ELSEVIER
Computer Networks 39 (2002) 165–183
COMPUTER NETWORKS
www.elsevier.com/locate/comnet

Optimal policy for label switched path setup in MPLS networks ☆

T. Anjali [a,*], C. Scoglio [a], J.C. de Oliveira [a], I.F. Akyildiz [a], G. Uhl [b]

[a] *Broadband and Wireless Networking Lab, School of Electrical and Computer Engineering, Georgia Institute of Technology, 250, 14th Street, Atlanta, GA 30332, USA*

[b] *Swales Aerospace & NASA Goddard Space Flight Center, Beltsville, MD 20705, USA*

Received 19 November 2001; accepted 22 November 2001

Responsible Editor: I.F. Akyildiz

Abstract

An important aspect in designing a multiprotocol label switching (MPLS) network is to determine an initial topology and to adapt it to the traffic load. A topology change in an MPLS network occurs when a new label switched path (LSP) is created between two nodes. The LSP creation involves determining the route of the LSP and the according resource allocation to the path. A fully connected MPLS network can be used to minimize the signaling. The objective of this paper is to determine when an LSP should be created and how often it should be re-dimensioned. An optimal policy to determine and adapt the MPLS network topology based on the traffic load is presented. The problem is formulated as a continuous time Markov decision process with the objective to minimize the costs involving bandwidth, switching, and signaling. These costs represent the trade-off between utilization of network resources and signaling/processing load incurred on the network. The policy performs a filtering control to avoid oscillations which may occur due to highly variable traffic. The new policy has been evaluated by simulation and numerical results show its effectiveness and the according performance improvement. A sub-optimal policy is also presented which is less computationally intensive and complicated. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: LSP establishment; LSP re-dimensioning; MPLS; MPLS network topology

1. Introduction

In recent years there has been active research in the field of multiprotocol label switching (MPLS) and an increasing number of networks are supporting MPLS. MPLS is a switching technology to forward packets based on a short, fixed length identifier called *label*. Using indexing instead of long address matching, MPLS achieves fast forwarding. Labels are used as indices of a table that contains the connection path. An MPLS network

☆ This work was supported by NASA Goddard. The work of J.C. de Oliveira was also supported in part by CAPES (The Brazilian Ministry of Education Agency).

* Corresponding author.

E-mail addresses: tricha@ece.gatech.edu (T. Anjali), caterina@ece.gatech.edu (C. Scoglio), jau@ece.gatech.edu (J.C. de Oliveira), ian@ece.gatech.edu (I.F. Akyildiz), uhl@rattler-e.gsfc.nasa.gov (G. Uhl).

1389-1286/02/$ - see front matter
PII: S1389-1286(01)00308-5

consists of label switched paths (LSPs) and edge/core label switch routers (LSRs). The LSRs store the label translation tables. Core LSRs provide transit services in the middle of the network while edge LSRs provide an interface with external networks. Packets with identical label are forwarded on the same LSP. LSPs are virtual unidirectional paths established from the sender to the receiver [1]. An extension of the Resource reSerVation Protocol (RSVP) is used to establish and maintain LSPs in the backbone [2]. One of the most significant applications of MPLS is traffic engineering (TE) [3], since LSPs can be considered as virtual traffic trunks that carry flow aggregates generated by packet classification.

Packet classification is also performed in the differentiated services (DiffServ) model which is based on classifying and aggregating individual micro-flows, at the edge of the network, into one of several behavioral aggregates (BAs) [4]. A per-hop behavior (PHB) defines the service a packet should receive in the network. Currently, packets are treated at each router based on two standard PHBs [5,6]:

- *Expedited forwarding* (*EF*): minimizes delay and jitter. It provides the highest level of aggregate QoS. Any traffic exceeding the traffic profile is discarded.
- *Assured forwarding* (*AF*): has four classes and three drop precedences within each class. Traffic compliant with the pre-negotiated traffic profile is delivered with a higher probability than the non-compliant traffic.

There are some open research problems related to the DiffServ model:

- How can the DiffServ model be extended to heterogeneous networks?
- How can the network resource utilization be improved by performing traffic engineering at DiffServ class level?

For both cases, a solution can be provided by using MPLS, after defining a mapping between DiffServ classes and LSPs. To the best of our knowledge, this mapping solution is still an open research problem. Towards this end, we will define class types and then map them to virtual MPLS networks. Each virtual MPLS network will have its own topology which will be independent of other virtual networks. This will provide better resource utilization by performing traffic engineering at DiffServ level. Also the LSPs can be mapped over a pure-MPLS (non-DiffServ) network extending DiffServ mapping to heterogeneous networks.

Following the IETF suggestions, we define class-types as the set of traffic trunks with same bandwidth constraints. Three class-types stand out and each can be carried on a virtual MPLS network by itself, e.g.,

- MPLS net1 as Class type 0, i.e., best effort (BE),
- MPLS net2 as Class type 1, i.e., *EF* (for real time traffic),
- MPLS net3 as Class type 2, i.e., *AF* 1 and 2 (for low loss classes).

