Performance Modelling and Traffic Characterisation of Optical Networks

Harry Mouchos, Athanasios Tsokanos, and Demetres D. Kouvatsos

NetPen - Networks and Performance Engineering Research Unit
Informatics Research Institute (IRI),
University of Bradford, Bradford
West Yorkshire, BD7 1DP, UK
harry.mouchos@networks-simulation.com, atsokanos@teilam.gr,
d.kouvatsos@bradford.ac.uk

Abstract. A review is carried out on the traffic characteristics of an optical carrier's OC-192 link, based on the IP packet size distribution, traffic burstiness and self-similarity. The generalised exponential (GE) distribution is employed to model the interarrival times of bursty traffic flows of IP packets whilst self-similar traffic is generated for each wavelength of each source node in the optical network. In the context of networks with optical burst switching (OBS), the dynamic offset control (DOC) allocation protocol is presented, based on the offset values of adapting source-destination pairs, using preferred wavelengths specific to each destination node. Simulation evaluation results are devised and relative comparisons are carried out between the DOC and Just-Enough-Time (JET) protocols. Moreover parallel generators of optical bursts are implemented and simulated using the Graphics Processing Unit (GPU) and the Compute Unified Device Architecture (CUDA) and favourable comparisons are made against simulations run on general-purpose CPUs.

Keywords: Wavelength division multiplexing (WDM), Synchronous Optical Networking (SONET), optical burst switching (OBS) protocol, Just Enough Time (JET) protocol, Generalised Exponential Distribution (GE), bursty traffic, self-similar traffic, Compute Unified Device Architecture (CUDA), Parallel Processing, Wavelength Division Multiplexing (WDM), Dense-wavelength Division Multiplexing (DWDM), Optical Packet Switching (OPS), Optical Burst Switching (OBS), Self-Similarity, Long-Range Dependence (LRD), Generalised Exponential (GE) Distribution, Graphics Processing Unit (GPU).

1 Introduction

Optical networks with wavelength division multiplexing (WDM) have recently received considerable attention by the research community, due to the increasing bandwidth demand, mostly driven by Internet applications such as peer-to-peer networking and voice over IP traffic. In this context, several routing and wavelength reservation schemes, applicable to present and future optical networks, have been proposed [1], [2], [3], [4].

D. Kouvatsos (Ed.): Next Generation Internet, LNCS 5233, pp. 859–890, 2011.

More specifically, optical burst switching (OBS) [1] and [2] was proposed as an alternative to current schemes like SONET. The main feature of OBS is the separation of data burst transmission and the corresponding control information entitled Burst Head Packet (BHP). Each burst is preceded by its own BHP, which travels slightly ahead, configuring the switches and reserving a wavelength path for the upcoming burst. Several OBS protocols define the transmission time delay, called the offset, of a data burst following the BHP. A protocol such as the JET (Just Enough Time) [3] appears to outperform the TAG (Tell and Go)-based and JIT (Just in Time) protocols [3], [4]. However, there is still scope for further exploration in the OBS realm such as burst loss reduction and quality of service (QoS) provisioning. There is a requirement to develop networking protocols to efficiently use the raw bandwidth provided by the WDM optical networks [5].

In this tutorial a novel allocation protocol entitled the dynamic offset control (DOC) protocol is proposed in an attempt to tackle efficiently the aforementioned performance and QoS issues arising in the context of OBS networks. Similar proposals that provide feedback of the blocking probabilities to adapt the offset time have been proposed in the past, like [6]. However, even though [6] mentions Long-Range Dependence, it is never actually taken into consideration during the experiments. The paper assumes, exponentially distributed Optical Bursts arriving into the core optical network, with exponential Optical Burst sizes. The interarrival time of Optical Bursts is assumed to be exponentially distributed, and it depends on the number of available wavelengths (each of which would have OC-192 traffic Load - i.e. an aggregation of OC-192 traffic load per wavelength for each ingress edge node), the aggregation strategy employed at each edge node (based on the running protocol), the IP packet interarrival time distribution as well as Self-Similar Long-Range dependent traffic load, as well as the IP packet size distribution. Having multiple OC-192 traffic streams would definitely cause certain optical bursts to arrive into the core optical network simultaneously. This makes the Poissonian assumption incorrect. Even though it would be more logical to use GE [7] (albeit with different parameters) to model optical burst interarrival in the short time scales than the exponential distribution, this paper makes no assumptions on the bursts' interarrival times and aggregates and assembles optical bursts based on the measured IP packet sizes, the measured IP Packet interarrival times and the aggregation strategy chosen.

What [6] accomplishes (with the assumptions it makes), is to achieve fairness in the blocking probability for all source-destination pairs, regardless of the amount of hops in the light-path. This however, is accomplished at the expense of the lightpaths that don't have many hops (i.e. it increases the blocking probability in the bursts with a small number of hops to achieve fairness). This is accomplished by simply increasing offset times for the bursts that need to travel more hops on average, thus achieving less blocking. This, however, would have significant impact at the edge nodes' buffer length, which is not taken into account into the paper's analysis.

Other authors have considered to proactively drop bursts at the edge node, also based on blocking probability estimates [8]. However, burst arrivals are assumed to follow a Poisson process as well, and the blocking probability estimates are based on the Erlang-B formula. Even though [8] suggests that the blocking probability estimates need to use a different method if the arrival process is not Poisson, the paper does not provide nor does it investigate realistic traffic.

A Poissonian model is frequently considered in the literature. Yu, X. et al in [9] make the first proper attempt to get the characteristics of assembled traffic. The paper investigates several scenarios however the Poissonian assumption is again followed. No IP packet arrives at exactly the same time, IP packet interarrival time distribution and a fixed IP packet size is assumed. Even in the case where a variable IP packet size is considered, it assumes to be exponentially distributed instead of actually using the measured IP packet size distribution. Although [9] actually considers LRD in certain scenarios it is still affected by incorrect IP packet sizes and interarrival times. Realistic IP packet sizes and interarrival times are of vital importance when investigating characteristics of optical assembly.

The proposed algorithm in this paper does not rely solely on modifying offset times, but introduces the concept of having a preferred wavelength per source-destination pair, which adapts according to the network performance. Three different network scenarios were simulated, an arbitrary complex network, UK's educational backbone network JANET and the US NSFNET network with significant results.

The proposed DOC algorithm is extremely hard to model analytically as it makes absolutely no assumptions on IP packet sizes, IP interarrival times (even though GE is used to model time slots of less than 1 second, traffic load is self-similar in the simulation experiment). The nature of self-similarity and the proposed feedback algorithm makes it extremely difficult to track analytically.

The paper is organized as follows: Section 2 carries out traffic characterization of network traces taken by an OC-192 backbone network. Section 3 describes the OBS Network architecture considered in this paper. Section 4 presents the DOC OBS protocol. In section 5 the simulation and associated numerical results are shown. Section 6 describes the advantages of running simulations using a GPU. Conclusions and remarks for further work follow in Section 7.

2 Backbone Core Network Traffic Characterisation

Since 1995 many papers revealed and worked on Internet Traffic's self-similar characteristics. Since then, the research community employing simulations required generators of synthetic traces to match those revealed characteristics. It is proven that the traffic load per unit of time (bin) is self similar [10], [11]. Nevertheless, little research has been made on how to convert this self-similar traffic load into interarrival times for simulation purposes.

A couple of methods on how to convert Self-Similar traffic load into interarrival times are presented in [12]. However these methods resort to exponentially

distributed packet arrivals (albeit with various mean rates), underestimating burstiness in the short timescales.

Some light to this problem was also shed in [14] which introduces the following findings:

Packet arrivals appear Poisson at sub-second time scales: The packet's inter-arrival time follows an exponential distribution. In addition, packet sizes and interarrival times appear uncorrelated.

Internet traffic exhibits long-range dependence (LRD) at large time-scales: In agreement with previous findings, Internet traffic is proven to be LRD at scales of seconds and above.

The measurements of the above-mentioned paper were taken on CAIDA monitor located at a SONET OC-48 (2488.32 Mbps) link that belongs to MFN, a US Tier 1 Internet Service Provider (ISP).

In essence, the general consensus is that although the traffic load is self-similar, traffic at the sub-second level could easily be modelled by Poisson or Exponential (for interarrival times) distribution, but with different parameters per time slot (usually 1-second slots).

Conveniently, many researchers use the exponential or a fixed-rate arrival to model incoming Internet traffic to an optical edge node, taking advantage of the fact that IP packet arrivals could be Poisson distributed at sub-second time scales however some ignore LRD altogether. A good example is [15]. The Poissonian assumption however, remains simplistic, since it doesn't take into account packets that arrive simultaneously to the switch, i.e. burstiness in the small time domain (sub-second). (We consider simultaneous arrivals, packet arrivals that happen within one microsecond of each other).

An investigation into a high-bandwidth backbone core network was required to attempt to analyze the packet size distribution and interarrival times.
Anonymized traces were downloaded from CAIDA of an OC-192 link (9953.28 Mbps) [16].

The equinix-chicago Internet data collection monitor is located at an Equinix datacenter in Chicago, IL, and is connected to an OC192 backbone link (9953.28 Mbps) of a Tier1 ISP between Chicago, IL and Seattle, WA.

2.1 Hardware

The infrastructure consists of 2 physical machines, numbered 1 and 2. Both machines have a single Endace 6.2 DAG network monitoring card. A single DAG card is connected to a single direction of the bi-directional backbone link. The directions have been labeled A (Seattle to Chicago) and B (Chicago to Seattle) [16]. Both machines have 2 Intel Dual-Core Xeon 3.00GHz CPUs, with 8 GB of memory and 1.3 TB of RAID5 data disk, running Linux 2.6.15 and DAG software version dag-2.5.7.1. In a test environment both machines dropped less than 1% of packets with snaplen 48 at 100% OC192 line utilization, using a Spirent X/4000 packet generator sending packets with a quadmodal distribution, with peaks at 40, 576, 1500 and 4283 bytes [16].

2.2 Time Synchronization

Both physical machines are configured to synchronize their hardware clock via NTP. The DAG measurement cards have their own internal high-precision clock, that allows it to timestamp packets with 15 nanosecond precision. Every time a traffic trace is taken, The DAG internal clock gets synchronized to the host hardware clocks right before measurement starts. NTP accuracy is typically in the millisecond range, so at initialization the clocks on the individual DAG measurement cards can be off by a couple of milliseconds relative to each other. The precision (i.e. timing within the packet trace) within a single direction of trace data is 15 nanosecond for the DAG files, and 1 microsecond for PCAP files.

