Optimised Optical Burst Aggregation by Employing Dynamic State-Space Feedback Gain Scheduling Controller

Yixuan Qin, Zheng Lu, Georgios Zervas, Martin J. Reed, David K. Hunter, Reza Nejabati, Dimitra Simeonidou
School of Computer Science and Electronic Engineering, University of Essex,
Wivenhoe Park, Colchester, Essex CO4 3SQ, UK
Email: {yqin,zlu,gzerva,mjreed,dkhunter,rnejab,dsimeo}@essex.ac.uk

***Abstract*—This paper describes the design of a dynamic state-space feedback gain scheduling controller for OBS burst scheduling. It can meet upper layer application requirements, while protecting the network from incurring severe burst loss and end-to-end delay. Through the use of gain scheduling to provide dynamic shifting between different models, the controller provides more accurate modelling than a single linear system approximation. Moreover it employs a MIMO system to consider the interaction between the scheduler parameters, i.e. *TimeOut* and *MaxLenght* in order to provide more accurate modelling than a multiple SISO system approximation. Finally the proposed controller model is verified by the use of OPNET simulation.**

***Index Terms*—OBS, scheduling, aggregation, feedback control, gain scheduling, dynamic state-space, MIMO.**

I. Introduction

The main concept behind using feedback control in an OBS edge router is to employ the measured outputs from the core network (for example, burst delay and loss) to control the OBS edge router scheduler's behaviour, by adjusting aggregation parameters such as burst length threshold and aggregation time. In our studies of OBS edge and core routers, the burst delay and loss depend on the traffic load and control inputs (scheduler parameters).

The traffic load is characterised by the data rate and packet length distribution, however in existing implementations, it cannot be changed by the OBS edge nodes. A traditional OBS edge node system is an open loop system which cannot adapt to any changes in such traffic characteristics. However, the proposed dynamic state-space feedback gain scheduling model can adapt to this, and hence simplify the operation of the network.

The control inputs of the closed loop controller can be used to optimise the perfomance of the OBS network via the burst scheduler. The details of the control inputs will be addressed in Section III. By using feedback control, a method to set the control inputs dynamically to achieve optimised system output (i.e. minimum burst loss, minimum end-to-end delay) is described; the feedback control system model is constructed, formulated and verified, and the system has been shown to be stable and accurate, exhibiting short settling time. The performance of the dynamic state-space feedback gain scheduling controller is verified by simulation.

The rest of the paper is organized as follows. Section II describes and compares relevant existing work. Section III identifies the OBS system model via a dynamic state-space gain scheduling model. Section IV describes the design and verification of the gain scheduling feedback controller. Section V contains the OPNET simulation results, while Section VI concludes this paper.

II. Related Work

Although optical burst switching (OBS) [1] relaxes the implementation problems of optical packet switching (OPS) in terms of optical buffering, switching speed, synchronisation requirements and optical header processing speed, it also incurs a larger end-to-end delay and a higher packet loss rate. Considerable attention has been devoted to addressing and solving various issues in OBS networking. For example, Chen *et al*. proposed novel scheduling and signalling mechanisms to reduece the contention rate and improve the end-to-end delay [2]. These employ non-feedback based networking, where the ingress edge node has to regulate its own traffic through traffic scheduling or rerouting, without considering core network contention and delay, because the ingress edge nodes have no knowledge about the core network's status. Amin *et al*. proposed the use of wavelength conversion to reduce contention and improve end-to-end delay [3]. However, the mechanism fails when either there is no additional wavelength available at the requested output or the number of cascaded wavelength converters along the path from source to destination has reached the limit due to signal deterioration [4]. Farahmand *et al*. proposed a feedback-based contention avoidance mechanism, but it only regulates the transmission rate, which is not sufficient for higher layer protocols to request accurate and differentiated classes of service [5]. For example, an application may require low delay, low jitter and high priority, which will be addressed in this paper.

