Reconfiguration of MPLS/WDM Networks Using Simulation-based Markov Decision Processes

Pedram J. Fard, Richard J. La,
Kwangil Lee, Steven I. Marcus,
Mark Shayman
Electrical and Computer
Engineering Department
University of Maryland
College Park 20742
e-mail: pedram@eng.umd.edu

Abstract —

We present a new approach to the problem of logical topology reconfiguration in congested WDM networks using simulation-based multi-time scale Markov Decision Processes (MDP). Previous work considered total reconfiguration of the network, branch exchanges or addition and deletion of lightpaths when the traffic pattern changes. However, the special case when all or part of the network is congested has not been studied in detail before. We demonstrate that our approach performs well in both uncongested and congested regimes. We restrict the actions taken in reconfiguring the network to branch exchanges to limit the effect of reconfigurations on the number of dropped calls. At each time step a list of possible branch exchanges is formed, and using a heuristic algorithm this (usually long) list is trimmed down to a more manageable list. We use the MDP model to predict the traffic, and then a simulation based online algorithm is used to select one of the branch exchanges as the next action. We apply a policy improvement method to the heuristic policy to obtain a better policy than the heuristic policy.

Keywords – Dynamic traffic, simulations, control theory, stochastic processes, communication networks, optical networks.

I. Introduction

This paper investigates a new approach to the problem of online reconfiguration of the logical topology in WDM (Wavelength Division Multiplexing) networks in the presence of dynamic changes in traffic.

In this study we use an MPLS (Multi-Protocol Label Switching) over WDM infrastructure. Each node is an integrated node consisting of an optical cross connect (OXC) and a label switched router (LSR). This work extends to the case where some nodes only include OXCs with no LSR.

The lightpaths (optical channels) set up between the nodes form the logical topology of the network. In the logical topology, a lightpath is shown as a direct connection between the two end nodes no matter how many OXCs the optical signal goes through to reach the destination node. Each node has a limited number of interfaces that are capable of electronic to optical and optical to electronic conversion and traffic grooming. Each interface can handle one channel, and each channel is associated with one wavelength. We assume that there are as many wavelengths as necessary and if an interface is available a lightpath could be set up. In other words, the number of interfaces is the limiting factor, not the number of wavelengths. This work could be extended to the case where either the number of wavelengths or the number of interfaces is the limiting factor.

In order to reduce the disruption of traffic due to reconfiguration we consider branch exchanges only. We also implement traffic/call migration after each branch exchange such that calls on the lightpaths that are going to be deleted are migrated to other LSPs that are available between the S-D pair.

The MDP model used in this study uses the current traffic measurements and the time-of-day patterns to predict the traffic demand in the near future and the future rewards. Future look-ahead is needed mainly because we restrict the actions to branch exchanges (BEs). Since only one BE is allowed in each time step, to minimize disruption to the ongoing traffic, the action taken at time n not only determines the topology at time n, but also defines the possible topologies at $n+1$. This implies that a sequence of BEs selected by an algorithm may temporarily choose a logical topology that is not optimal for the current traffic demand as a transient topology in order to reach a good logical topology for the predicted demand.

A Multi-time Scale Markov Decision Process (MMDP) model [1] seems to provide a natural framework for this problem since we can distinguish at least three different types of actions that are taken with different frequencies. At the slowest time scale there are logical topology reconfigurations of the optical network, while the moderate time scale includes the Label Switched Path (LSP) reconfigurations that happen more frequently than logical topology reconfigurations. The fastest time scale deals with the distribution of the incoming calls among the existing LSPs between the corresponding S-D pairs; this occurs at the speed of call arrivals. In this study we select heuristic policies for the fast and moderate time scale.

At the slow time scale we first use a heuristic greedy policy to find branch exchanges that improve the logical topology. Then, we apply a policy iteration method to the heuristic policy to generate a rollout policy [2]. The rollout takes the greedy heuristic policy and modifies it, often resulting in an improved, non-greedy policy that utilizes the structure of the model and predicted future configurations and rewards.

