Modeling and Synthesis of Traffic in Optical Burst-Switched Networks

Ahmad Rostami and Adam Wolisz, *Senior Member, IEEE*

***Abstract*—We analyze burst assembly process as the main building block of the optical burst switching (OBS) paradigm. The analysis is performed for time-based, volume-based, as well as hybrid burst assemblers. Under the assumption that the process of packet arrival to the assembly buffer is Poisson, exact analytical expressions are derived for length and interdeparture time of bursts that are generated by these three classes of assembly algorithms. Furthermore, we consider the issue of generating burst trace, which arises during the performance evaluation of OBS networks through discrete-event simulation. In such a simulation study, a significant part of the simulation time, particularly in the case of a network with a large number of ingress nodes, is used by the implementation of the burst assembly algorithms. This is due to the fact that each data burst is a result of aggregating several short-length packets, which—in a straightforward approach—have to be individually generated and, afterward, "melted" into the burst. We present a novel approach to fast generation of bursts, which is based on the analytical models that are developed for burst length and burst interdeparture time distributions as well as an efficient generation technique (composition) supporting the generation of these distributions. The analysis is completed by numerical results that validate the accuracy of developed models and demonstrate the speedup gain of using proposed burst generation algorithms.**

***Index Terms*—Burst assembly, discrete event simulation (DES), optical burst switching (OBS), traffic modeling.**

I. Introduction

ADVANCES in optical technology and wavelength division multiplexing (WDM) have resulted in a huge amount of capacity available for transmission over optical links. Unfortunately, the possibilities of efficient usage of this capacity for data and multimedia traffic are frequently limited by the fact that switching techniques with fine switching granularity—like packet switching—cannot be efficiently realized in the optical domain using the currently available technology. The reason is mainly due to the fact that the switching times of optical devices are still too large compared with the average packet length and that there is no equivalent to RAM in the optical domain to realize the so-called store-and-forward switching technique. To address the issue, optical burst switching (OBS) has been recently proposed [1], [2] and has attracted much attention from the networking research community.

OBS is a switching paradigm that allows for dynamic allocation of resources in the optical domain at subwavelength granularity. It does so by combining three principles, namely, burst assembly at the edge, one-way out-of-band signaling, and cut-through switching in the core. Burst assembly refers to the process of aggregating small-size packets into bursts at ingress edges of the optical network. By increasing the size of data units, this aggregation makes it possible to relax the requirements on the speed of optical switching. Once a new data burst is ready for transmission at an ingress node, a signaling message is generated and released to the network ahead of the burst transmission. The data burst then follows the message an offset time after it has been sent without waiting for an acknowledgement, i.e., one-way reservation. The role of the signaling message is to inform all the switching nodes along the path of the arrival time of the burst so that they can configure their ports to switch the burst in a cut-through fashion upon its arrival. To that end, each signaling message is electronically processed at every intermediate node after passing through an optoelectrical conversion. However, in order that the data bursts can bypass such conversions at intermediate nodes, signaling messages are transmitted on dedicated WDM channels, i.e., out-of-band signaling.

Assembly algorithms change the statistical characteristics of input traffic that, in turn, could influence the performance of the network. To thoroughly analyze performance implications of the assembly process in OBS networks, the first step would be to study the burst assembly algorithm itself. Nevertheless, only few analytical studies exist in this direction, which have resulted in using simulation-based investigations as the main tool to understand the performance implications of different burst assembly algorithms. On the other hand, in a simulation-based experiment, the process of burst generation takes a significant amount of simulation time since each data burst can be created only by generating multiple individual packets. This issue becomes particularly crucial when OBS networks with a large number of ingress nodes have to be simulated over longer operational time periods. The most common approach to overcome the problem is to oversimplify it by ignoring the impact of burst assembly process on traffic characteristics and by simply assuming that the burst traffic has the same statistical characteristics as the packet traffic—in fact, it is usually assumed that bursts arrive at an OBS core node according to the Poisson process with burst length exponentially distributed. This approach, however, may undermine the credibility of simulation results and produce misleading

Manuscript received March 21, 2007; revised May 31, 2007. This work was supported in part by the European Commission within the Network of Excellence e-Photon/ONe+.

The authors are with the Telecommunication Networks Group (TKN), Technical University of Berlin, 10587 Berlin, Germany (e-mail: rostami@tkn.tu-berlin.de; awo@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JLT.2007.905891

results. In [3], we have demonstrated that the impact of the assembly process on the data loss rate at a core OBS node with small optical buffers may be as high as several orders of magnitude.

Following our work in [4] and to address the shortcomings in the available literature on analytical models for traffic characteristics inside the OBS network, in this paper, we focus on the burst assembly process as one of the main building blocks of the OBS architecture. The work presented in this paper consists of two parts.

- Analytical models are developed for length and interdeparture time distributions of the bursts that are generated by the most popular assembly algorithms. Namely, time-based, volume-based, as well as hybrid burst assembly algorithms are considered. The analytical models are exact; moreover, among others, they include the probability that in the time-based and hybrid assembly algorithms, padding has to be applied in order that the burst length meets a given minimum length requirement, and the probability that in the hybrid assembler, the timer expires.
- The models derived in the first part are synthesized, and a technique referred to as *composition technique* is applied to them to develop simple algorithms that mimic the real assembler's behavior by generating burst traffic of same statistical characteristics. The algorithms can be exploited as a burst traffic generator on a per-flow basis in discrete-event simulation (DES) models to accelerate the simulation.

The rest of this paper is structured as follows. In the following sections, the related works are reviewed, and assumptions that are made for the analysis are introduced. In Section II, popular algorithms for burst assembly and their operation are discussed. In Section III, we present our analytical models for probability density functions (pdfs) of interdeparture time and length of the bursts that are generated by different types of assembly algorithms. In Section IV, we apply the composition technique to the models of Section III to develop fast burst generators that can be used in simulation models. In Section V, we present numerical results that validate the accuracy of developed models and demonstrate that large speedup gain can be achieved by employing proposed burst generators. Final comments in Section VI conclude the work.

A. Related Work

Several studies have dealt with traffic characteristics and associated performance implications inside OBS networks [5]–[8]. Unfortunately, these studies do not present a complete analysis of the most important burst assembly algorithm that simultaneously applies both time and volume thresholds, i.e., hybrid assembly algorithm.

The probability generating function (*pgf*) of burst length has been derived in [5] under the assumption that the arrival process is the time-slotted Bernoulli. Then, based on the pgf, the distribution of burst length is approximated by standard distributions like the gamma distribution. The analysis, however, does not include the distribution of burst interdeparture time. In [6], analytical models were given for the distributions of burst length as well as burst interdeparture time for different burst assembly strategies; nevertheless, the work does not consider the hybrid assembly algorithm. The hybrid algorithm has been studied in [7], where it was claimed that the burst length distribution would have a very narrow range; therefore, only the distribution of the burst interdeparture time has been developed under the assumption that the burst length is fixed. In other words, the hybrid algorithm has been approximated by the volume-based algorithm. In [8], a variation of the time-based assembly algorithm is studied, in which at the end of each aggregation period, a length threshold is applied, and the contents of the buffer are accordingly assembled into a few bursts of a given length. The analysis provides the distribution of the number of bursts that are generated at the end of an assembly period.

Another related issue that has been widely discussed in the literature is the impact of burst assembly on traffic self-similarity. In [9], it was claimed that the self-similarity in the traffic that is injected to the OBS network can be reduced through the assembly process. However, in [10], it was demonstrated that this reduction is only related to short timescales and depends on the load offered to the assembly buffer. More importantly, in [11], it has been discussed that long-range dependence is not relevant for the performance evaluation of an OBS network with no buffering capacity.

B. Assumptions

In developing analytical models in this paper, we make the common assumption that packet-level traffic arrives at the assembly buffer according to the Poisson process, and packet lengths are exponentially distributed. The assumption of Poisson packet arrival can be justified taking into account that as we will observe in Section II, burst assemblers multiplex packet-level traffic of a large number of microflows. There are also some recent measurements of traffic in the Internet that suggest that at the subsecond timescales, packet arrivals in the core network follow the Poisson process [12]. We note here that the timescale that is associated with the burst assembly process is indeed far below one second.

