Optical Packet Switching with Packet Aggregation

Tamer Khattab, Amr Mohamed, Ayman Kaheel, and Hussein Alnuweiri[1]

University of British Columbia, Lab for Advanced Networking
Vancouver, BC, Canada
E-mail: tkhattab@ece.ubc.ca

Abstract: **In this paper we address the problem of maximizing the utilization of optical network bandwidth by proposing a technique called *packet containerization* at the edge of the optical network. This technique aggregates several packets into a single larger packet called the *container packet.* The container packet is then sent over the optical backbone network. The purpose of packet aggregation is to ensure that container packets have sufficient data to increase the bandwidth utilization of the optical links. To address quality-of-service requirements for delay sensitive applications, we reduce the packet delay inherent in the aggregation process by introducing an *aggregation timeout* threshold to guarantee an upper bound on the aggregation delay. Using simulation models supported by a mean-value mathematical analysis we show that the Containerization with Aggregation Timeout (CAT) algorithm significantly increases the utilization of the optical network bandwidth subject to time-out constraints.**

1 INTRODUCTION

Optical networks are evolving as the solution to the increasing demand for bandwidth in wide-area and metro-area packet networks. Although optical networks provide high bit-rate capacity, there are several challenges facing the use of optical networks as the backbone for packet networks such as the Internet. One of the major challenges is how to efficiently utilize the high bandwidth provided by optical links.

The main contribution of this paper is a new technique, called Containerization with Aggregation Timeout (CAT), that significantly increases the utilization of the optical network bandwidth subject to controllable time-out constraints. The two key aspects of CAT aim to achieve the following:

1. Maximize optical network bandwidth utilization, using the containerization concept.
2. Use the time-out threshold to control the maximum delay an individual packet can encounter due to containerization

CAT relies on aggregating data packets at the edge of the optical network into larger container packets which are transported across the optical backbone using either OPS techniques or Optical Cross Connect (OXC) devices with statically setup paths. The idea of aggregation has been proposed before in OBS [1]. In the case of OBS, however, a separation between packet data and packet header is always assumed and the main emphasis is on synchronization issues such as the use of offset time to separate the packet data from the header section [2][3]. In our model, the optical switching system does not require such separation between packet data and packet header in the container (aggregate) packets. Moreover, to simplify the model and simulate a more realistic world, we do not rely on the header of the container to determine the data path through the network backbone. In other words the backbone is based on OXC nodes rather than OPS nodes. This enables us to focus more on the gains achieved by aggregation as a means to increase bandwidth utilization of the optical backbone.

The next section describes the architecture of the network both from the physical and logical views. Section 3 gives a brief description of the CAT algorithm and how it fits into the proposed network architecture. In section 4, we establish a qualitative mathematical model for the purpose of validating the accuracy of our simulation models. Details of the simulation model parameters and assumptions are provided in section 5 followed by the numerical results of the simulation. We then conclude by emphasizing the outcomes deduced from the numerical results and provide our view of how this work can be extended in the future.

2 NETWORK ARCHITECTURE

2.1 Physical Network View

In order to achieve higher efficiency while keeping the network architecture at a low level of complexity, we have chosen to design our network physical architecture in a hierarchical manner. This will allow for separating the

[1] This work has been supported by The "*National Sciences and Engineering Research Council of Canada*" (NSERC) and the "*Canadian Institute for Telecommunications Research*" (CITR)

aggregation capable (more complex but slower) Optical Packet Switching (OPS) devices from the forwarding only (less complex but faster) Optical cross-connect (OXC) devices.

The physical architecture of the network shown in Figure 1 shows the location of the different switching devices and how they are interconnected together using point to point links to form a sample modeled network. As can be seen from the figure the network consists of five OPS edge devices. Three of the OPS devices are acting as ingress edge devices, while the other two OPS devices are acting as egress edge devices.

In the architecture shown in Figure 1 every ingress OPS is serving two identical independent sources, while each of the egress OPS devices is connected to one receiving station. This allows for simulating most of the possible source to destination permutations.

