International Journal of Engineering and Technology Research
Vol. 1, No. 7, August 2013, PP: 107 - 115, ISSN: 2327-0349 (Online)
Available online at www.ijeatr.org

Research article

A Survey of Burst Assembly Algorithms for Optical Burst Switching (OBS)

Mmoloki Mangwala and O.O. Ekabua

Department of computer science
North West University, Private Bag X2046, Mmabatho, 2735
Tel: +27 18 3892111, Fax: +27 18 3925775

E-mail: {22335285,obeten.ekabua}@nwu.ac.za

Abstract

A lot of research has focused on the need to develop an all-optical next generation network. That is, the development of a network where both transmission and switching are all performed in the optical domain. Optical Burst Switching (OBS) has so far received more attention than its predecessors for such a solution. Burst assembly is one the most crucial elements of OBS. The objective of this paper is to survey the concepts of OBS, burst assembly algorithms presented in literature and the effects of burst assembly on traffic self similarity. After the survey, this paper attempts to provide some directions on future research.

Index terms – Optical Burst Switching, Burst assembly algorithm, Self similarity

1. INTRODUCTION

The abundance of bandwidth brought about by the introduction optical fibre has resulted in a lot of research of optical switching paradigms. The three main paradigms presented in literature are Optical Circuit Switching (OCS), Optical Packet Switching (OPS) and Optical Burst Switching (OBS) [1]. OBS is regarded as the best solution as it has the advantages of both OPS and OCS, and eliminates their disadvantages.In the OBS paradigm, only a few control channels (e.g. one per fibre) go through O/E/O conversion [2]. Given that the data is switched all-optically at burst level, data transparency and statistical multiplexing can be achieved concurrently. The fundamental principle of OBS is that at ingress node, packet headers are separated from data payload. Data payloads destined to the same output port are aggregated and assembled into one data packet called a "burst" of a certain size [3]. The extracted header (control packet) is then sent ahead of its burst with a given offset time. This is to give the control packet enough time to configure the router so that when the burst arrives, the router will be ready. The control packet and the burst are sent in separate dedicated channels. OBS takes advantage of both the huge capacity in fibre for switching/transmission and the sophisticated processing capability of electronics. It is able to achieve cost reduction and leverage the technological advances in both optical and electronic worlds, which makes it a viable technology for the next generation optical Internet. The success of OBS relies much on burst assembly. If proper burst assembly algorithms are employed, burst loss can be minimised and busty traffic can be smoothed.

International Journal of Engineering and Technology Research
Vol. 1, No. 7, August 2013, PP: 107 - 115, ISSN: 2327-0349 (Online)
Available online at www.ijeatr.org

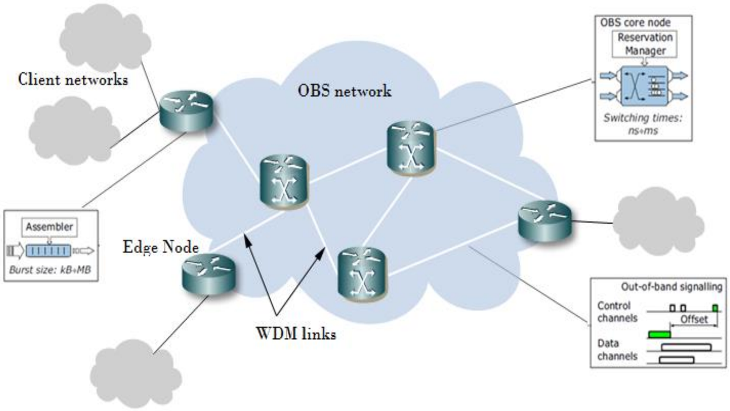

Figure 1: OBS network architecture [3]

The rest of this paper is organized as follows; section 2 discusses the fundamental concepts of optical burst switching, burst assembly and algorithms are presented in section 3. Section 4 presents the effects of burst assembly on traffic self-similarity and section 5 concludes this paper