Topology reconfiguration in optical networks has been studied under different assumptions. In [4] and [5], a total reconfiguration of the network is studied. A disadvantage of this approach is that a total reconfiguration causes traffic disruption during the transition period. We use small steps (branch exchanges) in addition to a migration scheme that mitigates the adverse effects of a total topology reconfiguration.

The number of branch exchanges needed to go from one topology to another is minimized in [9]. This is only useful if the target topology is known. But, they do not propose a target topology. Also in [3] branch exchanges are used for reconfiguration of the logical topology. However, since they do not use any predictions of traffic pattern the resulting policy is reactive as opposed to the proactive policy proposed here.

The problem of designing the logical topology for a known traffic pattern has been studied in [11], [7], and [8]. However, the exact traffic pattern is not known beforehand in practice. We define a traffic model that has two parts. The first part is deterministic and reflects the time-of-day variations in the traffic demand. This could be the average behavior of the network over many days. The second part is random and reflects the uncertainty of our prior knowledge of traffic, which is the day-to-day variation of traffic.

In [12] the algorithm assumes that there are wavelengths and interfaces available so that it is possible to set up new lightpaths when needed and remove lightpaths when they are significantly under-utilized. This approach works well when the network is not congested. However, when the network is congested this scheme may run out of resources. If other related lightpaths are not lightly utilized (which is the case during congestion) the algorithm is unable to make any adjustments to the topology to reduce the congestion in the network.

The rest of the paper is organized as follows: Section II gives all definitions and assumptions. Section III defines the algorithms used at different time steps. Section IV shows our simulation results and Section V concludes the paper.

II. Definitions and Assumptions

We consider N integrated nodes in the MPLS/WDM network. Each node uses its interfaces and the underlying physical network to connect to other nodes using lightpaths. Each lightpath connects the two end nodes with optical-only signals. S-D pairs that are not connected directly with a lightpath should go through two or more lightpaths with OEO (optical-electronic-optical) conversion in between.

Reconfiguration of the logical topology is allowed only through BEs that are defined on two lightpaths. A BE is formed by swapping one of the two end nodes of a lightpath with one of the end nodes of another lightpath. Hence, each pair of lightpaths has two potential BEs. We assume the lightpaths to be bidirectional and source and destination of a lightpath are interchangeable. The interfaces include one transmitter and one receiver, and the traffic matrix is assumed to be symmetric.

Given the logical topology, LSPs route the traffic from one end node to another. An LSP is a sequence of lightpaths that connects a S-D pair. To limit the search for the LSPs we search for a fixed number of LSPs with the least number of hops. In this study we search for three LSPs with the smallest end-to-end delay for each S-D pair.

The traffic matrix is defined by T, where T_{ij} represents the rate of call arrivals for the S-D pair ij. Each call is assumed to be constant bit rate (CBR). This rate is denoted by C.

Associated with each LSP, LSP^l_{ij}, there is an end-to-end delay, d^l_{ij}, between nodes i and j. Here l denotes a unique index for each LSP. The end-to-end delay for an LSP consists of two components: A constant part, $d^l_{c,ij}$, which is the propagation delay along the path that the LSP travels to go from the source to the destination, and a variable part, $d^l_{v,ij}$, which is the queuing delay at the source node. We assume per LSP queuing, which means that each LSP has a dedicated bandwidth and the queuing delay is only encountered at the source of the LSP and the packets do not encounter any queuing delay at the intermediate nodes. For numerical studies we model the queuing delay using the following formula:

$$d^l_{v,ij} = \frac{1}{L^l_{ij} - Cu^l_{ij}} \tag{1}$$

where L^l_{ij} is the assigned bandwidth, and u^l_{ij} is the number of calls on LSP l between S-D pair ij.

We define $d_{0,ij}$ to be the minimum delay between S-D pair ij among all possible topologies. This would be the best direct path (no OEO conversion) between this S-D pair.