The group of OPS devices constitute the edge network layer. In the edge network the ingress OPS devices aggregate the incoming packets into containers before sending them to the backbone network, while the egress OPS devices perform the packet de-containerization function on the received packet containers.

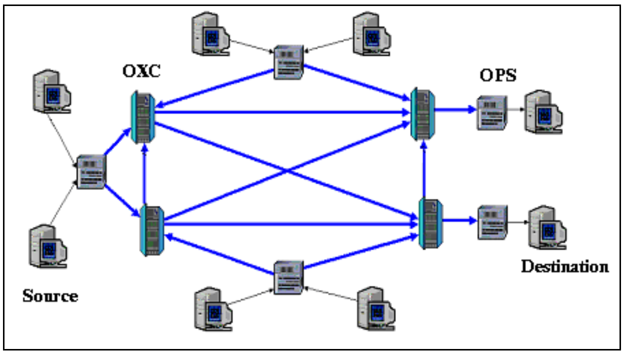

Figure 1 Physical network view

The backbone network consists of four OXC's connected in a full mesh pattern. All the links shown with the thick lines above are optical links capable of carrying up to four separate wavelengths. The transmission speed per wavelength is approximately 2.3 Gbps (OC-48). The links from sources to OPS devices and those from OPS devices to destination nodes are single wavelength links with the same transmission speed.

The OPS devices perform the switching and aggregation functions based on the destination address of the input packet. On the other hand, the OXC devices perform switching based only on the input port and input wavelength of the signal. This makes the OXC devices faster and completely transparent to the data, while being less dynamic. This shortage of dynamic behavior in OXC devices is the main reason of lower efficiency of bandwidth usage when classical packet switching is applied at the edge devices. In order to overcome these inefficiencies, aggregation is performed at the edge devices before sending the data to the core network.

2.2 Logical Network View

To completely describe the architecture of the network, a logical network view must be supplied. The logical view (Figure 2) explains how light paths are configured between sources and destinations over the physical network setup. The establishment of light-paths is done by configuring the OXC devices. This configuration can either be done dynamically through a path setup request initiated by the edge devices before sending an aggregate container packet or statically by the network operator. In our model we use the latter approach. The configuration is done by setting up the OXC to switch a certain wavelength coming from a certain port to another predefined wavelength going over a pre-selected port. By doing this operation for all the wavelengths over all the ports across all the OXC devices, light-paths are established between different OPS devices. These light-paths are seen by the OPS devices as if they are the physical connections [4]. This is why we call this the logical network view.

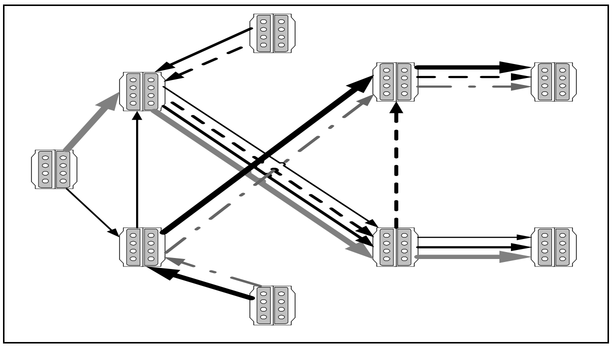

Figure 2 Logical network view

3 PACKET AGGREGATION

In principle, the function of the aggregation process is to assemble the arriving packets into aggregate container packets, which are sent by the OPS to the backbone network. Two parameters control this assembling operation, namely container size (B) in bits and aggregation timeout (T) in seconds. The maximum container size controls the maximum number of packets in the aggregate container. If the incoming traffic arrival rate is high enough, the maximum container size will be reached in a reasonable time. However, if the traffic has a low arrival rate, the container under assembly might have to wait for a relatively long time till the maximum number of packets is reached. In order to avoid large aggregation delays, we use the aggregation timeout parameter. Therefore, the maximum assembly delay an aggregate container can encounter is the aggregation timeout period. .

The aggregation model algorithm works as follows. When a packet arrives at a switch, its destination or egress OPS is determined from its destination address. If there is already a container under assembly destined to the same destination, the packet is added into the container. Otherwise, a new container is created and then the packet is encapsulated into it. The container structure is shown in Figure 3. As can be seen from the figure, a small header is attached to the container. This header consists of two fields, number of packets encapsulated in the container and the address of the destination egress node of the container.