2. OPTICAL BURST SWITCHING CONCEPTS

An OBS [1] network is made up of electronic edge nodes and optical core nodes. These are interconnected by Dense Wavelength Division Multiplexing (DWDM) links [4]. When data from client networks arrives at the ingress or edge nodes, it is aggregated and assembled into optical bursts and control packet (CP) using various mechanisms. Data payloads destined to the same egress are aggregated into a burst of size determined by the assembly mechanisms. Data bursts to the same destination can share one CP. The CP has all necessary information for forwarding/ routing the data burst such as burst class, length, destination, QoS attributes etc. The CP is sent ahead of its data burst with an offset time. This is to allow for the processing of the CP and for the reconfiguration of the switch fabric. The data burst and the CP are sent on separate channels. Data burst payload is transmitted all-optically while CP undergoes O/E/O conversion. The CP is processed in an electronic controller of the switching node. The controller performs several functions including burst forwarding, resources reservation, contention resolution, QoS provisioning etc [5].

2.1 OBS Node Architecture

An OBS network has two types of edge nodes, that is, the ingress and egress edge node. The ingress edge node is responsible for moulding the client's data signals to the format used by OBS network. That is, the ingress edge node is responsible for aggregation of client data. It then assembles several packets destined to the same egress edge node into a larger burst often referred to as "super packet" and also generates a data burst control packet. The ingress can also setup offset time which is used between the burst control packet and the burst data payload [6]. Figure 2.a shows an OBS ingress edge node. The egress edge node has a de-aggregation unit which disassembles the burst data payload before it is transmitted to the client network. The egress edge node actually performs the opposite of the ingress edge node.

OBS core node is located in the core of the OBS network. OBS core node has the following functional blocks; an input interface, switch fabric, electronic switch controller and an output interface. An OBS core node is shown in figure 2 (b). The major role of the input interface is to extract control and data channels. The control packets extracted from the control channel carry control signals/information [1,2]. Only control information is converted into electrical form. Control packets are processed by the switch controller. Most importantly, the switch controller makes a burst forwarding look-up table and transmission reservation for incoming data payload. The next section discusses burst assembly and burst assembly mechanisms/ algorithms. Burst assembly is one of most important functions performed by the ingress edge node.

International Journal of Engineering and Technology Research
Vol. 1, No. 7, August 2013, PP: 107 - 115, ISSN: 2327-0349 (Online)
Available online at www.ijeatr.org

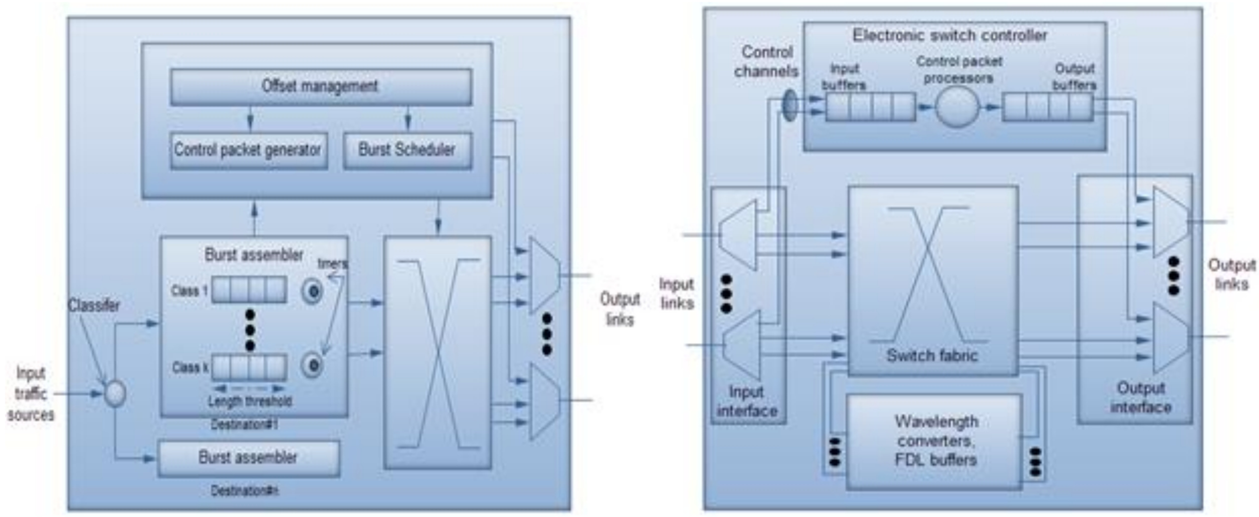

Figure 2: (a): OBS Ingress Node [6] (b): OBS Core Node [6]