Pantaleo *et al*. discussed a two independent single-in, single-out (SISO) based application-aware scheduling controller to adjust the aggregator parameters (control inputs), i.e. *burst length* and *delay time* [6]. Indeed, both aggregator parameters affect the system status (plant outputs), i.e. *delay, jitter* and *burst loss*, at the same time and interact with each other, for which SISO is not suitable and will yield inaccurate results

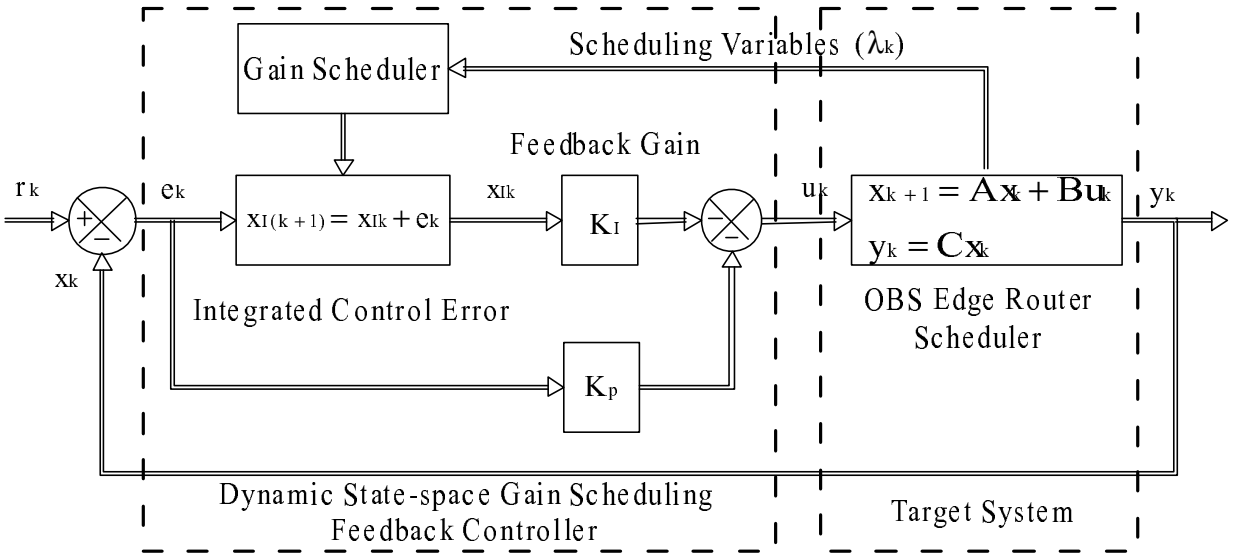


Fig. 1. Dynamic state feedback control system.

for the target system and model. Consequently, in this paper a dynamic multiple-in, multiple-out (MIMO) model is employed in order to yield more accurate performance. Both Huang and Maach *et al*. employed a feedback controller to optimise plant performance, also they all implied that the linearised controller can be approximately correct when functioning in a region close to the operating point [7], [8]. That is true when the system works at a narrow working range, for example, either dedicated to fast switch, for example, high-cost, low-dimension, but fast (1 ns) SOA-MZI based switch (where the burst size is approximately hundreds of kilobits) or slow switch, for example, low-cost, commercial, high-dimension, slow MEMS based switch (10 ms) (in which the burst size is approximately hundreds of megabits), but it is not true with hybrid switch architectures [9] which have more than one operating point. In this paper, dynamic gain scheduling [10] is employed to solve this problem, and it will be discussed in depth in Section III and Section IV.

III. System Identification

Prediction of the system response to varying control inputs is crucial for accurate OBS burst scheduler design. In order to model the implementation procedure accurately but simply in order to make it more feasible, it is assumed that a nonlinear function can be approximately represented by a linear function around the operating point. For each operating point, there is a linear difference equation to represent the system model. Because of the dynamics of the system's working range due to users' on-the-fly requirement of either fast or slow switching, the proposed feedback control system model must employ gain scheduling to choose the appropriate operating point dynamically according to the current control inputs.

Two control inputs are defined:

- *TimeOut* (*TO*): The maximum time that a burst is allowed to wait in the scheduler buffer. This is calculated by measuring the time from the first packet arrival in the scheduler buffer until the time from which the burst starts to be sent out.
- *MaxLength* (*ML*): The maximum burst length before it is sent out. Calculated by counting after the last burst sent, the total number of bytes in all packets stored in the scheduler buffer.

Two system outputs are defined:

- *Loss*: The loss/contention probability at the core node. This is calculated by dividing the number of lost bursts by the total number of burst received at the core, and updating this calculation every five bursts.
- *Delay*: The time taken to arrive the destination edge node. This is calculated as the sum of the delay in the ingress edge node, the switching time in the core nodes and the link transmission delay. This is also re-calculated and updated every five bursts.

The target system is the OBS edge router burst scheduler, as shown in the right-hand side of Figure 1. Note that the double lines in this figure indicate vector-valued variables.

