) a constant time interval T_b after the arrival of the first segment; time T_b is called the *burstification period*. The TCP segments that enter the burstifier during a burstification period form a burst. We assume no limit on burst length, i.e. a burst can contain any number of TCP segments.

With regard to the deburstifier, we refer to the logical model shown in Fig. 4. The entering burst is disaggregated and the packets it contains are inserted into a FIFO queue, from which they are extracted to be transmitted over the access path.

As far as network impairments are concerned, we model the forward (TCP source $\rightarrow$ TCP receiver) and backward (TCP receiver $\rightarrow$ TCP source) access network paths as lossless links with end-to-end delay equal to d seconds and with bit-rates equal to B_a bit/s (B_a is hereafter called the *access bandwidth*). The purpose of these assumptions is to isolate the contribution of the correlation effect due to burstification: modeling the presence of access network losses and delay variability would have altered overall TCP performance, making it harder to evaluate the correlation effect .

The forward and backward paths through the OBS network are modeled as lossy links characterized by a transit time of T_p seconds and a bandwidth so high that the burst transmission time is negligible with respect to T_p. As we are considering a bufferless OBS network and assuming that the offset time is constant, the transit time T_p is also constant and is equal to the sum of the propagation and offset times. To make the analysis easier, the burst loss within the OBS network is assumed to be Bernoulli distributed with a parameter, p, representing the probability that the reservation of optical resources needed for burst switching fails.

As far as the TCP connection is concerned, let us assume that: i) the receiver advertises a maximum congestion window (cwnd) of W_m segments at connection establishment; ii) the receiver returns an ACK for every received segment, i.e. it does not employ a delayed ACK mechanism [28]; iii) the user applications at both ends of the TCP connection are always ready to send and receive data-units; iv) all TCP segments have the same size L, measured in bits; we assume that during a TCP connection, segments are predominantly of fixed length L; we therefore ignore the effects of variability in segment sizes.

The parameters above are summarized in Tab. 1 and are considered to be constant during the lifetime of the TCP connection.

Tab. 1: Summary of model parameters

Parameter	*Definition*
d	Access network delay (sec)
B_a	Access network bit-rate (bps)
p	Burst loss probability in the OBS network
T_p	OBS network end-to-end transit delay (sec)
T_b	Burstification period (sec)
W_m	Maximum TCP congestion window (segments)
L	TCP segment size (bit)

B.2 Impact of burstification on TCP performance

Let us now discuss the impact of burstification on the performance of TCP Reno. In the connection model presented in the previous section, there are two burstification effects: i) an increase in end-to-end round trip time (RTT); ii) the introduction of correlation among segment loss and segment delivery events.

As far as concerns increased RTT, the burstification process introduces an extra delay in the transfer of each segment; each segment has to wait for the expiration of the burstification period T_b before it can be forwarded by the EN. This extra delay component increases the RTT and, consequently, the value of the Retransmission Time-Out (RTO). It is easy to understand that the increase in the RTT leads to a lower send-rate for the TCP connection [(2)]; we refer to this degradation as a *delay penalty*.

Now, we focus our attention on the correlation effect introduced by burstification. First, let us define three classes of TCP connections: *fast*, *medium* and *slow*.

Definition 1: a fast-class connection is characterized by an access bandwidth (B_a) so high that the burst contains all the segments (or ACKs) of the TCP congestion window.

---

(2) This is strictly true for a *RTT limited* TCP source [23], in which the time needed to transmit the maximum congestion window is always less than or equal to RTT.

Definition 2: a slow-class connection is characterized by an access bandwidth so low that the burst contains at most one TCP segment (or ACK).

Definition 3: a connection is considered to belong to the medium-class if it does not belong to either the fast-class or the slow-class.

In the following, we give the name "fast-class (slow-class or medium-class) source" to the source in a fast-class (slow-class or medium-class) connection.

Let R be the maximum number of segments transmitted by the TCP source during a burstification period T_b; that is the minimum of: the maximum value of the TCP cwnd (W_m) and the maximum number of segments of size L that can be transmitted by the source during the burstification period T_b with access bandwidth B_a. This can be expressed by,

$$R = \min\left\{W_m, \max\left\{\left\lfloor B_a\ T_b / L \right\rfloor, 1\right\}\right\} \tag{1}$$

where $\lfloor x \rfloor$ represents the floor function of x.

In the connection model we are considering, a source belongs to the fast-class if R=W_m [(3)]; the slow-class if R=1; the medium-class if 1<R<W_m.

Let us now consider, for these three connection classes what happens when a burst is lost or when a burst is successfully delivered.

1) Slow-class connections

For a slow-class connection, the loss of a forward burst determines at most the loss of a single segment; this event is likely to be recovered by fast recovery/retransmit mechanisms, which cause the TCP congestion window to be halved. As we assume that burst loss events are statistically independent, segment loss events will be statistically independent as well; therefore, on the average, a slow-class TCP connection will go into fast retransmit/recovery every 1/p segments sent. Unlike

---

[(3)] Under our assumptions of an unshared burstifier and fixed network delays, the ACKs inter-arrival time is equal to the segment inter-arrival time. As a consequence, if all cwnd segments are contained in a single forward burst (i.e., R=W_m), all related ACKs will be contained in a single reverse burst.

the loss of a forward burst, the loss of a backward burst with high probability does not cause a reduction of the congestion window. As the lost burst contains a single ACK, the cumulative property of TCP acknowledgements means that the next delivered ACK will also confirm the reception of segments related to the lost ACKs. However, it is worth noting that the loss of ACKs may have a slight slowing effect on the increase of the congestion window and, in some cases, may compromise the start of fast retransmit/recovery mechanisms, forcing the source into time-out. As will be shown below, these penalties introduced by ACK loss may be ignored, if the burst loss probability is less than 10^{-1}.

2) Fast-class connections

Remarkable differences can be observed in the case of a fast-class connection. The loss of a forward burst causes the same effect as the loss of a whole TCP congestion window. The lost segments will be recovered by means of the time-out mechanism that throttles the congestion window to one. A similar effect on the congestion window will be observed if a backward burst is lost: the lost burst contains all the ACKs for the congestion window; hence, the TCP connection goes into time-out. It is clear that the presence of the burstifier introduces a high degree of correlation among segment loss events, which prevents fast recovery/retransmit mechanisms from operating correctly. When, on the other hand, both forward and related backward bursts are successfully delivered, the TCP source experiences a concentrated successful delivery that quickly reopens the congestion window. Therefore, the cumulative impact of the loss of a burst on a fast-class connection depends on the composition of these two opposite effects. The resulting TCP time-out expiration probability p_f is equal to $1-(1-p)^2$; so, on the average, a time-out event occurs every $\overline{cwnd}/p_f$ segments sent, where $\overline{cwnd}$ represents the mean value of the congestion window, measured in segments.

3) Medium-class connections

For medium-class connections, the TCP reaction to burst loss depends on several conditions that make the analysis difficult. Intuitively, the behavior of a medium-class connection is somewhere between that of fast-class and slow-class connections: the higher the number of segments per burst, the closer the behavior will resemble that of a fast-class connection.

Fig. 5 shows three examples of variation in congestion window size (*cwnd*) for slow-, medium- and fast-class connections.

Lost segments in fast-class connections are recovered by means of the RTO mechanism; therefore, *cwnd* decreases to one after each burst loss. Nevertheless, the great number of successfully delivered segments between two consecutive time-outs quickly reopens the congestion window and keeps it near its maximum value (e.g., W_m=128 in this figure).

On the contrary, segment losses in slow-class connections are mainly recovered by means of fast recovery and fast retransmit mechanisms, which do not throttle the congestion window as when the RTO expires, but the smaller number of successfully delivered segments between two consecutive fast retransmit/recoveries keeps the current value of the congestion window significantly below its maximum value. As expected, the size of the congestion window for medium-class connections is intermediate between the sizes for fast- and slow-class connections.

B.3 The TCP Reno Send-Rate Model

In this section, we present two analytical models for computing the TCP send-rate for slow- and the fast-class sources, respectively.

We define an *R-source* as a TCP source able to transmit at most R segments during a burstification period T_b.

Adopting the same line of reasoning as [23], we define $N(t,R)$ as the number of segments actually transmitted by an R-source in the time interval $[0,t]$, $t>0$; let $B(t,R)=N(t,R)/t$ be the actual send-rate for an R-source over a time interval $[0,t]$. Note that $N(t,R)$ is the number of segments transmitted, irrespective of their successful delivery, i.e. $N(t,R)$ does not take into account possible segment losses due to transmission errors or network congestion. We define the actual long-term steady-state send-rate $B(R)$ of an R-source as:

$$B(R) = \lim_{t\to\infty} B(t,R) = \lim_{t\to\infty} \frac{N(t,R)}{t} \quad (2)$$

In the rest of this section, we first determine the increase in RTT and RTO due to the burstification process; we then analytically evaluate send-rates for slow- and fast-class sources, i.e. $B(1)$ and $B(W_m)$. Finally, we show, by means of simulations, that a medium-class source, i.e. $B(1<R<W_m)$, achieves an intermediate send-rate.

Referring to the network model in Fig. 2, let us define:

- RTT : the average value of the round trip time for the TCP connection, i.e. the time interval between the transmission of a segment and the reception of the related ACK;
- $RTTVAR$: the standard deviation of the round trip time;
- RTO : the average value of the "first" retransmission time-out, i.e. the retransmission time-out without any *backoff* duplication [24].

The values of these variables include delays contributed by burstifiers. However, in order to better evaluate the impact of the burstification process on TCP performance, it is useful to introduce a supplementary set of variables in which the delay introduced by burstifiers is not considered. So, let us define:

- RTT_0 : the average value of the round trip time in the absence of burstifiers;
- $RTTVAR_0$: the standard deviation of the round trip time in the absence of burstifiers;
- RTO_0 : the average value of the "first" retransmission time-out in the absence of burstifiers.