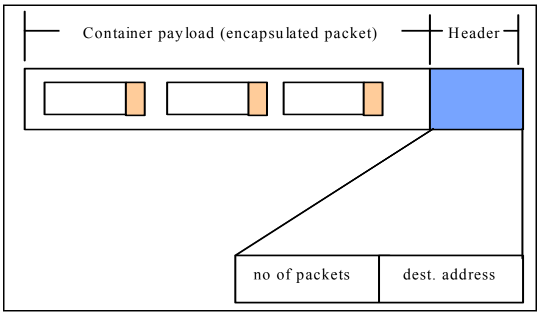

Figure 3 Container packet structure

If the maximum container size is reached or the aggregation timeout period has elapsed, the container is marked as *ready* to be sent, and a new container is created for the following packets. Subsequently, if the destination wavelength at the destination port is free, the container is converted to the optical domain and sent. Otherwise, the container is queued in an electrical domain FIFO queue until the desired wavelength becomes free at which point it is sent to the backbone network. Figure 4 shows the finite state machine model of the OPS containerization engine that satisfies our algorithm.

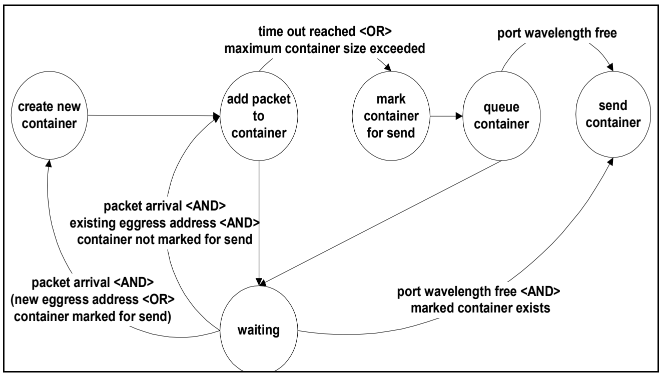

Figure 4 Finite state machine for aggregation process

Finally, when a container arrives at the network egress OPS, it is converted to the electrical domain, and its packets are de-capsulated, queued into a FIFO queue and then sent to their respective destination nodes.

4 MODEL ANALYSIS

The main results in this paper have been produced using simulation models. However, a mean value qualitative analysis will be given in this section to verify the simulation results. To simplify analysis, we treat the network as if the event arrival processes are constant rate. We will use the mean values of the arrival random processes as representatives for the hypothetical deterministic arrival processes with constant rates. Given that we have small inter-arrival time between events relative to the simulation time, the number of events is large enough to yield sufficient accuracy for representing the overall system behavior with the averages of the random processes.

4.1 End to End Delay

In our model, there are three types of delays a packet encounters in its path from the source to the destination: the aggregation delay (D_a) in the ingress OPS, the transmission delay (D_T), and the queuing delay (D_q) at the output of the egress OPS after the de-containerization process.

The packet arrival process is assumed to be a Poisson process[2] with mean value λ packets/sec and the packet size is fixed with size (L) bits. Accordingly, the aggregation delay (D_a) can be calculated as follows:

$$D_a = \frac{1}{N}\sum_{i=1}^{N} D_i$$

Where D_i is the aggregation delay for packet number i in the aggregation queue, and N is number of packets in the aggregate container packet. N can have one of two possible values depending on the arrival rate (λ), the maximum container size (B) and the aggregation timeout (T). In order to find these values we need to consider the two possible situations for a container to be completed and marked for sending. These two situations are timeout is reached first and maximum container size is reached first. If the timeout is reached first, N will be given by:

$$N = \lambda T$$

If the maximum container size is reached first, then

[2] In this case the packet inter-arrival time is an exponentially distributed process with mean value $1/\lambda$.