The training data is obtained via OPNET simulation which will be addressed later in Section V to identify the system model. It consists of observations of the control inputs to the system and the corresponding outputs. Moreover, it is represented by a sequence of vector tuples $\{\tilde{\mathbf{u}}_k,\ \tilde{\mathbf{x}}_k\}$ $(k \in [1, N+1])$, (in this paper, vectors are denoted by boldface lowercase letters; matrices are denoted by boldface uppercase letter; others are denoted by normal font letters.) of which $\tilde{\mathbf{u}}_k$ represents the control inputs, i.e. *TO* and *ML*, at the sample time k. The unit of sample time k is the time used by the OBS core router to receive 5 bursts; $\tilde{\mathbf{x}}_k$ represents the corresponding outputs, i.e. *Loss* and *Delay* at the sample time k. The mean value of the training data is represented by $(\bar{\mathbf{u}},\ \bar{\mathbf{x}})$, which is assumed to be the operating point. The offset values are $\mathbf{u}_k = \tilde{\mathbf{u}}_k - \bar{\mathbf{u}}$, $\mathbf{x}_k = \tilde{\mathbf{x}}_k - \bar{\mathbf{x}}$ and are used to model the

target system. In order to model and express concisely the relationships between the inputs and outputs in the MIMO system with considering the interference between inputs, the state-space model is used to describe system dynamics. The initial state-space model used for the target system in this work is shown in Equation (1),

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \\ \mathbf{y}_k = \mathbf{C}\mathbf{x}_k \end{cases} \tag{1}$$

$$\mathbf{x}_k = \begin{bmatrix} \widetilde{loss}_k - \overline{loss} \\ \widetilde{delay}_k - \overline{delay} \end{bmatrix}, \quad \mathbf{u}_k = \begin{bmatrix} \widetilde{TO}_k - \overline{TO} \\ \widetilde{ML}_k - \overline{ML} \end{bmatrix}$$

In Equation (1), $\mathbf{x}_k$ is an 2×2 vector of state variables, $\mathbf{u}_k$ is an 2×2 vector of inputs, $\mathbf{y}_k$ is an 2×1 vector of measured outputs. For simplicity, in this paper, the output variables are also the state variables, meaning that

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

When identifying the slow switch, we get

$$\mathbf{A} = \begin{bmatrix} 0.5864 & -0.6866 \\ -0.0838 & 0.4875 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -1.9732 & 3.9918 \\ 0.2792 & 2.0979 \end{bmatrix}$$

When identifying the fast switch, we get

$$\mathbf{A} = \begin{bmatrix} 0.6893 & -0.7812 \\ -0.0983 & 0.5923 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -2.8343 & 4.9398 \\ 0.9321 & 2.0391 \end{bmatrix}$$

In the above **A** and **B** are obtained by applying the training data to the model which is represented in Equation 1 and using least-squares regression [11]. The assessment of the target system model is summarised in Table I in which the commonly used normalised root-mean-square error ($NRMSE$), variability (R^2) and correlation coefficient (CC) are defined below (Define $\hat{y}$ is the predicted system outputs):

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{N}\sum_{k=1}^{N}(y-\hat{y})^2}}{y_{\text{max}} - y_{\text{min}}}$$

$$\text{R}^2 = 1 - \frac{\text{var}(y-\hat{y})}{\text{var}(y)}$$

$$\text{CC} = \frac{\sum_{k=1}^{N}(y-\hat{y})u_k}{\sqrt{\text{var}(y-\hat{y})\text{var}(u_k)}}$$

after applying this to our model, we get,

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{N}\sum_{k=1}^{N}(y_{k+1} - ay_k - bu_k)^2}}{y_{k\text{max}} - y_{k\text{min}}} \tag{2}$$

$$\text{R}^2 = 1 - \frac{\text{var}(y_{k+1} - ay_k - bu_k)}{\text{var}(y_{k+1})} \tag{3}$$

TABLE I
TARGET SYSTEM MODEL ASSESSMENT.