Remembering the TCP Reno rules for RTO evaluation [21], and considering the fact that, in the connection model shown in Fig. 2, variation in round trip time is due exclusively to burstifiers, we obtain:

$$RTT_0 = 4d + 2T_p \quad {}^{(4)} \tag{3}$$

$$RTTVAR_0 = 0 \tag{4}$$

$$RTO_0 = RTT_0 + 4RTTVAR_0 = RTT_0 \tag{5}$$

As the delay experienced by a segment and the returning ACK within the burstifier is bounded in the interval $[0,T_b]$, the value *RTT* is bounded by RTT_0+2T_b. Moreover, assuming $RTTVAR \approx RTTVAR_0$ (5), i.e. the delay variation introduced by the burstifiers is not so heavy as to significantly increase the standard deviation of the round trip time and, consequently, the value of "first" retransmission time-out, we obtain for RTT and RTO:

$$RTT \approx (1+2\alpha)\,RTT_0 \tag{6}$$

$$RTO \approx (1+2\alpha)RTT_0 \tag{7}$$

where $\alpha = T_b / RTT_0$.

In these expressions $(1+2\alpha)$ represents the effect of the burstifiers on the average round trip time and on the average "first" retransmission time-out. This factor can thus be considered as a measure of the *delay penalty* introduced by the burstification process.

1) Send-rate for slow-class TCP sources: B(1)

On the basis of the assumptions discussed in the previous section, it follows that a slow-class source experiences independent segment losses and that, if the loss probability is sufficiently low (e.g. $p<10^{-1}$), the effects of the loss of ACKs may be ignored. Under these assumptions, the TCP connection model (Fig. 2) can be analyzed using the approach described in [23,25,26]. In particular,

---

(4) For the sake of simplicity, we ignore the contribution of segment transmission time, this assumption (i.e. $L / B_a = 4\,d + 2\,T_b$) holds for the whole of our discussion.

(5) In the following the symbol '≈' means 'about equal to'

we utilize the send-rate (expressed by (32) in [23] with b=1), B_{ku}, to evaluate the slow-class TCP send-rate $B(1)$. Hence:

$$B_{ku}(W_m, RTT, p, RTO) =$$

$$\begin{cases} \dfrac{\dfrac{1-p}{p} + E[W_u] + \hat{Q}(E[W_u])\dfrac{1}{1-p}}{RTT(\dfrac{E[W_u]}{2}+1) + \hat{Q}(E[W])RTO\dfrac{f(p)}{1-p}} & \text{for } E[W_u] < W_m \\ \dfrac{\dfrac{1-p}{p} + W_m + \hat{Q}(W_m)\dfrac{1}{1-p}}{RTT(\dfrac{W_m}{8}+\dfrac{1-p}{pW_m}+2) + \hat{Q}(W_m)RTO\dfrac{f(p)}{1-p}} & \text{otherwise} \end{cases} \tag{8}$$

where

$$E[W_u] = 1 + \sqrt{\frac{8(1+p)}{3p}+1} \ ; \ \ \hat{Q}(u) \approx \min\left\{1, \frac{3}{u}\right\} \ ; \ \ f(p) = 1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6$$

Applying (6), (7) and (8), the slow-class TCP send-rate $B(1)$ can be written as:

$$B(1) = B_{ku}(W_m, RTT_0(1+2\alpha), p, RTT_0(1+2\alpha)) \tag{9}$$

It is worth noting that the model derived from [23] assumes that the TCP sources are *round trip time limited*, i.e. the time needed to transmit a number of segments equal to the maximum congestion window (W_m) is always less than or equal to RTT, i.e.

$$\frac{W_m L}{B_a} \leq RTT \tag{10}$$

Taking into account (6) and the fact that, for a slow-class TCP source, $B_a T_b / L \leq 1$, the condition of existence of a round trip time limited slow-class source, i.e.

$$T_b \leq \frac{RTT_0}{W_m - 2} \tag{11}$$

2) Send rate for fast-class TCP sources: $B(W_m)$

In this section we develop a model of TCP congestion control and the RTO recovery mechanism that captures the correlation effect introduced by the burstifier on a fast-class source.

Considering that, for a fast-class TCP source, $R=W_m$, definitions (6) and (10) imply that a fast-class source is always round trip time limited; consequently, as in [23], we model the TCP behavior as a succession of "rounds". During the *j*-th round a TCP source transmits a number of segments equal to the current value of the congestion window (*cwnd*); once all the *cwnd* segments have been sent, the source waits for reception of the related ACKs. The next segment will not be transmitted until either i) the first ACK for one of the previously transmitted segments is received, or ii) the retransmission time-out (RTO) expires. The start of the transmission of the next segment signals the end of the *j*-th round and the beginning of the (*j*+1)-th round.

It should be remembered that in a fast-class TCP source, all segments transmitted during a round are contained in a single forward burst and, if the forward burst is successfully delivered, all of the related ACKs generated by the TCP receiver will be contained in a single backward burst.

We define a *successful round*, as a round in which both the forward and the backward bursts are successfully delivered. On the contrary, a round in which either the forward or the backward bursts are lost is defined as a *lossy round*. The probability p_f of a lossy round is given by:

$$p_f = 1-(1-p)^2 \tag{12}$$

where, the probability that a successful round occurs is 1-p_f .

Three kinds of rounds can be defined: i) *slow start rounds*; ii) *congestion avoidance rounds*, and iii) *backoff rounds*. Each can be either successful or lossy.

As established by TCP rules, during a slow start round, a TCP source transmits all the segments allowed by the value of the congestion window; after each successful round, it doubles the value of *cwnd* until the value of the slow start threshold (*ssth*) is reached (Fig. 6). This begins a sequence of congestion avoidance rounds. During a congestion avoidance round, a TCP source transmits the segments allowed by the value of the congestion window; after each successful round, *cwnd* is increased by one segment until the maximum value of *cwnd* (i.e. W_m) is reached.

When a lossy round occurs, the RTO expires and the TCP source goes into slow start, setting *cwnd*=1 and *ssth* equal to one half of the value assumed by the congestion window in the previous round; the RTO is doubled, i.e. the backoff coefficient (*bck*) is set to 2. If this "first retransmission" round is also lossy (Fig. 7), after the expiration of the RTO, the TCP source sets *cwnd*=1, *ssth*=2 and doubles *bck*. If a sequence of lossy rounds occurs, a sequence of backoff rounds takes place; in each round, the TCP source maintains *cwnd*=1 and *ssth*=2, doubling the value of *bck*, until it reaches 64. When the lossy round sequence is broken by a successful round, the TCP source sets *cwnd*=2 and a sequence of slow start rounds begins.

In order to simplify the model, we assume that, after a successful round, the backoff coefficient (*bck*) used for the RTO evaluation is always set to one. This is not completely consistent with the rules in [24], which recommend that the backoff coefficient should be reset to one, only when all previously lost segments have been successfully retransmitted. However, the close fit between the model and simulation results makes this hypothesis reasonable.

In line with these considerations, the dynamic behavior of a fast-class TCP source can be modeled by a Markov chain $\{S^{\phi}_{cwnd,ssth,bck}\}$ in which the generic state $S^{\phi}_{cwnd,ssth,bck}$ of the source is characterized by the type ϕ of the current round [slow start (ϕ=ss), congestion avoidance (ϕ=ca), and backoff (ϕ=bo)], and by the values of *cwnd*, *ssth* and *bck*.

Three families of round, and consequently three categories of state, can be defined. In what follows we will consider the terms "state" and "round" to be interchangeable.

– $S^{ss}_{cwnd,ssth,bck}$ identifies a slow start round; it should be noted that, as previously explained, for this kind of round the only possible values of *bck* are 1 and 2, (i.e. *bck*=1, 2);

– $S^{ca}_{cwnd,0,1}$ identifies a congestion avoidance round; as previously explained, for this kind of state the value of *ssth* is irrelevant (so the subscript "0" should be interpreted as meaning "don't care"), *bck* always assumes the value 1 (i.e. *bck*=1);

– $S^{bo}_{1,2,bck}$ identifies a backoff round, as previously explained, for this type of round *cwnd*=1 and *ssth*=2.

Let us define:

- $N(S^{\phi}_{cwnd,ssth,bck})$: the number of segments emitted in the state $S^{\phi}_{cwnd,ssth,bck}$ (ϕ=*ss*, *ca*, *bo*);
- $\overline{T}(S^{\phi}_{cwnd,ssth,bck})$: the mean time for which the source is in state $S^{\phi}_{cwnd,ssth,bck}$ (ϕ=*ss*, *ca*, *bo*), i.e. the mean duration of the associated round.

Assuming an integer number of rounds over the interval (0,*t*), it is easy to prove that the send rate $B(W_m)$ for a fast-class TCP source is:

$$B\left(W_m\right) = \lim_{t\to\infty} \frac{N(t,W_m)}{t} = \frac{\sum_{\phi}\sum_{cwnd}\sum_{ssth}\sum_{bkc} \Pr\{S^{\phi}_{cwnd,ssth,bck}\} N\left(S^{\phi}_{cwnd,ssth,bck}\right)}{\sum_{\phi}\sum_{cwnd}\sum_{ssth}\sum_{bck} \Pr\{S^{\phi}_{cwnd,ssth,bck}\}\ \overline{T}(S^{\phi}_{cwnd,ssth,bck})} \tag{13}$$

In order to compute (13), we identify the states $S^{\phi}_{cwnd,ssth,bck}$ of the Markov chain characterizing the dynamic behavior of the TCP source and the relevant transition probabilities and evaluate the steady state probabilities $\Pr\{S^{\phi}_{cwnd,ssth,bck}\}$; finally we evaluate the quantities $N(S^{\phi}_{cwnd,ssth,bck})$ and $\overline{T}(S^{\phi}_{cwnd,ssth,bck})$.