2.3 CAIDA Traffic Analysis

IP Packet Size Distribution. In 2003, official packet size distribution measurements were monitored and recorded over standard IP internet traffic on specific dates at specific times [17]. In particular, three Cisco routers have been used for this purpose and the measurements have been taken at a daily peak on a five minute average on March 2003, whilst another, but more reliable, IP packet length distribution has been captured from 39 trace files between May 13th 1999 at 19:13:46 PDT and May 19th 1999 at 13:02:20 PDT. The latest distribution of IP packet sizes was seen at the NASA Ames Internet Exchange (AIX) by CAIDA and since they contain contributions from the different workloads carried by the network at different times of day, they should represent more of an average picture of the packet size distribution than any other individual trace. However, both of the above sources demonstrate very important aspects of internet traffic. Specifically, the majority of the packets seen are 40, 576 and 1500 bytes and all the internet traffic packets are traced at the range between 23 and 1500 bytes. More than 65% of the packets have been traced at a smaller than 576 bytes size and 50% of the total byte volume belongs to the 1500 bytes packet size.

The analysis, however, of the data collected by the equinix-chicago Internet data collection monitor of an OC192 backbone link (9953.28 Mbps) prove that the IP packet size distribution has changed significantly.

Although the IP Packet size distribution is way too granular to describe in detail, roughly, about 50% of the IP packets appear to have a size of around 40 bytes, about 30% of the IP packets appear to have a size of roughly 1500 bytes, with the remaining ranging from 40 to 1500 bytes. The actual Probability Density and the Cumulative Probability of the Packet Sizes are shown (c.f., Fig. 1 and Fig. 2).

Burstiness. As mentioned before there is a probability that a number of packets will arrive simultaneously (i.e. with a time difference of less than 1 micro second) forming batches of IP packets arriving. Distribution Fitting using MATLAB revealed that the batch sizes throughout the traces' duration can be described by the Geometric Distribution:

$$CDF = 1 - (1 - p)^k \tag{1}$$

and

$$PDF = (1 - p)^{k-1} p, k \in \{1, 2, 3, ...\} \tag{2}$$

Interarrival Times. Fitting of the Exponential Distribution to the interarrival times has been conducted with MATLAB is shown in (c.f., Fig. 4). It was proven that the exponential distribution accurately fits the measurements: The fitting

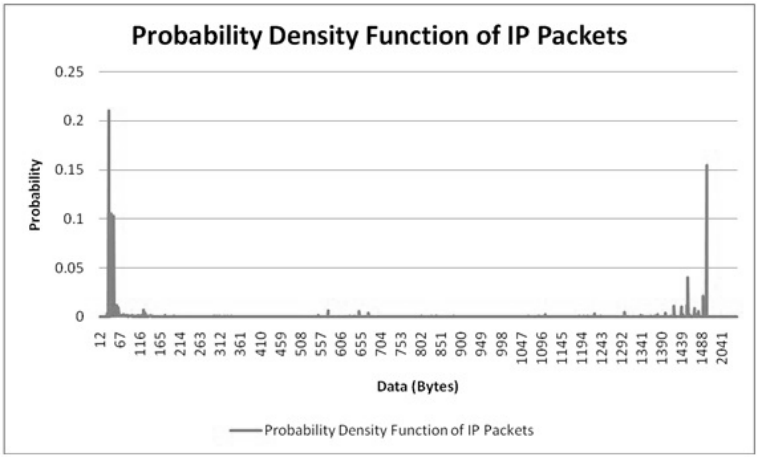

Fig. 1. Probability Density of IP Packet Sizes resulted from the analysis of the OC-192 traffic traces

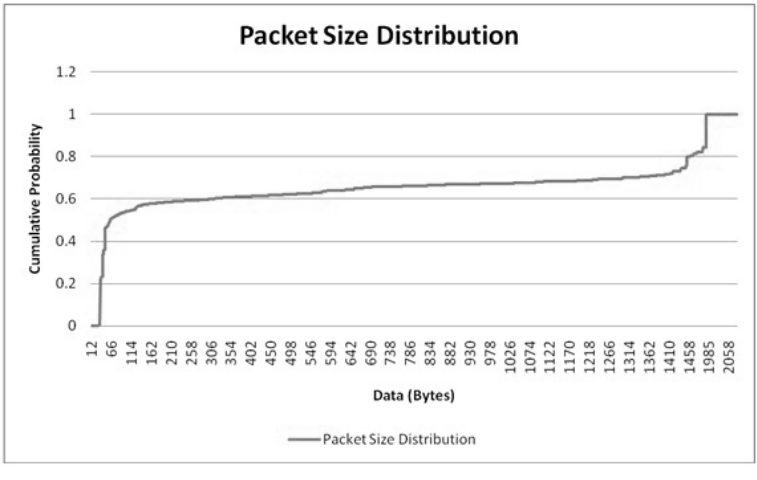

Fig. 2. Cumulative IP Packet Size Distribution resulted from the analysis of the OC-192 traffic traces

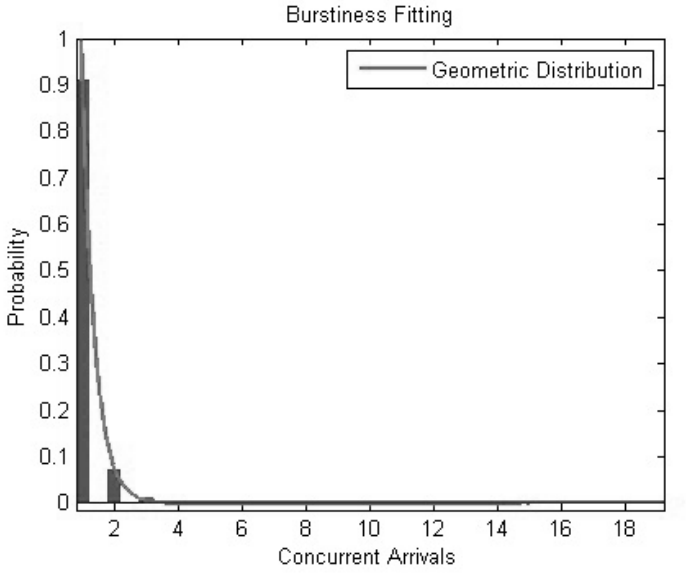

Fig. 3. Burstiness fitting with the Geometric distribution

shown in the image above, was calculated to have 99.98% accuracy (i.e. $R^2 = 0.999878489995854$)[1].

The average batch size estimated over the entire trace duration is 2.235122.

Distribution fitting was conducted for each 1-second duration slot in the trace duration. With an accuracy of above 98% the exponential distribution fits the interarrival time distribution albeit with different mean for each 1-second interval.

Therefore at sub-second scales we have exponentially distributed batch arrivals, with a geometrically distributed batch size. These findings suggest that the Generalized Exponential Distribution [7], [18] is ideal to model arrival of IP Internet Traffic on a high bandwidth backbone link, at sub-second intervals.

Load Traffic Self-Similarity. It has been well established in the literature that network traffic load shows self - similar traffic characteristics [19], [20], [21]. It is, therefore, required to analyse the data provided by CAIDA [16] for Self-Similarity.

The cumulative traffic (in Bytes) per second for over 3700 seconds of cumulative traffic was analysed. Using the wavelet method to estimate the self-similarity degree (Hurst Parameter), resulted in a highly self-similar LRD traffic with Hurst = 0.935 (95% Confidence Interval [0.876, 0.994]) (c.f., Fig. 5).

[1] The coefficient of determination, is a good measure of how well the chosen distribution fits the given data. It must lie between 0 and 1, and the closer it is to 1, the better the fit. It is symbolised as R^2.

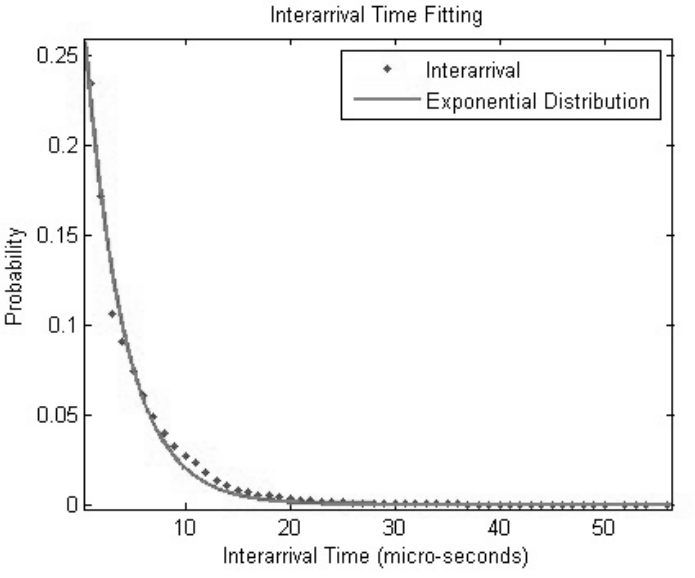

Fig. 4. Interarrival time fitting with the Exponential distribution

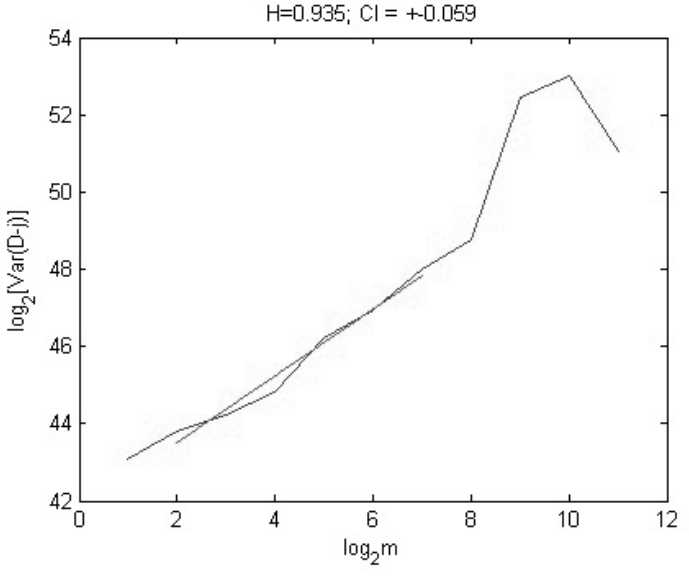

Fig. 5. Estimation of the Hurst parameter using the wavelet method

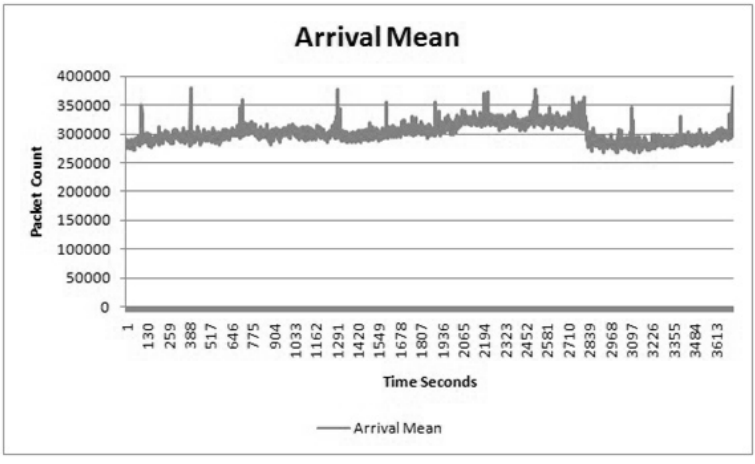

Fig. 6. Estimated MAR per time slot over the entire duration of the trace *packetcount/second*

This confirms previous findings of past research attempts by the community [22], [23], [24], [25].