Model Evaluation		$NRMSE$	R^2	CC
Slow Switch	Burst Loss	15.19%	0.6440	6.5091
	Delay	3.3%	0.9926	2.9483
Fast Switch	Burst Loss	12.13%	0.7340	7.9391
	Delay	4.5%	0.9813	5.3433

$$\text{CC} = \frac{\sum_{k=1}^{N}(y_{k+1} - ay_k - bu_k)u_k}{\sqrt{\text{var}(y_{k+1} - ay_k - bu_k)\text{var}(u_k)}} \tag{4}$$

$NRMSE$ is a frequently-used measure of the differences between values predicted by a model or an estimator and the values actually observed from the parameter being modeled or estimated. Most often, though not always $NRMSE$ is expressed as a percentage, where a smaller value indicates less residual variance. As shown in the Table I, the $NRMSE$ of system output *Delay* is very good, less than 5% for both slow switch and fast switch models. The $NRMSE$ of system output *Burst Loss* is higher, but still well acceptable. R^2 is used to qualify accuracy of the proposed model, it ranges from 0 to 1, with a value close to 1 implying an accurate fit. As shown in the Table I, the R^2 of system out *Delay* is very good, being almost 1. The R^2 for *Burst Loss* is slight lower than broadly accepted, at 0.8. Fortunately, the third measure of the model, correlation coefficient, CC, shows strong relationship between the system inputs, i.e. *TO*, *ML* and the system outputs, i.e. *Delay*, *Burst Loss*, which also proves the accuracy of the training data. From the results shown in Table I, the target system model which is represented by Equation (1) has been shown to be accurate.

IV. Feedback Control System Model

Application-aware OBS (which has been addressed comprehensively in [12]) provides service plane applications with the ability to request not only network resources (for example, computing power, bandwidth and wavelength) but also network characteristics (for example, data delay, jitter and burst size). If the requested network characteristics are not suitable or can not be met for the current network state, the application may not be given the QoS it requires, for example, longer end-to-end delay and/or higher burst loss. The feedback controller can provide a compromise between application requirements and network performance, which at some extent satisfy the application requirements without dramatically affecting the network status. Also the feedback controller can protect the target system, i.e. OBS edge router burst scheduler, which is characterised by Equation (1) from disturbances, i.e. traffic load variations. As discussed above, this is in order to balance the trade-off between application requirements and higher delay or loss. Also to compensate for the disturbance, a feedback controller is used. In order to seek zero steady-state error and a moderate fast transient response time, a controller much

like a proportional-integral (PI) control dynamic state feedback controller is employed (Figure 1). This will determine the new values of the burst scheduler's parameters, i.e., *TO* and *ML* (defined in Section III) offset based on the current control error $\mathbf{e}_k$ and the corresponding integrated control error $\mathbf{x}_{Ik}$ which both are defined in Subsection IV-A. The gain scheduler can shift between two control systems (fast switch or slow switch) based on network status or higher layer application requests. This information is abstracted into Scheduling Variables (λ_k), then sent to the gain scheduler as shown in Figure 1.

A. Feedback Controller Design

Firstly, define the reference signal as $\mathbf{r}$. As shown in the left side of Figure 1, the control errors are offset from the operating point (desired maximum burst loss and end-to-end delay) and are quantified as follows by substituting the second sub-equation of Equation (1),

$$\begin{aligned} \mathbf{e}_k &= \mathbf{r} - \mathbf{y}_k \\ &= \mathbf{r} - \mathbf{x}_k \end{aligned} \tag{5}$$

By combining the first sub-equation of Equation (1) and Equation (5), the dynamics of the control error is

$$\begin{aligned} \mathbf{e}_{k+1} &= \mathbf{r} - \mathbf{x}_{k+1} \\ &= \mathbf{r} - \mathbf{A}\mathbf{x}_k - \mathbf{B}\mathbf{u}_k \\ &= \mathbf{A}\mathbf{e}_k - \mathbf{B}\mathbf{u}_k + (\mathbf{I} - \mathbf{A})\mathbf{r} \end{aligned} \tag{6}$$

The control effort represents to what extent adjustments must be made to the control inputs (*TO* and *ML* offset) in order to regulate the state variables (measured burst loss and end-to-end delay). In other words, the control effort determines the quantity of the control inputs. A smaller control error implies more accurate, shorter settling time and overshoot. A smaller control effort implies smaller control inputs, i.e. *TO* and *ML* that do not have to change much. The trade-off between control effort and control error is as follows: a smaller control error requires a larger control effort, but a smaller control effort lead to a larger control error. In order to optimise this trade-off, linear quadratic regulation (LQR) [13] is employed, which provides an automated way of finding an appropriate state-space feedback controller rather than iteratively refining the pole placement model. $[\mathbf{K}_p\ \mathbf{K}_I]$ is computed to minimise the performance index J defined as

$$J = \sum_{k=0}^{\infty} \left[\mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k\right] \tag{7}$$

