MPC-based Management of Computing and Wavelength Resources in Optical Grid Networks

GENKI MATSUI, KIMINAO KOGISO, TAKUJI TACHIBANA, KENJI SUGIMOTO
Nara Institute of Science and Technology
Graduate School of Information Science
8916-5 Takayama, Ikoma, Nara 630-0192
JAPAN
{genki-m, kogiso, takuji-t, kenji}@is.naist.jp

Abstract: In this paper, we propose a dynamic management method with model predictive control (MPC) so that the number of transmitted tasks can be decided based on dynamics of lightpath establish/release process. The proposed method utilize the number of lightpaths that has been established as the dynamics for deriving the number of tasks. This derivation is performed by solving the quadratic programming (QP). By using this method, it is expected that both the computing and network resources can be utilized effectively. We evaluate the performance of the proposed method with Matlab and compare its performance with the performance of PID-based method. In numerical examples, we investigate how computing and network resources can be utilized more effectively by using our proposed method. In addition, we investigate the computation time of our proposed method and the PID-based method.

Key–Words: optical grid, resource management, model predictive control, quadratic programming, computing resources, wavelength resources

1 Introduction

In Grid computing, a massive amount of data can be processed by using geographically dispersed computing resources. Currently, the grid computing has been utilized in scientific, engineering, and business applications. Here, the recent large-scale scientific applications require more expensive and powerful computing resources such as super computers, experimental facilities, and massive storage systems. On the other hand, in the existing grid computing, computer resources are connected via Internet and the high-speed and large-bandwidth connections cannot be provided. As a result, it is hard to process the recent scientific application by using the existing grid computing.

Recently, optical grid has been proposed and implemented in some projects [1], [2], [3]. In optical grid, end-to-end connections called lightpaths are established from a source node to a destination one (See Fig. 1). A massive amount of data can be transmitted with the established lightpaths. Therefore, optical grid is expected to be utilized for processing the recent scientific applications.

Here, computing resources are shared by multiple grid clients in optical grid networks, as is the case with the conventional grid computing. In addition, wavelength resources (network resources) are also shared by multiple clients. Therefore, in optical grid networks, both computing resource and network resource have to be managed dynamically so that grid clients can share those resources effectively.

In order to manage the computing and network resources effectively, [4] has proposed a dynamic resource management based on proportional integral derivative (PID) control. In this method, the number of tasks that are transmitted to resource site with PID control so as to utilize computing resource effectively. Moreover, lightpaths are established or released dynamically based on the output signal of PID control so that network resource can be utilized effectively. Here, in general, it takes some time to establish or release a lightpath. Therefore, frequent lightpath establishment and release degrades the performance of resource management. However, in the PID-based method, dynamics of lightpath establishment and release can not been considered. In this paper, we propose a new dynamic management method with model predictive control (MPC) [5] so that the number of transmitted tasks can be decided based on a dynamics of lightpath establish/release process. The proposed method utilize the number of lightpaths that has been established as the dynamics for deriving the number of tasks. This derivation is performed by solving the quadratic programming (QP). By using this method, it is expected that both the computing and network re-

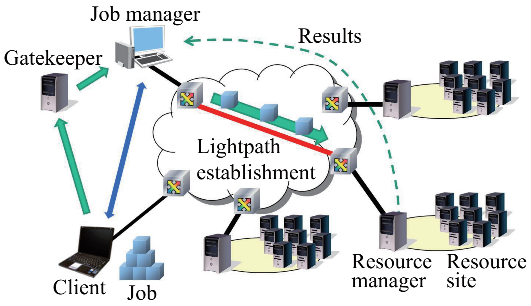


Figure 1: Computing Resource Management.

sources can be utilized effectively.

We evaluate the performance of the proposed method with Matlab and compare its performance with the performance of PID-based method. In numerical examples, we investigate how computing and network resources can be utilized more effectively by using our proposed method.

The rest of the paper is organized as follows. Section 2 shows how computing and network resources are managed in optical grid and introduces some related works. In Section 3, we explain our proposed method with QP-based MPC for dynamic resource management method. Numerical examples are shown in Section 4 and finally, conclusion are presented in Section 5.

2 Resource Management in Optical Grid

2.1 Management Policy

Figure 1 shows how a grid job is executed and how computing resources are managed in optical grid networks. Here, in this paper, we consider the execution of a grid job which consists of numerous independent tasks such as parameter search application [6]. As shown in this figure, once a client generates a job, which consists of independent tasks, the generated job is send to a computing site via lightpath and executed.