Let us begin with the slow start states (ϕ=*ss*). In a $S^{ss}_{cwnd,ssth,bck}$ state a TCP source transmits in slow-start with *bck*≤2; hence:

$$N(S^{ss}_{cwnd,ssth,bck}) = cwnd \tag{14}$$

As for the range of values that *cwnd* can assume, it is obvious that, in a slow start state, *cwnd*≤*ssth*. Due to the slow-start congestion window doubling mechanism, the only possible values for *cwnd* are powers of two or the final value *cwnd*=*ssth*; so, for a slow start state $cwnd \in \left\{ \min\left\{ ssth,\ 2^{k} \right\}, k = 1,2,3,... \right\}$.

Having fixed a value for *ssth*, we define the *ssth-th slow start sub-chain* as the set of $S^{ss}_{cwnd,ssth,bck}$ states obtained by varying *cwnd* within the set of allowed values. Due to the limitations on the possible values of *cwnd*, the number of states of a *ssth*-th slow-start chain is equal to $1 + \left\lceil \log_2\left(ssth\right) \right\rceil$, where $\left\lceil x \right\rceil$ represents the ceiling function of *x*.

As far as the range of values of *ssth* is concerned, it is easy to recognize that: $2 \le ssth \le \left\lfloor \frac{W_m}{2} \right\rfloor$, hence, the total number of slow-start states is equal to $\sum_{i=2}^{\lfloor W_m/2 \rfloor} 1 + \left\lceil \log_2\left(i\right) \right\rceil$.

As far as the mean duration of a slow-start round is concerned, it should be noted that if *cwnd*=1, the generic $S^{ss}_{1,ssth,bck}$ round follows a lossy round (Fig. 6); hence *bck*=2 and the mean round duration is: i) *RTT*, if $S^{ss}_{1,ssth,2}$ is a successful round; ii) 2·*RTO*, if $S^{ss}_{1,ssth,2}$ is a lossy round.

Furthermore, if *cwnd*>1, the generic $S^{ss}_{cwnd,ssth,bck}$ round follows a successful round; hence *bck*=1 and the relevant mean round duration is: i) *RTT*, if $S^{ss}_{cwnd,ssth,1}$ is a successful round; ii) *RTO*, if $S^{ss}_{cwnd,ssth,1}$ is a lossy round.

As a consequence, the mean time duration $\overline{T}(S^{ss}_{cwnd,ssth,bck})$ of a $S^{ss}_{cwnd,ssth,bck}$ round is given by:

$$\overline{T}(S^{ss}_{cwnd,ssth,bck}) = \begin{cases} RTT\left(1\text{-}p_f\right) + 2\ RTO\ p_f & cwnd = 1 \\ RTT\left(1\text{-}p_f\right) + RTO\ p_f & otherwise \end{cases} \qquad (15)$$

Let us now examine congestion avoidance rounds (ϕ=*ca*). During a $S^{ca}_{cwnd,0,1}$ round, the TCP source transmits a number of segments given by:

$$N\left(S^{ca}_{cwnd,0,1}\right) = cwnd \qquad (16)$$

According to TCP rules, in this set of states, the minimum value of *ssth* is 2. Thus, $3 \le cwnd \le W_m$.
We call the set of congestion avoidance states, the *congestion avoidance sub-chain*.

As far as concerns the mean duration of a congestion avoidance round, we may apply the same reasoning as in the case of $S^{ss}_{cwnd,ssth,bck}$ with *cwnd*>1. Therefore, the mean time duration of a $S^{ca}_{cwnd,0,1}$ round is:

$$\overline{T}(S^{ca}_{cwnd,0,1}) = RTT\left(1-p_f\right)+RTO\ p_f \tag{17}$$

Finally, when a backoff round $S^{bo}_{1,2,bck}$ occurs, the TCP source doubles the backoff coefficient *bck* and tries to retransmit the segments lost in the previous round. Hence, the number of segments $N(S^{bo}_{1,2,bck})$ when the source is in this state and the relevant mean time duration $\overline{T}(S^{bo}_{1,2,bck})$ are given by:

$$N(S^{bo}_{1,2,bck})=1 \tag{18}$$

$$\overline{T}(S^{bo}_{1,2,bck}) = RTT\left(1-p_f\right)+bkc\cdot RTO\ p_f \tag{19}$$

The TCP backoff mechanism, and the value *bck*=2 used in slow-start states, imply that the allowed values for *bck* are {4,8,16,32,64}. In the following, we refer to the set of backoff states as the *backoff sub-chain*.

Summarizing, we have defined the number of segments transmitted and the mean time duration for each type of round. Now we focus on the transitions between states. For the sake of clarity, we first examine a specific example; we then examine the general case by generalizing the observations from the example.

Fig. 8 depicts the state transition diagram for W_m=8. In this case, we have three slow-start sub-chains, associated with *ssth*=2, 3, 4, one congestion avoidance sub-chain and one backoff sub-chain. Let us follow the state transition sequence for the round time behavior shown in Fig. 6.

In the left most round, the TCP source is in slow-start with *cwnd*=2, *ssth*=4 and *bck*=1; the associated state is thus $S^{ss}_{2,4,1}$. Since this round is successful, in the next one the TCP source doubles *cwnd*, i.e. the source hops from $S^{ss}_{2,4,1}$ to $S^{ss}_{4,4,1}$. This round is successful too, but, since *cwnd* has reached the slow-start threshold (*ssth*), in the next round the TCP source increases cwnd by 1 and enters the congestion avoidance phase. As a consequence, in the state transition diagram we have a hop from $S^{ss}_{4,4,1}$ to $S^{ca}_{5,0,1}$. During the $S^{ca}_{5,0,1}$ round there are no losses, so a new congestion avoidance round starts with *cwnd*=6; i.e. there is a hop from $S^{ca}_{5,0,1}$ to $S^{ca}_{6,0,1}$ and subsequently, for the same reason, from $S^{ca}_{6,0,1}$ to $S^{ca}_{7,0,1}$. The $S^{ca}_{7,0,1}$ round is a lossy one, so, the round ends at the expiration of the RTO. In the next round, the TCP source throttles its *cwnd* to one and sets *ssth*=$\lfloor 7/2 \rfloor$=3. As a consequence, there is a hop from $S^{ca}_{7,0,1}$ to $S^{ss}_{1,3,2}$ on the transition diagram. The "first retransmission" round $S^{ss}_{1,3,2}$ is a successful one; hence, the source hops to the $S^{ss}_{2,3,1}$ state, and so on.

Conversely, as shown in Fig. 7, when the "first retransmission" round $S^{ss}_{1,3,2}$ is a lossy one, the next round is a backoff round with *bck*=4; i.e. the source hops from $S^{ss}_{1,3,2}$ to $S^{bo}_{1,2,4}$. If the $S^{bo}_{1,2,4}$ round is also lossy, once the RTO has expired, the TCP source sets *bck*= 8 and tries to retransmit the segments. This retransmission attempt forms the $S^{bo}_{1,2,8}$ round. This is successful; once the ACKs have been received, the TCP source begins to transmit in slow-start with *cwnd*=2 and *ssth*=2. This implies a transition from $S^{bo}_{1,2,8}$ to $S^{ss}_{2,2,1}$, and so on.

Generalizing the state transition diagram in Fig. 8 , it can be seen that the Markov chain that describes the evolution of a fast-class TCP source is composed of:

- one slow-start sub-chain for every possible value of *ssth*, i.e. $\lfloor W_m/2 \rfloor$;
- one congestion avoidance sub-chain;
- one backoff sub-chain;

Fig. 9 shows the possible state transitions from a generic state in the slow start, congestion avoidance and backoff sub-chains. The relevant transition probabilities are indicated in square brackets; the terms in round brackets indicate the possible conditions under which the relevant transition may occur.

By utilizing the generic elements shown in Fig. 9, it is possible to build the complete Markov chain governing the dynamic behavior of the TCP source and to determine the steady state probabilities $\Pr\{ S^{\phi}_{cwnd,ssth,bck} \}$.

Using the value of $\Pr\{ S^{\phi}_{cwnd,ssth,bck} \}$ with (14), (15), (16), (17), (18), (19) we can use (13) to calculate the fast-class send rate.

B.4 Modeling Analysis Numerical Results

In this section we first validate the proposed TCP model using a simulation campaign [27]; we then demonstrate the main result of this paper, namely that, given the model parameters (W_m, T_b, RTT, p), the higher the maximum number R of segments in a burst, the higher is the TCP source send rate.

In order to better understand these results, it is worth recalling that the TCP sources are round trip time limited. Thus by (1), once W_m, T_b and L are fixed, the potential send rate (achievable for p=0) is equal to $(W_m \cdot L)/RTT$, independently of the value of B_a. Hence, any difference between the send-rates for two sources with different B_a can only be due to the correlation effect previously discussed.

Fig. 10 plots the TCP send rate against the burst loss probability (p), assuming T_b=3 ms, RTT_0=600 ms, W_m=128 and a segment size L=512 byte.

As far as the slow-class send rate, i.e. $B(1)$, is concerned, for $p<10^{-1}$ there is a close fit between the analytical results provided by (9) and the results from simulation. Though this cannot be seen in Fig. 10, for values of loss probability greater than 10^{-1} the curve generated by the analytical model is an upper bound for the actual send-rate. This effect is due to the fact that the model ignores the loss of ACKs.

Regarding the fast-class send rate, i.e. $B(W_m)$, we note the strong correlation between the analytical results provided by (13) and the simulated ones.