In addition, the exponential assumption for packet length distribution simplifies our analysis and leads us to derive exact analytical models that can be easily synthesized to develop burst generation algorithms. Nevertheless, through simulation, we will present results for the case where arriving IP packets have real measurement-based trimodal distribution. We will show that the developed models also provide a good approximation for this case.

II. Burst Assembly in OBS

The burst assembly process is one of the distinguishing features of OBS architecture that could largely influence the performance of the network. To generate bursts, each OBS ingress node contains a burst assembly unit. The unit receives packet-level traffic at its input and aggregates them into bursts. In its simplest form, an assembly unit is composed of a packet

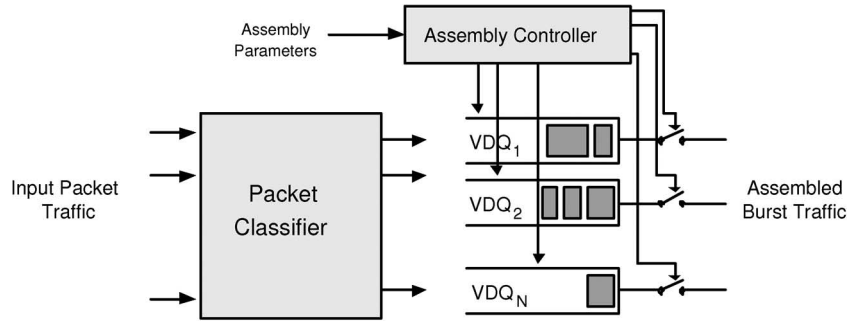

Fig. 1. Burst assembly unit in an ingress OBS node.

classifier, an assembly controller, and several assembly buffers (see Fig. 1). Once a packet arrives to the unit, its destination address (and possibly the QoS class to which it belongs) is checked by the packet classifier. According to the result of the classification, the packet is enqueued in one of the available virtual destination queues (VDQs), where each VDQ is a buffer dedicated to packets destined to a certain egress node and is referred to as the assembly buffer. Therefore, the unit should contain one assembly buffer per destination (and per QoS class). Note that the term destination here refers to an egress node in the OBS domain; thus, packets of the same attributes from different microflows can be multiplexed into the same assembly buffer.

The assembly controller is responsible for making the decisions regarding when contents of each assembly buffer should be aggregated into a burst and released to the network. The controller takes care of scheduling burst generations based on several criteria, some of which are imposed by the network, e.g., maximum and minimum burst length, and others are imposed by QoS requirements of incoming packet traffic, e.g., maximum delay that a packet can tolerate in an assembly buffer. Accordingly, various proposals have been investigated for this purpose (see, e.g., [10] and [13]). The controller may apply different algorithms to different assembly buffers.

In general, burst assembly algorithms may be classified into three major categories, namely, time-based, volume-based, and hybrid algorithms. In the time-based algorithm, the controller is equipped with a timer. Once a packet arrives to an empty assembly buffer, the corresponding timer is set to a time threshold $T_{\rm Th}$. This threshold is determined considering the maximum delay that a packet can tolerate in the ingress node. Then, as soon as the timer expires, all packets in the buffer are aggregated into a burst and sent out. The timer is deactivated when the buffer is emptied. If the length of the burst generated in this way is below a given level $L_{\min}$, padding has to be used to fulfill the minimum burst length requirements. The value of $L_{\min}$ is dictated by the network architecture and depends on the ratio between the number of data and the control channels of WDM links in the network. Specifically, $L_{\min}$ has to be selected large enough to avoid possible conflicts between reservation messages of different data bursts over the control channel [13].

In the volume-based algorithm, the controller checks the aggregate length of packets in the assembly buffer each time a new packet arrives. As soon as the aggregate length exceeds a predefined threshold $L_{\rm Th}$, all packets in the buffer are assembled into a new burst. This threshold must be larger than the minimum burst length requirement of the network and is usually selected with respect to maximization of utilization of resources over the network [13]. Algorithms of this category are usually not recommended for delay-sensitive traffic because there is no guarantee on the upper-bound delay that a packet may experience in the buffer.

Alternatively, in the hybrid assembly algorithm, the control unit keeps track of both the aggregate volume of packets in the buffer and the time elapsed since the first packet has arrived. That is, the timer is set to $T_{\rm Th}$ once a packet arrives and finds the buffer empty, and the length of the queue is compared against a length threshold $L_{\rm Th}$ upon each new arrival. Then, a new burst will be generated when either the timer goes off or the volume threshold is exceeded. In either case, the timer will be deactivated after the buffer is emptied. In an algorithm of this category, packet arrival rate determines which criterion, between time and volume, will be used to generate a new burst at a given time. That is, if the arrival rate is below a specific level, a new burst will be generated $T_{\rm Th}$ units of time after the first packet has arrived. However, if the arrival rate is heavy enough, bursts of length $L_{\rm Th}$ will be generated back-to-back so that no packet will encounter a maximum assembly delay of $T_{\rm Th}$. In the former case, it is likely that the timer expires while the aggregate length is less than the minimum burst length requirement. In that case, padding has to be used.

III. Burst Traffic Modeling

In this section, we consider modeling of burst traffic generated by the time-based, volume-based, and hybrid assemblers. Consider a single assembly buffer in an OBS ingress node. Let packets arrive to the buffer according to the Poisson process with rate λ, and also the packet length be independent and identically distributed according to the exponential distribution with mean μ^{-1}, i.e., the offered load to the assembler is $\rho = \lambda\mu^{-1}$. In the following analysis, we let X and Z be random variables denoting the length and interdeparture time of the bursts leaving the assembler's output, respectively.

A. Time-Based Algorithm

Consider a time-based assembly algorithm with the assembly parameter and the minimum burst length requirement set to $T_{\rm Th}$ and $L_{\min}$, respectively. It was shown in [4] that for the case

where there is no minimum requirement on the burst length, the pdf's of the burst length and the burst interdeparture time are, respectively, given by

$$f_{\mathrm{T}}(x) = \sum_{n=0}^{\infty} \frac{\mu(\mu x)^n}{n!} e^{-\mu x} P_{\mathrm{T}}[N=n], \qquad (x>0) \quad (1)$$

and

$$g_{\mathrm{T}}(z) = \lambda e^{-\lambda(z-T_{\mathrm{Th}})}, \qquad (z \geq T_{\mathrm{Th}}) \quad (2)$$

where

$$P_{\mathrm{T}}[N=n] = \frac{(\lambda T_{\mathrm{Th}})^n}{n!} e^{-\lambda T_{\mathrm{Th}}}, \qquad (n=0,1,\ldots). \quad (3)$$

The probability that the burst length is smaller than $L_{\min}$, i.e., padding is triggered, would be equal to

$$\begin{aligned} P_{\mathrm{T,Pad}} &= P(X < L_{\min}) \\ &= \int_0^{L_{\min}} f_{\mathrm{T}}(x)dx \\ &= \sum_{n=0}^{\infty} \frac{\gamma(n+1, \mu L_{\min})}{n!} P_{\mathrm{T}}[N=n] \end{aligned} \quad (4)$$

where $\gamma(a,x) = \int_0^x t^{a-1}e^{-t}dt$ is the incomplete gamma function. Therefore, the pdf of the burst length after padding would be given by

$$\begin{aligned} f_{\mathrm{T,Pad}}(x) &= P_{\mathrm{T,Pad}}\delta(x - L_{\min}) \\ &\quad + \sum_{n=0}^{\infty} \frac{\mu(\mu x)^n}{n!} e^{-\mu x} P_{\mathrm{T}}[N=n], \qquad (x \geq L_{\min}) \end{aligned} \quad (5)$$

where $\delta(\cdot)$ is the Dirac delta function.