General Observations. Fitting of the exponential distribution on the interarrival times was conducted for each 1-second duration slot of the trace duration, as well as fitting of the geometric distribution for the batch sizes for the same 1-second duration slots.

The mean arrival rate (MAR) and the Squared Coefficient of Variation (SCV) is calculated for each second of the entire duration of the backbone traces. The estimations are shown below:

Arrival Mean Estimation (per slot). In (c.f., Fig. 6) the estimations of the MAR lambda (measured in Bytes/second) that resulted from the exponential distribution fitting is shown for the entire duration of the trace. The total traffic load self-similar characteristics require a variable mean rate which is proven by the CAIDA traffic trace analysis. Researchers may use this variable rate to accurately model the interarrival times of packets for each time slot. However, the exponential distribution is not sufficient to model batches of packets arriving at the same time.

Each estimation of the SCV was calculated with accuracy of above 98%[2]. In (c.f., Fig. 6) the estimations are shown for the SCV C_{α}^{2}, that resulted from the Geometric Distribution fitting during the entire duration of the trace. The fitting was performed on the average batch size (i.e., simultaneous arrivals)[3] for each time slot.

[2] With $R^2 > 0.98$.

[3] Arrivals are considered simultaneous if the occur within 1 μsec.

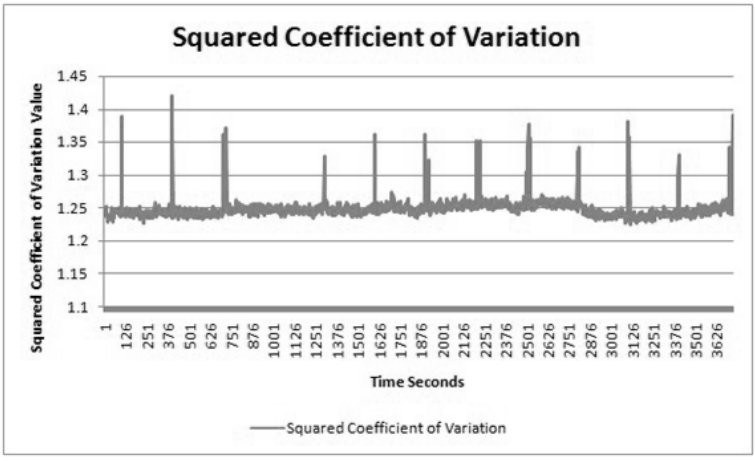

Fig. 7. Estimated SCV per time slot over the entire duration of the trace (value/second). Values of greater than 1, indicate burstiness in the short-time scales.

Summary. Traffic Characterisation was conducted on traces taken by CAIDA of an OC-192 Backbone Core Network. Results have shown that even in the short time scales (sub-second duration) some burstiness persists. This could be better modelled by the Generalised Exponential Distribution as it maintains the analytical tractability of the Exponential Distribution but does not underestimate burstiness. For each time slot, the estimated SCV is greater than 1, indicating batches of simultaneous arrivals in the node. On average over the entire period the value of the SCV was estimated at 1.534340.

To validate the estimation of λ and C_α^2, (c.f., Fig. 7) shows a comparison between the measured traffic from the CAIDA traces and the traffic resulted by the estimation. It is obvious that the traffic loads match accurately. Multiple aggregated streams arrive to an optical backbone node, converted, demultiplexed and transmitted through several wavelengths over the same fiber. Potentially this could increase the average batch size of simultaneous arrivals significantly. The GE distribution is proven to be ideal for modelling heavier traffic loads by simply using higher values for the C_α^2.

3 OBS Network Architecture

An OBS network uses one-way reservation protocols, sending a control packet to configure the switches along a path, followed by a data burst without waiting for an acknowledgement for a successful connection establishment. Source nodes have an offset time depending on the destination node. OBS networks employ several different protocols for bandwidth reservation and vary in architectures.

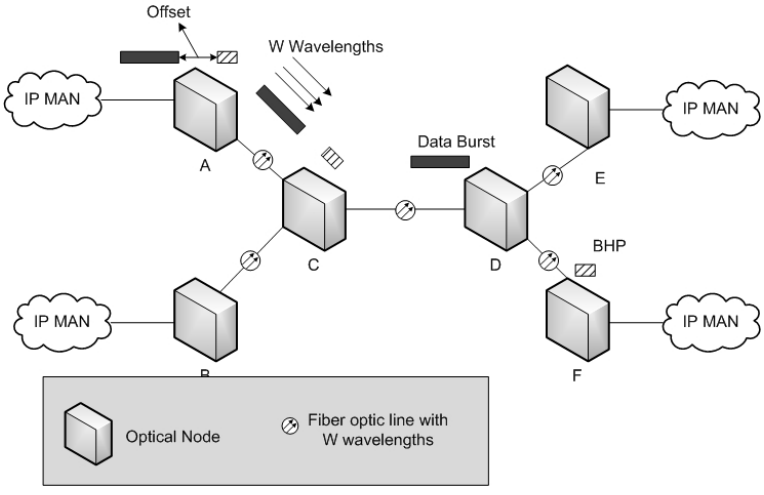

Fig. 8. A simple OBS network architecture

3.1 Just-Enough-Time (JET)

For illustration purposes, a simple OBS network model with two source and two destination nodes is displayed in (c.f., Fig. 8). Source nodes A and B transmit data bursts into the network using the OBS JET protocol with random destinations (E and F). Every source node transmits data bursts after an offset time $Offset_{JET}$ since the transmission of the control packet (also called Burst Header Packet - BHP).

In JET no buffering is required at the intermediate nodes, due to the fact that the bursts are stored in buffers at the source node, and they are transmitted after an offset time large enough to allow the control packet to be processed.

Another attractive feature of JET is the delayed reservation. Bandwidth is reserved only from the moment the burst arrives till the moment it departs the intermediate node. These times are specified in the control packet. In JET the control packet needs to include the burst length and the remaining value of the offset time $Offset_{JET}$. The control packet is timestamped with its arrival time at each intermediate node along with its expected transmission. In addition the control packet will carry an up-to-date value of $Offset_{JET}$ to the next node.

In case, Fiber Delay Lines exist in the intermediate nodes, they can be entirely utilized for the purpose of resolving conflicts instead of waiting for the control packet to be processed [26].

However, due to the high costs, it is more likely that no buffering will be available in the core optical network. Because of the lack of buffering burst dropping is quite probable. To tackle this, classifying bursts according to a priority system is suggested in [26]. This suggestion involves adding an additional offset time T' to the base $Offset_{JET}$ of the burst of a high-priority class. To completely isolate one priority class from another, the authors suggest that the high priority class'

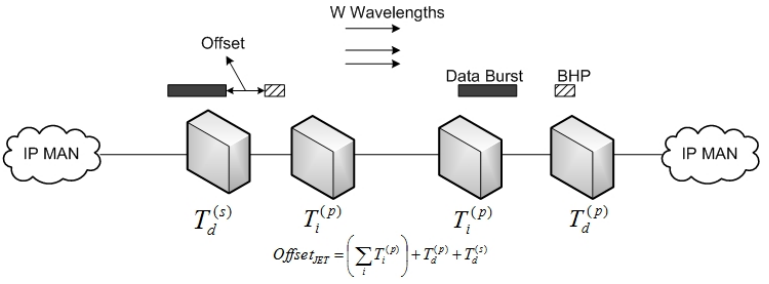

Fig. 9. Offset Time Calculation

offset time to have a value of a multiple of the lower priority class' average burst size. Blocking (or loss) of data bursts occurs in node C.

Under the JET protocol, a fixed offset time is used which is calculated in a manner that takes into account the processing delays of the BHP at the intermediate switches. Let T_i be the processing delay of the BHP at an intermediate switch i, $T_d^{(p)}$ be the processing delay of the BHP at the destination switch and $T_d^{(s)}$ denote the time to setup and configure the destination switch. Clearly, the $Offset_{JET}$ for JET is determined by

$$Offset_{JET} = \left(\sum T_i^{(p)}\right) + T_d^{(p)} + T_d^{(s)} \tag{3}$$

Let $\{i\}$ be the set of intermediate switching nodes belonging to a path from a source node to a destination node. The offset time calculation for the JET protocol is illustrated in Fig. 9 for a path that includes two intermediate switching nodes between the source and the destination of the data burst. The offset time needs to be long enough to account for the processing time of the BHP at the two intermediate nodes and the destination, plus the setup time at the destination [27].

3.2 Routing and the Control Wavelength

Let's assume that every optical fiber link of the network has W wavelengths. One wavelength is used for the transmission of BHP and control packets while the (W-1) remaining wavelengths are used to transmit bursts.

The control wavelength uses Time Division Multiplexing (TDM) for every node in the network, including the intermediate and destination nodes, in order to avoid collisions by assigning a time slot to every node. A source node transmits a BHP containing information about the destination address, burst size and offset of the upcoming burst. To achieve the wavelength reservation for the upcoming burst, the BHP travels through the control wavelength to every intermediate node on its path until it reaches its destination.

3.3 Incoming Traffic Characteristics

For each Optical Network Topology considered, all edge nodes act as both ingress and egress nodes. This means that all nodes situated at the edge of the optical

network, both transmit and receive optical bursts. Each node is considered to have 4 incoming and 4 output ports. Each port-pair is transmitting and receiving on a separate wavelength. Each fiber in the optical network is considered to have 4 wavelengths.

After the analysis of the CAIDA traffic, no assumptions are made during the simulation experiments regarding optical burst interarrival times. For each ingress edge node, OC-192 traffic is assumed for each wavelength. Self-similar traffic load for each wavelength is generated based on the traffic measurements, and converted into interarrival times. The IP packets arriving, are assembled into optical bursts according to destination, and are stored in electronic buffers (each buffer associated with a specific destination node) prior to their conversion into the optical domain. The optical burst assembly strategy chosen is using both a time limit as well as a burst length limit. After the optical bursts are assembled they are transmitted into the core optical network according to the specific OBS protocol.