These virtual networks are layered on top of the physical network as illustrated in Fig. 1. The capacity of each physical link is partitioned among different MPLS networks, and a maximum capacity (fixed percentage of the total link capacity) is assigned to each partition. The unused reserved bandwidth can then be used for BE traffic. The design and management of the above MPLS networks are a fundamental key to the success of the DiffServ–MPLS mapping. However, many problems such as *the definition of the network topology*, *LSP dimensioning*, *LSP setup/tear-down procedures*, *LSP routing*, and *LSP adaptation for incoming resource requests*, need to be solved. The off-line network design methods, which use a priori knowledge of traffic demand, are not suitable for MPLS networks [7] due to the high unpredictabi-

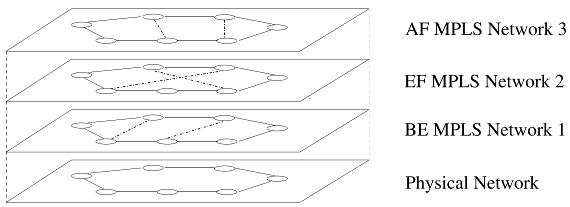


Fig. 1. Virtual MPLS networks.

lity of the Internet traffic. A fully connected MPLS network, where every pair of LSRs is connected by a direct LSP, is very inefficient [8] due to the high signaling cost and the management of a large number of LSPs. The signaling cost is of the order of N^2, where N is the total number of routers.

Two different approaches, *traffic-driven* and *topology-driven*, can be used for MPLS network design. In the *traffic-driven* approach, the LSP is established on demand according to a request for a flow, traffic trunk or bandwidth reservation. The LSP is released when the request becomes inactive. In the *topology-driven* approach, the LSP is established in advance according to the routing protocol information, e.g., when a routing entry is generated by the routing protocol. The LSP is maintained as long as the corresponding routing entry exists, and it is released when the routing entry is deleted. The advantage of the traffic-driven approach is that only the required LSPs are setup, while in the topology-driven approach, the LSPs are established in advance even if no data flow occurs.

A simple LSP setup policy based on the traffic-driven approach has been proposed in [8], in which an LSP is established whenever the number of bytes forwarded within one minute exceeds a threshold. This policy reduces the number of LSPs in the network; however, it has very high signaling costs and needs high control efforts for variable and bursty traffic as in the case of a fully connected network.

In an earlier paper [9], we have suggested a threshold-based policy for LSP setup. It provides an on-line design for MPLS network depending on the current traffic load. The proposed policy is a traffic-driven approach and balances the signaling and switching costs. By increasing the number of LSPs in a network, the signaling costs increase while the switching costs decrease. In the policy, LSPs are setup or torn down depending on the actual traffic demand. Furthermore, since a given traffic load may change depending on time, the policy also performs filtering in order to avoid oscillations which may occur in case of variable traffic.

In this paper, we introduce a new LSP setup/redimensioning policy and prove that the optimal policy is a threshold policy, using the Markov decision process (MDP) [10] theory. In Section 2, the LSP setup problem is formulated and solved, and the policy structure is described. The optimal policy is derived in Section 3 and the sub-optimal policy *least one-step cost* is given in Section 4. The implementation issues are described in Section 5 along with the numerical results and comparison of the optimal policy with the sub-optimal policy. Conclusions are given in Section 6.

2. The setup problem of label switched paths

When a bandwidth request arrives between two nodes in a network that are not connected by a direct LSP, the decision about whether to establish such an LSP arises. In this section, we will first describe the model formulation and then obtain a decision policy which governs the decisions at each instant.

2.1. Model formulation

We now describe the system under consideration. Let $G_{\mathrm{ph}}(N,L)$ denote a physical IP network with a set of N routers and a set of physical links L. We define the following notation for $G_{\mathrm{ph}}(N,L)$:

- $l(i,j) \in L$: physical link between routers i and j.
- $C_{\mathrm{ph}}(i,j)$ for $i, j \in N$: total link capacity of $l(i,j)$.
- $h(i,j)$ for i, $j \in N$: number of hops between nodes i and j.

We introduce a virtual "induced" MPLS network $G(N,\mathscr{L})$, as in [3], for the physical network $G_{\mathrm{ph}}(N,L)$. This virtual MPLS network $G(N,\mathscr{L})$ consists of the same set of routers N as the physical network $G_{\mathrm{ph}}(N,L)$ and a set of LSPs, denoted by $\mathscr{L}$. We assume that each link $l(i,j)$ of the physical network corresponds to a default LSP in $\mathscr{L}$ which is non-removable. The other elements of $\mathscr{L}$ are the LSPs (virtual links) built between non-adjacent nodes of $G_{\mathrm{ph}}(N,L)$ and routed over $l(i,j)$'s. Note that G is a directed graph and $\mathscr{L} \supseteq L$. In other words, the different MPLS networks (for different class types) are built by adding virtual LSPs to the physical topology when needed. In this paper, we will use the terms graph, network and topology interchangeably for the physical and MPLS networks G and G_{ph}, respectively.

We define the following notation for $G(N, \mathscr{L})$:

- $\mathrm{LSP}(i,j) \in \mathscr{L}$: LSP between routers i and j (when they are not physically connected).
- $\mathrm{LSP}_0(i,j) \in \mathscr{L}$: default LSP between routers i and j (when they are physically connected).
- $C(i,j)$ for $i,\ j \in N$: total capacity of $\mathrm{LSP}(i,j)$ ($C(i,j) = 0 \iff \mathrm{LSP}(i,j)$ not established).
- $A(i,j)$ for $i,\ j \in N$: available capacity on $\mathrm{LSP}(i,j)$ ($A(i,j) = 0 \iff \mathrm{LSP}(i,j)$ fully occupied).
- $B(i,j)$ for $i,\ j \in N$: total bandwidth reserved between routers i and j. It represents the total traffic between router i as the source and router j as the destination.