3. BURST ASSEMBLY ALGORITHMS

One of the most important issues in the design and implementation of OBS networks is the burst assembly mechanism used at edge nodes. Burst assembly is a procedure whereby incoming packets from various sources are aggregated into bursts at the edge of an OBS network. The switching unit forwards the arriving packets to burst assembly units. Packets which are bound for the same output lines are processed into one burst assembly unit [7]. The burst assembly strategy used influences the traffic characteristics in the network such as traffic burstiness and traffic self-similarity [8]. The strategy implemented also determines the end to end performance of the network. The two main objectives of a burst assembly strategy/mechanism are [7,9].

- To reduce the packet burstification delay so that the overall delay can be reduced and
- To increase the burst size – increasing the burst size will result in the reduction of the number of bursts produced, hence the reduction in the associated processing overhead at the core nodes.

Figure 3 gives an illustration of the performance of a “good” and a “bad” burst assembly mechanism/algorithm. The two objectives above contradict one another in the sense that increasing the burst size will also increase burstification delay.

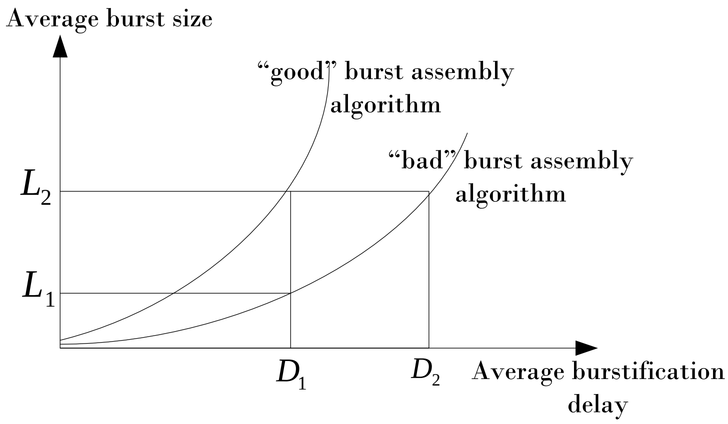

Figure 3: Performance of a "good" and a "bad" burst assembly algorithm [10]

It is therefore imperative that one strikes a balance of these two objectives when implementing a burst assembly strategy. According to [10], such a balance depends on the quality of service requirements of the users and capabilities of the backbone nodes”.

A number of burst assembly algorithms have been discussed in literature. The traditional or rather basic ones are time threshold based algorithms and burst length-based (size threshold based) algorithms [11] [12]. Another burst assembly algorithm which was proposed is the mixed or hybrid time/burst length algorithm. Other variations of these algorithms were also presented in literature [10]. In recent years there have been quite a number of proposals of burst assembly algorithms which are based on traffic prediction/forecasting. In the next sub-section an overview of these algorithms is given.

3.1 Time-based Algorithms

With time-based algorithms [13], a timer is started at the start of each burst assembly cycle. When a determined fixed time T is reached, all packets that would have arrived during that period would be assembled into a burst. In this scheme a minimum burst size L_{min} is set. Therefore all bursts generated should be of size equal to L_{min} or more. If the burst size is less, then it is padded to size equal to L_{min} [13]. Careful consideration should be made when choosing the value of T as large values may lead to unnecessary packet delays at the edge. The other problem is that if the value is small, many small bursts will be generated and that would lead to high control overhead at the core nodes

3.2 Burst Length-based Algorithm

Burst length based algorithm also known as size threshold algorithm was discussed in [11]. Unlike the time-base algorithm, this algorithm uses a fixed burst size L to decide the generation of a burst. Once the L has been reached a burst is assembled and sent to the corresponding output port. The shortcoming of this scheme is that depending on traffic, it may take long for L to be reached and result in undesirable delays at the edge node. When traffic is high, with a smaller value of L, many bursts will be produced resulting in high control overhead at the core nodes.