4 The Dynamic Offset Control (DOC) Allocation Protocol

Like JET, the Dynamic Offset Control (DOC) Allocation Protocol, possesses the qualities that made JET so attractive. DOC requires no buffering at the intermediate nodes and it has the delayed reservation feature [38]. However, due to its control packet mechanism, no additional information needs to be embedded in the control packet at each node on the path.

Each source node transmits a BHP containing information about the destination address, burst size and offset of the upcoming burst. The initial offset at every source node is calculated in the same manner as in the JET protocol [26], [30]. When a burst arrives at its destination or when a collision is detected at some intermediate node, the destination or intermediate node will, respectively, transmit a control packet in their own time slot to the source node with information regarding the arriving or blocked burst. This information includes the burst's destination, offset time and size. In this fashion, the source node calculates progressively a blocking percentage for every source - destination pair.

4.1 Problem Statement

Protocol Formulation
Notation
Given:

- $LimPer_{sd}$: The Blocking Tolerance for the source - destination pair (s, d).
- $OffsetStep$: This value is used to increment or decrement the offset value for each source - destination Pair $Offset_{sd}$ up to its $MAX_{Offset_{sd}}$ value or down to $MIN_{Offset_{sd}}$ value.
- $Hurst$: Self-Similarity Degree.

– $TopologyInfo$: Topology information which includes: actual topology, number of wavelengths and routing.

Variables:

– $BPer_{sd}$: Blocking percentage for the source - destination pair (s, d).
– $Offset_{sd}$: The current offset value for the source - destination pair (s, d).
– $INIT_{Offset_{sd}}$: The default/initial offset value for the routing path of the source - destination pair (s, d). The initial offset value is a value between the minimum offset value and the maximum offset value for DOC. In the JET protocol, the initial offset value is also the minimum.
– $MIN_{Offset_{sd}}$: The minimum offset value for the routing path of the source - destination pair (s, d). The minimum offset value is the minimum time required for all intermediate nodes to process the BHP packets.
– $MAX_{Offset_{sd}}$: The maximum offset value for the routing path of the source - destination pair (s, d). The maximum is calculated based on the QoS requirements of the applications running on the network.
– $PWave_{sd}$: Describes the preferred wavelength to use to transmit bursts from the source node s to the destination Node d. This value is adapted according to each source - destination pair's traffic parameters by the DOC protocol.

Objectives:
Achieve Higher Throughput
Maintain Mean Queue Lengths
Higher Utilisation

4.2 Outline of the Algorithm

A summary relating to the operational aspects of the DOC protocol is presented in a stepwise fashion below.
Begin
Input Data: *TopologyInfo,* $LimPer_{sd}$, $OffsetStep$, $Hurst$

Step 1: Calculate initial offset $INIT_{Offset_{sd}}$, minimum offset $MIN_{Offset_{sd}}$ and maximum offset $MAX_{Offset_{sd}}$ at each source node for each destination node as in JET protocol; for each node i, i = 1, 2, ... , N. Use the first available wavelength to transmit bursts through.

Step 1.1: Generate Self-Similar Traffic Load and employ the GE distribution to model the interarrival times of data bursts per time slot;

Step 1.2: Define at each source node initial offset values for each destination node;

Step 1.3: Bursts are transmitted through the preferred wavelengths $PWave_{sd}$ unless they are busy transmitting, in which case the next available one is chosen.

Step 2: If a data burst arrives successfully at a destination node, then a control packet is sent to the source node to update $BPer_{sd}$;

Step 2.1: If $BPer_{sd} < LimPer_{sd}$

Step 2.1.1: If $Offset_{sd} > MIN_{Offset_{sd}}$ then $Offset_{sd} = Offset_{sd} - OffsetStep$ otherwise $Offset_{sd} = MIN_{Offset_{sd}}$

Step 2.2: If $BPer_{sd} = LimPer_{sd}$

Step 2.2.1: $Offset_{sd} = INIT_{Offset_{sd}}$

Step 2.3: If $BPer_{sd} > LimPer_{sd}$

Step 2.3.1: If $Offset_{sd} < MAX_{Offset_{sd}}$ then $Offset_{sd} = Offset_{sd} + OffsetStep$ otherwise

Step 2.3.2: $Offset_{sd} = INIT_{Offset_{sd}}$ and set a new preferred wavelength $PWave_{sd}$

Step 3: If a data burst is blocked at an intermediate node, then a control packet is sent to the source node to update $BPer_{sd}$;

Step 3.1: If $BPer_{sd} < LimPer_{sd}$

Step 3.1.1: If $Offset_{sd} > MIN_{Offset_{sd}}$ then $Offset_{sd} = Offset_{sd} - OffsetStep$ otherwise $Offset_{sd} = MIN_{Offset_{sd}}$

Step 3.2: If $BPer_{sd} = LimPer_{sd}$

Step 3.2.1: $Offset_{sd} = INIT_{Offset_{sd}}$

Step 3.3: If $BPer_{sd} > LimPer_{sd}$

Step 3.3.1: If $Offset_{sd} < MAX_{Offset_{sd}}$ then $Offset_{sd} = Offset_{sd} + OffsetStep$ otherwise

Step 3.3.2: $Offset_{sd} = INIT_{Offset_{sd}}$ and set a new preferred wavelength $PWave_{sd}$

End.

4.3 DOC Algorithm Description

The key idea of the proposed DOC protocol is to keep the loss rate between the source-destination pair to the minimum [38]. Optical bursts are assembled and transmitted into the core optical network. Should a burst arrive at the destination node, or if a blocked data burst at an intermediate node occurs, a control packet will provide new data for the source node to calculate the new $BPer_{sd}$. If the blocking percentage drops under $LimPer_{sd}$ in case of a successfully transmitted data burst, it will start decreasing $Offset_{sd}$ by $OffsetStep$ unless the offset is equal to the $MIN_{Offset_{sd}}$ value for the source - destination pair. However, if the blocking percentage exceeds $LimPer_{sd}$ in case of a blocked data burst, then the node sets the pair's offset value to the initial offset value and increases it progressively by $OffsetStep$ until $BPer_{sd}$ drops again under $LimPer_{sd}$ or the offset reaches $MAX_{Offset_{sd}}$.

The "delay time to reaction" is a well-known problem which accompanies any feedback-based algorithm. To tackle this, source nodes in DOC adjust their offset values and preferred wavelengths based on percentage values of the performance metrics in question and thresholds that are specified as input to the algorithm. These percentage values, change according to feedback received and when these values cross certain thresholds the algorithm adjusts the offset values and preferred wavelengths. This "slow" adaptation prevents sudden reactions to delayed feedback where the situation has changed by the time the feedback is received.

Thus, $Offset_{sd}$ may be defined by

$$Offset_{sd} = \begin{cases} INIT_{Offset_{sd}} \leq Offset_{sd} \leq MAX_{Offset_{sd}}, \\ if BPer_{sd} > LimPer_{sd} \\ MIN_{Offset_{sd}} \leq Offset_{sd} < INIT_{Offset_{sd}}, \\ if BPer_{sd} \leq LimPer_{sd} \end{cases} \quad (4)$$

Although increasing the offset between the BHP packet and the burst helps towards lowering blocking, it may prove that it is not sufficient to decrease blocking between the two nodes. To this end, the source is required to choose an alternate route, which is implemented by introducing the concept of *preferred wavelength*.

4.4 Preferred Wavelength

The network transmits each burst on the first available wavelength through its corresponding output port, however, in DOC the concept of *preferred wavelength* is introduced. Each source node maintains a preferred wavelength for each destination node. The source node will first attempt to transmit the burst through the specified preferred wavelength. If the preferred wavelength is occupied (because of a transmission of another burst), the first available is chosen instead in a cyclic round-robin fashion.

Initially, the source nodes choose the first wavelength as the preferred for each destination node. If changing the offset is not enough to keep the Burst Loss percentage $LimPer_{sd}$ to a low value, then a new preferred wavelength $PWave_{sd}$ is chosen for this destination node. The whole purpose of choosing a preferred wavelength is for the various source-destination pairs to reach a state of balance that minimizes collisions.

5 The Simulation and Experimental Results

An event-driven simulator in CUDA (using an NVIDIA 8800 GTX), MATLAB and C# .NET was developed for the quantitative analysis of three types of optical network topologies, an arbitrary topology, JANET and NSFNET with source nodes transmitting towards all others having four wavelengths each and no wavelength conversion capabilities. The simulator uses a non approximation methodology which takes into account the actual self-similar traffic characteristics of an optical network analyzed in the beginning of this paper to demonstrate the effectiveness of the newly proposed DOC protocol and get unbiased results. The link lengths vary and are specified in each topology description. The IP packet traffic load is self-similar and generated for each wavelength of each source node. The IP packet interarrival times are distributed according to the GE distribution, with mean and SCV calculated on a time-slot basis. The Optical Burst Assembly Strategy employed is the Time - Length Constraint strategy with the size of the SONET OC-192 frame. Taking advantage of the GPU's processing power [29] facilitated simulation of the scenarios in high detail and speed.

The simulator generates bursts randomly for the entire duration of the simulation run, after aggregating incoming IP packets, with random destinations. During the simulation, the blocking percentage of every pair, the throughput and the buffer length at the source nodes are calculated.

5.1 Experimental Results

Network Topology Scenarios. Various topologies were used to perform a comparison between the JET and the DOC protocol. An arbitrary topology (tested in two modes, a simple mode with three source nodes and three destination nodes and a bidirectional mode with all edge nodes being source and destination nodes), JANET the UK Education Backbone network and the US' NSFNET network.

Arbitrary Topology. In the arbitrary topology three source nodes were initially considered (shown in c.f., Fig. 10 as A, B and C). Each source node had a choice of three destination nodes (shown in c.f., Fig. 10 as K, L and M).

Bidirectional Topology. The topology shown in c.f., Fig. 10 was used as the bidirectional topology. In this scenario, all edge nodes (A, B, C, K, L and M) were considered as both ingress and egress nodes. Each node could transmit bursts towards all other edge nodes.