We assume that all $\mathrm{LSP}_0(i,j)$ for $i,\ j \in N$ have large capacity and it is available to be borrowed by the other multi-hop LSPs that will be routed over the corresponding physical links $l(i,j)$. We introduce a simple algorithm for routing LSPs on $G_{\mathrm{ph}}(N,L)$ and bandwidth requests on $G(N,\mathscr{L})$. Each LSP must be routed on a shortest path in $G_{\mathrm{ph}}(N,L)$. We assume that the shortest path $P_{\mathrm{ph}}(i,j)$ between a source node i and destination node j is the minimum hop path in $G_{\mathrm{ph}}(N,L)$ and is denoted by

$$P_{\mathrm{ph}}(i,j) = \{l(i,u), \ldots, l(v,j)\}.$$

In the MPLS network, the bandwidth requests between i and j are routed either on the direct $\mathrm{LSP}(i,j)$ or on $P(i,j)$, which is a multiple-LSP path overlaying $P_{\mathrm{ph}}(i,j)$:

$$P(i,j) = \{\mathrm{LSP}_0(i,u), \ldots, \mathrm{LSP}_0(v,j)\}.$$

We also assume that $C_{\mathrm{ph}}(i,j)$ is sufficiently large for all $l(i,j)$ and whenever any LSP is re-dimensioned, it can borrow bandwidth from the physical links that it passes through.

The default and non-default LSPs can be explained with the help of Fig. 2. The dotted lines between nodes 1–4, 4–6, and 6–8 represent the default LSPs and the thick line between nodes 1–8 represents the direct LSP which is routed over the default LSPs.

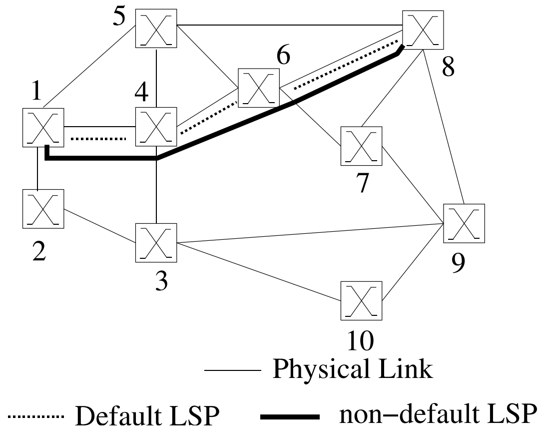


Fig. 2. MPLS network topology.

With the assumed routing algorithm, we can define the following two quantities:

- $B_L(i,j)$ for $i,\ j \in N$: part of $B(i,j)$ that is routed over $\mathrm{LSP}(i,j)$.
- $B_P(i,j)$ for $i,\ j \in N$: part of $B(i,j)$ that is routed over $P(i,j)$.

Note that $B(i,j) = B_L(i,j) + B_P(i,j)$ is the total of the bandwidth requests between i and j, $C(i,j) = A(i,j) + B_L(i,j)$ is the total capacity of $\mathrm{LSP}(i,j)$ and $B_P(i,j) = 0$ for default LSPs since $P(i,j)$ coincides with the $\mathrm{LSP}_0(i,j)$.

Let $S(i,j)$ be the set of all $\mathrm{LSP}(u,v)$ such that the corresponding shortest path $P_{\mathrm{ph}}(u,v)$ contains the link $l(i,j)$. The following condition must be satisfied:

$$\sum_{\mathrm{LSP}(u,v) \in S(i,j)} C(u,v) \leqslant \delta C_{\mathrm{ph}}(i,j), \tag{1}$$

where $\delta < 1$ is a maximum fraction of $C_{\mathrm{ph}}(i,j)$ that can be assigned to LSPs. Condition (1) means that the sum of capacity of all LSPs using a particular physical link on their path must not exceed a portion δ of the capacity of that physical link.

Definition 1 (*Decision instants and bandwidth requests*). We denote by t_m the arrival instant of a new bandwidth request between routers i and j for the amount $b(i,j)$. The instant t_m is called a *decision instant* because a decision has to be made to accommodate the arrival of the new bandwidth request.

We now describe the events that imply a decision. When a new bandwidth request $b(i,j)$ arrives in the MPLS network at instant t_m, the existence of a direct LSP between i and j is checked initially. For direct LSP between i and j, the available capacity $A(i,j)$ is then compared with the request $b(i,j)$. If $A(i,j) > b(i,j)$, then the requested bandwidth is allocated on that LSP and the available capacity is reduced accordingly. Otherwise, $C(i,j)$ can be increased subject to condition (1) in order to satisfy the bandwidth request.

On the other hand, if there exists no direct LSP between i and j, then we need to decide whether to setup a new LSP and determine its according $C(i,j)$. Each time a new LSP is setup or re-dimensioned, the previously granted bandwidth requests between i and j routed on $P(i,j)$ are re-routed on the direct LSP(i,j). However, this re-routing operation is only virtual, since, by our routing assumptions, both LSP(i,j) and $P(i,j)$ are routed on the physical network over the same $P_{\text{ph}}(i,j)$.