3.3 Time/ Length Hybrid Algorithm

To deal with the problems faced by the two schemes discussed in section 3.1 and 3.2, a hybrid of the two schemes was discussed in [12,14]. With the hybrid scheme a burst is assembled when either L or T is reached, whichever comes first. When T is reached first and when the burst size is not equal or more than the set L_{min}, then the burst size is padded to L_{min}. Several more variations of the traditional assembly algorithms that have been presented in literature are Data-length Time-lag Product-based assembly algorithm [15], Max-Time-Min-Max-Length assembly algorithm [7], Fixed-Time-Min-Length assembly algorithm [16], Extended-Timer-Based assembly algorithm [17], and others.

3.5 Burst Assembly Algorithms with traffic prediction

In a quest to further improve the performance of the burst assembly described above, a number of variants of the above burst assembly algorithms have been presented in literature. Most notable such algorithms use traffic prediction. Traffic prediction is very important when it comes to burst assembly parameters (burst length or time) choices. If these parameters are well chosen, burstification delay can be reduced greatly. A. Sideri and E.A. Varvarigos in [10] presented four burst assembly algorithms which uses linear prediction filter to predict the number of packet arrivals. M. Mangwala *et al* in [18] proposed a burst assembly algorithms which uses linear predictor with dynamic error compensation (L-PREDEC) [19] to predict on-coming traffic rate. The predicted rate is then used adjust the burst length threshold. The results in [18] show that the algorithm performs better than the traditional assembly algorithms. Similar algorithms were proposed in [20]. In the next sections a brief description of some of them is given.

3.5.1 Adaptive-Threshold with Fixed Maximum Time Limitation Algorithm (ATH-FMTL)

In [20] a burst assembly algorithm called Adaptive-Threshold with Fixed Maximum Time Limitation (ATH-FMTL) was proposed. This algorithm was proposed with the view of providing flexibility to the actual network traffic. This is the aspect which lacks in the traditional burst assembly algorithms discussed earlier. The aim of the ATH-FMTL algorithm is to achieve equilibrium between incoming packets to the transmission of the bursts. That implies that the rate of arrival packets at the ingress node corresponds to the rate at which bursts are formed and transmitted.

ATH-FMTL algorithm uses optimal burst length threshold and fixed maximum time limitation as a condition for burst creation. According to [20], "*The burst length thresholds are increased or decreased in case the burst queue*

size, at the time of burst generation, is larger than upper threshold or smaller than lower threshold, respectively."
The operation of ATH-FMTL is as follows: [20]

1. The packets arrive at the corresponding port and service class assembly queue becomes operative.
2. To classify the packets into the appropriate burst, the decision making is performed based on the fact that every packet has a delay tolerance that allows for flexibility during packet routing and on the assumption that no packet has a delay tolerance less that the amount of time it takes to route the packet through the OBS network, using the shortest route to its destination.
3. Each burst length is estimated at the end of *tp* (prediction time) according to the past burst length value and current arrival traffic.
4. Edge node determines the variable burst assembly duration (VBAD) by estimating burst size with current or previous load.
5. Control packet is sent to OBS core network at time τ ;
   $\tau = t_a - t_o$ (t_a : assembly time; t_o : offset-time)
6. It is assumed that wavelength conversion is available at every Label Switch Router (LSR) node in the core and a Just-Enough-Time reservation scheme is used.
7. The Latest Available Unused Channel with Void Filling (LAUCVF) scheduling scheme is used.

In [20] it was concluded that ATH-FMTL does not only increase the switching efficiency at the OBS core nodes, but also smoothens the input packet traffic and reduce the data loss to some extent. The main factor which was identified as the one which can minimize loss is being able to find an optimal threshold rage.

A. Sideri and E.A. Varvarigos [10], proposed three burst assembly algorithms which are based on the estimation of the expected number of packets arrivals. At the end of each time frame, a decision needs to be taken as to whether the burst should be sent out and a new burst assembly started or the edge node should wait for more packets from the next frame. For such a decision to be taken, a linear prediction filter to estimate the expected number $\hat{N}(n+1)$ of packets arrivals in the next frame $(n+1)$ is used, and check if any specified criteria is fulfilled. The three algorithms based on this principle were proposed in [10] and are discussed below.