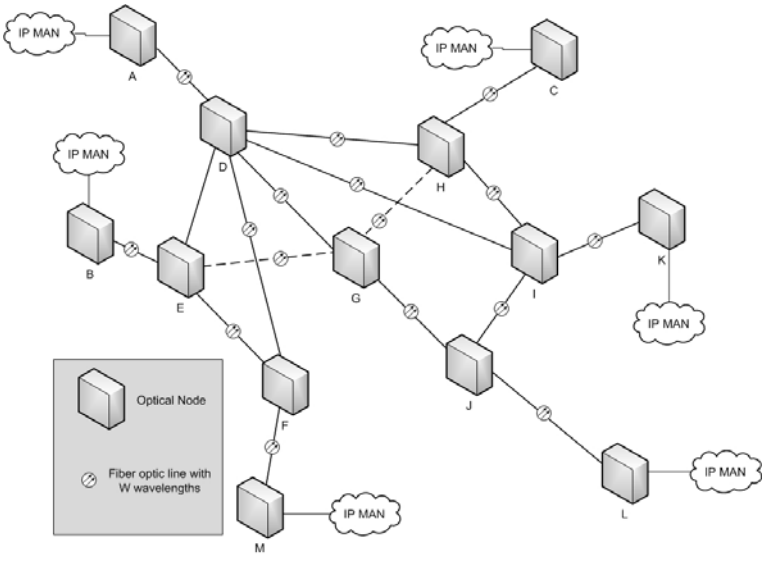

Fig. 10. Arbitrary Topology used for simulation purposes. This topology was also used in a bidirectional mode.

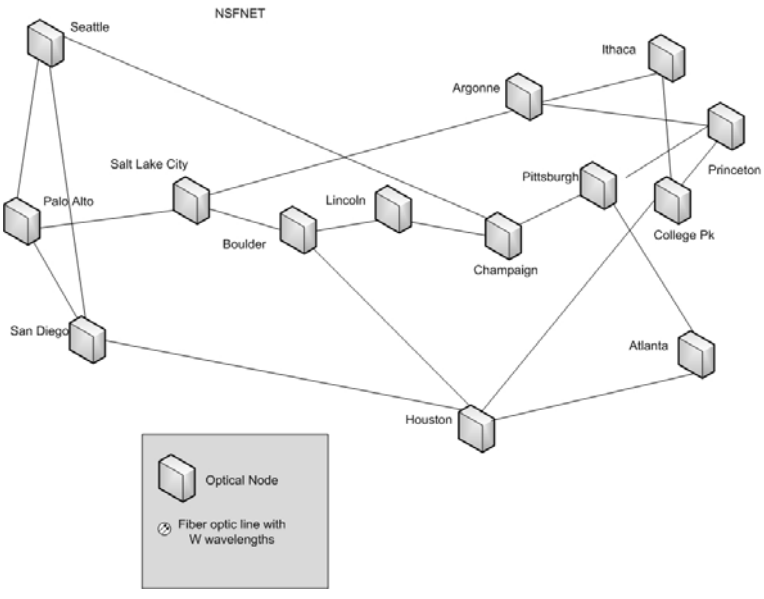

Fig. 11. The NSFNET T1 Network Topology

NSFNET Network Topology. The NSFNET Topology was also used as a network scenario (c.f. Fig. 11). All nodes in the network were considered as both ingress and egress nodes.

JANET Topology. Similarly to NSFNET, the JANET scenario (c.f., Fig. 12) used all nodes as both ingress and egress nodes.

Overall Blocking and Throughput. The mean Overall Blocking and Overall Throughput for all Topologies are shown in Fig. 13 and Fig. 14. It is obvious that for every topology scenario considered, the DOC protocol performed considerably better than the JET protocol. In Fig. 13 and Fig. 14 the average blocking and average throughput over the entire network are shown.

Port (Wavelength) Utilisation. Each source node transmits a burst towards its destination over a specific wavelength through an output port. The source ingress nodes each have 4 ports that transmit the bursts. Under the JET protocol, they transmit simply through the first available port. Certain wavelengths are over-utilised in comparison to the remaining ones and as a consequence there is a higher chance of collision, since a great number of bursts are traversing the same wavelengths.

The newly proposed DOC protocol, achieves fairness in terms of port utilisation by using various preferred wavelengths for each destination node. It is

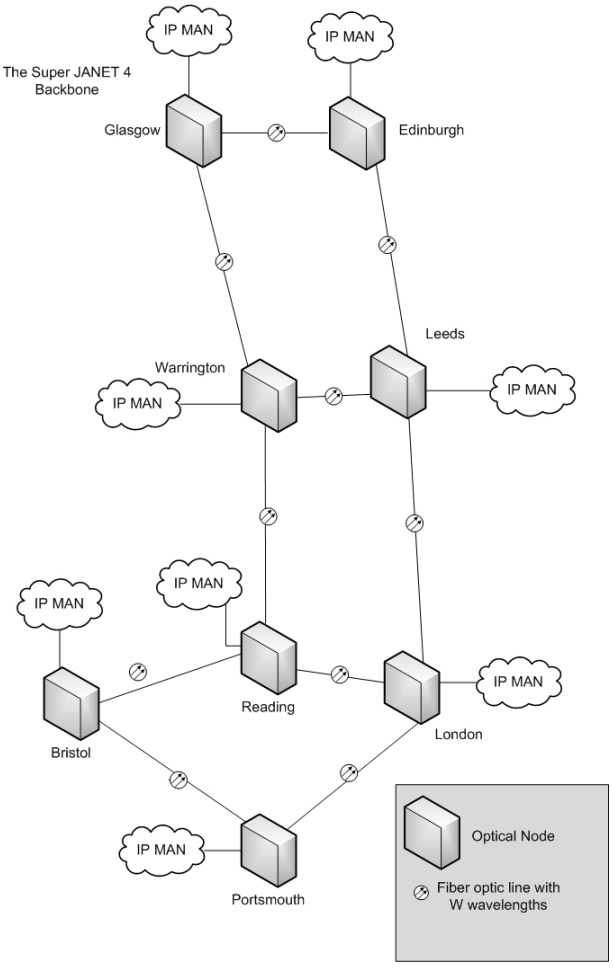

Fig. 12. The UKs Educational Network Super JANET 4

shown in Fig. 15 that on average most wavelengths will be better utilised under the DOC protocol in all the topologies investigated.

Mean Queue Length. Measurements were taken for each network topology scenario on the mean buffer length of each source node (measured in IP packet counts) under both protocols investigated, in order to investigate the effect of the variable offset values to the edge node buffers. It is obvious by the results (c.f., 16), that under DOC, the mean buffer lengths were not significantly affected by the offset value variability in comparison to JET. The blocking probability decrease achieved by DOC and the option of using preferred wavelengths for transmission, allows the offset values to remain at their minimum for the greatest part of the simulation duration.

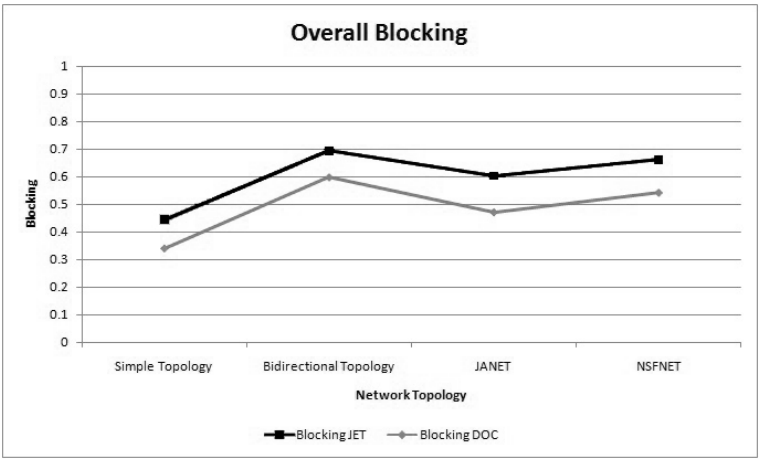

Fig. 13. Overall blocking probability comparison between JET and DOC for various network topologies

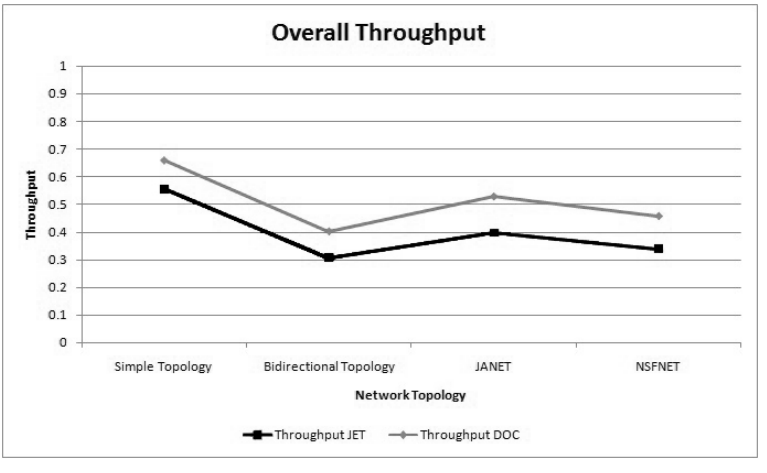

Fig. 14. Overall throughput probability comparison between JET and DOC for various network topologies

Blocking and Throughput Probability. Apart from the overall blocking, blocking measurements were taken on a source - destination pair basis. It is obvious (c.f., 17) that in most pairs, DOC decreased the blocking probability significantly since under DOC, source destination pairs adjust their offset values and preferred wavelengths based on performance metrics and feedback from the network.