We define three time scales associated with three levels of decision making. At the highest level (slow time scale), decisions about logical topology reconfiguration are made. The frequency of these decisions is on the order of once every few minutes. Each slow time scale step, denoted by n, consists of K moderate time scale steps. At the moderate time scale, bandwidth assignment for the LSPs and/or the set of LSPs can be changed. Each moderate time scale step, denoted by t, consists of J fast time scale steps. At the fast time scale incoming calls for S-D pairs are assigned to one of the available LSPs for that S-D pair. Fast time scale steps are identified by r.

In order to reduce the number of interrupted calls due to BEs, we allow call migration when resources are available.

II.A Reward functions

The reward function at the fast time scale step r is the sum of number of calls that are serviced through the network during a fast time step with a penalty for any additional delay compared to the shortest possible delay for each call. This reward function is defined as

$$R^f(r) = \sum_{i,j} \sum_{l} u^l_{ij}(r)(1 - \alpha(d^l_{c,ij} + d^l_{v,ij}(r) - d_{0,ij})) \tag{2}$$

where $u^l_{ij}(r)$ denotes the number of active calls between S-D pair ij through LSP l at fast time scale step r, and α is a constant that is used to convert the delays to a penalty that is deducted from the reward.

The reward for the moderate or slow time scale is the sum of the rewards for all fast time scale steps in the corresponding slow or moderate time scale step. $g(n)$ is the single step reward for the slow time scale, which is the sum of the fast time scale reward over all fast time scale steps in all moderate time scale steps in the slow time scale step n.

$$g(n) = \sum_{t} \sum_{r} R^f(r) \tag{3}$$

Our goal is to maximize the expected value of the slow time scale reward function (sum of all fast time scale rewards) by choosing the appropriate BEs, LSP selection and bandwidth assignment, and call assignment to LSPs:

$$\max E(\sum_{n} \sum_{t} \sum_{r} R^f(r)),$$

where E denotes expected value.

II.B Traffic Model

Our traffic model consists of two parts: a deterministic part that captures the time-of-day changes in the traffic at different parts of the network and a random part that represents the day-to-day variations in the traffic. Element T_{ij} of the traffic matrix represents the arrival rate of a Poisson process that models call arrivals for S-D pair ij. T_{ij} is defined as follows.

$$T_{ij}(t) = Y_{ij}(t)(1 + Z_{ij}(t))$$

where the deterministic part of traffic, $Y_{ij}(t)$, is a piecewise linear function that is known beforehand. $Z_{ij}(t)$ is the random part of the traffic for S-D pair ij. The elements of the traffic matrix change at the moderate time scale.

For the random part, we choose a traffic model that is more predictable during the early morning hours and as the day goes on, the traffic becomes less predictable, which means the variance of $Z_{ij}(t)$ grows higher at midday. Then again towards the end of the day the traffic becomes more predictable. This leads us to use a Brownian Bridge process for modelling the random part of traffic. The variance of the Brownian Bridge starts from zero and grows large at mid point and then decreases to zero at the end. To generate this process we approximate a Brownian motion by a random walk $X(t)$, then use the following equation to generate $Z(t)$ (see [13]).

$$Z(t) = X(t) - \frac{t}{t_e} X(t_e)$$

where t_e is the end-of-day time.

III. Policies at different time scales

In this study we concentrate on the slow time scale policy, which is the reconfiguration of the logical topology. We fix the moderate and fast time scale policies to the following heuristic policies, and their studies are left for the future.

III.A Fast time scale policy

In the fast time scale, we use a greedy heuristic algorithm that assigns the incoming calls to the LSP with available bandwidth and the least total delay. Since overloading the LSPs would result in large queuing delay and therefore a decline in the reward function (Equation (2)), we impose a limit on the number of calls accepted to an LSP given the corresponding bandwidth L. Once we reach this limit the additional incoming calls will be routed to the next available LSP. This limit is reached by maximizing Equation (2).

$$u_{max} = \frac{L}{C}\left(1 - \sqrt{\frac{\alpha'}{L}}\right) \qquad (4)$$

where α' is defined as

$$\alpha' = \frac{\alpha}{1 - \alpha(d_c - d_0)}$$

d_c, and d_0 are the corresponding constant delay and minimum delay in Equation (2).