B. Volume-Based Algorithm

For a volume-based algorithm with the assembly parameter L_{Th}, it was shown in [4] that the pdfs of the burst length and the burst interdeparture time are, respectively, given by

$$f_{\mathrm{V}}(x) = \mu e^{-\mu(x-L_{\mathrm{Th}})}, \qquad (x \geq L_{\mathrm{Th}}) \quad (6)$$

and

$$g_{\mathrm{V}}(z) = \sum_{n=0}^{\infty} \frac{\lambda(\lambda z)^n}{n!} e^{-\lambda z} P_{\mathrm{V}}[N=n], \qquad (z>0) \quad (7)$$

where

$$P_{\mathrm{V}}[N=n] = \frac{(\mu L_{\mathrm{Th}})^n}{n!} e^{-\mu L_{\mathrm{Th}}}, \qquad (n=0,1,\ldots). \quad (8)$$

C. Hybrid Algorithm

Consider a hybrid algorithm in which its minimum burst length requirement, volume threshold, and time threshold are set to $L_{\min}$, L_{Th}, and T_{Th}, respectively, where $L_{\min} < L_{\mathrm{Th}}$. As depicted in Fig. 2, once a packet enters an empty assembly buffer, the timer is set to T_{Th}. Therefore, the probability that the timer expires before the volume criterion L_{Th} is met can be expressed as

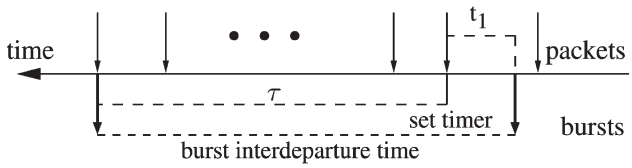

Fig. 2. Burst formation process under the hybrid assembly algorithm.

$$P_{T\mathrm{out}} = 1 - P(\tau < T_{\mathrm{Th}}) \quad (9)$$

where τ is the random variable characterizing the time that is needed until enough packets arrive to the assembly buffer so that a burst can be generated using the volume criterion. We recall from the volume-based assembler that the conditional pdf of τ given the number of arrivals follows the Erlang distribution, i.e.,

$$h(\tau|n) = \begin{cases} \delta(\tau), & n=0 \\ \frac{\lambda(\lambda\tau)^{n-1}}{(n-1)!} e^{-\lambda\tau}, & n \geq 1 \end{cases} \quad (10)$$

where n is the number of arriving packets whose aggregate length is L_{Th}. The probability function of n is given in (8). Note that $n=0$ is associated with the situation that the size of the first arrival to the buffer is larger than the volume threshold. In that case, the timer will be immediately deactivated after it is set; thus, τ is equal to zero. For the case of $n \geq 1$, we would have

$$\begin{aligned} P(\tau < T_{\mathrm{Th}}|n) &= \int_0^{T_{\mathrm{Th}}} \frac{\lambda(\lambda\tau)^{(n-1)}}{(n-1)!} e^{-\lambda\tau} d\tau \\ &= \frac{\gamma(n, \lambda T_{\mathrm{Th}})}{(n-1)!}, \qquad (n \geq 1). \end{aligned} \quad (11)$$

Therefore, P_{Tout} can be computed as

$$P_{T\mathrm{out}} = 1 - e^{-\mu L_{\mathrm{Th}}} - \sum_{n=1}^{\infty} \frac{\gamma(n, \lambda T_{\mathrm{Th}})}{(n-1)!} P_{\mathrm{V}}[N=n]. \quad (12)$$

Now, let us derive the pdf of the length of the bursts that are generated by the assembly buffer under study. The bursts that leave the assembly buffer before the timer expires would be larger than L_{Th} in length; thus, the pdf of their length would be equivalent to that of the bursts generated by the pure volume-based algorithm, as given in (6). On the other hand, if the timer does expire, the burst length would be smaller than L_{Th}. Therefore, the pdf of the burst length before possible padding would be

$$f_{\mathrm{H}}(x) = \begin{cases} P_{\mathrm{Tout}} f_a(x), & x < L_{\mathrm{Th}} \\ (1 - P_{\mathrm{Tout}})\mu e^{-\mu(x-L_{\mathrm{Th}})}, & x \geq L_{\mathrm{Th}} \end{cases} \quad (13)$$

where $f_a(x)$ is the pdf of the burst length when the timer expires, which is equal to the conditional pdf of $f_{\rm T}(x)$, as given by (1), given $(x < L_{\rm Th})$, i.e.,

$$\begin{aligned} f_a(x) &= f_{\rm T}(x|x < L_{\rm Th}) \\ &= \frac{f_{\rm T}(x)}{K_1}, \qquad (x < L_{\rm Th}) \\ &= \frac{1}{K_1}\left(\sum_{n=0}^{\infty} \frac{\mu(\mu x)^n}{n!} e^{-\mu x} P_{\rm T}[N=n]\right), \quad (x < L_{\rm Th}) \end{aligned} \tag{14}$$

where K_1 is the normalization factor and can be computed as [see (4)]

$$\begin{aligned} K_1 &= P(X < L_{\rm Th}) \\ &= \sum_{n=0}^{\infty} \frac{\gamma(n+1, \mu L_{\rm Th})}{n!} P_{\rm T}[N=n]. \end{aligned} \tag{15}$$

To calculate the probability that the padding is applied, we can write

$$\begin{aligned} P_{\rm H,Pad} &= P(X < L_{\min}) \\ &= \int_0^{L_{\min}} f_{\rm H}(x)dx \qquad (L_{\min} < L_{\rm Th}) \\ &= \int_0^{L_{\min}} P_{\rm Tout} f_a(x)dx \\ &= \frac{P_{\rm Tout}}{K_1}\left(\sum_{n=0}^{\infty} \frac{\gamma(n+1, \mu L_{\min})}{n!} P_{\rm T}[N=n]\right). \end{aligned} \tag{16}$$

Finally, the pdf of the burst length after padding is expressed in (17), shown at the bottom of the page.

Now, we turn to the burst interdeparture time. As depicted in Fig. 2, the burst interdeparture time is composed of two parts, namely, t_1, which is the time required until a packet arrives after the last burst has been sent out, and τ, which is the time between when the first packet arrives and when the new burst is generated, i.e.,

$$Z = t_1 + \tau. \tag{18}$$

For those bursts that are released due to the timer expiration, τ would be deterministic and equal to $T_{\rm Th}$; thus, the pdf of the burst interdeparture time would be equal to that of the pure time-based assembler, as given in (2). For other bursts, however, the interdeparture time would be equal to the sum of t_1 and τ given $\tau < T_{\rm Th}$. Therefore

$$\begin{aligned} g_{\rm H}(z) = P_{\rm Tout}\lambda e^{-\lambda(z-T_{\rm Th})}U(z-T_{\rm Th}) \\ + (1-P_{\rm Tout})g_a(z), \qquad (z>0) \end{aligned} \tag{19}$$

where $U(\cdot)$ is the unit step function. Also, $g_a(z)$ can be calculated by convolving the density functions of t_1 and τ as follows:

$$g_a(z) = s(t_1) * h(\tau|\tau < T_{\rm Th}) \tag{20}$$

where due to the Poisson packet arrival, the pdf of t_1 is given by $s(t_1) = \lambda e^{-\lambda t_1}$ $(t_1 \geq 0)$. Also, from (10) and (11), we would have

$$h(\tau) = e^{-\mu L_{\rm Th}}\delta(\tau) + \sum_{n=1}^{\infty} \frac{\lambda(\lambda\tau)^{n-1}}{(n-1)!} e^{-\lambda\tau} P_{\rm V}[N=n]. \tag{21}$$

Therefore, we can write

$$h(\tau|\tau < T_{\rm Th}) = \frac{h(\tau)}{P(\tau < T_{\rm Th})} \qquad (\tau < T_{\rm Th}) \tag{22}$$

where

$$P(\tau < T_{\rm Th}) = e^{-\mu L_{\rm Th}} + \sum_{n=1}^{\infty} \frac{\gamma(n, \lambda T_{\rm Th})}{(n-1)!} P_{\rm V}[N=n]. \tag{23}$$

Substituting (22) in (20) and solving the convolution, we will obtain (24), shown at the bottom of the page (see the Appendix).