Let t_n be the departure instant of a request for bandwidth allocation $b(i,j)$ routed on LSP(i,j). In this instant we need to decide whether or not to re-dimension LSP(i,j), i.e., reduce its capacity $C(i,j)$.

We assume that the events and costs associated with any given node pair i and j are independent of any other node pair. This assumption is based on the fact that the new bandwidth requests are routed either on the direct LSP between the source and destination or on $P(i,j)$, i.e., the other LSPs are not utilized for routing the new request. This assumption allows us to carry the analysis for any node pair and be guaranteed that it will be true for all other pairs. Under this assumption, we can drop the explicit (i,j) dependence of the notations. Also we assume that nodes i and j are not physically connected. For the default LSPs, there is a large amount of available bandwidth and they too borrow bandwidth, in large amounts, from the physical links, if needed.

Definition 2 (*Set of events*). For each router pair i and j in the MPLS network, e_m is the *event* observed at t_m.

- $e_m = 1$ if there is an arrival of a bandwidth request for amount b,
- $e_m = 0$ if there is a departure of a request of amount b from $B_P(i,j)$,
- $e_m = 2$ if there is a departure of a request b from $B_L(i,j)$.

Definition 3 (*Set of states*). For each router pair i and j in the MPLS network, we observe the system state when any event occurs. The state vector s_m at a given time instant t_m, $m = 0, 1, \ldots$ is defined as

$$s_m(i,j) = [A, B_L, B_P], \tag{2}$$

where A is the available capacity on LSP(i,j), B_L is the part of B that is routed over LSP(i,j) and B_P is the part of B routed on $P(i,j)$. Note that the state space $\bar{s}$, the set of all system states, is finite since A is limited by C which is in turn limited by the minimum of the link bandwidths on P_{ph}. B_L is limited by C and B_P by minimum of default LSP bandwidths on P_{ph}. Also note that states with non-zero $A(i,j)$ and $B_P(i,j)$ are possible because just before the instant of observation, some user request might have departed leaving available bandwidth in LSP(i,j). The state information for each LSP is stored in the first router of the LSP.

Definition 4 (*Set of extended states*). The state space $\bar{s}$ of the system can be extended by the coupling of the current state and the event.

$$S_m = \langle s_m, e_m \rangle. \tag{3}$$

The set $\overline{S}$ of extended states S_m is the basis for determining the decisions to be taken to handle the events.

Definition 5 (*Set of actions*). The decision of setting up or re-dimensioning LSP(i,j) when the event e_m occurs is captured by the binary action variable $a \in A = \{0,1\}$.

- $a = 1$ means that LSP(i,j) will be setup or re-dimensioned and the new value of its capacity is set as $C = B_L + B_P + b$, where b is considered negative if the event is a departure, either over LSP(i,j) or $P(i,j)$.
- $a = 0$ means that no action will be taken on the capacity of LSP(i,j).

Definition 6 (*Decision rules and policies*). A decision rule d_i provides an action selection in each state at a given decision instant t_i and a policy π specifies the decision rules to be used in each decision instant, i.e., $\pi = \{d_0(\overline{S}), d_1(\overline{S}), d_2(\overline{S}), \ldots\}$. If $d_i(\overline{S}) = d_j(\overline{S})\ \forall\ i$ and j, then the policy is stationary as the decision is independent of the time instant. For most of the possible system states, the decision rule can choose an action from the set $\{0,1\}$ but there are a few states where only one action is possible. Those states and corresponding actions are:

- $S_m = \langle [A, B_L, B_P], 1\rangle$ where $A > b \Rightarrow a = 0$ (the new request is routed on LSP(i,j)),
- $S_m = \langle [A, B_L, B_P], 0\rangle \Rightarrow a = 0$ (the request ending over $P(i,j)$),
- $S_m = \langle [A, B_L, 0], 2\rangle$ where $B_L = b \Rightarrow a = 1$ (LSP-(i,j) is torn down).

Definition 7 (*Cost function*). The incremental cost $W(S,a)$ for the system in state s, occurrence of the event e, and the taken action a is

$$W(S,a) = W_{\mathrm{sign}}(S,a) + W_{\mathrm{b}}(S,a) + W_{\mathrm{sw}}(S,a), \tag{4}$$

where $W_{\mathrm{sign}}(S,a)$ is the cost for signaling the setup or re-dimensioning of the LSP to the involved routers, $W_{\mathrm{b}}(S,a)$ is the cost for the carried bandwidth and $W_{\mathrm{sw}}(S,a)$ is the cost for switching of the traffic. The cost components depend on the system state and the action taken for an event.

The signaling cost $W_{\mathrm{sign}}(S,a)$ is incurred instantaneously only when action $a = 1$ is chosen for state S. It accounts for the signaling involved in the process of setup or re-dimensioning of the LSP. We consider that this cost depends linearly on the number of hops h in $P_{\mathrm{ph}}(i,j)$ over which the LSP is routed, plus a constant component to take into account the notification of the new capacity of the LSP to the network.

$$W_{\mathrm{sign}}(S,a) = a[c_s h + c_a], \tag{5}$$

where c_s is the coefficient for signaling cost per hop and c_a is the fixed notification cost coefficient. This cost is not incurred if $a = 0$.