3.5.2 Fixed Additional Packets Threshold Algorithm (N_{MIN} algorithm)

In this scheme, a lower bound N_{MIN} is defined. This is the bound on the number of future arrivals above which a decision to wait for more packets before sending a burst is assembled is made. After frame *n*, the estimate is compared with N_{MIN}, and if it is smaller than it, then the burst is assembled and sent with immediate effect or else it waits for another frame to be completed. In this scheme the burst is sent out at the end of the n^{th} frame if and only if;

$$\hat{N}(n+1) < N_{\min} \quad (3.1)$$

3.5.3 Proportional Additional Packets Threshold Algorithm (αL algorithm)

This scheme unlike the fixed additional packets threshold algorithm, uses a fraction of the current length of the burst as the threshold. The burst is assembled at the end of the n^{th} time frame if and only if

$$\hat{N}(n+1) < \alpha.\mathrm{L(n)} \quad . \quad (3.2)$$

where *L(n)* is the burst length, $\hat{N}(n+1)$ is the predictor's estimate for the number of packet arrivals expected during the frame (n+1) and α is the multiplicative parameter.

3.5.4 Average Delay Threshold Algorithm (T_A algorithm)

The Average Delay Threshold Algorithm was also proposed in [10]. The main objective was to improve the Average-Delay-based algorithm which was proposed in [21]. With this scheme the running average delay is calculated and the burst is assembled as soon as the average delay of the packets that comprise it reaches a threshold T_{AVE}. The drawbacks of the algorithm in [21] are;

- Calculating the running average introduces processing overhead, and

International Journal of Engineering and Technology Research
Vol. 1, No. 7, August 2013, PP: 107 - 115, ISSN: 2327-0349 (Online)
Available online at www.ijeatr.org

- Bursts may not be sent out at optimal time.

As a way of addressing these drawbacks, the T_A algorithm uses traffic prediction. The burstification delay is estimated at the end of each frame. The burst is sent out when the estimate exceeds the threshold value T_A. The average packet delay, *D(n)* of the packets in the burst assembly queue at the end of the frame is defined as;

$$D(n) = \frac{\sum_{i=1}^{L(n)} T_i(n)}{L(n)} \qquad (3.3)$$

where *L(n)* is the burst size at the end of frame *n*, $T_i(n) = n.\tau - t_i$ is the delay of the i^{th} packet. *D(n)* can also be computed using recursion as follows;

$$D(n) = \frac{L(n-1).D(n-1) + L(n-1).\tau + \sum_{i=1}^{N(n)} T_i(n)}{L(n-1) + N(n)} \qquad (3.4)$$

where *N(n)* is the number of packet arrivals during frame *n*. The estimated Average Packet Delay $\hat{D}(n+1)$ at the end of frame *n+1* is given by;

$$\hat{D}(n+1) = \frac{L(n).D(n) + \tau.L(n) + \hat{N}(n+1).\frac{\tau}{2}}{L(n) + \hat{N}(n+1)} \qquad (3.5)$$

A burst is assembled at the end of the n^{th} frame if and only if

$$\hat{D}(n+1) > T_A \qquad (3.6)$$

where T_A is the predefined threshold value.

3.4.5 Average Delay to Burst Size Ratio Improvement Prediction Algorithm (L_{MIN} algorithm)

The L_{MIN} algorithm uses traffic prediction to compute an estimate $\hat{DBR}(n+1)$ of Delay to Burst size Ratio (*DBR*) at the end of frame (*n+1*). The burst is sent if and only if the estimate is worse than the current *DBR (n)*. The *DBR (n)* at the end of frame *n* is given by;

$$DBR(n) = \frac{D(n)}{L(n)} = \frac{\sum_{i=1}^{L(n)} T_i(n)}{L^2(n)} \qquad (3.7)$$

or it can be calculated recursively as;

$$DBR(n) = \frac{L(n-1).D(n-1) + L(n-1).\tau + \sum_{i=1}^{N(n)} T_i(n)}{(L(n-1) + N(n))^2} \qquad (3.8)$$

$\hat{DBR}(n+1)$ at the end of frame (*n+1*) is found as follows;

$$\hat{DBR}(n+1) = \frac{L(n).D(n) + L(n).\tau + \hat{N}(n+1).\frac{\tau}{2}}{(L(n) + \hat{N}(n+1))^2} \qquad (3.9)$$

The burst is assembled and sent out at the end of frame *n* if and only if;

$$\hat{DBR}(n+1) < DBR(n) \,\&\&\, L(n) > L_{MIN} \qquad (3.10)$$

L_{MIN} threshold is used as a lower bound on the length of the bursts.