Accordingly, the burst interdeparture time can be expressed as in (25), shown at the bottom of the next page.

After having derived exact expressions for the pdf of the burst interdeparture time, we now take some practical considerations into account and approximate $P_{\rm Tout}$ and $g_{\rm H}(z)$ with two simpler expressions. In practice, the length threshold $L_{\rm Th}$ is usually

$$f_{\rm H,Pad}(x) = \begin{cases} P_{\rm H,Pad}\delta(x - L_{\min}) + P_{T\rm out} f_a(x), & L_{\min} \leq x < L_{\rm Th} \\ (1-P_{\rm Tout})\mu e^{-\mu(x-L_{\rm Th})}, & x \geq L_{\rm Th} \end{cases} \tag{17}$$

$$g_a(z) = \begin{cases} \frac{1}{P(\tau<T_{\rm Th})}\left(\lambda e^{-\mu L_{\rm Th}} e^{-\lambda z} + \sum_{n=1}^{\infty} \frac{\lambda(\lambda z)^n}{n!} e^{-\lambda z} P_{\rm V}[N=n]\right), & 0 < z \leq T_{\rm Th} \\ \frac{1}{P(\tau<T_{\rm Th})}\left(\lambda e^{-\mu L_{\rm Th}} e^{-\lambda z} + \sum_{n=1}^{\infty} \frac{\lambda(\lambda T_{\rm Th})^n}{n!} e^{-\lambda z} P_{\rm V}[N=n]\right), & z > T_{\rm Th} \end{cases} \tag{24}$$

much larger than μ^{-1} so that $e^{-\mu L_{\rm Th}} \simeq 0$. As a result, $P_{\rm Tout}$ can be approximated by

$$P_{\rm Tout} \simeq 1 - \sum_{n=1}^{\infty} \frac{\gamma(n, \lambda T_{\rm Th})}{(n-1)!} P_{\rm V}[N=n]. \tag{26}$$

Simplifying (25) and substituting $P_{\rm Tout}$ with the expression in (26), $g_{\rm H}(z)$ can be approximated by (27), shown at the bottom of the page.

IV. Generation of Burst Traffic in a Discrete-Event Simulator

In this section, we analyze the pdfs that were derived in the last section to develop efficient algorithms for burst traffic generation in a DES model. Let us start with the time-based burst assembly. The pdf in (5) is composed of two parts, where the second part includes the product of two functions, namely, the Erlang pdf with $(n+1)$ phases and mean $(n+1)\mu^{-1}$ and the Poisson distribution with mean $\lambda T_{\rm Th}$. Moreover, we notice that the second part in (5) constitutes a *convex* combination of an infinite number of Erlang pdf's, each with a different mean and number of phases. By definition, a pdf $f(y)$ is said to be a convex combination of other pdf's f_1, $f_2, \ldots$, if it can be written as

$$f(y) = \sum_{i=0}^{\infty} f_i(y) p_i \tag{28}$$

where $p_i \geq 0$, $\sum_{i=0}^{\infty} p_i = 1$, and each f_i is a pdf [14].

An interesting feature of a complex density function that can be expressed in the form of a convex combination of other simple density functions, like that in (5), is that its samples can be generated using the *composition* technique [14]. In this technique, each sample of a random variable Y with the pdf $f(y)$ is generated by 1) generating a positive random integer I such that $P(I=i) = p_i$, where $i = 0, 1, \ldots$, and 2) returning Y with the pdf $f_I(y)$. More specifically, a sample of the random variable with the pdf given in (5) can be easily generated in two steps. First, a positive random integer n is generated using the Poisson distribution with mean $\lambda T_{\rm Th}$; then, a random number is generated using the Erlang distribution with $(n+1)$ phases and mean $(n+1)\mu^{-1}$.

Algorithm 1 presents the pseudo-code of a simple algorithm that exploits the concept of the composition technique to generate burst traffic with the burst interdeparture time and burst length density functions that are given in (2) and (5), respectively. Referring to the algorithm $exp(\alpha)$, $Poisson(\beta)$, and $Erlang(n, n\theta)$ are exponential random number generator with mean α, Poisson random number generator with mean β, and Erlang random number generator with n phases and mean $n\theta$, respectively. The algorithm works as follows. First, the number of packets in the current burst is determined. Then, the length of the current burst is computed using the $Erlang$ random number generator and taking into account that the length should not be smaller than $L_{\rm min}$. Finally, interdeparture time is computed by generating an exponentially distributed random number and adding it to $T_{\rm Th}$. Following the same approach for the volume-based case, we will get the algorithm depicted in Algorithm 2, which works in a similar way that the time-based algorithm does.

To develop an algorithm for generating bursts with the hybrid policy, which is more complicated than the time- and volume-based cases, we use a combination of the first two algorithms, as depicted in Algorithm 3. The algorithm works as follows. In the first step, the value of $P_{\rm Tout}$ has to be computed using (12) or (26). Note that the value of $P_{\rm Tout}$ for a given set of input traffic and assembly parameters is fixed; thus, it is computed only once at the beginning of the simulation. After having computed t_1 (see Fig. 2), a sample is drawn from a uniform random number generator in the range (0, 1). The value of U will then be used to decide which procedure, between time-based and volume-based, has to be followed to generate the current burst. In the time-based case, care has to be taken so that the burst length is smaller than $L_{\rm Th}$. This is achieved through a conditional probability. Similarly, in the volume-based case, the burst interdeparture time minus t_1 must be smaller than $T_{\rm Th}$. In either case, the conditional probability is achieved through implementing a while loop.

The main advantage of using the presented algorithms over direct implementation of the burst assembly algorithms is simulation speedup. Specifically, if the assembly algorithm is directly implemented in a simulation model, the model first has to generate a large number of packets and then assemble them into bursts, where this process has to be repeated for every single burst. In other words, it would take $O(MN)$ time to generate N bursts, where M is the average number of packets

$$g_{\rm H}(z) = \begin{cases} P_{\rm Tout}\lambda e^{-\lambda(z-T_{\rm Th})} + \frac{1-P_{\rm Tout}}{P(\tau<T_{\rm Th})}\left(\lambda e^{-\mu L_{\rm Th}} e^{-\lambda z} + \sum_{n=1}^{\infty} \frac{\lambda(\lambda T_{\rm Th})^n}{n!} e^{-\lambda z} P_{\rm V}[N=n]\right), & z > T_{\rm Th} \\ \frac{1-P_{\rm Tout}}{P(\tau<T_{\rm Th})}\left(\lambda e^{-\mu L_{\rm Th}} e^{-\lambda z} + \sum_{n=1}^{\infty} \frac{\lambda(\lambda z)^n}{n!} e^{-\lambda z} P_{\rm V}[N=n]\right), & 0 < z \leq T_{\rm Th} \end{cases} \tag{25}$$

$$g_{\rm H}(z) \simeq \begin{cases} \left(1 - \sum_{n=1}^{\infty} \frac{\gamma(n,\lambda T_{\rm Th})}{(n-1)!} P_{\rm V}[N=n]\right) \lambda e^{-\lambda(z-T_{\rm Th})} + \sum_{n=1}^{\infty} \frac{\lambda(\lambda T_{\rm Th})^n}{n!} e^{-\lambda z} P_{\rm V}[N=n], & z > T_{\rm Th} \\ \sum_{n=1}^{\infty} \frac{\lambda(\lambda z)^n}{n!} e^{-\lambda z} P_{\rm V}[N=n], & 0 < z \leq T_{\rm Th} \end{cases} \tag{27}$$

per burst. This, however, implies that the more the average number of packets per burst is, the slower the corresponding simulation model works. This is fortunately not the case for our algorithms, in which the time needed to generate N bursts is $O(N)$. In the next section, we will numerically evaluate the speedup gain that is achieved using these algorithms.

Algorithm 1 Burst traffic generation according to the time-based algorithm with packet arrival rate λ, average packet length μ^{-1}, and assembly parameters $T_{\rm Th}$ and $L_{\rm min}$.