The other two components of Eq. (4) relate to the bandwidth (w_{b}) and switching (w_{sw}) cost rates, respectively.

$$W_{\mathrm{b}}(S,a) = \int_0^T w_{\mathrm{b}}(S,a)\,\mathrm{d}t,$$
$$W_{\mathrm{sw}}(S,a) = \int_0^T w_{\mathrm{sw}}(S,a)\,\mathrm{d}t,$$

where T is the time till the next event, i.e., until the system stays in state S.

We assume that the bandwidth cost rate $w_{\mathrm{b}}(S,a)$ to reserve $(B_L + B_P)$ capacity units depends linearly on $(B_L + B_P)$ and on the number of hops $h(i,j)$ in the physical shortest path over which the request is routed.

$$w_{\mathrm{b}}(S,a) = c_{\mathrm{b}} h (B_L + B_P), \tag{6}$$

where c_{b} is the bandwidth cost coefficient per capacity unit (c.u.) per time. Note that, from our routing assumption, the physical path is the same for LSP(i,j) and for $P(i,j)$ and thus the bandwidth cost rate depends only on the total carried bandwidth, irrespective of the fractions carried over different paths.

The switching cost rate $w_{\mathrm{sw}}(S,a)$ depends linearly on the number of switching operations in IP or MPLS mode and the switched bandwidth. The total number of switching operations is always h since the physical path is fixed. Whether these switching operations are IP or MPLS depends on the path chosen in the MPLS network. For B_L c.u. routed on LSP(i,j), we have 1 router performing IP switching and $(h-1)$ routers performing MPLS switching. For B_P c.u. routed on $P(i,j)$, we have h routers perform IP switching.

$$w_{\mathrm{sw}}(S,a) = [c_{\mathrm{ip}} + c_{\mathrm{mpls}}(h-1)]B_L + h c_{\mathrm{ip}} B_P, \tag{7}$$

where c_{ip} and c_{mpls} are the switching cost coefficients per c.u. per time in IP and MPLS mode, respectively. Summarizing, the signaling cost is incurred only at decision instants when $a = 1$, while the bandwidth and switching costs are accumulated continuously until a new event occurs.

Example 1. Here we illustrate how the state vector defined in Eq. (2) varies due to bandwidth request arrival and LSP setup. Consider a simple three node tandem network where node i is connected to node $i+1$, $i = 1, 2$. Suppose, at the initial instant t_0, the state vectors for the three nodes are given as follows (capacity is expressed in c.u.):

$$s_0(1,2) = (1000, 14, 0); \quad s_0(2,3) = (1000, 15, 0);$$
$$s_0(1,3) = (0, 0, 5).$$

Suppose that two alternative events occur at instant t_1:

EVENT A: A bandwidth request for 2 c.u. arrives between nodes 1 and 2 when system is in state s_0. Then $B_L(1,2)$ increases to 16 and $A(1,2)$ reduces by 2. So the new state vectors become:

$$s_A(1,2) = (998, 16, 0); \quad s_A(2,3) = (1000, 15, 0);$$
$$s_A(1,3) = (0, 0, 5).$$

EVENT B: A bandwidth request for 10 c.u. arrives between nodes 1 and 3 when system is in state s_0. We elaborate the two cases when $a = 1$ or $a = 0$.

Case 1 ($a = 1$): A direct LSP between nodes 1 and 3 is created and the new state vectors are

$$s_{B_1}(1,2) = (985, 14, 0); \quad s_{B_1}(2,3) = (985, 15, 0);$$
$$s_{B_1}(1,3) = (0, 15, 0).$$

The incremental cost from initial state $s_0(1,3)$ is calculated from Eq. (4) as:

$$\begin{aligned} W_1(S,a) &= W_b(S,a) + W_{sw}(S,a) + W_{sign}(S,a) \\ &= \{c_b \cdot 2 \cdot 15 + (c_{ip} + c_{mpls}) \cdot 15\}T \\ &\quad + 2c_s + c_a. \end{aligned}$$

Case 2 ($a = 0$): The request is routed on the 2-LSP path $P(1,3)$ and the new state vectors are

$$s_{B_0}(1,2) = (985, 29, 0); \quad s_{B_0}(2,3) = (985, 30, 0);$$
$$s_{B_0}(1,3) = (0, 0, 15).$$

The incremental cost, in this case, from initial state $s_0(1,3)$ is calculated from Eq. (4) as:

$$\begin{aligned} W_2(S,a) &= W_b(S,a) + W_{sw}(S,a) + W_{sign}(S,a) \\ &= \{c_b \cdot 2 \cdot 15 + (c_{ip} \cdot 2) \cdot 15\}T + 0. \end{aligned}$$

In the equations for $W_1(S,a)$ and $W_2(S,a)$, T is the average time between this event and the next event.

The set of all possible system states (Definition 4), events (Definition 2), actions (Definition 5) and associated costs (Definition 7) is given in Table 1. In the table, the node pair (i,j) is implicit and T is the time interval between the current event and the