Here, computing resources are shared by multiple clients, and hence the number of available computing resources changes over time [7]. Nevertheless, each job should be stably executed without interruption by sharing computing resources effectively. It is also preferable to complete the job execution as quickly as possible. To this end, the resource manager is required to store a constant number of tasks in its own buffer at all times [8].

Then, in optical grid networks, the tasks are transmitted from the job manager to the resource manager with lightpaths. Here, the number of installed wavelengths is limited. In addition, in optical grid, wavelength resources (network resources) are shared by multiple clients. Therefore, lightpaths should be established and released dynamically depending on the number of transmitted tasks so that wavelengths are not wasted.

2.2 Related Work

In terms of the computing resource management in conventional grid, [8] has proposed a dynamic management method so that a resource manager can store a constant number of tasks in its own buffer. In this method, feedback information about the number of tasks in the buffer is returned from the resource manager to the job manager. Then, based on the feedback information, the job manager adjusts the number of transmitted tasks dynamically with proportional integral (PI) control. By simulation using a Matlab and Simgrid that is a discrete time simulator for the Grid, it has shown that grid jobs are executed more stable than a conventional method, and the job execution can complete more quickly.

On the other hand, in terms of the wavelength resource management, many dynamic lightpath establishment methods have been proposed in the literature. These methods often utilize generalized multiprotocol label switching (GMPLS) [9]. With these GMPLS-based methods, lightpaths can be established and released dynamically on demand, and wavelength resources can be shared by many users effectively. These methods can manage computing resources or wavelength resources effectively in optical grid networks. Meanwhile, in optical grid networks, it is expected that both computing and network resources are shared by multiple clients effectively. Nevertheless, the existing methods can not manage both computing and wavelength resources at the same time.

Moreover, in this paper, we utilize MPC in order to manage computer and network resources. Here, MPC determines the current control action by solving online and it is utilized in order to control several types of applications [10], [11]. On the other hand, the optimal control problem in MPC has been more complex and larger recently. Therefore, QP is widely used to derive such a optimal control problem due to its short computation time.

In [10] this paper has utilized MPC for traffic coordination in railway systems, and has shown that proposed control with MPC would be able to recover from delays in an optimal way while breaking connections and letting some trains run faster than the planned speed profile.

In [11], this paper has proposed a neural network MPC of TCP flows and show the superior, transient,

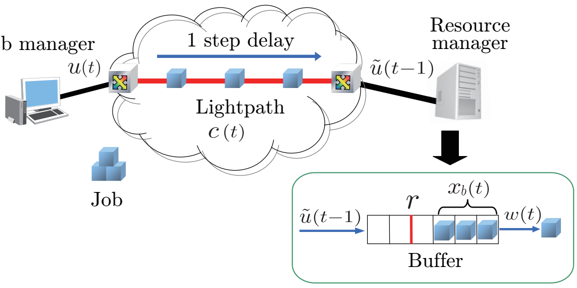


Figure 2: Configuration of optical grid.

steady state behavior and the general stability of MPC over the PI controller as it is applied to optimize the length of a TCP queuing window in a bottlenecked AQM link.

In this paper, we utilize QP with MPC for deriving the number of transmitted tasks quickly.

3 Proposed Dynamic Resource Management

As described in the previous section, in optical grid networks, both computer resources and wavelength resources have to be dynamically managed at the same time. To this end, we propose a dynamic resource management method where MPC is utilized.

Figure 2 shows a configuration of the optical grid, where a job manager transmits tasks to a resource manager with lightpaths. The proposed method is implemented in a job manager, and the job manager consists of a controller, a task transmission controller, and a lightpath controller. Once a client generates a job, a job manager is generated. The job manager selects a computing site where the job is executed and then it manages the job as shown in the following subsections.

3.1 Buffer in Resource Manager

At first, the job manager receives feedback information from the resource manager in terms of the number of tasks in its buffer (the initial value is zero). Let $x_b(t) \in \Re$ denote the number of tasks which are stored in the resource manager's buffer at time t. Here, x_b is given by

$$x_b(t+1) = x_b(t) - Tw(t) + T\tilde{u}(t-1), \quad (1)$$

where $w \in \Re$ $(w \geq 0)$ is the number of tasks that are processed on the computing site and $\tilde{u} \in \Re$ $(\tilde{u} \geq 0)$ is the number of tasks that are transmitted from the job manager to the buffer. Note that $t \in \mathcal{Z}^+$ denotes step ($\mathcal{Z}^+$ is the set of positive integers) and that $T > 0$ is sampling period. Then, $x_b \geq 0$ is finite and can be observed, $\tilde{u}(-1) = 0$.