$k \leftarrow$ number of bursts to be generated
λ, μ, $T_{\rm Th}$, $L_{\rm min} \leftarrow$ initialize
for $i := 1, i \leq k, i++$ **do**
  $n \leftarrow$ Poisson$(\lambda T_{\rm Th})$
  $x \leftarrow$ Erlang$(n+1, (n+1)\mu^{-1})$
  if $x < L_{\rm min}$ **then**
    $x \leftarrow L_{\rm min}$
  end if
  $z \leftarrow T_{\rm Th} +$ exponential(λ^{-1})
  wait (z)
  generate and send a burst of length x
end for

V. Evaluation and Discussion

In this section, the characteristics of traffic generated by the burst assemblers are studied through simulation and analysis. To validate the correctness of the analysis, we first simulated the system with the same assumptions made for the analysis. It was observed that the simulation results exactly match those obtained from the analysis as well as those achieved when the proposed burst generation algorithms are implemented. This was not surprising, as the analytical models derived in this paper are exact. As a result, and for the sake of brevity, we decided not to present these results here. Instead, we report on the results that validate the usage of our analytical models in the more realistic case of trimodal packet length distribution for the arrivals as measured in the real Internet. Additionally, simulation has been used to evaluate the speedup gain that can be achieved in a simulation experiment when the algorithms of Section IV are employed.

Algorithm 2 Burst traffic generation according to the volume-based algorithm with packet arrival rate λ, average packet length μ^{-1}, and assembly parameter $L_{\rm Th}$.

$k \leftarrow$ number of bursts to be generated
λ, μ , $L_{\rm Th} \leftarrow$ initialize
for $i := 1, i \leq k, i++$ **do**
  $n \leftarrow$ Poisson$(\mu L_{\rm Th})$
  $z \leftarrow$ Erlang$(n+1, (n+1)\lambda^{-1})$
  $x \leftarrow L_{\rm Th} +$ exponential(μ^{-1})
  wait (z)
  generate and send a burst of length x
end for

Table I shows the trimodal packet length distribution that has been used for the simulation [15]. All the simulation results that follow are obtained through explicitly implementing the

TABLE I
Trimodal Packet Length Distribution

Length (bytes)	Probability
40	0.5
552	0.3
1500	0.2

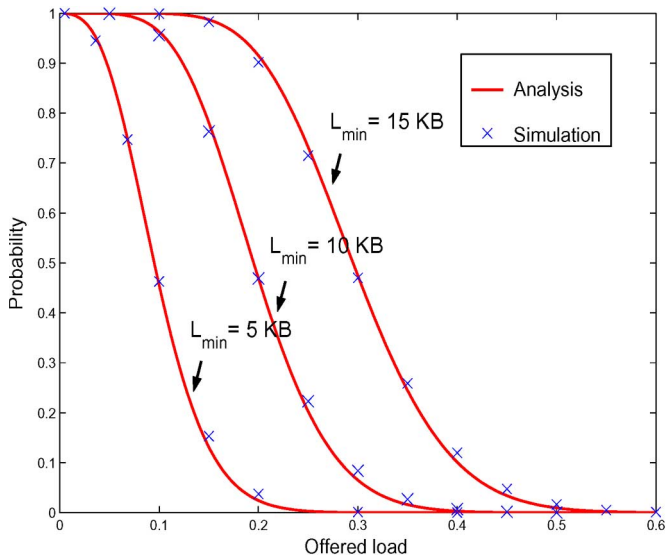

Fig. 3. Probability that the burst length is less than $L_{\rm min}$ as a function of load in the time-based assembler with $T_{\rm Th} = 4$ ms.

assembly algorithms in the discrete-event simulator OMNeT++ [16]. Input packets to the assembler are generated according to the Poisson process, and their length follows the trimodal distribution, as given in Table I. The results of the simulation are then compared with those obtained from the analytical models. In either case, the average packet length is set to 485.6 B. The capacity of the traffic flow that feeds the assembly buffer under consideration is assumed to be equal to 100 mb/s. We further decided to present the cumulative distribution functions (CDFs), instead of the pdfs, since they are easier to deal with.

Let us start with the time-based burst assembler. The assembly parameter $T_{\rm Th}$ is set to 4 ms. First, we evaluate the probability that generated bursts are smaller than a given minimum burst length requirement $L_{\rm min}$ and, thus, require padding. For a given assembly threshold, the load as well as $L_{\rm min}$ are the factors that influence this probability. Fig. 3 shows $P_{\rm T,Pad}$ as a function of load for different values of $L_{\rm min}$. As expected, the probability decreases with load. It is also seen that the results of the simulation match those of the analytical models very closely. To analyze the impact of $P_{\rm T,Pad}$ on the traffic characteristics, we consider the CDF of the burst length and the burst interdeparture time at both light and heavy loads while keeping $L_{\rm min}$ fixed at 15 kB (see Fig. 4). Plots of Fig. 4 clearly illustrate the impact of the padding on the distribution of the burst length. In fact, depending on the load and $L_{\rm min}$, fixed-size bursts could dominate the bursts generated by the assembler (see the case with $\rho = 0.3$ in Fig. 4). From the figure, it is also evident that the error introduced by the exponential packet length assumption is very narrow. The CDFs of the burst

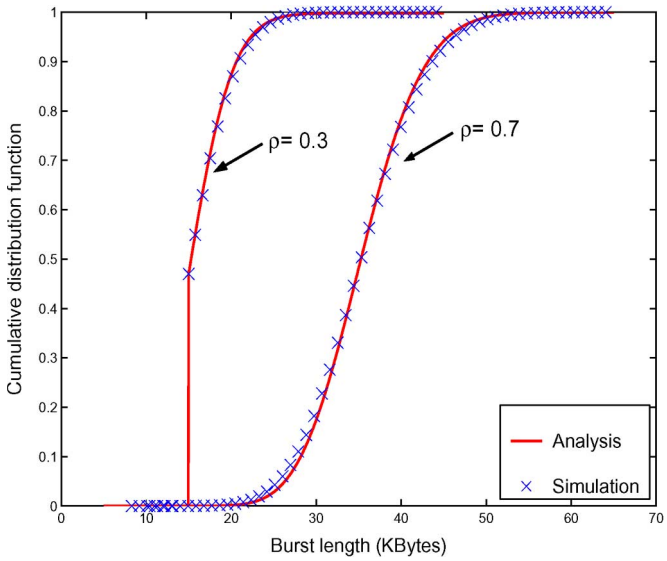

Fig. 4. CDF of the length of the bursts generated by the time-based assembler with $T_{\rm Th} = 4$ ms and $L_{\rm min} = 15$ kB.

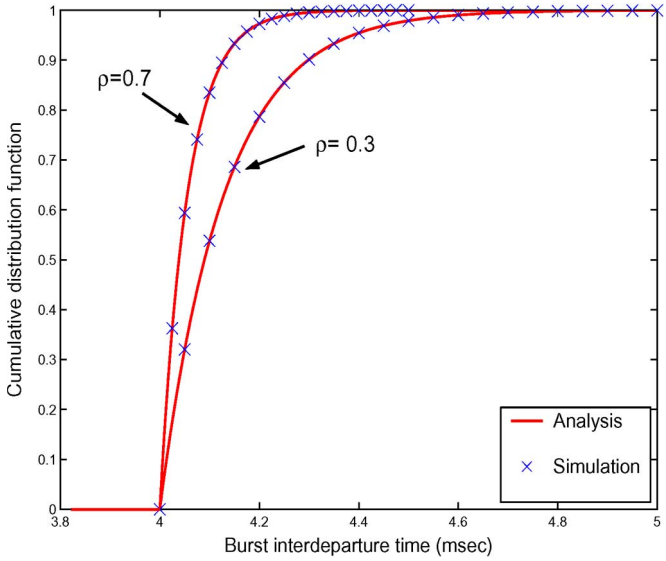

Fig. 5. CDF of the interdeparture time of the bursts generated by the time-based assembler with $T_{\rm Th} = 4$ ms and $L_{\rm min} = 15$ kB.

interdeparture time for the same setting are plotted in Fig. 5. As seen in the figure, the distributions provided by the analytical models exactly match those obtained from simulations. This is not surprising, as from (2), it is clear that the distribution of the burst interdeparture time is independent of the packet length distribution.