Table 1
Set of possible states

Old state	Action	New state	Cost
$\langle [A, B_L, B_P], 0 \rangle$	0	$\langle [A, B_L, B_P - b], e \rangle$ where $e \in \{0,1,2\}$	$T[hc_b\{B_L + B_P - b\}] + T[\{c_{ip} + (h-1)c_{mpls}\}B_L + hc_{ip}(B_P - b)]$
$\langle [A, B_L, B_P], 1 \rangle$ where $A \geqslant b$	0	$\langle [A - b, B_L + b, B_P], e \rangle$ where $e \in \{0,1,2\}$	$T[hc_b\{B_L + B_P + b\}] + T[\{c_{ip} + (h-1)c_{mpls}\} \times (B_L + b) + hc_{ip}B_P]$
$\langle [A, B_L, B_P], 1 \rangle$ where $A < b$	0	$\langle [A, B_L, B_P + b], e \rangle$ where $e \in \{0,1,2\}$	$T[hc_b\{B_L + B_P + b\}] + T[\{c_{ip} + (h-1)c_{mpls}\}B_L + hc_{ip}(B_P + b)]$
$\langle [A, B_L, B_P], 1 \rangle$ where $A < b$ and $Y = B_L + B_P + b$	1	$\langle [0, Y, 0], e \rangle$ where $e \in \{1,2\}$	$T[hc_bY] + T[\{c_{ip} + (h-1)c_{mpls}\}Y] + (c_sh + c_a)$
$\langle [A, B_L, B_P], 2 \rangle$	0	$\langle [A + b, B_L - b, B_P], e \rangle$ where $e \in \{0,1,2\}$	$T[hc_b\{B_L + B_P - b\}] + T[\{c_{ip} + (h-1)c_{mpls}\}(B_L - b) + hc_{ip}B_P]$
$\langle [A, B_L, B_P], 2 \rangle$ where $Y = B_L + B_P - b$	1	$\langle [0, Y, 0], e \rangle$ where $e \in \{1,2\}$	$T[hc_bY] + T[\{c_{ip} + (h-1)c_{mpls}\}Y] + (c_sh + c_a)$

next event. In all the cost formulations, the first component refers to the cost incurred for the bandwidth carried, second component refers to the cost for switching of the traffic over LSP(i,j) or $P(i,j)$, and third, if it exists, to the signaling cost.

3. Optimal LSP setup policy based on MDP

3.1. Optimization problem formulation

We propose a stochastic model to determine the optimal decision policy for LSP setup. The optimization problem is formulated as a continuous-time Markov decision process (CTMDP) [10]. The cost functions for the MDP theory have been defined in Definition 7. Following the theory of MDPs, we define the expected infinite-horizon discounted total cost, $v^\pi(S_0)$, with discounting rate α, given that the process occupies state S_0 at the first decision instant and the decision policy is π by:

$$v_\alpha^\pi(S_0) = E_{S_0}^\pi\left\{\sum_{m=0}^{\infty} \mathrm{e}^{-\alpha t_m}\left[W_{\text{sign}}(S_m,a) + \int_{t_m}^{t_{m+1}} \mathrm{e}^{-\alpha(t-t_m)}[w_\text{b}(S_m,a) + w_\text{sw}(S_m,a)]\,\mathrm{d}t\right]\right\}, \quad (8)$$

where t_0, $t_1,\ldots$ represent the times of successive instants when events occur and $W_{\text{sign}}(S_m,a)$ represents the fixed part of the cost incurred whereas $[w_\text{b}(S_m,a) + w_\text{sw}(S_m,a)]$ represents the continuous part of the cost between times t_m and t_{m+1}.

The *optimization objective* is to find a policy π^* such that:

$$v_\alpha^{\pi^*}(s) = \inf_{\pi\in\Pi} v_\alpha^\pi(s).$$

The optimal decision policy can be found by solving the related optimality equations [10] for each initial state S. We assume that the bandwidth requests arrive according to a Poisson process with rate λ and the request durations are exponentially distributed with rate μ. With our assumptions of a discounted infinite-horizon CTMDP, the optimality equations can be written as:

$$v(S) = \min_{a\in A}\left\{r(S,a) + \frac{\lambda+\mu}{\lambda+\mu+\alpha}\sum_{j\in\bar{S}} q(j\,|\,S,a)v(j)\right\}, \quad (9)$$

where $r(S,a)$ is the expected discounted cost between two decision instants and $q(j\,|\,S,a)$ is the probability that the system occupies state j at the subsequent decision instant, given that the system is in state S at the earlier decision instant and action a is chosen. From Eq. (4), $r(S,a)$ can be written as

$$r(S,a) = W_{\text{sign}}(S,a) + E_S^a\left\{\int_0^{\tau_1} \mathrm{e}^{-\alpha t}[w_\text{b}(S,a) + w_\text{sw}(S,a)]\,\mathrm{d}t\right\}, \quad (10)$$

where E_S^a represents the expectation with respect to the request duration distribution and τ_1 represents the time before the next event occurs.

With the Markovian assumption on request arrival and duration, the time between any two successive events (arrival of requests or departure of a request) is exponentially distributed with rate $(\lambda+\mu)$. Recalling that between two successive events the state of the system does not change, Eq. (10) can be rewritten as follows:

$$r(S,a) = W_{\text{sign}}(S,a) + [w_\text{b}(S,a) + w_\text{sw}(S,a)]E_S^a\left\{\int_0^{\tau_1} \mathrm{e}^{-\alpha t}\,\mathrm{d}t\right\} \quad (11)$$

$$= W_{\text{sign}}(S,a) + [w_\text{b}(S,a) + w_\text{sw}(S,a)]/(\alpha+\lambda+\mu). \quad (12)$$

Since the set of possible actions A is finite and $r(S,a)$ is bounded, it can be proved that the optimal policy π^* is stationary and deterministic [10].