$N = \left\lfloor \frac{B}{L} \right\rfloor$, where $\lfloor \chi \rfloor$ is the largest integer less than or equal to χ. Finally, the value of D_i is given by:

$$D_i = (i-1)\frac{1}{\lambda}, \qquad 1 \le i \le N$$

Accordingly, we have:

$$D_a = \frac{1}{N} \cdot \frac{N}{2}(N-1)\frac{1}{\lambda}$$

Hence:

$$D_a = \begin{cases} \frac{1}{2\lambda}\left(\left\lfloor \frac{B}{L} \right\rfloor - 1\right) & , \quad \lambda \ge \frac{1}{T}\left\lfloor \frac{B}{L} \right\rfloor \\ \frac{1}{2\lambda}(\lambda T - 1) & , \quad \lambda < \frac{1}{T}\left\lfloor \frac{B}{L} \right\rfloor \end{cases}$$

The transmission delay D_T is constant per packet and independent of the aggregation process. It is given by

$$D_T = \frac{L}{C},$$

where C is the output port speed per wavelength channel.

The last delay factor which is the queuing delay in the egress OPS (D_q) can be deduced in a manner very similar to the case of D_a with the extra consideration that more than one container packet could arrive at the node at the same time. Accordingly, we need to consider delay for each possible arrival scenario and take the average for that.

It can be easily shown that:

$$D_q = \begin{cases} \frac{D_T}{2}\left[\left\lfloor \frac{B}{L} \right\rfloor \frac{(M+1)}{2} - 1\right] & , \quad \lambda \ge \frac{1}{T}\left\lfloor \frac{B}{L} \right\rfloor, \\ \frac{D_T}{2}\left[\frac{\lambda T}{2}(M+1) - 1\right] & , \quad \lambda < \frac{1}{T}\left\lfloor \frac{B}{L} \right\rfloor \end{cases}$$

where M is the number of ingress OPS devices sending to the same egress OPS.

4.2 Channel Utilization

The other performance parameter that we are concerned with in our simulations is the channel bandwidth utilization (U). This parameter is defined as the average transmission rate of data on the channel (where average is taken over a period of time much larger than the inter-arrival time) divided by the link capacity C. The same result with a sufficient degree of accuracy can be obtained through dividing the average time the channel is used by the total time. To calculate the utilization according to the latter definition in practice we divide the packet transmission time (time channel is used) by the inter-arrival time between the current packet and the next one (total time) and take the average over all the packets passing through the system. Hence, one can write:

$$U = \frac{1}{N_p}\sum_{i=1}^{N_p}\frac{D_T}{I_i},$$

where I_i is the inter-arrival time between packet i and packet $i\text{-}1$, and N_p is the total number of packets passing through the channel during the period of calculation.

5 SIMULATION RESULTS

The network architecture shown in Figure 1 and Figure 2 was simulated using the OPNET simulation tool. The sources were configured to have Poisson arrivals with an average inter-arrival time equal to 2 μsec, unless otherwise is stated. The links are configured as OC-48 with link speed equal to approximately 2.3 Gbps per channel. Each link has four channels, with different wavelengths, except for the links coming from data sources to the ingress OPSs or the links coming from egress OPSs to destinations. The links coming from egress OPSs to destinations carry only one channel (wavelength). The packet length is chosen to be of fixed size equal to 1024 bits.