Algorithm 3 Burst traffic generation according to the hybrid algorithm with packet arrival rate λ, average packet length μ^{-1}, and assembly parameters $T_{\rm Th}$, $L_{\rm Th}$, and $L_{\rm min}$.

$k \leftarrow$ number of bursts to be generated
λ, μ, $T_{\rm Th}$, $L_{\rm Th}$, $L_{\rm min} \leftarrow$ initialize
$P_{\rm Tout} \leftarrow$ compute $P_{\rm Tout}$ from the model $(0 < P_{\rm Tout} < 1)$
for $i := 1, i \leq k, i++$ **do**
  $t_1 \leftarrow$ exponential(λ^{-1})
  $U \leftarrow$ uniform (0,1)
  if $U \leq P_{\rm T}$ {use time-based procedure} **then**
    $x \leftarrow L_{\rm Th}$
    while $x \geq L_{\rm Th}$ **do**
      $n \leftarrow$ Poisson$(\lambda T_{\rm Th})$
      $x \leftarrow$ Erlang$(n+1, (n+1)\mu^{-1})$
    end while
    if $x < L_{\rm min}$ **then**
      $x \leftarrow L_{\rm min}$
    end if
    $z \leftarrow T_{\rm Th} + t_1$
  else {use volume-based procedure}
    $z \leftarrow T_{\rm Th}$
    while $z \geq T_{\rm Th}$ **do**
      $n \leftarrow$ Poisson$(\mu L_{\rm Th})$
      if $n == 0$ **then**
        $z \leftarrow 0$
      else
        $z \leftarrow$ Erlang$(n, n\lambda^{-1})$
      end if
    end while
    $z \leftarrow z + t_1$
    $x \leftarrow L_{\rm Th} +$ exponential(μ^{-1})
  end if
  wait (z)
  generate and send a burst of length x
end for

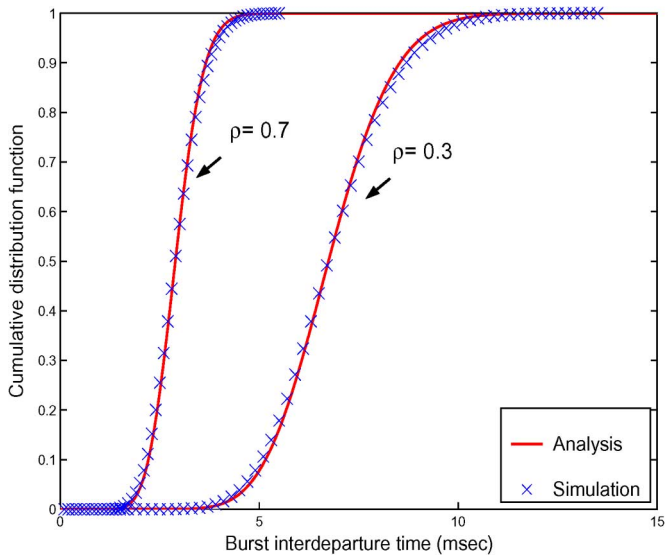

Fig. 6. CDF of the interdeparture time of the bursts generated by the volume-based assembler with $L_{\rm Th} = 25$ kB.

Fig. 6 depicts the CDF of the interdeparture time of the bursts that are generated by the volume-based assembler with the assembly threshold set to 25 kB. The results are shown at both light and heavy loads. We observe that as the load increases, the range of variation of the burst interdeparture times decreases. Also, there is a good match between analysis and simulation, which, again, justifies the exponential assumption for the distribution of packet length in developing the models. The corresponding burst length distributions are shown in Fig. 7. From (6), it is obvious that the burst length is

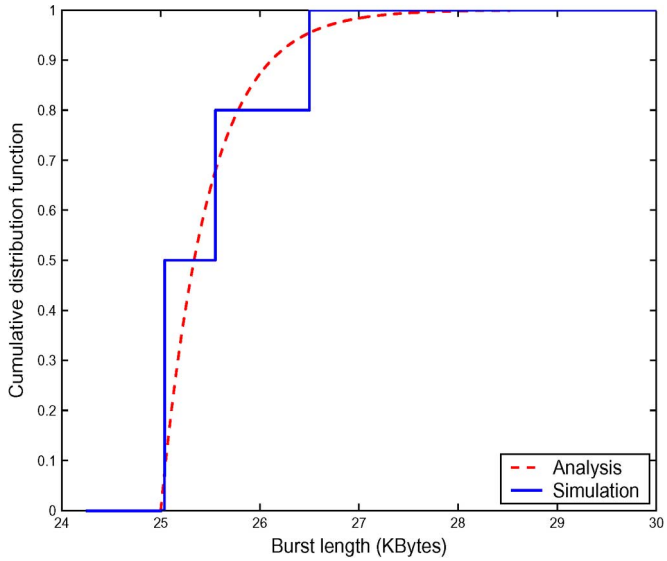

Fig. 7. CDF of the length of the bursts generated by the volume-based assembler with $L_{\mathrm{Th}} = 25$ kB.

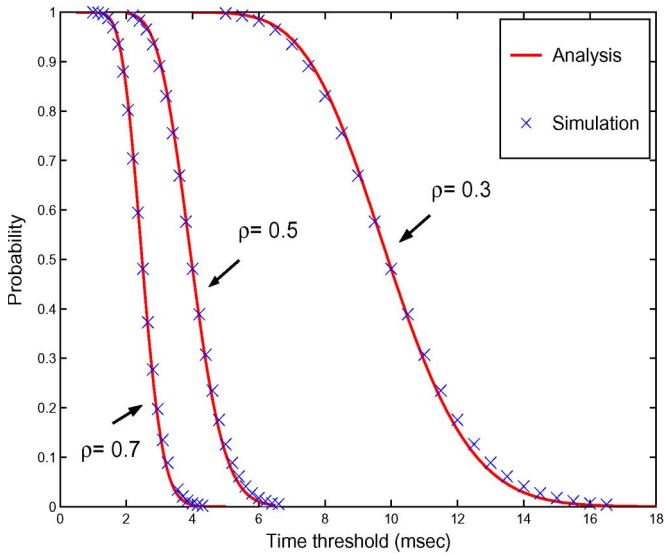

Fig. 8. Fraction of the bursts that are generated due to the timer expiration in a hybrid assembler with $L_{\mathrm{Th}} = 25$ kB.

independent from packet arrival rate. The CDF in the analytical case is a shifted exponential distribution, whereas it is a shifted trimodal distribution in the case of the trimodal packet arrival. The amount of shift in either case is the same and is equal to the assembly threshold. Taking into account that the assembly threshold is always far larger than a single packet length, the difference in the distributions of Fig. 7 can be regarded as negligible.

Let us now consider the hybrid assembly algorithm. The hybrid assembler applies a combination of time and volume criteria to decide for generating bursts; thus, it is important to see the interplay between these two criteria for different settings. In Fig. 8, values of P_{Tout} are plotted as a function of T_{Th} at different loads, with the length threshold L_{Th} set to 25 kB. It is seen that when the assembler is heavily loaded, a small change in T_{Th} can greatly affect the fraction of bursts that are

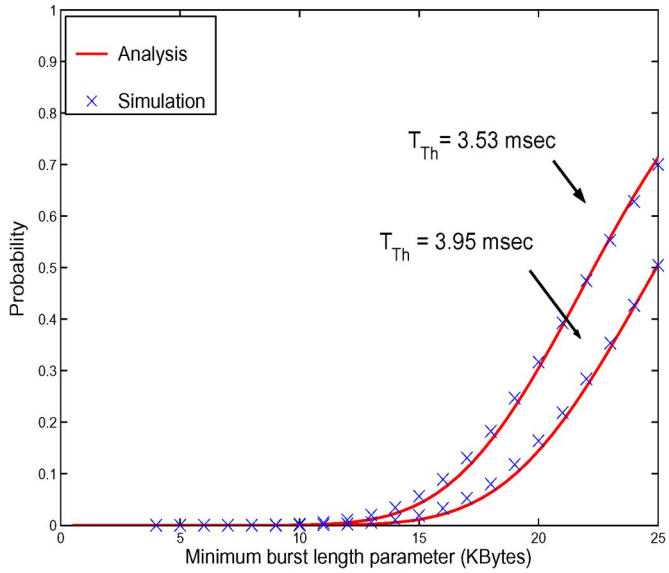

Fig. 9. Probability that the burst length is less than L_{min} in a hybrid assembler with $L_{\mathrm{Th}} = 25$ kB at load 0.5.