4 EFFECTS OF BURST ASSEMBLY ON TRAFFIC SELF-SIMILARITY

Traffic self-similarity is one characteristic of traffic which many researchers have shown interest in. Z. Sahinoglu and S. Tekinay in [22] stated that "Self-similarity is the property associated with fractals, which are objects whose appearances are unchanged regardless of the scale at which they are viewed". Self-similarity can lead to devastating effects in the networks if it is not handled with care. Self-similarity can result in high traffic loss in small buffered systems and can also lead to undesirable queuing delay in large buffered systems [23]. Studies in [24] and [25] have shown that burst assembly decreases traffic self-similarity. In their research, B. Kantarci, S. Oktug and T. Atmaca [24] made the following observations; 1) increasing the value of time threshold results in a decrease in the degree of self-similarity degree for the time based assembly technique, 2) larger bursts are dropped to give chance to smaller bursts to survive and 3) the degree of self-similarity remains the same when size based assembly technique is used. While sharing the same views as the observations in [24], G. Hu, K. Dolzer, and C. M. Gauger [23] states that except in the case of the time based assembly technique, all other burst assembly techniques do not have any impact on traffic self similarity and thus concluded that burst assembly does not reduce self-similarity. This conclusion is supported by the mathematical proof in [26]. However, S. Azodolmolky, A. Tzanakaki and I. Tomkos in their study of the impact of burst assembly algorithms in OBS networks with self-similar input traffic reported that burst assembly reduces self-similarity [25].

CONCLUSION

This paper presented concepts of optical burst switching and burst assembly algorithms. It is apparent that the use of traffic statistics to predict oncoming traffic is very important. Research shows that burst assembly algorithms where parameters i.e. time or length thresholds are selected based on traffic statistics perform better than the traditional ones. It is there for very important that future research also focus more on traffic prediction mechanisms.

REFERENCES

[1] Chen Yang, Qiao Chunming, and Yu Xiang, "Optical burst switching: a new area in optical networking research ," Network, IEEE , vol. 18, no. 3, pp. 16-23, May-June 2004.

[2] C. Qiao and M. Yoo, "Optical burst switching (OBS) – a new paradigm for an optical internet," J. High Speed Networks, vol. 8, no. 1, pp. 69-84, January 1999.

[3] and J.S. Pu-Fan Mo Yuhua Chen Turner, "Optimal Burst Scheduling in Optical Burst Switched Networks," journal of lightwave technology, vol. 25, no. 8, August 2007.

[4] J. M. Senior, Optical Fibre Communications: Principles and Practice, 2nd ed. [7] J. M. Senior, "Optical Fibre Communications: Principles and Practice", 2nd ed., ser. Intl. Series in Optoelectronics. Prentice Hall, 1992.: Prentice Hall, 1992.

[5] C. Gauger, J. Sp¨ath, and S. Bodamer K. Dolzer, "Evaluation of Reservation Mechanisms for Optical Burst Switching," AEU - International Journal of Electronics and Communications, vol. 55, no. 1, pp. 18-26, 2001.

[6] M. Nord, "Node design in optical packet and optical burst switching," in International Conference on Transparent Optical Networks , Warsaw, 2003, pp. 136 - 143.

[7] Y. Chen, and C. Qiao X. Yu, "Study of traffic statistics of assembled burst traffic in optical burst switched network," in Opticomm, 2002s, pp. 149–159.

[8] B.B. Mandelbrot, The Fractal Geometry of Nature, 1st ed. New York: W.H. Freeman and Company, 1982.

[9] J. Li, Y. Chen, and C. Qiao X. Cao, "Assembling TCP/IP packets in optical burst switched networks," IEEE, vol. 3, pp. 2808–2812, November 2002.

[10] Sideri Angeliki and A. Varvarigos and Emmanouel, "New Assembly Techniques for Optical Burst Switched Networks Based on Traffic Prediction," Computer Engineering and Informatics Dept, University of Patras, Greece, 2007.