At the first fast time scale step immediately after a branch exchange, the calls marked for migration in the slow time scale algorithm (see C) will be assigned to alternate LSPs that have reserved bandwidth for migrating calls (see B). At this point the migration of calls is complete.

		delay, available BW		
S-D	needed BW	LSP1	LSP2	LSP3
2	100	5,100	10,50	11,200
3	100	3,100	7,70	8,50
5	55	4,50	6,25	10,10
1	30	2,35	3,50	5,25
...	...	...	...	...

Table 1: Status of S-D pairs and their LSPs at the beginning, LSP 5 of S-D pair 2 has a common lightpath with LSP 3 of S-D pair 3.

		delay, available BW		
S-D	needed BW	LSP1	LSP2	LSP3
5	55	4,50	6,25	10,10
3	100	7,70	8,50	3,0
1	30	2,35	3,50	5,25
...	...	...	...	...

Table 2: Status of S-D pairs after assigning BW of 100 to S-D pair 2

III.B Moderate time scale policy

Moderate time scale policy takes the arrival rate measurements and the current number of calls on each LSP and the deterministic part of the traffic as inputs and assigns bandwidth to the LSPs.

This algorithm assigns bandwidth to the minimum-delay LSP (or LSPs) for each S-D pair. The list of LSPs available for each S-D pair is sorted based on end-to-end delay. The order in which the bandwidth is assigned to the S-D pairs is determined by the difference in delay between the first and second LSPs of each S-D pair. The reason behind this is that if two S-D pairs are competing for the bandwidth we would want to first assign the bandwidth to the S-D pair that would experience larger degradation in performance going from the first to the second LSP available. If a S-D pair has two LSPs with a small difference in delay, we can assign bandwidth to it later and if the bandwidth that could be used by the first LSP is assigned to other S-D pairs the second LSP would be available with little degradation in performance. This is displayed in an example below.

We assume S-D pairs 2 and 3 need 100 units of BW each and their corresponding best LSPs are competing for the same resources. If the BW is assigned to S-D pair 3, then 100 calls on S-D pair 2 suffer an extra 5 units of delay. If we assign the same BW to S-D pair 2, then 100 calls on S-D pair 3 will suffer only an extra 4 units of delay. Assignment of bandwidth to the S-D pair 2 and the change to the order of S-D pairs can be seen in Tables 1 and 2.

III.C Slow time scale policy

We define two policies for the slow time scale. The first one is a (greedy) heuristic policy that looks into the immediate reward (one step ahead). The second policy uses the first policy and makes policy improvement using online simulations to determine which one of the possible BEs is more likely to yield a better reward during the next few slow time scale steps. We call this a rollout policy.

We use a migration scheme that moves ongoing calls from LSPs that are being terminated, due to the BEs, to the remaining LSPs that have unused bandwidth. If alternate LSPs between the S-D pair do not have any unused bandwidth the calls are dropped.

Heuristic policy

Here we only provide a sketch of the algorithm. We consider every action in a set of admissible BEs (including the null action). For each of these BEs we calculate an estimate of the number of calls that would be assigned to each LSP based on the new topology and the estimate of the arrival rate. Based on this estimate we then find the rate of reward accumulation during the next slow time scale step and use that figure to compare different BEs, and the BE that results in the best rate of reward accumulation is selected.

The details of this algorithm are as follows:

a) Form the set B of admissible actions (BEs) using the algorithm described at the end of this section.

b) For each b_m in set B perform:

1. Find the resulting configuration after b_m.
2. For the S-D pairs that have lost some of their LSPs due to reconfiguration find corresponding 3 best LSPs that connect those S-D pairs. For the S-D pairs that have not lost any LSP but due to the new lightpaths setup, have better LSPs available, we add new and better LSPs and if the old and worse LSPs are not used by any calls we remove them. In this case we may keep more than 3 LSPs for a S-D pair until the calls depart from the old LSPs, when we could remove them from the list.
3. Mark for migration the calls on the LSPs that were terminated because of b_m.
4. Run moderate time scale policy to find the bandwidth assignments.
5. For each S-D pair, based on the traffic demand, we calculate the expected number of calls and assign calls to the LSPs in the order of their delay starting from the LSP with least delay up to the limit imposed by Equation (4) for each LSP and up to the predicted number of additional call arrivals $\tilde{E}_{ij}$.
6. The variable delay is calculated based on the number of calls assigned to each LSP.
7. Given the number of calls in each LSP calculate the following for the selected BE:

$$\theta = E(R^f(r_{ns}))$$

where r_{ns} denotes the first fast time scale step in the next slow time scale step and $E(R^f(r))$ is calculated by plugging in the estimated number of calls for each LSP into Equation (2).

c) Find the BE that results in the highest θ.

Rollout policy

The rollout policy applies a policy improvement method (called rollout) to the above heuristic policy in order to obtain a better policy [2]. At each decision time, n, (slow time scale) we make a decision analysis of all alternative actions, a_n^k. Each action corresponds to one of the admissible BEs. We are trying to maximize the reward, $J_n = E\{g_n(a_n^k) + J_{n+1}\}$. Where, $g_n(a_n^k)$ is the step reward for slow time scale step n given action a_n^k for that step. Since we do not know the exact value of the future rewards J_{n+1}, we estimate it using online simulations (here we run the simulations 5 steps into the future).

$$\max_k E\{g_n(a_n^k) + J_{n+1}\}$$

Thus, assuming that a given action is taken we let the system proceed from the state where it is after that action and use the heuristic policy to make all the subsequent decisions. We calculate the estimated total reward during these 5 steps. This reward is used as a measure of goodness for the original BE. Rollout policy picks the BE from the set of admissible BEs which results in the best 5-step reward.

Reducing the size of the action space in the slow time scale

We use the following algorithm to reduce the number of admissible actions to make the decision process computationally tractable for larger networks. Here we define $B(g)$ to be the set of all admissible BEs, and $B_1(g)$ is the resulting reduced set of admissible BEs.

There are two thresholds defined and used in this algorithm. These two thresholds, γ_l and γ_h are percentages of the lightpath capacity and are configurable.

1. Form set H such that it includes all S-D pairs that have a considerable bandwidth (more than $\gamma_l C$) on one of the congested lightpaths. We consider a lightpath congested if the utilization for that lightpath is more than γ_h.
2. From $B(g)$ add the null BE to $B_1(g)$.
3. Move to the next BE in $B(g)$.
4. If one of the two lightpaths resulting from this BE corresponds to a pair in H add this BE to $B_1(g)$ otherwise discard this BE.
5. Go to step 3.

IV. Numerical results

First we compare the rollout algorithm with the heuristic algorithm, an open-loop policy, and a no-action policy. In the open-loop policy, we use the deterministic part of the traffic to select a sequence of branch exchanges that is used to reconfigure the logical topology from the start to the end of simulation period. We call this an open-loop policy since we make no use of the measurements during the simulation. To form the open-loop sequence of branch exchanges, we let the logical topology for each one of the 50-minute periods defined in A be the topology with direct optical connection between S-D pairs with the highest traffic (subject to the constraint on the number of interfaces) based on the deterministic part of the traffic pattern. In transition periods, we manually find