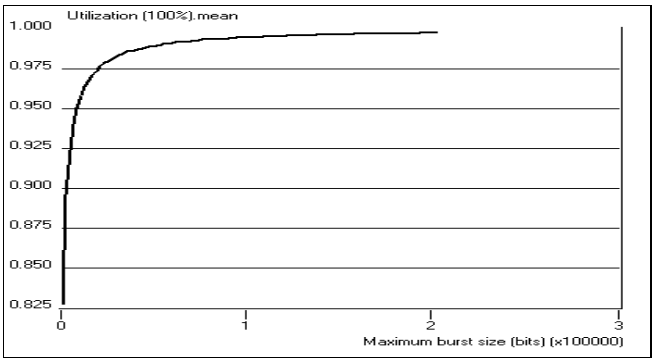

Figure 5 Effect of *B* on utilization for *T* = *1* sec

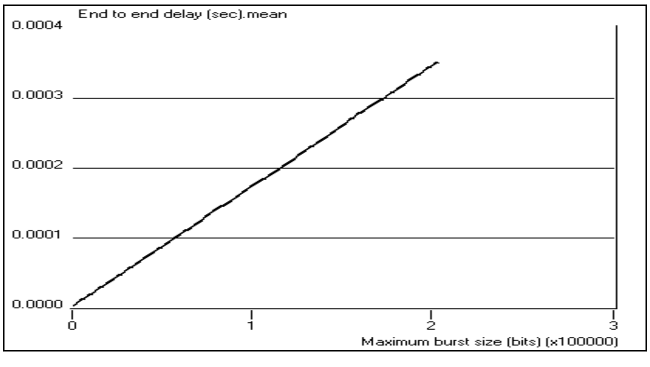

Figure 6 Effect of *B* on delay for *T* = *1* sec

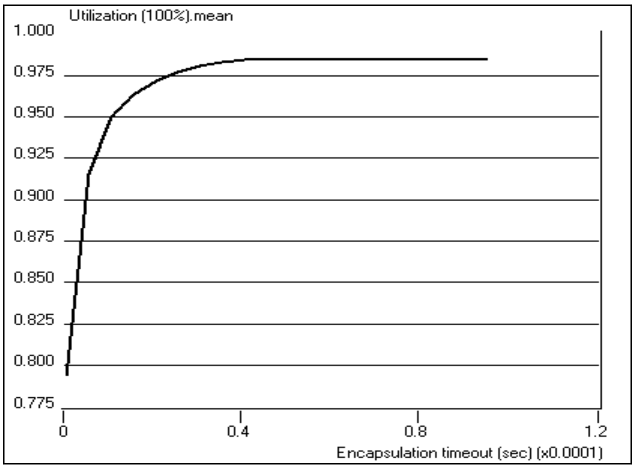

Figure 7 Effect of *T* on utilization for *B = 20480* bits

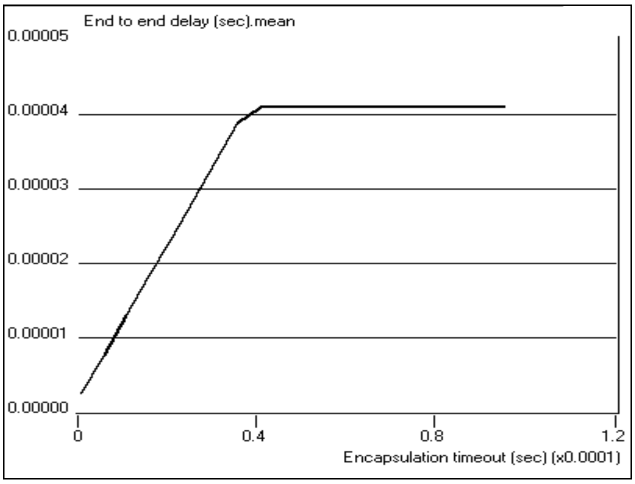

Figure 8 Effect of *T* on delay for *B = 20480* bits

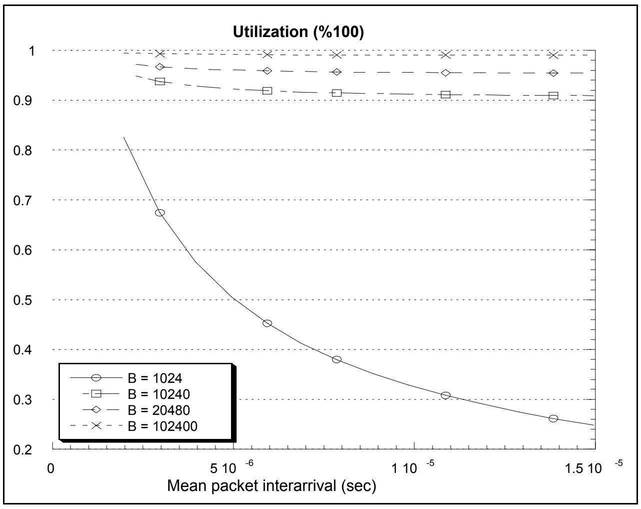

Figure 9 Utilization vs. *1/λ* for *T = 1* sec and different *B*

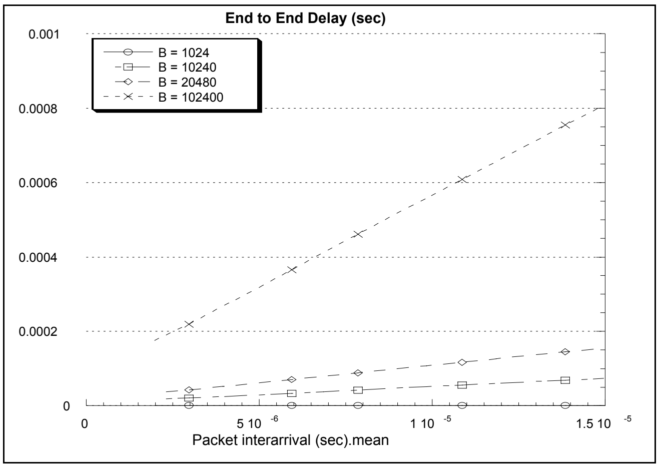