TABLE II
PARAMETERS USED TO EVALUATE THE HYBRID ASSEMBLER AND CORRESPONDING VALUES OF T_{out}

	Scenario I	Scenario II
L_{min} (kbytes)	10	18
T_{Th} (msec)	3.95	3.53
P_{Tout}	0.5	0.7

generated due to the timer expiration. This can be explained by recalling that in the pure volume-based burst assembler, the range of variations of the burst interdeparture time decreases with arrival rate, as depicted in Fig. 6.

Another important metric that has to be considered is the fraction of bursts that undergo padding before transmission, i.e., $P_{\mathrm{H,Pad}}$. The value of $P_{\mathrm{H,Pad}}$ is quite important because it is pertinent to the amount of overhead that is introduced by the setting of the assembler's parameters. Referring to (16), for a given length and time threshold, $P_{\mathrm{H,Pad}}$ depends on the value of L_{min}. Fig. 9 depicts the values of $P_{\mathrm{H,Pad}}$ as a function of L_{min} for two different values of T_{Th} at load $= 0.5$ and $L_{\mathrm{Th}} = 25$ kB. Note that at $L_{\mathrm{min}} = 25$ kB, $P_{\mathrm{H,Pad}}$ is equal to the whole fraction of bursts that are generated due to the timer expiration, i.e., P_{Tout}.

To study the impact of P_{Tout} and $P_{\mathrm{H,Pad}}$ on the distributions, two scenarios are considered, as depicted in Table II. In both scenarios, the offered load and L_{Th} are fixed and equal to 0.5 and 25 kB, respectively. Table II also shows the corresponding values of P_{Tout} when each of the scenarios is applied. In fact, we are more interested in the settings that result in a moderate value of P_{Tout} because in the extreme cases, the hybrid assembler collapses either to the time-based or the volume-based assembler. Distributions of the length of the bursts at the output of the hybrid assembler are plotted in Fig. 10. The burst length distributions in Fig. 10 can be divided into two regions, which is in accordance with the expression derived in (17).

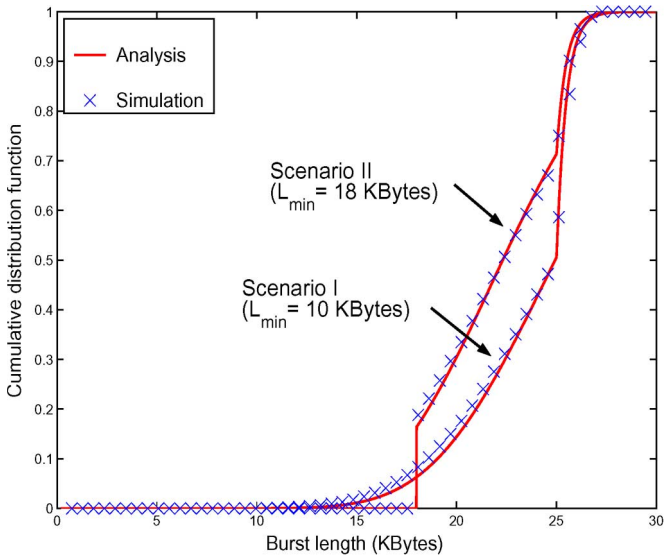

Fig. 10. CDF of the length of the bursts generated by the hybrid assembler with $L_{\mathrm{Th}} = 25$ kB at load 0.5.

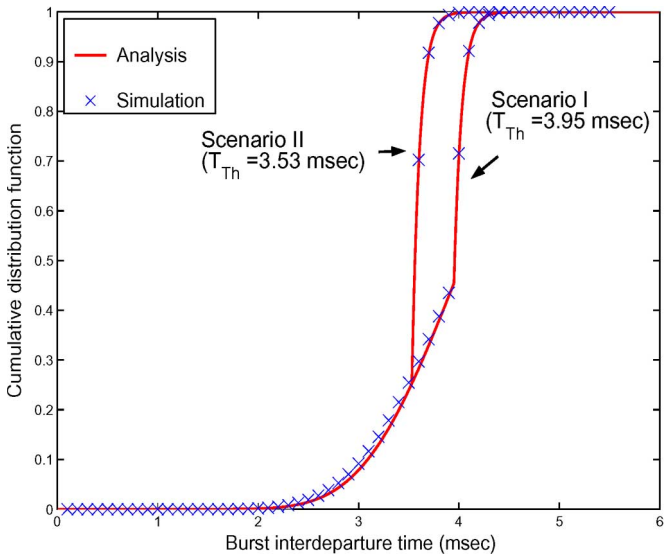

Fig. 11. CDF of the interdeparture time of the bursts generated by the hybrid assembler with $L_{\mathrm{Th}} = 25$ kB at load 0.5.

The first region, i.e., the burst length between $L_{\min}$ and L_{Th}, is associated with those bursts that are generated when the timer goes off. However, the second region, i.e., the burst length greater than L_{Th}, is associated with bursts that are generated because the length threshold is exceeded; thus, their length has a very limited variation. The burst length variation in the first region, however, depends on $L_{\min}$, as seen in Fig. 10.

Distributions of burst interdeparture time corresponding to the scenarios of Table II are shown in Fig. 11. Since the load and L_{Th} are fixed in both scenarios, the corresponding interdeparture times only differ for the times greater than 3.53 ms, which is the smallest time threshold between the two scenarios. Similar to the burst length, the distribution of the burst interdeparture time can also be divided into two regions, with the border of the regions being at T_{Th}. Moreover, it is seen that as the border is shifted to the left, i.e., the time threshold is decreased, the range of variations of the burst interdeparture time reduces.

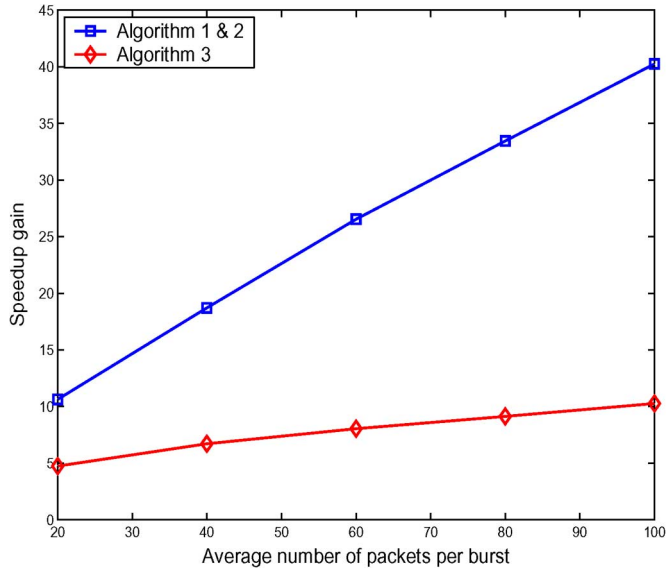

Fig. 12. Speedup gain of using the burst generator algorithms over direct simulating of the assembly algorithms for generating 10^7 bursts as a function of the average number of packets per burst.

Observations of Figs. 8–11 again confirm that the analytical models provide a very accurate estimation for the more realistic case of trimodal packet length distribution. In fact, in all cases, we deal with the random variable resulting from aggregating a large number of independent trimodal random variables. It can be intuitively understood that in that case, the major player is the first moment of the individual packet length, and the role of higher moments decreases with the number of aggregated packets.