Fig. 10 also shows the send rate $B(R)$ for a medium-class TCP source (i.e. $1<R<W_m$). As expected, $B(R)$ assumes intermediate values between the values for fast and the slow class sources. The higher the value of R, the higher is the value of $B(R)$.

In general, we conjecture that, for $R_1 < R_2$.

$$B(1) \le B(R_1) \le B(R_2) \le B(W_m) \tag{20}$$

In order to explain conjecture (20), let us introduce the concept of *correlation benefit* (C_b). With reference to the TCP connection model in Fig. 2, we define *correlation benefit* as the ratio between the TCP send rate $B(R)$ and the TCP send rate associated with a slow-class source $B(1)$, i.e.

$$C_b(R) = \frac{B(R)}{B(1)} \tag{21}$$

Applying (9) and (13), it is easy to prove that in the case of fast-class sources ($R=W_m$) the correlation benefit is independent of RTT. The same result holds for many of our simulation runs with medium-class sources (data not shown).

In Fig. 11 we plot values for C_b against the burst loss probability (p) for specific values of R (R=20, 50, 100, 128); Fig. 12 plots C_b against p for W_m=64, 128 and R=50, 64, 128.

An examination of Fig. 11 and Fig. 12 shows that [6]:

i) the higher the value of R, the higher is $C_b(R)$;

ii) the maximum values for $C_b(R)$ are centered around $p \simeq 1/W_m$; for extreme values of burst loss probabilities, i.e. p=0, p=1, $C_b(R)$ has a value of one.

---

[6] These conclusions can be rigorously demonstrated if we use the simplified TCP model proposed in [22] .

iii) when W_m^1 and W_m^2 are two values for the maximum congestion window, such that $W_m^1 < W_m^2$, producing correlation benefits $C_b^1(R)$ and $C_b^2(R)$, for $p > 1/W_m^1$, $C_b^1(R)$ is roughly equal to $C_b^2(R)$; in fact, for $p > 1/W_m^1$ the loss probability is so high that it does not allow values of $cwnd > W_m^1$. Hence, the send-rate does not change, even for higher values of W_m (see Fig. 12);

iv) the correlation benefit may give rise to a significant increase in send rate in the region of loss around $p \simeq 1/W_m$. Fig. 11 and Fig. 12 show that, in the cases examined, the fast-class send rate can reach about 6 times the rate for the slow-class.

C. PRACTICAL SIGNIFICANCE OF THE MODELING RESULT

To demonstrate the existence of the correlation benefit the previous sections have relied on a simplified TCP connection model that eliminates several network issues while highlighting the correlation benefit itself. For this reason, the results obtained should be interpreted in qualitative rather than strictly quantitative terms. It is thus reasonable to ask what is the significance of these results in a real network environment. This question is particularly important if we consider that the hypothesis of fixed burst loss probability (p) is a crude simplification of loss behavioral in real networks.

The aim of this section is to provide evidence of correlation benefit in real network environments, while at the same time identifying specific limitations in our analytical model.

To achieve this goal we have conducted simulations of the showcase network scenario reported in Fig. 13. In this scenario we have an OBS backbone network connecting eleven couples of edge nodes by means of a link between two core nodes. All the links are bidirectional and support four wavelengths at 155.42 Mbps. Output wavelength contentions within nodes are solved using the Just Enough Time protocol and wavelength conversion [19]. The link between the core nodes is a bottleneck for the network, in the sense that network losses take place only within these nodes. To compare the results of this simulation with the modeling results in Fig. 10, we consider an end-to-end propagation delay (i.e., RTT_0) of 600 ms, a burstification period T_b of 3 ms, a maximum congestion window W_m equal to 128 segments and a segment size L of 512 bytes; OBS offset time is ignored.

At both ends of the link, the electronic access network consists of ten access campuses and routers. Each campus is connected to a specific access router (7) that interfaces the edge node at 621.68 Mbps. Campus #i consists of five TCP sources, five TCP receivers, one background self-similar traffic source [14] with Hurst parameter equal to 0.7 and one background traffic receiver (8). The five TCP sources and receivers interface the router with access bandwidths, B_a, of 1 Gbps, 100 Mbps, 10 Mbps, 1 Mbps, and 500 Kbps respectively. The background source and receiver interface the router with 621.68 Mbps of bandwidth. With respect to the analysis in Section B.3, the source with B_a =1 Mbps can be considered as a round trip time limited slow-class source, whereas the send-rate for the 500 Kbps source is limited by the access bandwidth.

Finally, two couples of TCP sources and receivers are directly connected to a dedicated edge node, which we refer to as a "direct-source/receiver". The direct-source does not share a burstifier and has infinite access bandwidth. Apart from burst loss modeling, it faces the same network impairments as the fast-class source in the connection model in Fig. 2. Analyzing the performance of the direct-source connection will be useful to validate the effectiveness of the Bernoulli-distributed loss model in Section B.

Transport level connections can be summarized as follows: the TCP source #j for campus #i is connected, on the other side of the network, to the TCP receiver #j for campus #i; the background traffic source for campus #i is connected, on the other side of the network, to the background traffic receiver for campus #i; finally, direct-sources are connected to direct receivers on the other side of the network. In short, the network scenario is symmetrical.

---

(7) The packet queue for the access router is unlimited.

(8) The background traffic load is the parameter we manipulate to change the burst loss probability.

Such a showcase scenario allows us to exploit correlation benefit taking into account: i) the relationship between load and loss on the OBS network, which in the modeling analysis was less stringent, due to the assumption of a Bernoulli-distributed probability of burst loss; ii) the sharing of the burstifier [(9)].

It is only the latter that produces effects on obtainable performance, other than those already highlighted by the modeling analysis in Section B.3. In the connection model described in Fig. 2, the TCP source has a dedicated burstifier; hence, with enough access bandwidth, it is able to include all the cwnd segments (or ACKs) within a single burst. This characterizes the source as fast-class. Let us now consider the behavior of this source in the presence of a shared burstifier. It may occur that the burstification period starts with the arrival of a data packet sent by another source and expires during the bulk arrival of the cwnd segments (or ACKs) for the source. In this case, these segments will be assembled into two bursts instead of one. Following burstification, if there are no network impairments, these two groups of segments will be interrupted by a burstification period, hence, they will never be assembled into a single burst. However, in the event of a burst loss, TCP flow control mechanisms will reduce the number of segments sent, thereby increasing the probability of aggregation into a single burst [(10)]. In other words, after a burst loss, the source renews its capability to assemble all the cwnd segments into a single burst. The same reasoning applies to medium-class sources.

Fig. 14 reports the probability density function (pdf) for the number of segments assembled in a single burst by a TCP connection in the absence of network loss (i.e., regime cwnd is equal to 128). As expected, the pdf for the direct-source is concentrated at the maximum value for cwnd (i.e., 128), the entire cwnd lying within a single burst. In the connection model in Fig. 2, an access

[(9)] Queuing in the access router also allows us to take account of limited delay variation in the access network.

[(10)] Delay variation in the access network may push groups of segments closer together or further apart, leading to re-aggregation of segments into a single burst or a group separation. In the simulations examined here, these events are of marginal importance with respect to more important effects.

bandwidth of 1 Gbps is enough to consider the source as fast-class; in the showcase scenario, on the other hand, burstifier sharing effects prevent this and the pdf is spread around values below maximum cwnd. Increased access bandwidth nonetheless leads to an increase in the average number of segments per burst. To conclude, the sharing of the burstifier reduces the ability of the TCP source to aggregate segments; conversely, shorter cwnd transmission time means that this ability increases with increased access bandwidth. Consequently, in real networks, it is hard to achieve a fast-class connection, in the sense of a TCP source-receiver couple always able to place all cwnd segments and ACKs within a single burst. The modeling results for fast-class connections (13) have thus to be considered as an upper bound.

Fig. 15 compares average TCP send-rates (11) for simulation sources, with modeled fast-class (13) and slow-class (9) connections. We observe a good fit between send-rates for the direct-source and the fast-class model and between the 1 Mbps source and the slow-class model. Due to the burstifier sharing effect, the other sources demonstrate intermediate performance, which, as expected, increases with increases in access bandwidth.

Now, let us discuss the effectiveness of the Bernoulli loss modeling assumption in Section B.3 to model the TCP send-rate for fast-class (13) and slow-class sources (9). As previously mentioned, apart from burst losses, the direct-source in Fig. 13 faces the same network impairments as a fast-class source, for which (13) holds. Similarly the 1 Mbps source is roughly similar to the slow-class source, for which (9) applies. The assumption that losses follow a Bernouilli distribution is validated, at least in our simple showcase scenario, by the close fit between send-rates for the direct-source and the fast-class source, with regard to (13), and between send-rates for the 1 Mbps source and the slow-class source, with regard to (9).

Finally, Fig. 16 shows the simulated correlation benefit. Due to burstifier sharing, the correlation benefit experienced by the sources is well below the upper bound for a fast-class connection, while maintaining the form and properties envisioned by the modeling analysis at the end of Section B.4.

---

(11) The average send-rate for homologous sources.

We observe 1.4 ÷ 3.5 fold improvements in send-rate, implying that correlation benefit could be practically significant.

D. CONCLUSIONS

In this paper, we have investigated the relationship between burstification and the TCP Reno send-rate in an OBS IP optical network. The analysis has identified an interesting phenomenon – referred to here as correlation benefit – capable of increasing the TCP send-rate.

Correlation benefit arises from time correlations among data unit (segment or ACK) loss events and data unit delivery events induced by the burstification mechanism in the OBS network. The number of data units aggregated in the same burst depends on the relationship between the source access bandwidth and the burstification period. The results obtained show that the higher the number of data units aggregated in a burst, the higher is the correlation benefit; the correlation benefit is maximum for loss probabilities equal to the inverse of the maximum congestion window and vanishes as the loss probability approaches the extreme values of 0 and 1.