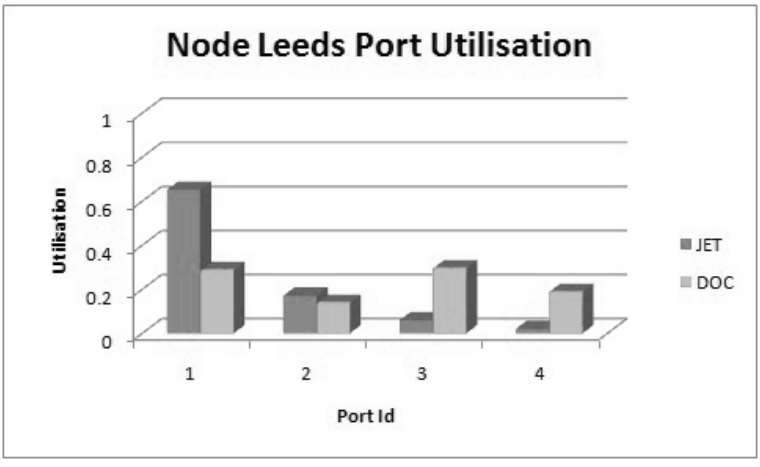

Fig. 15. JANET topology wavelength utilisation for the 'Leeds' source node

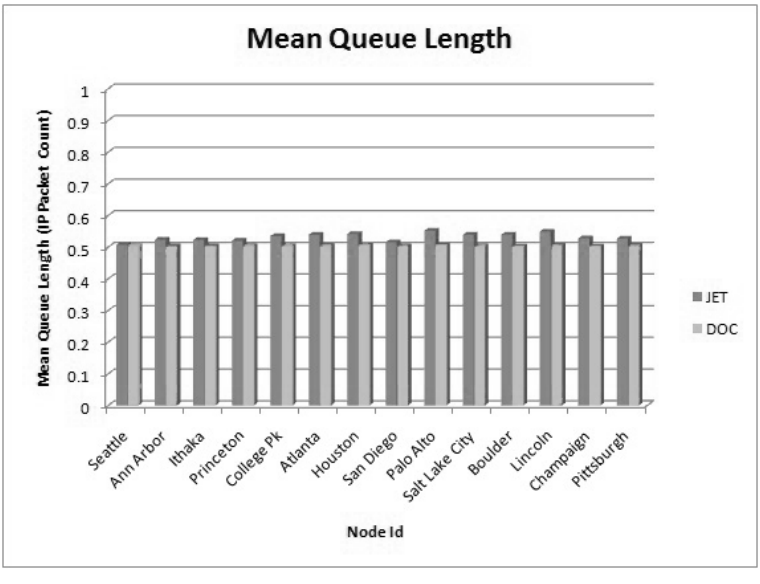

Fig. 16. Mean queue length for source nodes in NSFNET Topology network

Similarly, throughput measurements on a source-destination pair basis were calculated. All the network topologies achieved much higher throughput when running under the proposed DOC protocol (c.f., 18).

DOC vs. Multiclass OBS JET networks. Note that depending on QoS policies for the various networks, it might be justifiable for JET to incorporate

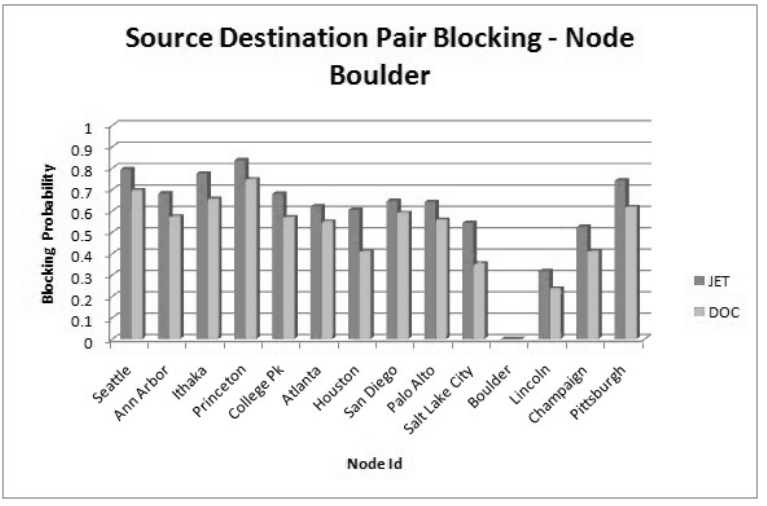

Fig. 17. Source - destination Pair Blocking Probability Comparison in NSFNET topology scenario for the source node at 'Boulder' between the JET and DOC protocol

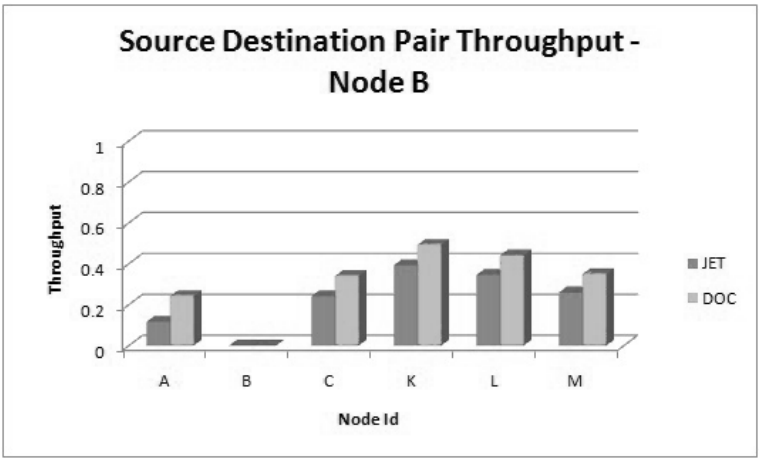

Fig. 18. Source - destination Pair Throughput Probability Comparison in NSFNET topology scenario for the source node at 'Boulder' between the JET and DOC protocol

higher offset values in order to allow control packets added time to be more effectively processed so that network resources are properly allocated, as for example in the case of high-priority bursts [31].

Below there are sample comparisons between JET and DOC, where JET's offset value is higher than the minimum, for demonstrative purposes. It is demonstrated that DOC outperforms JET under such a scenario as well. It's worth

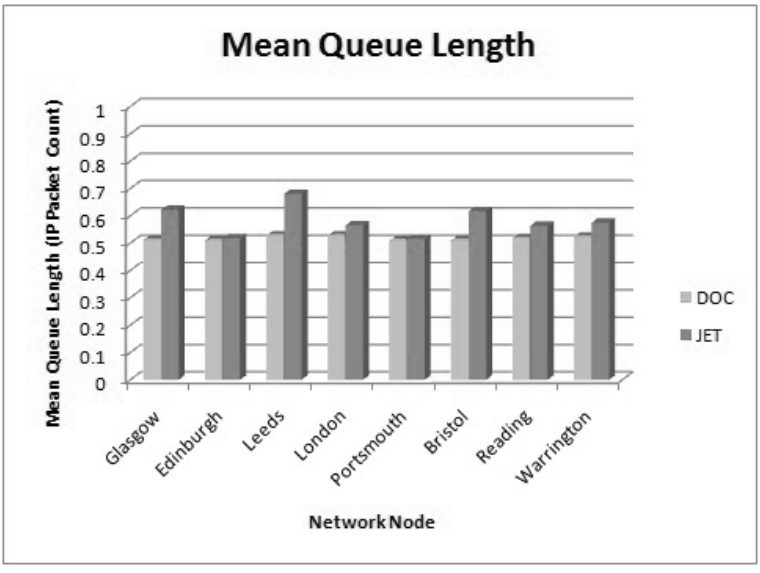

Fig. 19. Mean queue length for source nodes in JANET Topology network

noting that although the Mean Queue Length rises when JET uses a higher offset value, DOC manages to keep the Mean Queue Length lower while also achieving a decrease in blocking.

6 Simulation with the GPU

Due to self-similarity's high complexity and the associated analytical difficulty, simulation became the most favourable tool to evaluate the performance of networks [39]. Consequently, many self-similar trace generators were invented towards the creation of synthetic traces of network traffic such as On/Off Sources and M/G/∞ [37].

The inherent complexity of optical networks is an additional obstacle to the speed and duration of the simulations. Due to WDM / DWDM, current optical fiber speeds in WANs exceed several Tbps, when each wavelength's speed is 10 - 40 Gbps. A simulation would practically need millions of simulated optical bursts and for each optical burst a large number of IP Packets. Therefore, a Uniform Random Number Generator with an extended period is necessary. One such huge-period generator was proposed in [32], called Mersenne Twister (MT), which has a massive prime period of $2^{19937} - 1$.

However, the computational requirements remain extremely taxing when running an optical network simulation that is fed by self-similar LRD traffic flows on an average home computer. Thus, supercomputers and other parallel systems are usually chosen for these types of simulations that potentially have very high costs and are not always readily available.

6.1 Simulating on a GPU

Modern CPUs have multiple cores, which means in essence that more than one processor exists within the same chipset. Today, CPUs with four cores have increased performance by allowing more processes to be handled simultaneously. However, even if the first steps towards multiple processors in one chipset have been made, it is hardly enough.

Alternatively, graphics cards created by NVidia [33] may be employed for the simulations to run on, instead of the actual CPU of the system. These Graphics Processing Units (GPUs) have proven to be a good cost-effective solution to the lack of processing power problem.

Differences between CPU and GPU. In the past five years, a lot of progress has been made in the field of graphics processing, giving birth to graphics cards that have a large amount of processors as well as memory sufficient to allow their use to other fields than simply processing and depicting graphics. The processing power (floating-point operations per second) superiority of modern graphics cards (GPUs) to that of the CPU with NVidia's GeForce 8800 GTX reaches almost 340 GFlops whilst newer models like the GeForce GTX 280, can reach almost 900 GFlops(c.f., [29]). Nevertheless, even if the GPU appears to have significantly more power than the conventional CPU, it is worth noting that the CPU is capable of handling different kinds of processes quickly, while the GPU is only capable of processing a specific task very fast. This latter task needs to be in the form of a problem composed of independent elements, due to the large parallelization of GPUs (c.f., [29]).

Note that the GeForce 8800 GTX is equipped with 128 scalar processors divided into 16 groups - called multiprocessors - of 8. The calculation power of such GPU was shown in [34] to be clearly superior to the CPU. However, the difference is not that large unless this power is efficiently utilized [39].

GPU Architecture. Via the CUDA framework, the GPU is exposed as a parallel data streaming processor, which consists of several processing units. CUDA applications have two segments. One segment is called "kernel" and is executed on the GPU. The other segment is executed on the host CPU and controls the execution of kernels and transfer of data between the CPU and GPU [29].

Several threads that run on the GPU run a kernel. These threads belong to a group called block. Threads within the same block may communicate with each other using shared memory and may not communicate with threads of another block. There is a hierarchical memory structure, where each memory level has different size, restrictions and speed [29].

A disadvantage of the GPU was that it adopted a 32-bit IEEE floating-point numbers, which is well lower than the one supported by a general-purpose CPU. However, the recently released NVIDIA 280 series supports double precision floating point numbers [39].

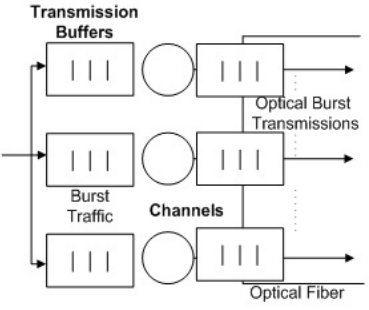

Fig. 20. Ingress node that transmits optical bursts into the core optical network, multiplexing several wavelengths (decomposed)

6.2 Numerical Results

Simulating a decomposed optical edge (c.f., Fig. 20) node would require separate self-similar generators for each transmission rate. This transforms optical burst generation into a problem that can be parallelized into multiple processor cores. However, running each of these self-similar generators on a single CPU, would introduce a significant overhead as the self-similar generators can be extremely time-consuming. Assuming the time required for these generators can be minimized, this would greatly increase the efficiency of optical packet/burst generation simulations.