Figure 10 Delay vs. *1/λ* for *T=1* sec and different *B*

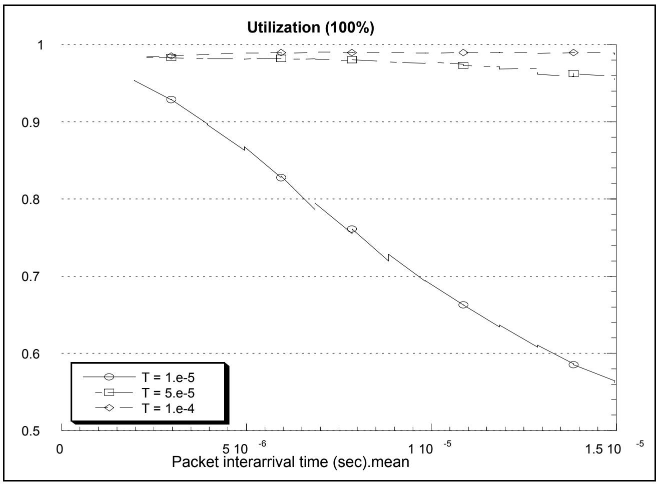

Figure 11 Utilization vs. *1/λ* for *B = 20480* bits and different *T*

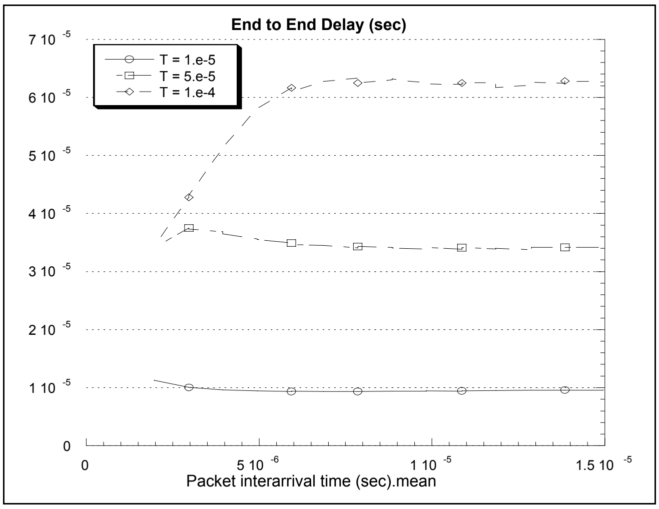

Figure 12 Delay vs. *1/λ* for *B = 20480* bits and different *T*

Figure 5 and Figure 6 clearly indicate that as *B* increases the utilization increases which makes a larger burst size more favorable. On the other hand, as *B* increases the packet end-to-end delay increases, which is the downside of a larger *B*.