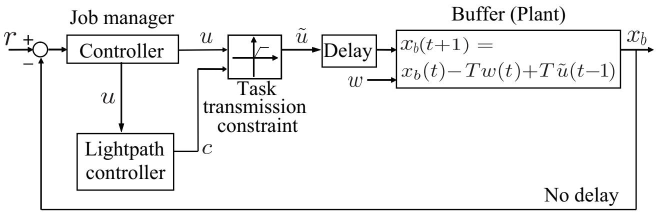


Figure 3: Block diagram.

In this paper, we assume that the resource manager returns the information about $x_b(t)$ to the job manager (see Step 1 in the previous subsection). Just after the job manager receives the feedback information, the job manager transmits the tasks so that $x_b(t)$ is kept close to a reference value r. Here, we assume that the transmission delay of tasks from the job manager to the resource manager is constant, equal to 1 step. This is because lightpaths are used only for the transmission of tasks. On the other hand, we assume that the transmission delay of feedback information is zero due to the simplicity.

3.2 Determination of number of transmitted tasks with QP-based MPC Controller

Next, the controller determines the number of tasks that will be transmitted to the resource manager $u(t)$ so that x_b is kept close to reference value r. In our proposed method, the number of transmitted tasks are determined by using QP-based MPC. Here, Fig. 3 shows a block diagram of our proposed method. In Fig. 3, our proposed QP-based MPC is used in the controller of the job manager.

Now, we define the bandwidth of a lightpath as β, the number of established lightpaths that are available at time t as $n(t)$. In this case, the maximum number $c(t)$ of tasks that can be transmitted is given by $c(t) = n(t)\beta$. As a result, the job manager can transmit at most $c(t)$ tasks. Therefore, the number of transmitted tasks are constrained by $c(t)$. Note that $c(t)$ is a dynamic constraint condition of this system, and it changes depending on the establishment and release of lightpaths.

From the above, a control problem for $u(t)$ can be

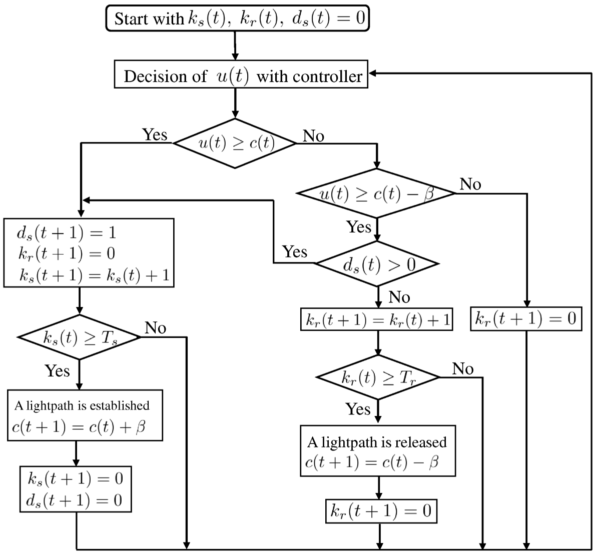


Figure 4: Dynamic lightpath establishment and release algorithm.

formulated as the following QP optimization problem:

$$\min_{\{u_{t+k}\}_{k\in\mathcal{N}}} J(u_t, x(t)) = \min_{\{u_{t+k}\}_{k\in\mathcal{N}}} \sum_{k\in\mathcal{N}} \left\| y_{t+k|t} - r \right\|_{Q_k,2},$$

$$\begin{aligned} \text{s.t.} \quad x_{t+k+1|t} &= Ax_{t+k|t} + B_1 u_{t+k} + B_2 w_{t+k|t}, \\ y_{t+k|t} &= Cx_{t+k|t}, \\ 0 \leq u_{t+j} &\leq c(t), \quad \forall k \in \mathcal{N}, \end{aligned}$$

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ T \end{bmatrix}, B_2 = \begin{bmatrix} -T \\ 0 \end{bmatrix}, C = [1 \quad 0].$$

where N is prediction horizon, and $\mathcal{N} = \{0, 1, \cdots, N-1\}$.