After having discussed the characteristics of the burst traffic, we now study the speedup gain that is achieved when using the algorithms of Section IV in a DES model. The proposed algorithms are based on exact analytical models that have been developed in Section III; thus, the traffic that is generated using such algorithms exactly matches that of the analytical models. Therefore, here, we only present and discuss the speedup gain of such algorithms over a straightforward implementation of the assembly algorithms in a DES model.

For this purpose, traces of burst traffic are generated through the direct approach as well as the proposed algorithms, and the corresponding simulation times are compared with each other. Both algorithms are implemented in OMNeT++, in which all basic random number generators required for implementing the algorithms are available as built-in functions. We have measured and compared the time needed for each of the approaches to generate 10^7 bursts on a Pentium IV 3.2-GHz processor with 1024-MB RAM. Fig. 12 presents the speedup gain that has been measured versus the average number of packets per burst. It is seen that the speedup gain for the pure time-based and volume-based algorithms could be well beyond an order of magnitude even for short bursts, and it increases with the average number of packets per burst. The reason is that in the direct implementation, as discussed before, simulation time increases with the number of packets per burst. The speedup

gain for the hybrid algorithm is also quite significant; however, it is smaller than those of the first two algorithms. In fact, the difference between the speedup gains of Algorithm 1(2) and Algorithm 3 is that in the latter case, generating the conditional probabilities requires going through a while loop that, in turn, increases the simulation time.

VI. Conclusion

In this paper, we developed analytical models for the distributions of length and interdeparture time of bursts that leave the assembly buffer in an ingress OBS node. The models are exact for the case that the packet length is exponentially distributed; however, through simulation, it was shown that they also provide a very good approximation for the case of the trimodal packet length distribution. Through numerical examples, it was shown that varying the assembly parameters can significantly influence the burst traffic characteristics. This, together with the fact that burst traffic characteristics have large impacts on the performance of OBS networks, emphasizes the importance of the proper selection of the assembly parameters.

Additionally, as an important application of the models, we presented novel traffic generators to be used in DES models. The main benefit of the proposed generators is the large speedup gain, which, by the way, increases with the average number of packets per burst. In the specific example considered in this paper, a speedup gain as high as 40 has been observed.

Appendix

$$\begin{aligned} g_a(z) &= s(t_1) * h(\tau|\tau < T_{\rm Th}) \\ &= \int_0^\infty s(z-\theta)h(\theta|\theta < T_{\rm Th})d\theta \\ &= \frac{1}{P(\tau < T_{\rm Th})} \int_0^\infty \lambda e^{-\lambda(z-\theta)} \\ &\quad \times \left(e^{-\mu L_{\rm Th}}\delta(\theta) + \sum_{n=1}^{\infty} \frac{\lambda(\lambda\theta)^{n-1}}{(n-1)!} e^{-\lambda\theta} P_{\rm V}[N=n]\right) d\theta. \end{aligned} \tag{29}$$

In the case where $z \le T_{\rm Th}$, we would have

$$\begin{aligned} g_a(z) &= \frac{1}{P(\tau < T_{\rm Th})} \\ &\quad \times \left(\lambda e^{-\mu L_{\rm Th}} e^{-\lambda z} + \sum_{n=1}^{\infty} \frac{\lambda(\lambda z)^n}{n!} e^{-\lambda z} P_{\rm V}[N=n]\right) \\ &\qquad (0 < z \le T_{\rm Th}). \end{aligned} \tag{30}$$

Otherwise

$$\begin{aligned} g_a(z) &= \frac{1}{P(\tau < T_{\rm Th})} \\ &\quad \times \left(\lambda e^{-\mu L_{\rm Th}} e^{-\lambda z} + \sum_{n=1}^{\infty} \frac{\lambda(\lambda T_{\rm Th})^n}{n!} e^{-\lambda z} P_{\rm V}[N=n]\right) \\ &\qquad (z > T_{\rm Th}). \end{aligned} \tag{31}$$

References

[1] C. Qiao and M. Yoo, "Optical burst switching (OBS)—A new paradigm for an optical Internet," *J. High Speed Networks*, vol. 8, no. 1, pp. 69–84, 1999.

[2] J. Turner, "Terabit burst switching," *J. High Speed Networks*, vol. 8, no. 1, pp. 3–16, 1999.

[3] A. Rostami and A. Wolisz, "Impact of edge traffic aggregation on the performance of FDL-assisted optical core switching nodes," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2007.

[4] A. Rostami and A. Wolisz, "Modeling and fast emulation of burst traffic in optical burst-switched networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2006, vol. 6, pp. 2776–2781.

[5] K. Laevens, "Traffic characteristics inside optical burst switched networks," in *Proc. Optical Networking Communications (Opticomm)*, 2002, pp. 137–148.

[6] M. V. Rodrigo and J. Goetz, "An analytical study of optical burst switching aggregation strategies," in *Proc. Broadnets, Workshop OBS*, 2004.

[7] X. Yu, Y. Chen, and C. Qiao, "A study of traffic statistics of assembled burst traffic in optical burst switched networks," in *Proc. Optical Networking Commun. (Opticomm)*, 2002, pp. 149–159.

[8] X. Mountrouidou and H. Perros, "Characterization of the burst aggregation process in optical burst switching," in *Proc. Networking*, 2006, pp. 752–764.

[9] A. Ge, F. Callegati, and L. S. Tamil, "On optical burst switching and self-similar traffic," *IEEE Commun. Lett.*, vol. 4, no. 3, pp. 98–100, Mar. 2000.

[10] X. Yu, J. Li, X. Cao, Y. Chen, and C. Qiao, "Traffic statistics and performance evaluation in optical burst switched networks," *J. Lightw. Technol.*, vol. 22, no. 12, pp. 2722–2738, Dec. 2004.

[11] M. Izal and J. Aracil, "On the influence of self-similarity on optical burst switching traffic," in *Proc. IEEE Globecom*, 2002, vol. 3, pp. 2308–2312.

[12] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido, "A nonstationary Poisson view of Internet traffic," in *Proc. IEEE INFOCOM*, 2004, pp. 1558–1569.

[13] Y. Xiong, M. Vandenhoute, and H. Cankaya, "Control architecture in optical burst-switched WDM networks," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 10, pp. 2062–2071, Oct. 2000.

[14] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, 3rd ed. New York: McGraw-Hill, 2000.

[15] C. Fraleigh *et al.*, "Packet-level traffic measurements from the Sprint IP backbone," *IEEE Network*, vol. 17, no. 6, pp. 6–16, Nov./Dec. 2003.

[16] *Discrete Event Simulation System OMNeT++*. [Online]. Available: http://www.omnetpp.org

[17] H. Akimaru and K. Kawashima, *Teletraffic Theory and Applications*, 2nd ed. New York: Springer, 1999.

Ahmad Rostami received the B.Sc. degree (with honors) in electrical engineering from Shahed University, Tehran, Iran, in 2000 and the M.Sc. degree in electrical engineering from Amir-Kabir University of Technology, Tehran, in 2003. He is currently working toward the Ph.D. degree at the Technical University of Berlin, Berlin, Germany.

Since May 2004, he has been a Research Assistant with the Telecommunication Networks Group (TKN), Technical University of Berlin. From 2001 to 2003, he was an R&D Engineer at the Iran Telecommunication Research Center, Tehran. His research interests are in optical WDM networks, with a focus on traffic modeling and performance-related issues of OPS/OBS networks.

Adam Wolisz (M'93–SM'99) received the B.S., Ph.D., and Habilitation degrees from the Silesian University of Technology, Gliwice, Poland, in 1972, 1976, and 1983, respectively.

After a period with Polish Academy of Sciences (until 1990) and GMD-Fokus, Berlin, Germany (1990–1993), he joined Technische Universität Berlin, in 1993, where he is the Chaired Professor for Telecommunication Networks and the Executive Director of the Institute for Telecommunication Systems. He is also an Adjunct Professor with the Department of Electrical Engineering and Computer Science, University of California, Berkeley. His research interests are in architectures and protocols of communication networks. He is currently focusing mainly on wireless/mobile networking and sensor networks.