Looking at Figure 7 it can be seen that as the aggregation timeout increases the network utilization increases until *T* is equal to 40 µ*sec,* and beyond this point *T* has no effect on the network utilization. This can be understood if we observe that when *T* reaches *40* µ*sec,* it allows the maximum burst size to be reached, at which point *B* becomes the controlling factor. The same result can be seen from Figure 8. As the aggregation timeout increases, the delay increases, this is because a larger *T* allows for a larger aggregation delay and when *T* reaches *40* µ*sec,* the delay becomes constant as the role of *B* becomes the dominant factor.

Taking a closer look at Figure 9 one can compare the utilization values for different maximum container sizes including the case when aggregation is not used (Maximum burst size set to 1024 bit which is one packet). It is clear that the utilization is directly proportional with λ. Furthermore, the OPS with aggregation capability performs much better in terms of utilization as we increase the container size *B*. This is expected as we have shown in Figure 7 and Figure 8.

Figure 10 on the other hand, shows one of the most interesting results of this model. This result is the fact that the delay is directly proportional to the quantity $(1/\lambda)\cdot B$ as we have shown by the mathematical analysis in the previous section. Each line on the graph has a slope directly proportional to *B* and the value of the delay decreases with λ.

In order to explain the curves in Figure 11 and Figure 12 We have to remember that the delay as explained before is equal to sum of three terms; aggregation delay, transmission delay, and the queuing delay at the egress OPS. Now consider the case of $T=10$ µ*sec*. When $1/\lambda=2$ µ*sec* the container will have 5 packets. Accordingly, the transmission delay and egress queuing delay are large (2.2, and 4.18 µ*sec* respectively), and both are proportional to the container size. Therefore, when λ decreases, the number of packets in the container decreases and both the transmission delay and the egress queuing delay decrease. This explains why the delay in this case decreased. This continues till $1/\lambda$ becomes larger than or equal to 5 µ*sec*. At this point the container will always contain one packet and all delay terms will be constant causing the saturation part of the graph.

For the case when $T=50$ µ*sec,* we can see that at the start when λ is very large the average delay increases with the decrease of λ. This is because at that time the effective factor is the maximum container size, which causes the delay to increase with decreasing λ. Then there is a zero slope part, which can be explained by the fact that when the delay reaches its maximum at $1/\lambda=2.5$ µ*sec* ($T=50$ µ*sec* allows 20 packets to be aggregated), the delay will begin decreasing as the average timeout becomes the dominant term in the delay.

6 CONCLUSION

In this paper we have proposed an algorithm called CAT that enhances the efficiency of OPS devices when used as edge devices for a network core consisting of OXC switches. We have also presented simulation results as well as a mean value analytical model for the delay and utilization of networks using our proposed containerization algorithm.

Our simulation as well as analytical results show that the CAT algorithm promises better network utilization. Of course this does not come for free, as can be seen from the increase in the end-to-end delay. However, the simulation results show that the network operator can carefully choose the aggregation parameters such that the utilization can be increased without significantly increasing the end-to-end delay. This can be done by avoiding the points where the delay reaches its maximum values for a given utilization value.

The work presented in this paper can be extended in the future to include QoS mechanisms [5] and see how these mechanisms can be tuned or modified to enhance the performance of optical networks. A further extension to this work is to exploit different switching architectures and see how these architectures can be modified to enhance the overall performance of the network.

7 REFERENCES

[1] Chunming Qiao and Myungsik Yoo, Optical Burst Switching (OBS) – a New Paradigm for an Optical Internet, Journal of High Speed Networks, No.8, 1999

[2] Ilia Baldine et al, JumpStart: A Just-in-Time Signaling Architecture for WDM Burst-Switched Networks, IEEE Communications Magazine, Feb. 2002

[3] Klaus Dolzer et al, Evaluation of Reservation Mechanisms for Optical Burst Switching, International Journal of Electronics and Communications (AEÜ), Vol.55 No.1, Jan 2001.

[4] R. Ramaswami and K. Sivarajan, Optical Networks: A Practical Perspective, Morgan Kaufmann, 1998.

[5] Ayman Kaheel, Tamer Khattab, Amr Mohamed, and Hussein Alnuweiri, "Quality-of-Service Mechanisms in IP-over-WDM Networks", accepted for publication in IEEE Communications Magazine.