3.2. Transition probability function

The transition probabilities $q(j\,|\,s,a)$ in Eq. (9) for our model are related to the transition probabilities in an M/M/1 queue and are given by

Eq. (13). In the equation, $P_{\mathrm{Ar}}(A)$ denotes the probability that the amount of bandwidth requested by the arrival is less than or equal to A and P_{D} is the probability that a connection that is departing was routed over $P(i,j)$. We know that the probability of the next event being a departure is given by $(\mu/(\lambda+\mu))$.

$$q(j\,|\,S,a) = \begin{cases} (\lambda/(\lambda+\mu)), & S=\langle A,B_L,B_P,0\rangle,\quad a=0,\quad j=\langle A,B_L,B_P-b,1\rangle,\\ P_{\mathrm{D}}, & S=\langle A,B_L,B_P,0\rangle,\quad a=0,\quad j=\langle A,B_L,B_P-b,0\rangle,\\ (\mu/(\lambda+\mu))-P_{\mathrm{D}}, & S=\langle A,B_L,B_P,0\rangle,\quad a=0,\quad j=\langle A,B_L,B_P-b,2\rangle,\\ \\ (\lambda/(\lambda+\mu)), & S=\langle A,B_L,B_P,1\rangle,\quad a=0,\quad j=\langle A-b,B_L+b,B_P,1\rangle\ \ (A\geqslant b),\\ P_{\mathrm{D}}, & S=\langle A,B_L,B_P,1\rangle,\quad a=0,\quad j=\langle A-b,B_L+b,B_P,0\rangle\ \ (A\geqslant b),\\ (\mu/(\lambda+\mu))-P_{\mathrm{D}}, & S=\langle A,B_L,B_P,1\rangle,\quad a=0,\quad j=\langle A-b,B_L+b,B_P,2\rangle\ \ (A\geqslant b),\\ \\ (\lambda/(\lambda+\mu)), & S=\langle A,B_L,B_P,1\rangle,\quad a=0,\quad j=\langle A,B_L,B_P+b,1\rangle\ \ (A<b),\\ P_{\mathrm{D}}, & S=\langle A,B_L,B_P,1\rangle,\quad a=0,\quad j=\langle A,B_L,B_P+b,0\rangle\ \ (A<b),\\ (\mu/(\lambda+\mu))-P_{\mathrm{D}}, & S=\langle A,B_L,B_P,1\rangle,\quad a=0,\quad j=\langle A,B_L,B_P+b,2\rangle\ \ (A<b),\\ \\ (\lambda/(\lambda+\mu)), & S=\langle A,B_L,B_P,1\rangle,\quad a=1,\quad j=\langle 0,B_L+B_P+b,0,1\rangle\ \ (A<b),\\ (\mu/(\lambda+\mu)), & S=\langle A,B_L,B_P,1\rangle,\quad a=1,\quad j=\langle 0,B_L+B_P+b,0,1\rangle\ \ (A<b),\\ \\ (\lambda/(\lambda+\mu)), & S=\langle A,B_L,B_P,2\rangle,\quad a=0,\quad j=\langle A+b,B_L-b,B_P,1\rangle,\\ P_{\mathrm{D}}, & S=\langle A,B_L,B_P,2\rangle,\quad a=0,\quad j=\langle A+b,B_L-b,B_P,0\rangle,\\ (\mu/(\lambda+\mu))-P_{\mathrm{D}}, & S=\langle A,B_L,B_P,2\rangle,\quad a=0,\quad j=\langle A+b,B_L-b,B_P,2\rangle,\\ \\ (\lambda/(\lambda+\mu)), & S=\langle A,B_L,B_P,2\rangle,\quad a=1,\quad j=\langle 0,B_L+B_P-b,0,1\rangle,\\ (\mu/(\lambda+\mu)), & S=\langle A,B_L,B_P,2\rangle,\quad a=1,\quad j=\langle 0,B_L+B_P-b,0,2\rangle,\\ \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

3.3. Optimality equations

The optimality Eq. (9) can be explicitly written for all possible states by obtaining $r(S,a)$ from Eq. (12) and $q(j\,|\,S,a)$ from Eq. (13) as follows:

$$v(\langle A,B_L,B_P,0\rangle) = \frac{hc_{\mathrm{b}}(B_L+B_P-b)+hc_{\mathrm{ip}}(B_P-b)+DB_L}{\alpha+\lambda+\mu} + \frac{\lambda+\mu}{\alpha+\lambda+\mu}J, \tag{14}$$

$$v(\langle A,B_L,B_P,1\rangle) = \frac{hc_{\mathrm{b}}(B_L+B_P+b)+hc_{\mathrm{ip}}B_P+D(B_L+b)}{\alpha+\lambda+\mu} + \frac{\lambda+\mu}{\alpha+\lambda+\mu}K \quad \text{for } A\geqslant b, \tag{15}$$

$$v(\langle A,B_L,B_P,1\rangle) = \min\left\{\frac{hc_{\mathrm{b}}(B_L+B_P+b)+hc_{\mathrm{ip}}(B_P+b)+DB_L}{\alpha+\lambda+\mu} + \frac{\lambda+\mu}{\alpha+\lambda+\mu}L,\ c_s h+c_a + \frac{hc_{\mathrm{b}}(B_L+B_P+b)+D\{B_L+B_P+b\}}{\alpha+\lambda+\mu} + \frac{\lambda+\mu}{\alpha+\lambda+\mu}M\right\} \quad \text{for } A<b, \tag{16}$$