When J is minimized, the state $\mathbf{x}_k$ which is defined in Equation (1) as *Burst Loss* and End-to-end *Delay* will be zero or minimized; this will in turn guarantee that the whole OBS burst scheduler system is *stable*. $\mathbf{Q}$ and $\mathbf{R}$ are the weighting matrices which quantify the control error and control effort respectively. Define

$$\mathbf{Q} = \begin{bmatrix} q_{11} & 0 & 0 & 0 \\ 0 & q_{22} & 0 & 0 \\ 0 & 0 & q_{33} & 0 \\ 0 & 0 & 0 & q_{44} \end{bmatrix}$$

as a 4×4 vector because deployed PI controller model coefficients $\mathbf{A}$ will become a 4×4 vector which will be addressed soon in Equation (10). q_{11} and q_{22} weight $\mathbf{e}_k$ (offset of the value of burst loss and end-to-end delay) and are both chosen to be unity to provide equal costs. q_{33} and q_{44} weight $\mathbf{x}_{Ik}$ (the integrated value of $\mathbf{e}_k$) and a value of $q_{33} = q_{44} = 4$ is chosen for a fast switch, in order to influence the choice of feedback gain $\mathbf{K}$ more. Although a larger value of $\mathbf{Q}$ implies larger feedback gain, if it is too large, system oscillation will result. As a result $q_{33} = q_{44} = 45$ is chosen for slow switch, in order to provide a better trade-off. Define

$$\mathbf{R} = \begin{bmatrix} r_{11} & 0 \\ 0 & r_{22} \end{bmatrix}$$

A larger value of $\mathbf{R}$ implies lower control effort. r_{11} weights the control effort of *TO*, while r_{22} weights the control effort of *ML*. The maximum value of *TO* is of the magnitude of $10^2 \mu s$ for a fast switch, and $10^2 ms$ for a slow switch; the maximum value of *ML* is of the magnitude of 10^5 for a fast switch, and 10^8 for a slow switch. In order normalise the range difference between *TO* and *ML*, $\mathbf{R}$ must be carefully chosen. $r_{11} = 1/(10^2)^2, r_{22} = 1/(10^5)^2$ for a fast switch, while $r_{11} = 1/(10^2)^2, r_{22} = 1/(10^8)^2$ for a slow switch.

Secondly, the integrated control error, denoted by $\mathbf{x}_{Ik}$, is shown in Equation (8),

$$\mathbf{x}_{I(k+1)} = \mathbf{x}_{Ik} + \mathbf{e}_k \tag{8}$$

The feedback controller will dynamically determine the value of $\mathbf{u}_k$ based on the current and past values of $\mathbf{e}_k$, which is represented by the control law of Equation (9).

The state vector is now augmented to $[\mathbf{e}_k\ \mathbf{x}_{Ik}]^\top$, the control law is

$$\mathbf{u}_k = -[\mathbf{K}_p \quad \mathbf{K}_I] \begin{bmatrix} \mathbf{e}_k \\ \mathbf{x}_{Ik} \end{bmatrix} \tag{9}$$

in which $\mathbf{K}_p$ is the feedback gain of $\mathbf{e}_k$, and $\mathbf{K}_I$ is the gain associated with $\mathbf{x}_{Ik}$. Combining Equation (6) and (8), the augmented state-space model is

$$\begin{aligned} \begin{bmatrix} \mathbf{e}_{k+1} \\ \mathbf{x}_{I(k+1)} \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{e}_k \\ \mathbf{x}_{I(k)} \end{bmatrix} + \begin{bmatrix} -\mathbf{B} \\ \mathbf{0} \end{bmatrix} \mathbf{u}_k + \begin{bmatrix} \mathbf{I} - \mathbf{A} \\ \mathbf{0} \end{bmatrix} r \\ &= \left(\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{I} & \mathbf{I} \end{bmatrix} - \begin{bmatrix} -\mathbf{B} \\ \mathbf{0} \end{bmatrix} [\mathbf{K}_p\ \mathbf{K}_I] \right) \begin{bmatrix} \mathbf{e}_k \\ \mathbf{x}_{Ik} \end{bmatrix} + \begin{bmatrix} \mathbf{I} - \mathbf{A} \\ \mathbf{0} \end{bmatrix} r \\ &= \left(\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{I} & \mathbf{I} \end{bmatrix} - \begin{bmatrix} -\mathbf{B} \\ \mathbf{0} \end{bmatrix} [\mathbf{K}_p\ \mathbf{K}_I] \right) \begin{bmatrix} \mathbf{e}_k \\ \mathbf{x}_{Ik} \end{bmatrix} \end{aligned} \tag{10}$$

The reference signal r in Equation (10) is set to zero in order to keep the target system as close as possible to the operating point.