It should be emphasized that these results apply to the specific case of lossless access networks, and significant increases in send-rate, for burstification periods in the order of ms, can be expected only for high-speed access (e.g., from tens to thousands of Mbps).

E. REFERENCES

[1] B. Mukherjee, “Optical Communication Networks”, McGraw-Hill Series on Computer Communications, 1997

[2] M. Listanti , V. Eramo, R. Sabella, “Architectural and technological issues for future optical Internet networks”, *IEEE Communications Magazine*, Vol. 38, No. 9, September 2000, pp. 82-92

[3] S. Yao, B. Mukherjee, S. Dixit, "Advances in Photonic Packet Switching: An Overview", *IEEE Communications Magazine*, Vol. 38, No. 2, February 2000, pp. 84-94

[4] Callegati, M. Casoni, C. Raffaelli, B. Bostica “Packet Optical Networks for High-Speed TCP-IP Backbones”, *IEEE Communications Magazine*, Vol. 37, No. 1, January 1999, pp. 124-129

[5] P. Gambini et al., “Transparent Optical Packet Switching: Network Architecture and Demonstrators in the KEOPS Project”, IEEE Journal on Selected Area in Communications, Vol. 16, No. 7, September 1998, pp. 1245 -1259

[6] C. Qiao, ”Labeled optical burst switching for IP-over-WDM integration”, *IEEE Communications Magazine*, Vol. 38, No. 9, September 2000, pp. 104-114

[7] C. Qiao, M. Yoo “A Novel Switching Paradigm for Buffer-less WDM Networks”, *Proceedings of Optical Fiber Communication Conference (OFC)*, Paper ThM6, Feb. 1999, pp.177-179.

[8] Y. Chen, J. Turner “WDM Burst Switching for Petabit Capacity Routers”, Proceedings of Milcom, 1999

[9] J. Turner, “Terabit Burst Switching”, *Journal of High Speed Networks*, Vol.8, No.1, 1999, pp. 3-16

[10] C. Qiao, M. Yoo, “Choices, Features and Issues in Optical Burst Switching (OBS)”, *Optical Networking Magazine*, Vol.2, April 1999.

[11] F. Callegati, A.C. Cankaya, Y. Xiong, M. Vandenhoute, “Design issues of optical IP routers for Internet backbone applications”, *IEEE Communications Magazine* , Vol. 37, No. 12 , Dec. 1999, pp. 124-128

[12] J. Xiong, M.Vandenhoute, A.C. Cankaya, “Control Architecture in Optical Burst-Switched WDM Network”, *IEEE Journal on Selected Areas in Communication*, Vol. 18, No. 10, October 2000

[13] M. Yoo, C. Qiao, S. Dixit, “QoS performance of optical burst switching in IP-over-WDM networks”, *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 10, October 2000, pp. 2062-2071

[14] A. Ge, F. Callegati, L.S. Tamil, “On optical burst switching and self-similar traffic” *IEEE Communications Letters* , Vol. 4, No. 3, March 2000, pp. 98-100

[15] A. Detti, M. Listanti, “Application of Tell & Go and Tell & Wait Reservation Strategies in a Optical Burst Switching Network: a Performance Comparison”, *Proceedings of IEEE International Conference on Telecommunication (ICT)*, Vol.2, pp. 540-548, June 2001

[16] M. Neuts, H. Le Vu, M. Zukerman, "Performance Analysis of Optical Composite Burst Switching", *IEEE Communications Letters*, month 2002, pp. 100-102

[17] A. Detti, V. Eramo, M. Listanti, “Performance evaluation of a new technique for IP support in a WDM optical network: optical composite burst switching (OCBS)”, *IEEE Journal of Lightwave Technology*, Vol. 20, No. 2 , Feb. 2002, pp. 154 –165.

[18] John Y. Wei, Ray I. McFarland, “Just-In-Time Signaling for WDM Optical Burst Switching Networks”, *IEEE Journal of Lightwave Technology*, Vol. 18, No. 12, December 2000, pp. 2019-2037

[19] M. Yoo and C. Qiao, “Just-enough-time(JET): A High Speed Protocol for Bursty Traffic in Optical Networks”, Dig. of IEEE/LEOS Summer Topical Mtgs. Technologies for a Global Info. Infrastructure, Aug. 1997, pp. 26–27

[20] R.B. Cohen. “Grid computing and the growth of the Internet“, http://www.ggf.org/meetings/ggf4/Speaker_Sres/Grid%20%Forum%2041.ppt

[21] W. Stevens, “TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms”, RFC2001, January 1997.

[22] A. Detti, M. Listanti, “Impact of Segments Aggregation on TCP Reno Flows in Optical Burst Switching Networks”, *Proceedings of IEEE INFOCOM 2002*, Vol. 3, pp. 1803 – 1812

[23] J. Padhye, V. Firoiu, D.F. Towsley, J.F. Kurose, “ Modeling TCP Reno performance: a simple model and its empirical validation”, *IEEE/ACM Transactions on Networking*, Vol.8, No. 2, April 2000, pp. 133 –145

[24] V. Paxson, M. Allman, “ Computing TCP's Retransmission Timer”, RFC2988, November 2000

[25] T. Lakshman, U. Madhow, “The performance of TCP/IP for network with high bandwidth-delay product”, *IEEE/ACM Transactions on Networking*, Vol. 5, No. 3, June 1997, pp. 336-350

[26] M. Mathis, J. Semke, J. Mahdavi, T. Ott, “The macroscopic behavior of the TCP congestion avoidance algorithm”, *ACM/SIGCOMM Comput. Commun. Rev.*, Vol 27, No. 3, July 1997

[27] "Network Simulator 2" [OnLine] developed by Lawrence Berkeley Network laboratory and University of California Berkeley; http://www.isi.edu/nsnam/ns/

[28] W. Richard Stevens, " TCP/IP Illustrated,Volume 1: The Protocols", Addison-Wesley Professional Computing Series