By solving the above QP problem, the optimal sequence $\{u^*_{t+k}\}_{k\in\mathcal{N}}$ is derived. Finally, the optimal number of transmitted tasks $u(t) = u^*_t$ is determined.

3.3 Lightpath Controller

Then, the lightpath controller performs our proposed dynamic lightpath establishment and release processes. Figure 4 shows how a lightpath is established or released by the lightpath controller. As shown in this figure, the lightpath controller compares the number $u(t)$ of tasks in the buffer with the maximum number $c(t)$ of tasks that can be transmitted.

Table 1: Maximum and average value of w.

Time t[s]	0∼10	10∼20	20∼30	30∼40	40∼50	50∼60
γ [Gbit/s]	0.30	0.40	1.0	0.7	0.85	0.6
$E[w]$ [Gbit/s]	0.15	0.20	0.50	0.35	0.425	0.30

Table 2: Maximum and average value of w with A.

Time t[s]	0∼10	10∼20	20∼30	30∼40	40∼50	50∼60
γ [Gbit/s]	A	2A	3A	4A	5A	6A
$E[w]$ [Gbit/s]	0.5A	A	1.5A	2A	2.5A	3A

Once $u(t) = c(t)$ is satisfied at time t, the lightpath controller starts to establish a new lightpath. Here, $T_s \in \mathcal{Z}^+$ denotes the lightpath establishment time and $k_s(t)$ denotes a counter. If $k_s(t)$ becomes equal to T_s, the lightpath establishment process is completed and a new lightpath is established. Here, we assume that T_s is a constant.

On the other hand, in terms of the lightpath release, $T_r \in \mathcal{Z}^+$ denotes the threshold for lightpath release. Moreover, $d_s(t) = 1$ means that a lightpath has been established at time t. If $u(t) \leq c(t) - \beta$ is satisfied and $d_s(t) = 0$, counter $k_r(t)$ increase by one. When $k_r(t)$ becomes equal to T_r, an established lightpath is released. However if $k_r(t) < T_r$ and lightpath establishment process starts, $d_s(t)$ becomes one and $k_r(t)$ becomes zero. This denotes that the lightpath establishment has a higher priority over the lightpath release process. When T_r is set to be small (large), a lightpath is (not) released frequently.

Therefore bandwidth $c(t)$ is determined according to the following process

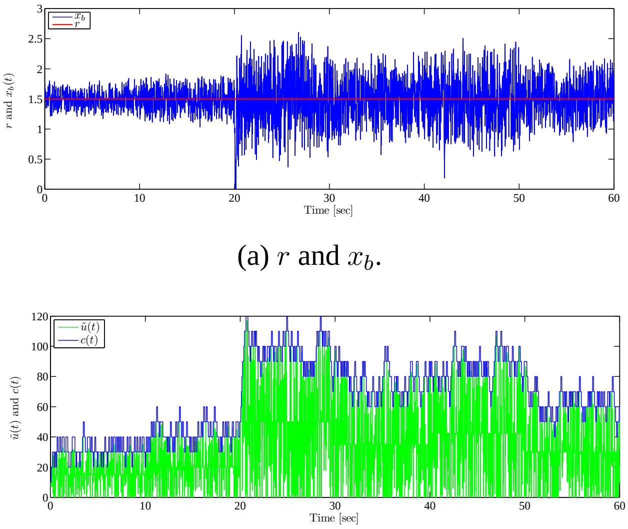
$$c(t + T_s) = c(t) + \beta, \quad \text{if} \quad u(t) = c(t), \tag{2}$$

$$\begin{aligned} c(t + T_r) = c(t) - \beta, & \\ \text{if} \quad u(t+i) < c(t+i) - \beta, & \\ \forall i \in \{0, \cdots, T_r - 1\}. & \end{aligned} \tag{3}$$

In this algorithm, lightpaths are established and released dynamically by comparing output signal $u(t)$ of controller with the maximum number $c(t)$ of transmitted tasks. Therefore, it is expected that computing resources and network resources are effectively managed simultaneously.