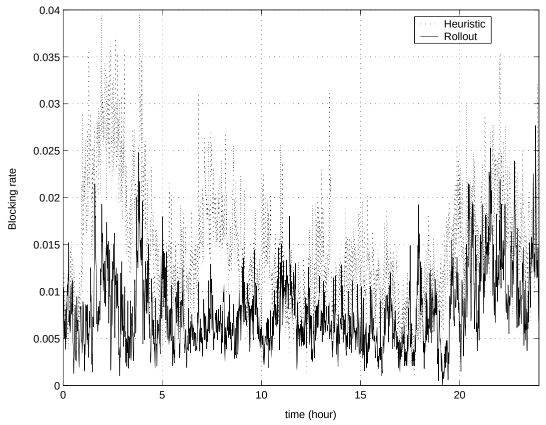

Figure 1: Blocking rate for rollout and heuristic policies

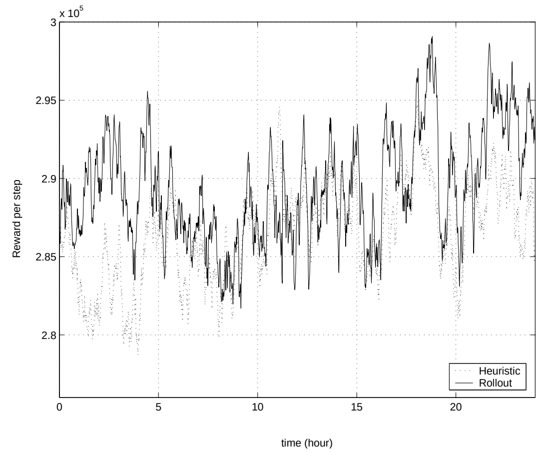

Figure 2: Reward per step for rollout and heuristic policies

a sequence of branch exchanges to transition from one logical topology to another.

The no-action policy is a policy with no actions (static topology). We select the logical topology to be the one with direct optical connection between S-D pairs with highest initial traffic (same as open-loop).

IV.A System parameters

In our simulations we use a 10 node network and each node has 3 pairs of router interfaces that can be used to connect to other nodes. So each node has a direct bi-directional lightpath to 3 other nodes and is connected to other nodes by multihop LSPs. Hence, there are 45 S-D pairs and 15 lightpaths.

The value of α is selected to be 100 (sec^{-1}). This determines how the delay affects the reward in Equation (2). The value of α also affects the threshold we set for call acceptance in Equation (4). The capacity of each lightpath is selected to be 36,000 packets/second, and each call has a rate of 20 packets/second. This means that the maximum capacity of a lightpath is 1,800 calls. Duration of each call has an exponential distribution with mean duration of 3 minutes. We run our simulations over a period of 24 hours. The threshold parameters γ_l and γ_h are selected to be 5% and 78%, respectively. The results are based on the average over 10 sample paths.

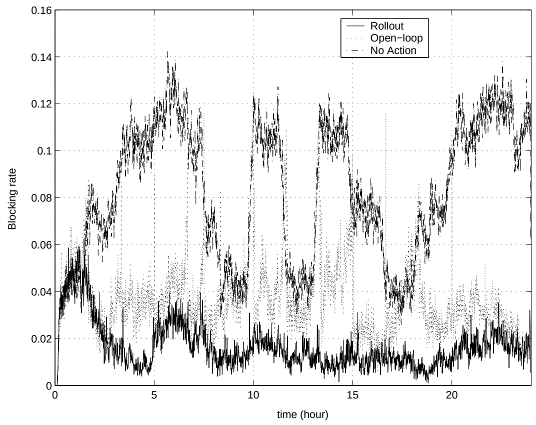

Figure 3: Blocking rate for rollout, open-loop and no-action policies

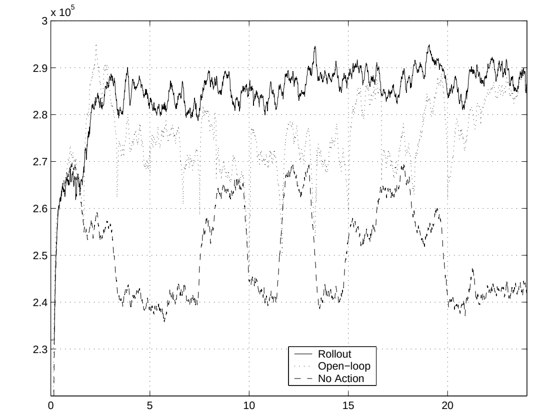

Figure 4: Reward per step for rollout, open-loop and no-action policies

IV.B Rollout vs. heuristic

We compare the performance of the rollout policy with that of the heuristic policy. Figures 1 and 2 show this comparison for the blocking rate and the reward per step respectively. We can see that rollout policy improves the heuristic policy by as much as 5% at times. This is because while the greedy heuristic policy takes myopic actions, the rollout policy can take the future reward over next several slow time steps into account to find a *sequence* of BEs to reach a better logical topology. Sometimes the heuristic policy becomes stuck at a local optimum, whereas the rollout policy may be able to avoid being stuck at such a local optimum.