The aims of the experiments, are to observe whether the use of a GPU for traffic generators can improve their efficiency and minimize their time requirements. Three types of generators were chosen, a generator based on the GE distribution for lightpath session arrivals, an LRD generator based on the M/G/∞ delay system and an LRD generator based on On/Off sources. The length of the generated samples was limited specifically to reveal potential bottlenecks when running these generators on the GPU versus the CPU.

Simulations of a decomposed optical edge (c.f., Fig. 20) node were conducted on an NVidia 8800 GTX. For this investigation many different scenarios were considered, in order to provide a deeper insight on the GPU's capabilities on various test cases.

Lightpath session arrivals were generated based on the GE-type distribution, so that batches of sessions are taken into consideration. A total of 24,002,560 lightpath sessions were created, initially on one block on the GPU and on three types of general purpose CPUs, single, dual and quad core (c.f., Fig. 21). Initially, measurements were taken by increasing the number of threads that simultaneously ran on the same block.

The results showed that by using only one block (i.e., one multiprocessor) on the GPU, the overall simulation duration was significantly longer when the thread number was lower than 8. As the threads were increasing, the performance improvement was significant, surpassing even a Quad core CPU.

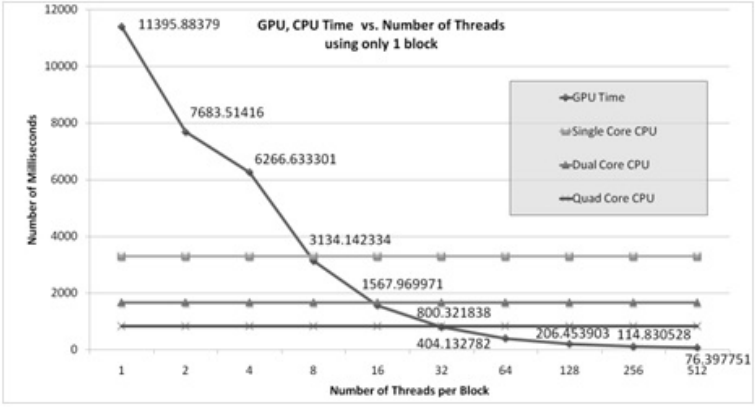

Fig. 21. Simulation duration in milliseconds, of GPU and CPU simulations vs. number of simultaneous threads per block

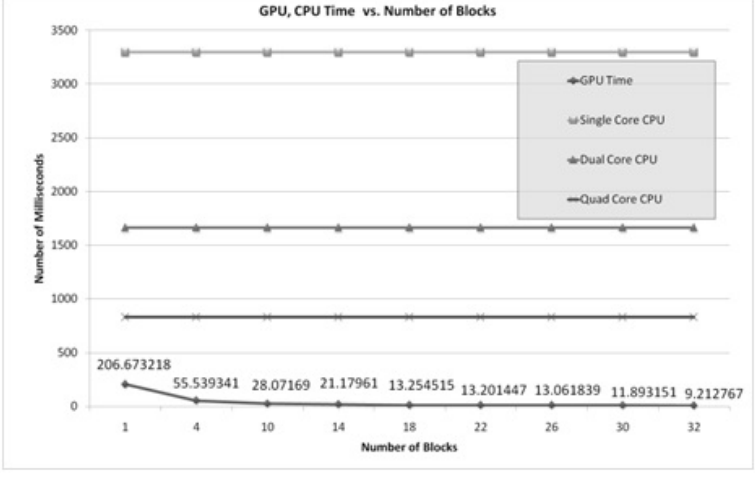

Fig. 22. Performance comparisons between the GPU and different types of CPUs. Graph shows duration in milliseconds versus number of blocks

The same experiment was conducted, while increasing the number of blocks processing the kernel at the same time. This essentially increased the parallelization of the generator, increasing dramatically the performance. To this end, a parallel version of Mersenne Twister was employed to allow for simultaneous random number generators on each of the 190 wavelengths [35].

As the number of blocks is increasing, significant performance improvement is observed, as more blocks use more multiprocessors (c.f., Fig. 22). Note that a total of 16 blocks are required for all multiprocessors of the NVidia 8800 GTX to be used and up to 32 to be properly utilized for maximum performance.

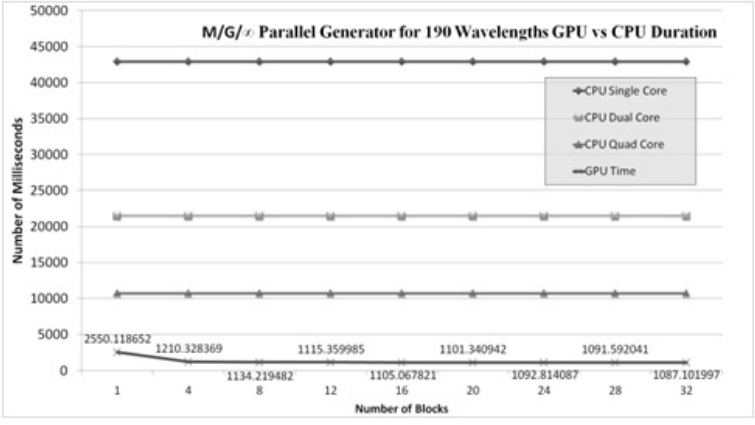

Fig. 23. Performance comparisons between the GPU and different types of CPUs. Graph shows duration in Milliseconds versus number of blocks. Number of blocks start with 1 block using 16 Threads and continues increasing the blocks employing 128 threads per block.

However, as mentioned in the beginning of this tutorial, the Poisson arrival process and the compound Poisson arrival process are not suitable to simulate network traffic for an OBS Network. Therefore, self-similarity generators need to be implemented since most are computational intensive. The M/G/∞ generator was chosen specifically because it is not easily implemented on a parallel system, to allow increased complexity of calculations for each wavelength. To this end, an optical edge node is being simulated having 190 wavelengths. Each wavelength produces streams of optical bursts, with a traffic load that exhibits self-similar properties. For demonstration purposes the generated samples per wavelength were limited to 302084, for a total of 58000128 bursts for the entire node. For each wavelength a series of self-similar traces is generated based on simulating individual M/G/∞ delay systems. The results shown in (c.f., Fig. 23) indicate that the GPU performs even better when the complexity of the problem is high in comparison to the CPU, provided it has been parallelized enough to efficiently utilize the GPU's resources. This observation was also validated in [34].

An experiment with On/Off Sources was also conducted. This method is inherently easy to implement on a parallel system. It involves generating traffic from each wavelength of the fiber independently on a parallel system and then aggregating them into a multiplexed stream. Aggregation, however, needs to run on a single core since it requires access to all generated traffic streams. Essentially this creates a bottleneck, which limits the overall performance of the GPU. To illustrate this, the amount of traffic for each wavelength was limited. As shown in Fig. 24 by increasing the number of threads per block and by using only one block, the GPU eventually surpasses the quad core CPU once the number of threads exceeds 128. Nevertheless, the performance difference is not that great. This is due to the aggregation overhead.

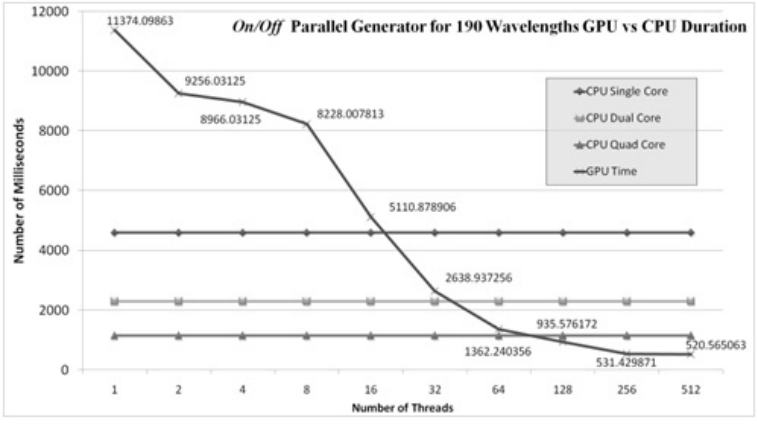

Fig. 24. Performance Comparisons between the GPU and different types of CPUs. Graph shows duration in Milliseconds versus number of threads per block.

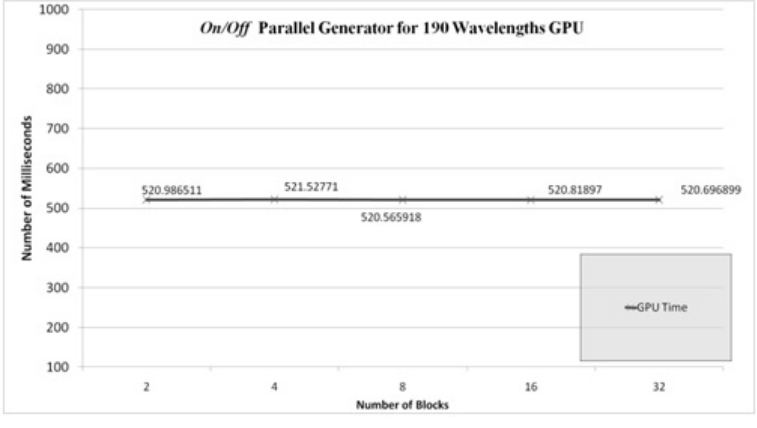

Fig. 25. Performance of GPU while increasing blocks, for a small amount of traffic traces

Furthermore, it is shown in Fig. 25 that by increasing the number of blocks, no real difference in performance, is observed since the number of samples is too small whereas the aggregation overhead too great.

Increasing the number of samples would actually increase the performance difference between the CPU and the GPU, while at the same time diminishing the aggregation's overhead effect. This leads to the conclusion that without proper parallelization, the GPU remains significantly underutilized, thus making the general-purpose CPU the main choice for processes that were programmed without considering a parallel system.

7 Conclusions and Further Work

The traffic characterization of an OC-192 backbone network link was conducted, from traces retrieved by CAIDA [16]. The suitability of the GE distribution, with variable parameters based on the self-similar traffic slot bin was demonstrated and differences in the IP packet size distribution were presented. These traffic characteristics were used in the OBS protocol simulations to prevent biased results. The exposition showed that although the traffic load is self-similar, nevertheless burstiness still occurs in the short-time scales.