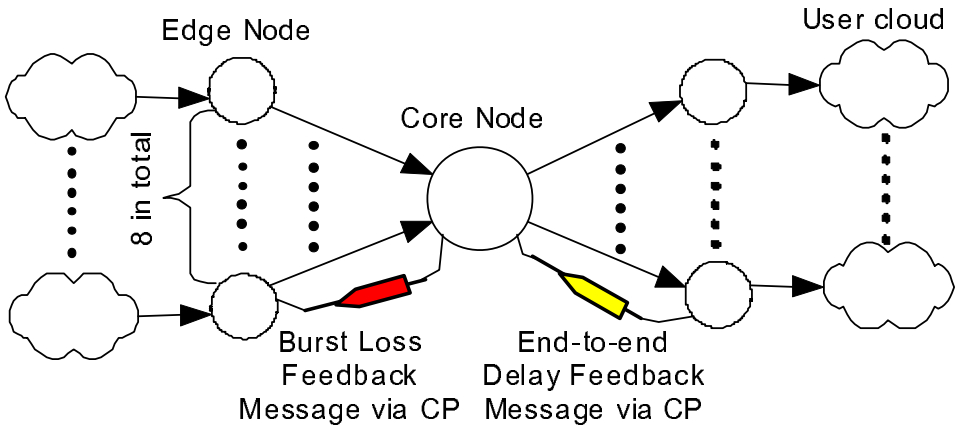


Fig. 2. Simulation scenario.

TABLE II
DYNAMIC STATE-SPACE FEEDBACK CONTROLLER SYSTEM MODEL ASSESSMENT.

	$k_s = -4/\log r$	$M_p = r^{\pi/\theta}$
Slow Switch	10.6248	0.0030
Fast Switch	10.6255	0.0204

Given the orignial values of **A** and **B**, the augmented state-space model, and carefuly chosen **Q** and **R**, by applying the LQR algorithm (for example, `dlqr` command in Matlab), the feedback control gains are obtained.
For a slow switch,

$$\mathbf{K} = \begin{bmatrix}\mathbf{K}_p & \mathbf{K}_I\end{bmatrix} = \begin{bmatrix}-0.6286 & 1.2739 & -0.3308 & 0.6294 \\ 0.0437 & 0.4577 & 0.0440 & 0.3111\end{bmatrix}$$

For a fast switch,

$$\mathbf{K} = \begin{bmatrix}\mathbf{K}_p & \mathbf{K}_I\end{bmatrix} = \begin{bmatrix}-0.2513 & 0.8293 & -0.1627 & 0.3941 \\ 0.1631 & 0.3177 & 0.0744 & 0.2261\end{bmatrix}$$

Expanding Equation (9), the system control law is,

$$\begin{bmatrix}TO_k \\ ML_k\end{bmatrix} = \begin{bmatrix}u_{1k} \\ u_{2k}\end{bmatrix} = -[\mathbf{K}_p \quad \mathbf{K}_I]\begin{bmatrix}\mathbf{e}_k \\ \mathbf{x}_{Ik}\end{bmatrix}$$

$$= -[\mathbf{K}_p \quad \mathbf{K}_I]\begin{bmatrix}e_{loss(k)} \\ e_{delay(k)} \\ x_{loss,I(k)} \\ x_{delay,I(k)}\end{bmatrix}$$

B. Model verification

It is common to verify a system's settling time (k_s) to see how long it takes to reach the steady state. For example, it is important to predict how long it takes the system to return close to the operating point every time the OBS scheduler parameters (*TO*, *ML*) change. If it is too slow, the OBS scheduler may neither have enough time to respond the new upper layer application requirements nor to respond accurately. However, if it is too fast, then it will become prone to overshoot (M_p). By expanding the characteristic polynomial of Equation 10 which is shown below, the dominant pole may be found.