IV.C Rollout vs. open-loop and no-action

Figures 3 and 4 compare the performance of the rollout, open-loop, and no-action policies. We can see that at times the open-loop and no-action policies have similar reward per step and blocking rates. Those are the periods during which the deterministic traffic pattern is similar to the initial traffic. Also note that at the start and towards the end of the day the reward per step and the blocking rate for the rollout policy become similar to those of the open-loop policy. This is because the random part of the traffic, a Brownian Bridge process, starts with a zero variance and goes back to zero variance at the end. When the variance of the random part becomes small the rollout policy has no clear advantage over the policy that uses only the deterministic part of the traffic.

V. Conclusion

We have demonstrated that the rollout policy improves the heuristic policy by looking farther into the future. It performs better than the open loop policy due to the use of measurements. The comparison of the rollout algorithm with the no-action algorithm shows how badly a static logical topology would perform given the type of traffic used in our experiment.

With this method we can exploit the time-of-day changes in the traffic as part of the deterministic traffic pattern and at the same time take advantage of the MDP model to predict the traffic in the future and to take into account future rewards when making current decisions.

References

[1] Hyeong Soo Chang, et al,"Multitime scale Markov decision processes," IEEE Transactions on Automatic Control, Vol 48, June 2003, 976-987.

[2] D. Bertsekas, "Dynamic Programming and Optimal Control," pp. 314-344, 2nd edition, Athena Scientific, 2000.

[3] A. Narula-Tam, and E. Modiano, "Dynamic Load Balancing in WDM Packet Networks With and Without Wavelength Constraints," IEEE Journal on slelcted areas in Communications, Vol 18, No 10, pp1972-1979, Oct. 2000.

[4] John Wei, et al, "Network Control and management for the next generation Internet," IEICE Transactions on Communications, Vol E83-B, No. 10, pp. 2191-2209, Oct 2000.

[5] Kevin H. Liu, et al, "Performance and testbed study of topology reconfiguration in IP over WDM networks," IEEE Transactions on Communications, Vol 50, pp. 1662-1679, Oct 2002.

[6] Ilia Baldine, George N. Rouskas, "On the Design of Dynamic Reconfiguration Policies for Broadcast WDM Networks," Proceedings of SPIE '98 Conference on All-Optical Networking: Architecture, Control, and Management Issues, vol 3531, pp. 146-157.

[7] F. Ricciato, S. Salsano, A. Belmonte, M. Listanti, "Offline Configuration of a MPLS over WDM Network under Time-Varying Offered Traffic," INFOCOM 2001.

[8] J. Labourdette and A. Acampora, "Logically rearrangeable multihop lightwave networks," IEEE Transactions on Communications, vol. 39, Aug. 1991.

[9] J. Labourdette, G. Hart, A. Acampora, "Branch-Exchange Sequences for Reconfiguration ofLightwave Networks," IEEE Transaction on Communications, Vol. 42 No. 10, pp. 2822-2832, Oct 1994.

[10] D. Bertsekas , R. Gallager, "Data Networks," Prentice Hall, 1992.

[11] R. Ramaswami and K. Sivarajan, "Design of Logical Topologies for Wavelength-Routed Optical Networks," IEEE Journal on Selected Areas in Communications, Vol 14, No. 5, pp. 840-851, Jun 1996.

[12] A. Gencata, B. Mukherjee, "Virtual-Topology Adaptation for WDM Mesh Networks Under Dynamic Traffic," Proc. of IEEE INFOCOM, Jun. 2002.

[13] S. Ross, "Stochastic Processes," 2nd edition, John Wiley & Sons Inc., 1996