4 Numerical Examples

In this section, we evaluate the performance of the proposed method by using Matlab. In the following, unit time T is set to 0.01 s, and simulation time is 60 s. The buffer size in the resource manager is 3.0 Gbits, and the reference value r of the number of tasks in the buffer is 1.5 Gbits. The initial value of the number



(a) r and x_b.

(b) c and $\tilde{u}$.

Figure 5: Time response.

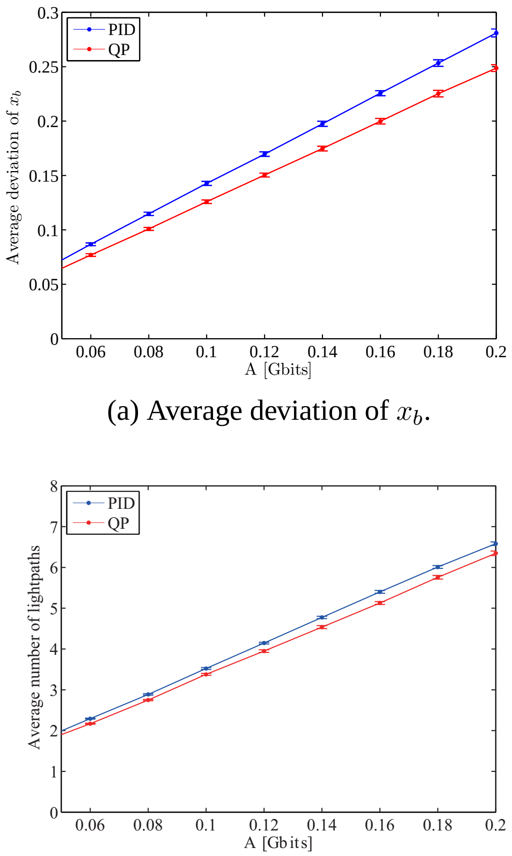
$x_b(0)$ of tasks in the buffer is set to 0.4 Gbits, and the initial value $c(0)$ of constraint is set to 10 Gbps.

Moreover, bandwidth β of a wavelength is set to 10 Gbps, and the maximum number of available wavelengths is 12. The initial value $n(0)$ of the number of established lightpath is set to one. The lightpath establishment time T_s is set to 50 ms and control parameter T_r for lightpath release is set to 50 ms. In terms of the computing resources, the number $w(t)$ of tasks that are processed on a computing site is distributed uniformly on [0, $\gamma(t)$] Gbits. $\gamma(t)$ changes over time t according to Table 1.

Here, the controller does not have the complete information about $w(t)$. Therefore we assume that the controller has the information about the time average $E[w]$ of $w(t)$. As a result, $w(t)$ in equation (1) is replaced by $E[w]$.

4.1 Time response of x_b and c

Figures. 5(a) and (b) show time responses of $x_b(t)$ and $c(t)$, respectively, when QP-based MPC method is used. From Fig. 5(a), we can find that our proposed method tries to make x_b close to r. In Fig. 5(b), the value of constraint c is dynamically increased and decreased by processing of lightpath establishment and release. In addition, we can find that the number of transmitted tasks is determined to satisfy the value of constraint c. From these figures, we can find that our proposed method determine appropriate number of tasks that transmitted and establish and release lightpath depending on the number $w(t)$ of tasks that are processed.



(a) Average deviation of x_b.

(b) Average number of lightpaths.

Figure 6: Performance comparison against γ.

4.2 Performance comparison against γ

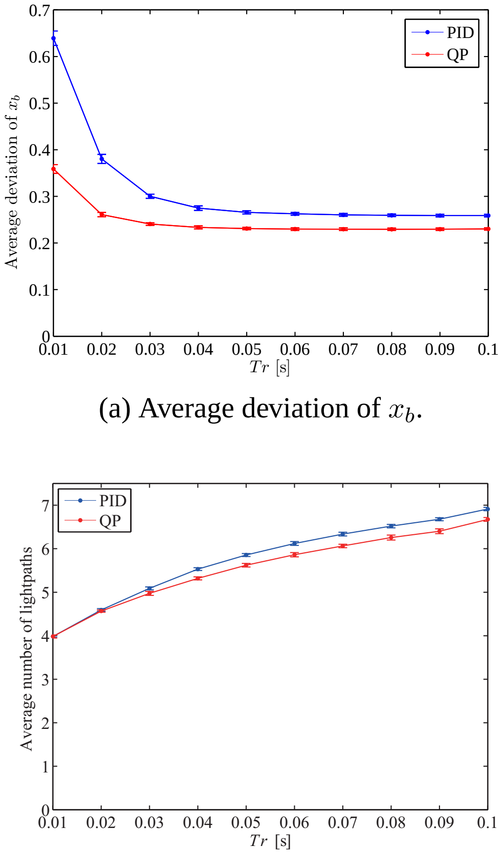
In this subsection, we compare the performance of the proposed method with that of PID method. Here, γ changes over time t according to Table 2. As A becomes large, the number of tasks that can be processed increases.