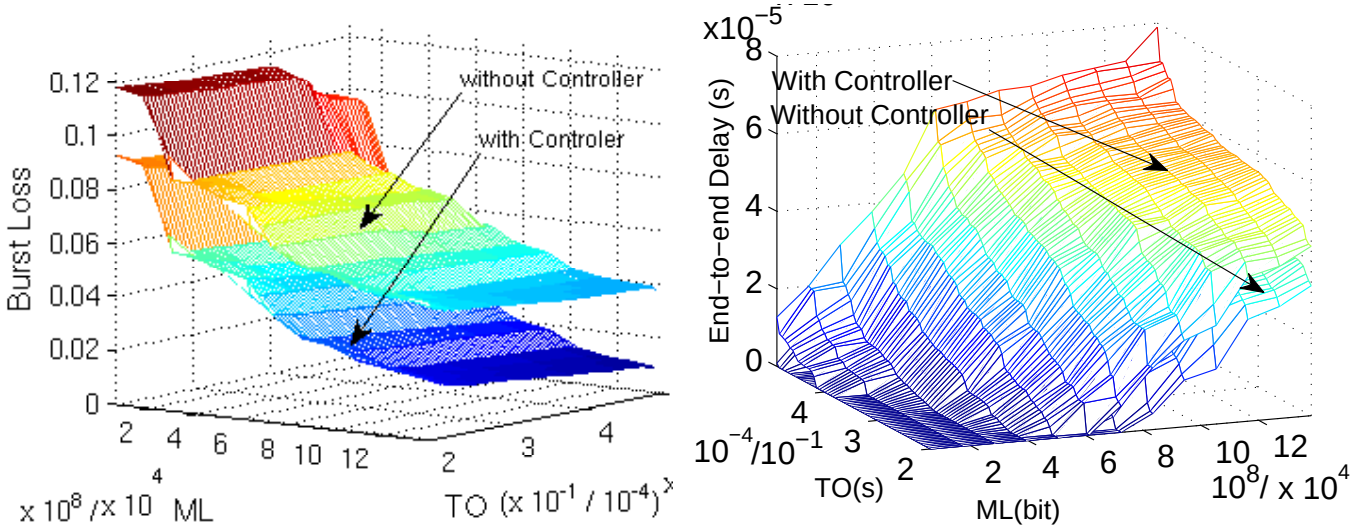
$$det\left(z\mathbf{I} - \begin{bmatrix}\mathbf{A} & \mathbf{0} \\ \mathbf{I} & \mathbf{I}\end{bmatrix} + \begin{bmatrix}-\mathbf{B} \\ \mathbf{0}\end{bmatrix}[\mathbf{K}_p\ \mathbf{K}_I]\right)$$

$r = 0.6863$, $\theta = 0.304$ for a fast switch, while $r = 0.6863$, $\theta = 0.204$ for a slow switch. The corresponding k_s and M_p for a slow and a fast switch respectively are shown to be good in Table II. k_s could be smaller if the static or pre-compensation based feedback controller were deployed, but neither of those have the capacity to reject disturbance.

V. SIMULATION RESULTS

The network is assumed in normal usage to have a accumulated load of 0.6 in the core nodes. There are 8 edge nodes connecting to the ingress links on the core nodes, and there are another 8 edge nodes at the egress as shown in Figure 2. Each edge node is connected to a cloud of users. In order to get a clear characterization of the proposed controller, only one traffic direction is considered in the simulations. At the edge node, the mean packet size is 1370 bytes, i.e. a common video streaming application packet size. The Hurst Parameter of the packet size is 0.7 as broadly used [14], and the Fractal Onset Time Scale is 1.0 for a self-similar distribution. The feedback information is gathered when every 5 bursts are received in order to yield a more reasonable response time for the controller and also keep the load caused by the additional mechanism as low as possible. Then the feedback information is transmitted back to the responding edge node via the control plane.

Simulation results in Figure 3a show that the burst loss rate decreases as either *TO* or *ML* increases. *ML* dominates the effect of burst loss when compared to the effect of *TO*. The proposed OBS burst scheduler employing a dynamic state-space feedback gain scheduling controller always has lower



(a) Burst Loss v.s. *TO* and *ML* in both fast and slow switch

(b) End-to-end Delay v.s. *TO* and *ML* in both fast and slow switch

Fig. 3. Dynamic state-space feedback gain scheduling OBS scheduling controller performance.

burst loss than a conventional OBS burst scheduler. This holds true for both the fast switch model and the slow switch model.