[11] K. Haridoss, and J. Jue V. Vokkarane, "Threshold-Based Burst Assembly Policies for QoS Support in Optical Burst-Switched Networks," in Proc. Opticomm, 2003, pp. 125-136.

[12] Nina Skorin-Kapov, "Multicast routing and wavelength assignment in WDM networks: a bin packing approach," Journal Of Optical Networking, vol. 5, no. 4, pp. 266-279, April 2006.

[13] Burak,Oktug, Sema F.,Atmaca, Tülin Kantarci, "“Performance of OBS techniques under self-similar traffic based on various burst assembly techniques," Elsevier, vol. 30, no. 2, pp. 315-325, January 2007.

[14] J. Li, Y. Chen and C. Qiao X. Cao, "Assembling TCP/IP Packets in Optical Burst," in Proc. IEEE Globecom, 2002.

[15] Yuan Chi, Huang Junbin, Li Zhengbin, and Xu Anshi, "A Novel Burst Assembly Algorithm for OBS networks Based On Data-length Time-lag Product," in Asia-Pacific Conference on Communications, Perth, 2005.

[16] F. Callegati, and L.S. Tamil A. Ge, "On Optical Burst Switching and Self-Similar Traffic," IEEE Communications Letters, vol. 4, no. 3, pp. 98-100, March 2000.

[17] K. Dolzer, "Assured Horizon - A New Combined Framework for Burst Assembly and Reservation in Optical Burst Switched Networks," in NOC , 2002.

[18] Mmoloki Mangwala, Boyce B Sigweni, and Bakhe M Nleya, "Efficient burst assembly algorithm with traffic prediction," in SATNAC, 2011.

[19] Y. Wei, J. Wang, and C.Wang, "A traffic prediction based bandwidth management algorithm of a future internet architecture," in International Conference on Intelligent Networks and Intelligent Systems, Icinis, 2010, pp. 560-563.

[20] Amit Kumar Garg, "Analysis of burst/packet assembly techniques in high-speed optical switching network," Optik-International Journal for Light and Electron Optics, vol. 122, no. 7, pp. 616-619, April 2011.

[21] E. Varvarigos and K. Vlachos K. Christodoulopoulos, "A new Burst Assembly Scheme based on the Average Packet Delay and its Performance for TCP Traffic," Optical Switching and Networking, vol. 4, no. 3-4, pp. 200-2012, November 2007.

[22] Z. Sahinoglu and S. Tekinay, "On Multimedia Networks: Self-Similar Traffic and Network Performance," IEEE COMMUNICATIONS MAGAZINE, vol. 37, no. 1, pp. 48-52, 1999.

[23] Guoqiang Hu, Klaus Dolzer, and Christoph M. Gauger, "Does Burst Assembly Really Reduce the Self-Similarity," in OFC, 2003, pp. 124-126.

[24] Burak Kantarci, Oktug Sema, and Tulin Atmaca, "Analysing the Effects of Burst Assembly in Optical Burst Switching under Self-similarity Traffic," in Advanced Industrial Conference on Telecommunications, 2005.

[25] Siamak Azodolmolky, Anna Tzanakaki, and Ioannis Tomkos, "Study of the Impact of Burst Assembly Algorithms in Optical Burst Switched Networks with Self-Similar Input Traffic," ICTON, pp. 36-40, 2006.

[26] A. Neidhardt, F. Huebner, and A. Erramilli, "Shaping and Policing of Fractal Traffic," IEICE Transactions on Communications, vol. E81-B, no. 5, pp. 858-869, 1998.

Mangwala, Mmoloki received his Bachelors of Science degree in Computer Engineering in 2007 from University of KwaZulu Natal and Masters of Science degree in 2011 from North West University. His research interests include Next Generation Networks and e-Government. He is currently a PhD candidate at North West University working on Next Generation Networks.

Ekabua, Obeten O. is a Professor and Departmental Chair of the Department of Computer Science in the North-West University, Mafikeng Campus, South Africa. He holds BSc(Hons), MSc and PhD degrees in Computer Science in 1995, 2003 and 2009 respectively and has since 1998 been lecturing in the University of Calabar, Nigeria. His research interest is software measurement and maintenance, Cloud and GRID Computing, Cognitive Radio Networks, Security Issues and Next Generation Networks