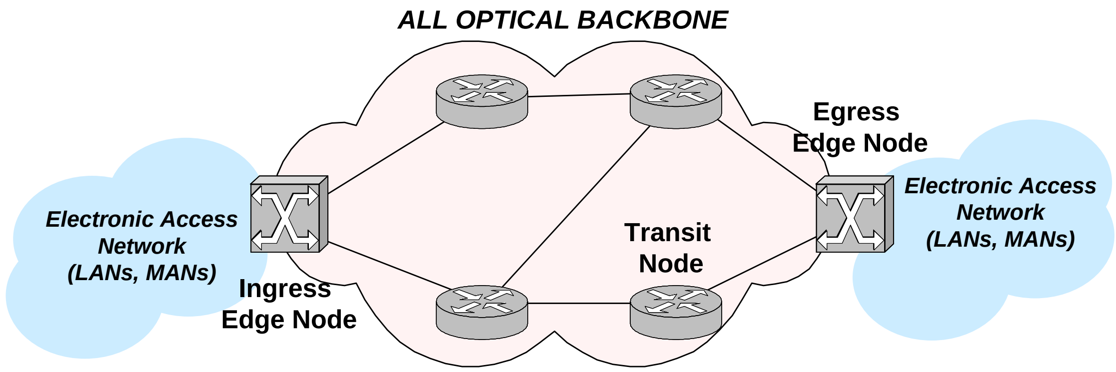

Fig. 1. All optical IP network scenario

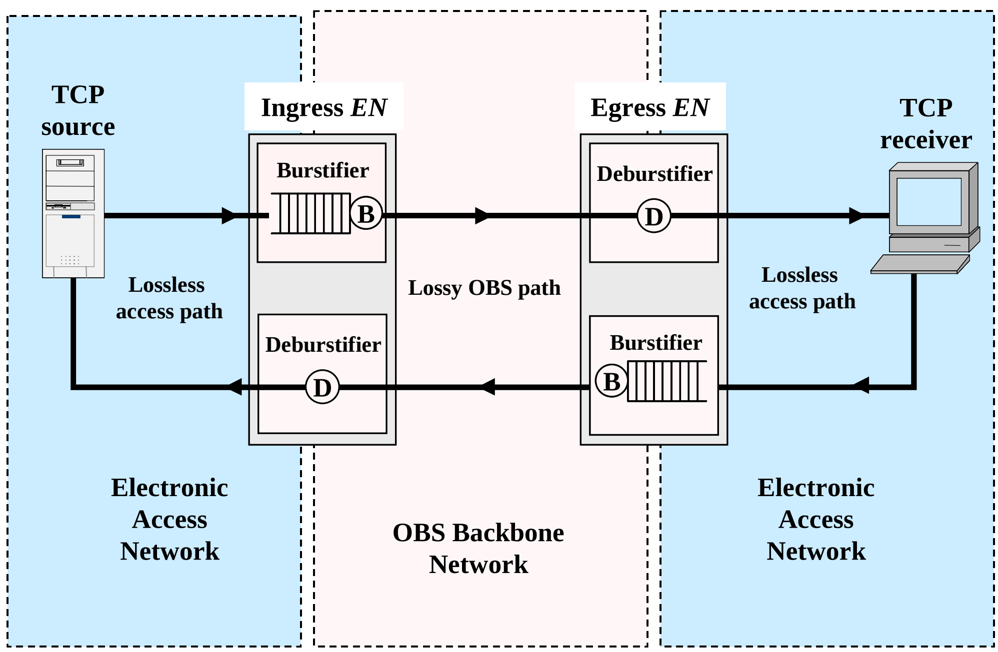

Fig. 2. TCP connection model

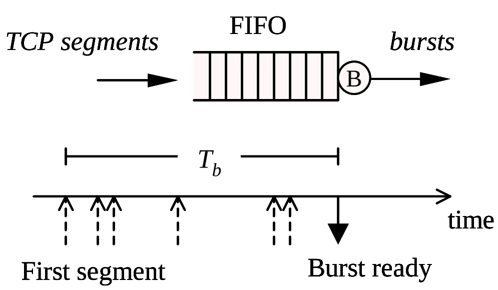

Fig. 3. burstifier logical sketch

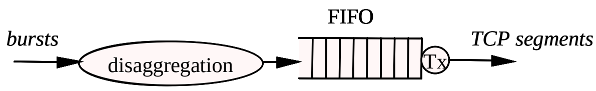

Fig. 4. deburstifier logical sketch

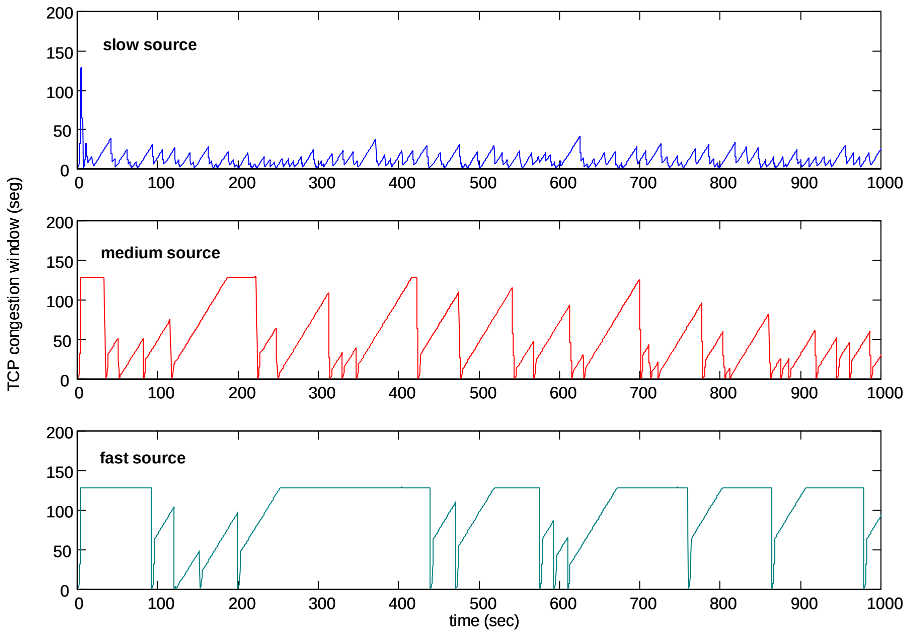

Fig. 5. Examples of variations in the TCP congestion window (cwnd) with slow, medium and fast-class sources (W_m = 128)

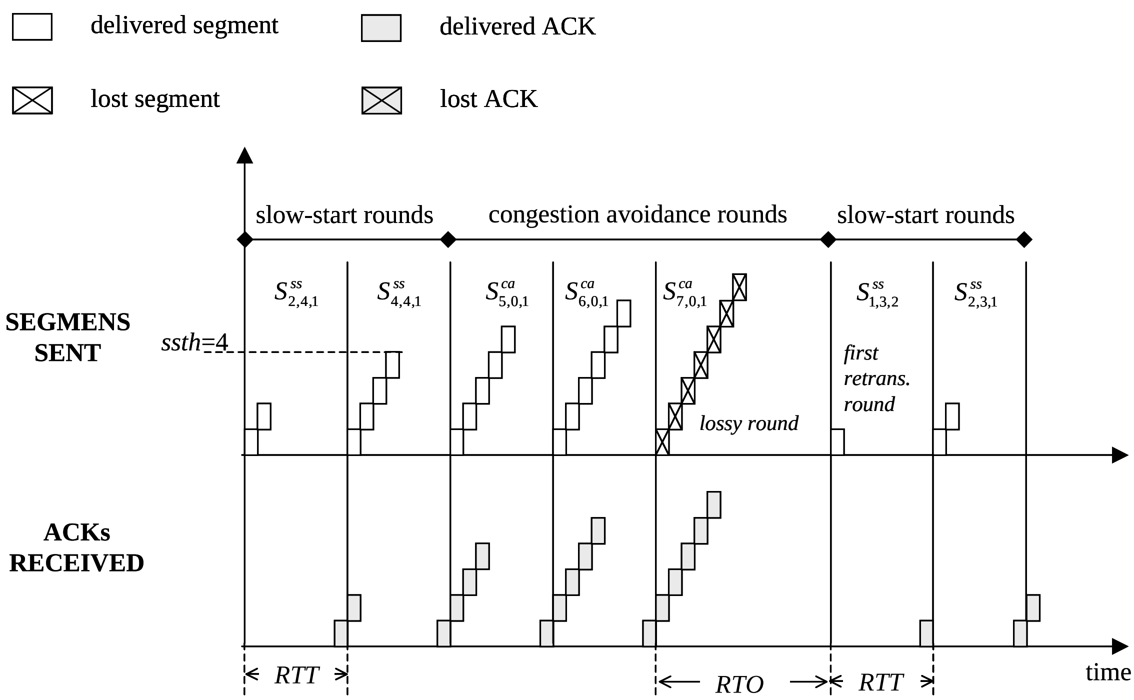

Fig. 6 – Example of round time sequences in which the first retransmission round is successful

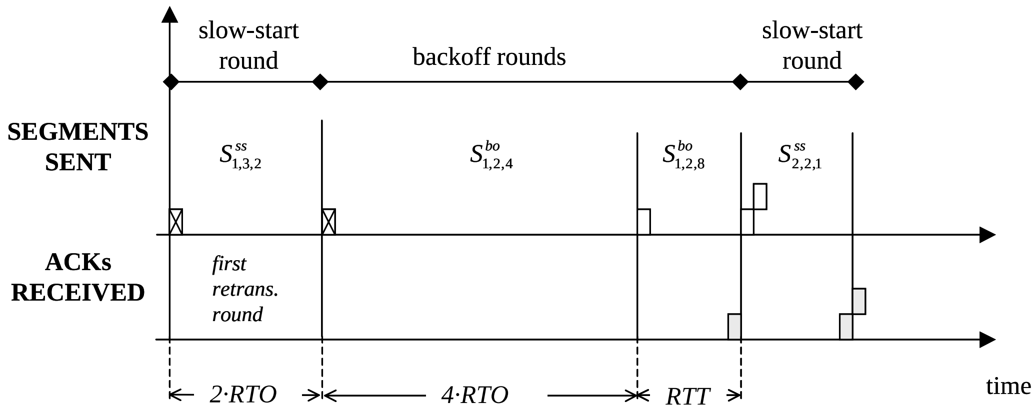

Fig. 7 - Example of round time sequences in which the first retransmission round is lossy