Fig. 6(a) and (b) show the average of the deviation of x_b against r and the average number of the lightpath, respectively. From Fig. 6(a), we can find the average of the deviation of the proposed method is smaller than that of PID method regardless of A. In addition, in Fig. 6(b), the average number of lightpaths is smaller than that of PID method regardless of A. Therefore, our proposed method with QP-based MPC is more effective than the PID method.

4.3 Performance comparison against T_r

Then, we also compare the performance of the proposed method with that of PID method. Here, threshold of the lightpath release T_r changes from 0.01 s to 0.1 s.

Fig. 7(a) and (b) show the average of the deviation of x_b against r and the average number of lightpaths, respectively. From Fig. 7(a), it is clear that the average of the deviation of the proposed method is smaller than that of PID method regardless of T_r. In addition, in Fig. 7(b), the average number of lightpaths is smaller than that of PID method. Therefore, it is clear that our proposed method is more effective than PID method.



(a) Average deviation of x_b.

(b) Average number of lightpaths.

Figure 7: Performance comparison against T_r.

5 Conclusion

In this paper, we proposed the dynamic resource management method with QP-based MPC which is implemented in a job manager. In the proposed method, QP-based MPC is utilized in order to manage both computing resources and network resources. We evaluated the performance of our proposed method with simulation. From the numerical examples, we found that the proposed method can manage computing resource effectively by using a smaller number of lightpaths than the conventional PID method.

References:

[1] N. Rao, W. Wing, S. Carter, and Q. Wu, “UltraScience Net: Network Testbed for Large-Scale Science Applications,” *IEEE Commun. Mag.*, vol. 43, no. 11, pp. S12-S17, Nov. 2005.

[2] T. Lehman, J. Sobieski, and B. Jabbari, “DRAGON: A Framework for Service Provisioning in Heterogeneous Grid Networks,” *IEEE Commun. Mag.*, vol. 44, no. 3, pp. 84-90, Mar. 2006.

[3] X. Zheng, M. Veeraraghavan, N. Rao, Q. Wu, and M. Zhu, “CHEETAH: Circuit-Switched High Speed End-to-End Transport Architecture Testbed,” *IEEE Commun. Mag.*, vol. 43, no. 8, pp. S11-S17, Aug. 2005.

[4] T. Tachibana, K. Kogiso, and K. Sugimoto, “Dynamic Management of Computing and Network Resources with PID Control in Optical Grid Networks,” in *Proc. ICC 2008*, pp. 396–400, May 2008.

[5] M. Morari, and J.H. Lee, “Model Predictive Control: Past, Present and Future,” *Computers and Chemical Engineering*, vol. 23, no. 5, pp. 667-682, May 1999.

[6] H. Casanova, G. Obertelli, F. Berman, and R. Wolski, “The AppLeS Parameter Sweep Template: User-Level Middleware for the Grid,” *in Proc. Super Computing 2000*, Nov. 2000.

[7] F. Berman, G. Fox, and T. Hey, Eds, *Grid Computing: Making the Global Infrastructure a Reality*, 2003.

[8] H. Osaki, S. Watanabe, and M. Imase, “On Dynamic Resource Management Mechanism Using Control Theoretic Approach for Wide-Area Grid Computing,” in *Proc. IEEE Conf. on Control Applications 2005*, pp. 891–896, Aug. 2005.

[9] E. Mannie, Ed., “Generalized Multiprotocol Label Switching (GMPLS) Architecture,” *IETF RFC 3945*, Oct. 2004.

[10] B. D. Schutter and T. J. J. van den Boom, “Model predictive control for railway networks,” *in Proc. 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM’ 01)*, July 2001.

[11] K. Rahnamai, K. Gorman and A. Gray, “Model Predictive Neural Control of TCP Flow in AQM Network,” *in Proc. IEEE NAFIPS 2006*, June 2006.