The performance of the proposed DOC allocation protocol for high performance OBS networks without wavelength converters was also investigated. Different traffic demands amongst the nodes of the optical network were taken into account and a dynamic updating of the offset was adopted based on the occurrence of blocked bursts and successful transmissions. Preferred wavelengths were also chosen on a source - destination basis. Numerical evaluation results based on simulation were devised focusing on the performance metrics of throughput, mean queue length and blocking probability. Self-similar traffic was generated for each wavelength of every source node in all network topologies employed. Moreover, the MAR and SCV of GE-type interarrival times of IP packets were calculated for each time slot. It was observed in many measurements taken that both the overall and each source - destination pair blocking probabilities under the DOC protocol were greatly decreased in comparison to those associated with the JET protocol. In addition, the DOC protocol achieved fairness in terms of wavelength utilisation and moreover, it was experimentally shown that this protocol, based on a dynamic control of the offset and preferred wavelength, did not increase buffer length requirements at each source node in comparison to those of the JET protocol.

The DOC protocol could be extended to take into consideration the case of broken links of its topology. This will necessitate the creation of a new feature that could be added on top of the variable offset values and the preferred wavelength system. This will cause the OBS network under DOC to pick a new preferred route for each source - destination pair if neither the varying offset values nor the preferred wavelength managed to keep a low value for the associated blocking probability. Choosing an alternate route may allow an OBS network to bypass the broken links in the network achieving, therefore, higher throughput in contrast to other OBS networks. Moreover, statistical analysis of the offset values needs to be conducted whilst further investigation is required under other scenarios where the thresholds of the performance metrics of the network are also adjusted according to their attributes, as appropriate.

The simulations in this tutorial were executed on an NVidia 8800 GTX graphics card acting as a parallel system. Results showed a GPU can provide significantly faster results in comparison to the CPU, provided that the optical node simulation is decomposed properly and the simulation is designed for a parallel system. This allows independent systems based on different wavelengths to be simulated simultaneously. Note that simulations need to have high complexity and duration in order to justify the use of a GPU. However, it is required that

simulation designers develop their simulations efficiently; otherwise worse performance may be experienced. In the future, more and more processing cores are expected to appear and the research community needs to create cost-effective algorithms that take advantage of their fast multiprocessing potential.

Moreover, it is essential to develop an optimal a hybrid simulation approach that utilises in full the available technology and allows multithreaded segments to be calculated on the GPU and single core bottlenecks on the CPU, respectively. In addition a new royalty-free standard, named Open Computing Language (OpenCL), needs further consideration as it is gaining popularity for cross-platform, parallel programming of modern processors found in personal computers, servers and handheld/embedded devices. OpenCL is nowadays adopted by more and more operating systems and platforms, whilst drivers were issued by NVIDIA for the OpenCL to work with CUDA. It can be used to greatly improve speed and responsiveness for a wide spectrum of applications in numerous market categories from gaming and entertainment to scientific and medical software [36].

References

1. Amstutz, S.R.: Burst Switching - an introduction. IEEE Communications Magazine, 36–42 (1983)
2. Listanti, M., Eramo, V., Sabella, R.: Architectural and Technological Issues for Future Optical Internet Networks. IEEE Communications Magazine 38(9), 82–92 (2000)
3. Yoo, M., Qiao, C.: Just-Enough-Time (JET): a high speed protocol for bursty traffic in optical networks. Vertical-Cavity Lasers, Technologies for a Global Information Infrastructure, WDM Components Technology, Advanced Semiconductor Lasers and Applications, Gallium Nitride Materials, Processing and Devices 11(15), 26–27 (1997)
4. Dolzer, K., Gauger, C., Spath, J., Bodamer, S.: Evaluation of reservation mechanisms in optical burst switching networks. AEU Int. J. of Electronics and Communications 55(1) (2001)
5. Verma, S., Chaskar, H., Ravikanth, R.: Optical Burst Switching: A Viable Solution for Terabit IP Backbone. IEEE Network, 48–53 (2000)
6. Tan, K., Mohan, G., Chua, K.C.: Link Scheduling State Information Based Offset Management for Fairness Improvement in WDM Optical Burst Switching Networks. Computer Networks 45(6), 819–834 (2004)
7. Kouvatsos, D.D.: Entropy Maximization and Queueing Network Models. Annals of Operation Research 48, 63–126 (1994)
8. Zhang, Q., Vokkarane, V., Jue, J., Chen, B.: Absolute QoS Differentiation in Optical Burst-Switched Networks. IEEE Journal on Selected Areas in Communications 22(9) (2004)
9. Yu, X., Chen, Y., Qiao, C.: Study of traffic statistics of assembled burst traffic in optical burst switched networks. In: Proceedings, Optical Networking and Communication Conference (OptiComm), Boston, MA (2002)
10. Park, K., Willinger, W.: Self-Similar Network Traffic and Performance Evaluation. John Wiley and Sons, Chichester (2000)

11. Willinger, W., Taqqu, M., Erramilli, A.: A Bibliographical Guide to Self-Similar Traffic and Performance Modeling for Modern High-Speed Networks. Stochastic Networks: Theory and applications. Oxford University Press, Oxford (1996)
12. Jeong, H.-D.J., Pawlikowski, K., McNickle, D.C.: Generation of Self-Similar Processes for Simulation Studies of Telecommunication Networks. Mathematical and Computer Modelling 38, 1249–1257 (2003)
13. Jeong, H.-D.J., Pawlikowski, K., McNickle, D.C.: Generation of Self-Similar Processes for Simulation Studies of Telecommunication Networks. Mathematical and Computer Modelling 38, 1249–1257 (2003)
14. Karagiannis, T., Molle, M., Faloutsos, M., Broido, A.: A Nonstationary Poisson View of Internet Traffic. In: INFOCOM 2004, 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1558–1569 (2004)
15. de Vega Rodrigo, M., Goetz, J.: An analytical study of optical burst switching aggregation strategies. In: Proceedings of the Third International Workshop on Optical Burst Switching, WOBS (2004)
16. Shannon, C., Aben, E., Claffy, K., Andersen, D.: The CAIDA Anonymized 2008 Internet Traces - 19/06/2008 12:59:08 - 19/06/2008 14:01:00 (2008), CAIDA, `http://www.caida.org/passive/passive_2008_dataset.xml` (retrieved December 10, 2008)
17. CAIDA. The Cooperative Association for Internet Data Analysis (2003), `http://www.caida.org/research/traffic-analysis/AIX/` (Retrieved from CAIDA)
18. Kouvatsos, D.D.: Maximum Entropy and the G/G/1 Queue. Acta Informatica 23, 545–565 (1986)
19. Crovella, M., Bestavros, A.: Self-Similarity in World-Wide Web Traffic: Evidence and Possible Causes. In: Proc. ACM SIGMETRICS 1996 (1996)
20. Paxson, V., Floyd, S.: Wide Area Traffic: The Failure of Poisson Modeling. In: Proc. ACM SIGCOMM 1994 (1994)
21. Popescu, A.: Traffic Analysis and Control in Computer Communications Networks. Blekinge Institute of Technology, Stockholm (2007) (preprint)
22. Beran, J.: Statistics for Long-Memory Processes. Chapman and Hall, Boca Raton (1994)
23. Abry, P., Veitch, D.: Wavelet Analysis of Long Range Dependent Traffic. IEEE Transactions on Information Theory (1998)
24. Stallings, W.: High-Speed Networks and Internets: Performance and Quality of Service, 2nd edn. Prentice-Hall, Englewood Cliffs (2002)
25. Taqqu, M., Teverovsky, V.: On Estimating the Intensity of Long-Range Dependence. Finite and Infinite Variance Time Series. Boston University USA, Boston (1996)
26. Qiao, C., Yoo, M.: Choices, Features and Issues in Optical Burst Switching (OBS). Optical Networking Magazine 2 (1999)
27. Xu, L., Perros, H.G., Rouskas, G.N.: A Simulation Study of Access Protocols for Optical Burst-Switched Ring Networks. In: Proceedings of Networking (2002)
28. Aysegul, G., Biswanath, M.: Virtual-Topology Adaptation for WDM Mesh Networks Under Dynamic Traffic. IEEE/ACM Transactions on Networking 11, 236–247 (2003)
29. NVIDIA. NVIDIA CUDA Compute Unified Device Architecture, Programming Guide. NVIDIA (2007), `http://www.nvidia.com/object/cuda_develop.html`
30. Dolzer, K., Gauger, C.: On burst assembly in optical burst switching networks - a performance evaluation of Just-Enough-Time. In: Proceedings of the 17th International Teletraffic Congress (ITC 17), Salvador (2001)

31. Barakat, N., Sargent, E.H.: Analytical Modeling of Offset-Induced Priority in Multiclass OBS Networks. IEEE Transactions on Communications 53(8), 1343–1352 (2005)
32. Matsumoto, M., Nishimura, T.: Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator. ACM Transactions on Modelling and Computer Simulation (TOMACS) 8(1), 3–30 (1998)
33. Hu, G., Dolzer, K., Gauger, C.: Does Burst Assembly Really Reduce the Self-Similarity. In: Optical Fiber Communications Conference, OFC 2003, vol. 1, pp. 124–126 (2003)
34. Triolet, D.: NVidia CUDA: Preview - BeHardware (2007), http://www.behardware.com/art/imprimer/659
35. Feldmann, A.: Characteristics of TCP Connection Arrivals. In: Self-Similar Network Traffic and Performance Evaluation, pp. 367–397. Wiley Interscience, Hoboken (2000)
36. Khronos Group.: OpenCL - The open standard for parallel programming of heterogeneous systems, http://www.khronos.org/opencl/
37. Park, K., Willinger, W.: Self-Similar Network Traffic: An Overview. In: Park, K., Willinger, W. (eds.) Self-Similar Network Traffic and Performance Evaluation, pp. 1–38 (2000)
38. Mouchos, C., Tsokanos, A., Kouvatsos, D.D.: Dynamic OBS Offset Allocation in WDM Networks. Computer Communications (COMCOM) - The International Journal for the Computer and Telecommunications Industry, Special Issue on 'Heterogeneous Networks: Traffic Engineering and Performance Evaluation 31(suppl. 1) (to appear mid, 2010), (in Press Corrected Proof), ISSN 0140-3664, doi: 10.1016/j.comcom.2010.04.009, http://www.sciencedirect.com/science/article/B6TYP-4YVY769-1/2/e3903ceb381e6d5f30adf33d1824281a (available online April 16, 2010)
39. Mouchos, C., Kouvatsos, D.D.: Parallel Traffic Generation of a Decomposed Optical Edge Node on a GPU, in Traffic and Performance Engineering for Heterogeneous Networks. In: Kouvatsos, D.D. (ed.) Performance Modelling and Analysis of Heterogeneous Networks, ch. 20, Aalborg, Denmark. Series of Information Science 7 Technology, vol. 2, pp. 417–439. River Publishers (2009), ISBN 978-87-92329-18-9