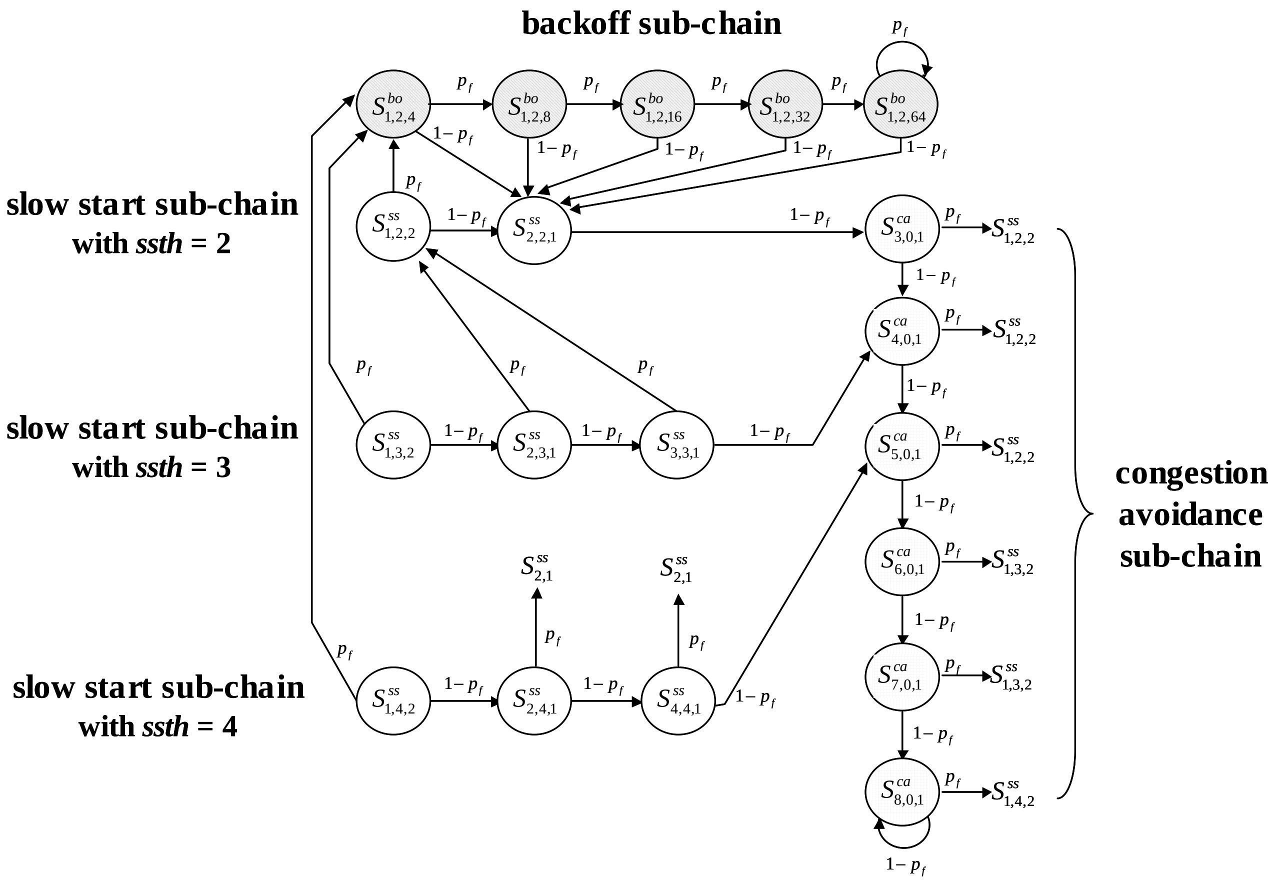

Fig. 8 – Sample state transition diagram for a fast TCP source with W_m=8

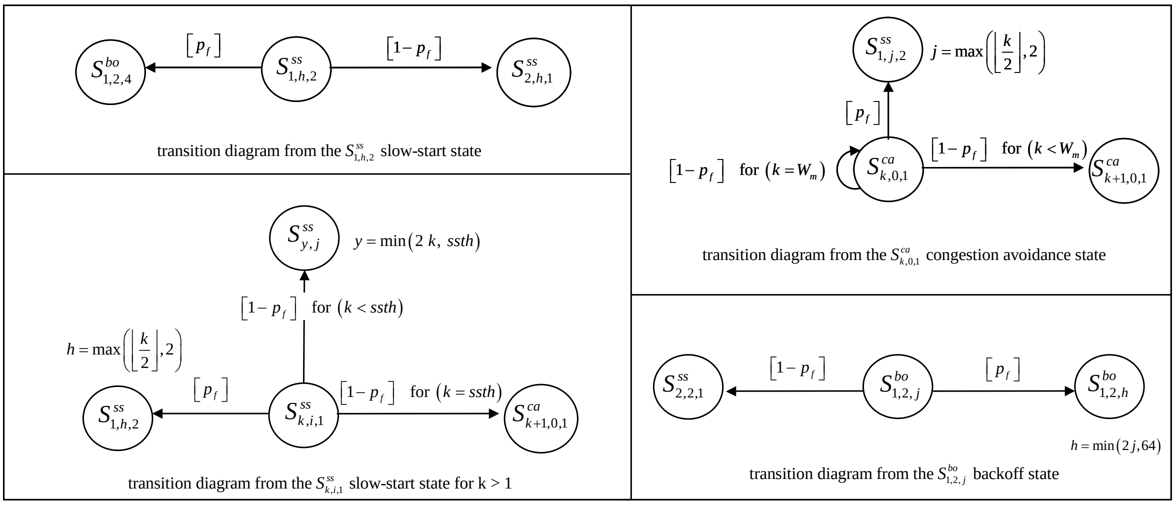

Fig. 9 – General state transition diagrams for a fast TCP source

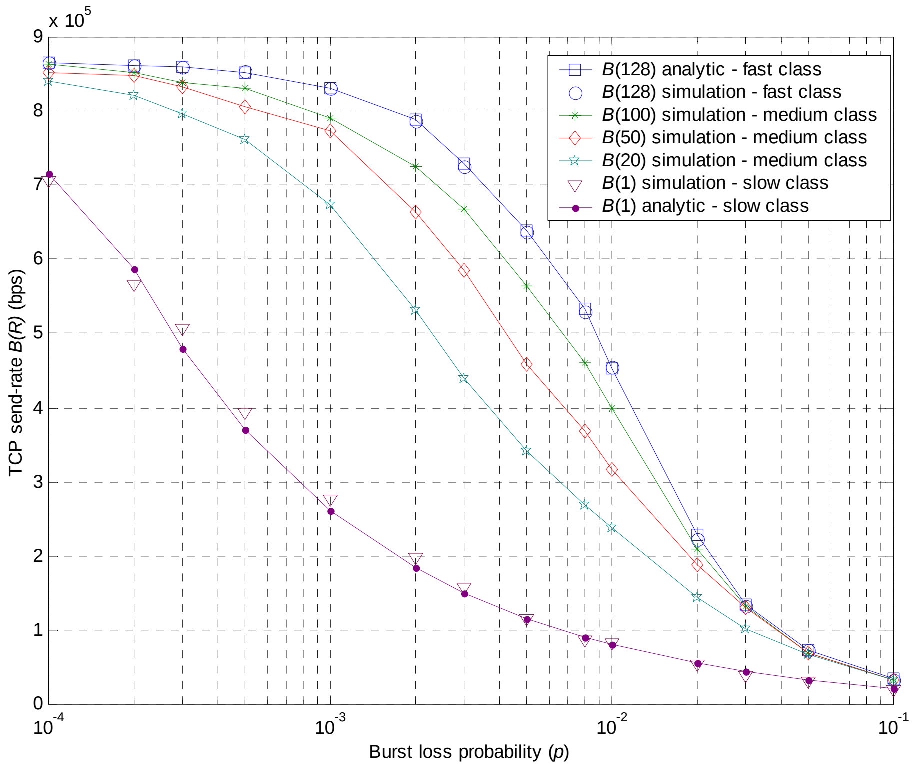

Fig. 10 - TCP send rate $B(R)$ vs. burst loss probability (p) for T_b=3 ms, RTT_0=600 ms, W_m=128, L=512 byte, for different maximum number of segments per burst (R)

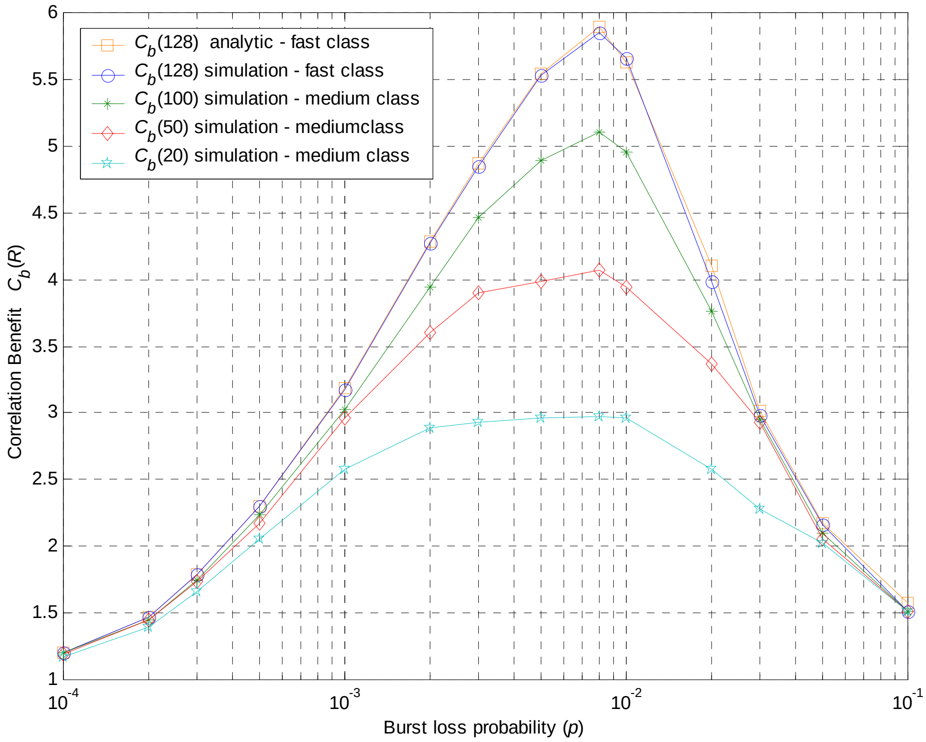

Fig. 11- Correlation benefit $C_b(R)$ vs. burst loss probability (p) with W_m=128 for several values of maximum number of segments per burst (R)

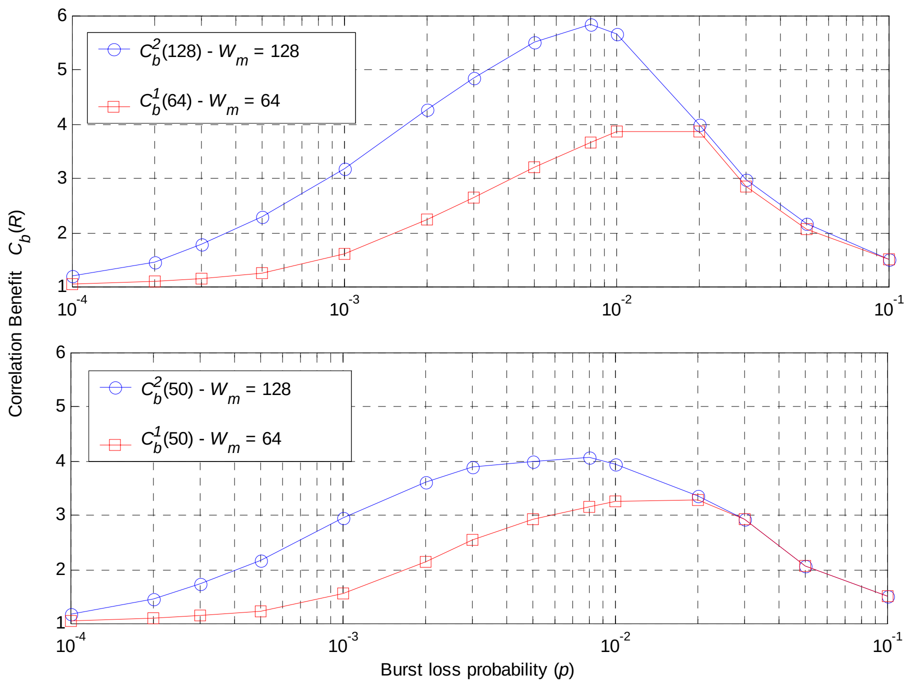

Fig. 12 - Correlation benefit $C_b(R)$ vs. burst loss probability (p) with W_m=128 and W_m=64 for a fast class source (R= W_m) and a medium-class source with R=50