Simulation results in Figure 3b show that the end-to-end delay is decreases as either *TO* or *ML* decrease. *ML* and *TO* have a comparable effect on end-to-end delay. The proposed OBS burst scheduler employing a dynamic state-space feedback gain scheduling controller always has lower end-to-end delay than a conventional OBS burst scheduler. This holds true for both the fast switch model and the slow switch model.

VI. Conclusions

The proposed dynamic state-space feedback gain scheduling controller for OBS burst scheduling has superior burst loss and end-to-end delay performance to a conventional OBS burst scheduler. While it meets requests based upon upper layer requirements for class of service, it also prevents the network from incurring severe burst loss and end-to-end delay due to unreasonable requests which are issued without regard for the network's status. The gain scheduling technology provides the OBS scheduling feedback controller with the ability to shift between different controller systems when the network status changes or, for example, upper layer application requests require either slow or fast switching on the fly. It also applies a MIMO model, in order to consider the interaction between two OBS burst scheduling parameters (*TO* and *ML*), thus providing more accurate modelling than a multiple SISO system approximation.

Acknowledgements

The work described in this paper was carried out with the support of the BONE-project ("Building the Future Optical Network in Europe"), a Network of Excellence funded by the European Commission through the 7th ICT-Framework Programme.

References

[1] C. Qiao and M. Yoo, "Optical Burst Switching (OBS) A New Paradigm for an Optical Internet," *Journal of High Speed Networks*, vol. 8, no. 1, pp. 69–84, Jan. 1999.

[2] Y. Chen, J. S. Turner, and P. Mo, "Optimal Burst Scheduling in Optical Burst Switched Networks," *Jounal of Lightwave Technology*, vol. 25, no. 8, pp. 1883–1894, August 2007.

[3] A. A. Amin and *et al*. "40/10 Gbps Bit-rate Transparent Switching and Contention Resolving Wavelength Conversion in an Optical Burst Router Prototype," in *Proceeding of ECOC 2006*, Cannes, France, Oct. 2006.

[4] X. Gao, M. Bassiouni, and G. Li, "Evaluating the Impact of Wavelength Conversion Cascading on the Performance of Optical Routing Algorithms," in *Proceeding of Communications, Internet, and Information Technology*, St. Thomas, USVI, USA, 2006.

[5] F. Farahmand, Q. Zhang, and J. P. Jue, "A Feedback-Based Contention Avoidance Mechanism for Optical Burst Switching Networks," in *3rd International Workshop on Optical Burst Switching*, San Jose, CA, October 2004.

[6] A. Pantaleo, Y. Qin, G. Zervas, R. Nejabati, D. Simeonidou, and A. Pattavina, "Application-aware Ingress Interface for Dynamic OBS Networks," in *European Conference on Optical Communication (ECOC 2008)*, Brussels, Belgium, September 2008, p. Tu.1.C.4.

[7] H. Anpeng and X. Linzhen, "A Novel Segmentation and Feedback Model for Resolving Contention in Optical Burst Switching," *Photonic Network Communications*, vol. 6, no. 1, pp. 61–67, July 2003.

[8] A. Maach, G. von Bochmann, and H. T. Mouftah, "Congestion Control and Contention Elimination in Optical Burst Switching," *Journal of Telecommunication Systems*, vol. 27, no. 2-4, pp. 115–131, October 2004.

[9] G. Zervas, L. Sadeghioon, D. Klonidis, Y. Qin, R. Nejabati, and D. Simeonidou, "Demonstration of Novel Multi-Granular Switch Architecture on an Application-Aware End-to-End Multi-Bit Rate OBS Network Testbed," in *Proceedings of ECOC, post deadline, 2007*, 2007.

[10] W. Leithead, "Survey of Gain-Scheduling Analysis and Design," *International Journal of Control*, vol. 73, pp. 1001–1025, 2000.

[11] A. Gelman and J. Hill, *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, 2006.

[12] G. Zervas, Y. Qin, R. Nejabati, D. Simeonidou, A. Campi, W. Cerroni, and F. Callegati, "Demonstration of Application Layer Service Provisioning Integrated on Full-Duplex Optical Burst Switching Network Test-bed," in *Proceedings of OFC, post deadline, 2008*, 2008.

[13] P. Dorato, C. Abdallah, and V. Cerone, *Linear-Quadratic Control: An Introduction*. Prentice-Hall, 1995.

[14] M. Kulikovs and E. Petersons, "Remarks Regarding Queuing Model and Packet Loss Probability for the Traffic with Self-Similar Characteristics," *International Journal of Computer Science*, Spring 2008.