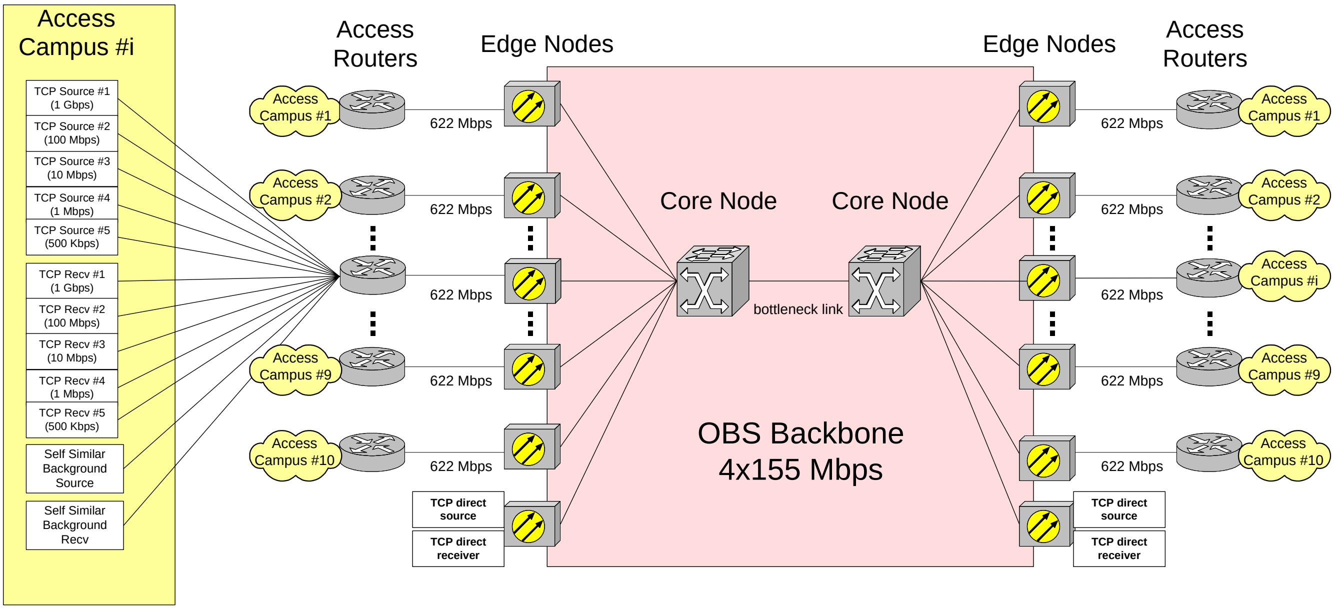

Fig. 13 – Showcase network scenario

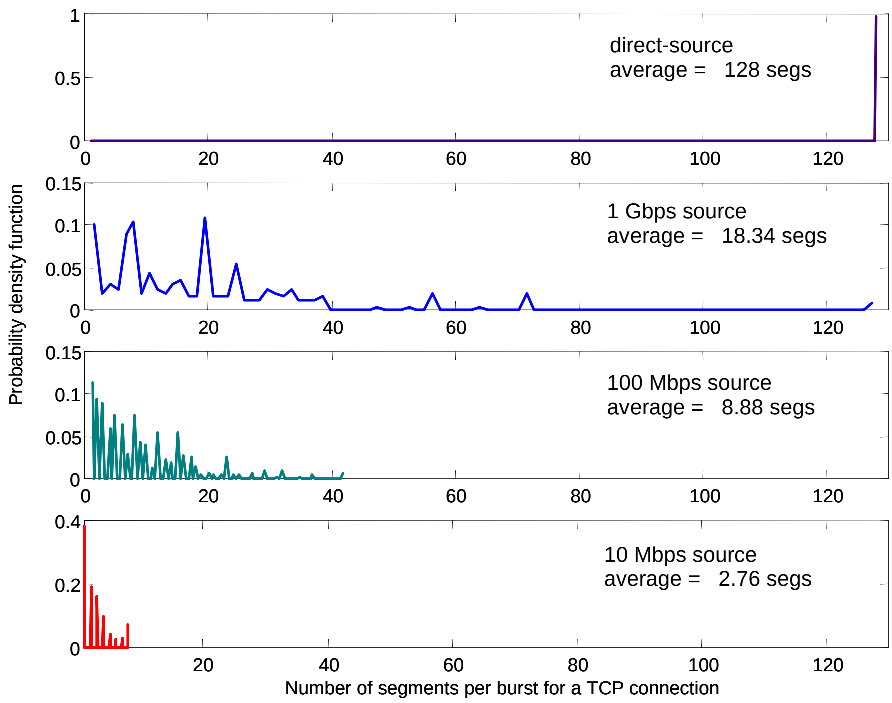

Fig. 14 – Probability density function for the number of segments inserted within a single burst by the TCP connection, for different access bandwidths (direct-source, 1 Gbps, 100 Mbps, 10 Mbps), in the showcase network scenario in the absence of burst loss.

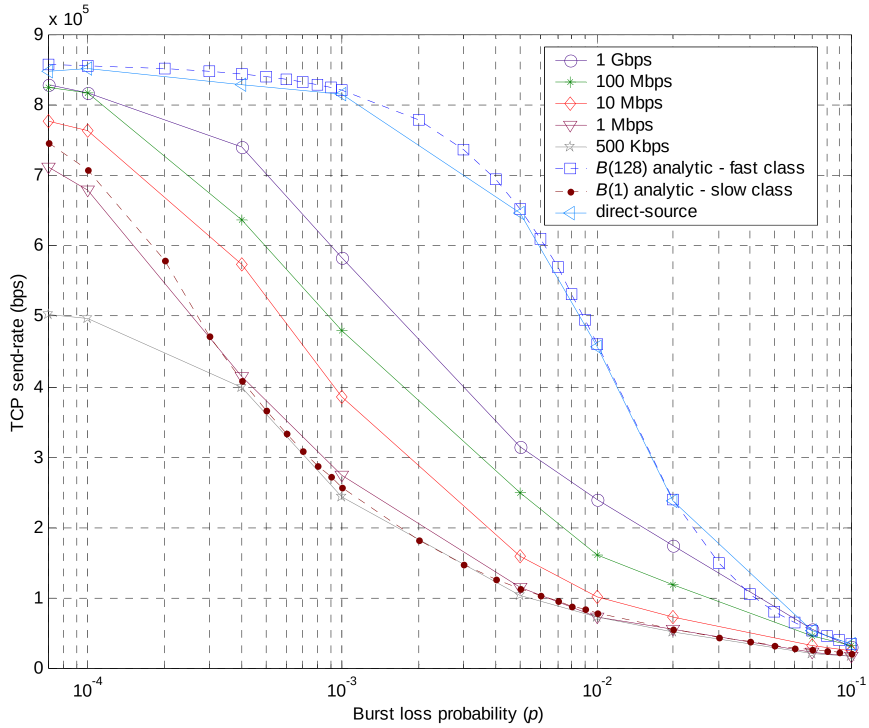

Fig. 15 - TCP send rate vs. the burst loss probability (p) for several values of access bandwidth in the showcase network scenario

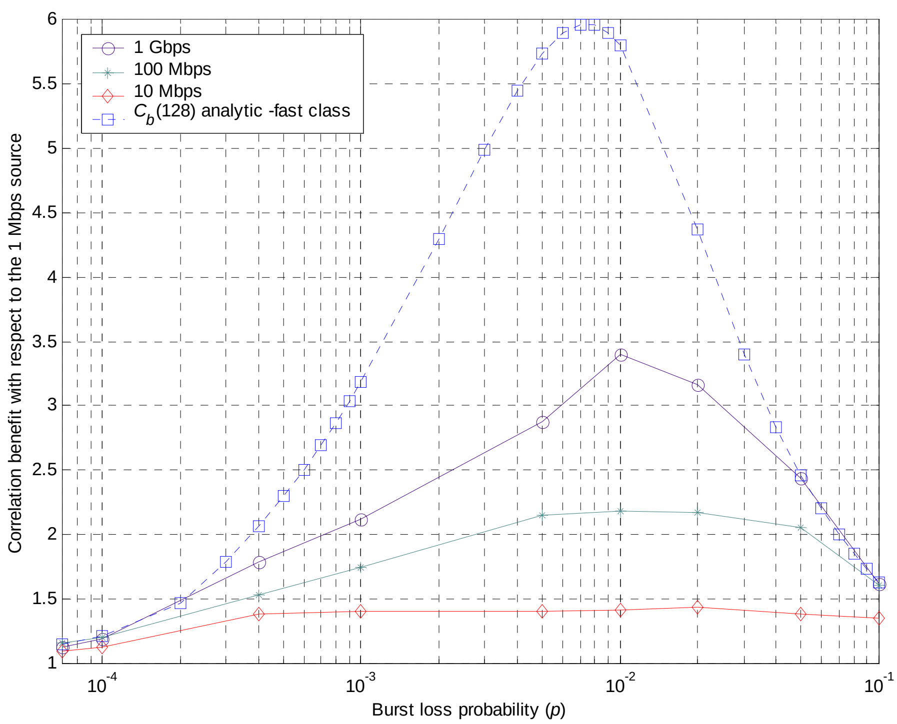

Fig. 16 – Ratio between TCP send-rate (1 Gbps, 100 Mbps, 10 Mbps) and TCP send rate (i.e., Correlation Benefit) for the 1 Mbps source versus the burst loss probability in the showcase network scenario