$$v(\langle A, B_L, B_P, 2\rangle) = \min\left\{ \frac{h c_{\mathrm{b}}(B_L + B_P - b) + h c_{\mathrm{ip}} B_P + D(B_L - b)}{\alpha + \lambda + \mu} + \frac{\lambda + \mu}{\alpha + \lambda + \mu} X, c_s h + c_a + \frac{h c_{\mathrm{b}}(B_L + B_P - b) + D(B_L + B_P - b)}{\alpha + \lambda + \mu} + \frac{\lambda + \mu}{\alpha + \lambda + \mu} Y \right\}, \tag{17}$$

where

$$D = \{c_{\mathrm{ip}} + (h-1)c_{\mathrm{mpls}}\},$$

$$J = \left[\frac{\lambda}{\lambda + \mu} v(\langle A, B_L, B_P - b, 1\rangle) + P_{\mathrm{D}} v(\langle A, B_L, B_P - b, 0\rangle) + \left\{ \frac{\mu}{\lambda + \mu} - P_{\mathrm{D}} \right\} v(\langle A, B_L, B_P - b, 2\rangle) \right],$$

$$K = \left[\frac{\lambda}{\lambda + \mu} v(\langle A - b, B_L + b, B_P, 1\rangle) + P_{\mathrm{D}} v(\langle A - b, B_L + b, B_P, 0\rangle) + \left\{ \frac{\mu}{\lambda + \mu} - P_{\mathrm{D}} \right\} v(\langle A - b, B_L + b, B_P, 2\rangle) \right],$$

$$L = \left[\frac{\lambda}{\lambda + \mu} v(\langle A, B_L, B_P + b, 1\rangle) + P_{\mathrm{D}} v(\langle A, B_L, B_P + b, 0\rangle) + \left\{ \frac{\mu}{\lambda + \mu} - P_{\mathrm{D}} \right\} \times (\langle A, B_L, B_P. + b, 2\rangle) \right],$$

$$M = \left[\frac{\lambda}{\lambda + \mu} v(\langle 0, B_L + B_P + b, 0, 1\rangle) + \frac{\mu}{\lambda + \mu} v(\langle 0, B_L + B_P + b, 0, 2\rangle) \right],$$

$$X = \left[\frac{\lambda}{\lambda + \mu} v(\langle A + b, B_L - b, B_P, 1\rangle) + P_{\mathrm{D}} v(\langle A + b, B_L - b, B_P, 0\rangle) + \left\{ \frac{\mu}{\lambda + \mu} - P_{\mathrm{D}} \right\} v(\langle A + b, B_L - b, B_P, 2\rangle) \right],$$

$$Y = \left[\frac{\lambda}{\lambda + \mu} v(\langle 0, B_L + B_P - b, 0, 1\rangle) + \frac{\mu}{\lambda + \mu} v(\langle 0, B_L + B_P - b, 0, 2\rangle) \right].$$

By substituting Eq. (14) into Eqs. (15)–(17), we obtain the simplified optimality equations as given below.

Optimality equations:

$$v(\langle A, B_L, B_P, 0\rangle) = \frac{hc_{\mathrm{b}}(B_L + B_P - b) + hc_{\mathrm{ip}}(B_P - b) + DB_L}{\alpha + \lambda + \mu} + \frac{\lambda + \mu}{\alpha + \lambda + \mu} J, \tag{18}$$

$$v(\langle A, B_L, B_P, 1\rangle) = \frac{h c_{\mathrm{b}}(B_L + B_P + b) + h c_{\mathrm{ip}} B_P + D(B_L + b)}{\alpha + \lambda + \mu} + \frac{\lambda + \mu}{\alpha + \lambda + \mu} K \quad \text{for } A \geqslant b, \tag{19}$$

$$v(\langle A, B_L, B_P, 1\rangle) = \min\{v(\langle A, B_L, B_P + 2b, 0\rangle), c_s h + c_a + v(\langle 0, B_L + B_P + b, b, 0\rangle)\} \quad \text{for } A < b, \tag{20}$$

$$v(\langle A, B_L, B_P, 2\rangle) = \min\{v(\langle A + b, B_L - b, B_P + b, 0\rangle), c_s h + c_a + v(\langle 0, B_L + B_P - b, b, 0\rangle)\}. \tag{21}$$

3.4. The optimal policy

The solutions of the four optimality equations (18)–(21) give the optimal values $v^*(A, B_L, B_P, e)$ of expected infinite-horizon discounted total costs. From the optimality equation (20), we derive that for the state $S = \langle A, B_L, B_P, 1\rangle$ where $A < b$, the optimal action would be

$$a^*\langle A, B_L, B_P, 1\rangle = \begin{cases} 1 & c_s h + c_a < v^*(\langle A, B_L, B_P + 2b, 0\rangle) - v^*(\langle 0, B_L + B_P + b, b, 0\rangle), \\ 0 & \text{otherwise}, \end{cases} \tag{22}$$

Algorithm:
1. Set $v^0(S) = 0$ for each state $S \in (\bar{S})$. Specify $\varepsilon > 0$ and set $n = 0$.
2. For each $S \in (\bar{S})$, compute $v^{n+1}(S)$ by substituting $v^n(S)$ on the right hand side of equations (18), (19), (20) and (21).
3. If $\| v^{n+1} - v^n \| < \varepsilon$, go to step 4. Otherwise, increment n by 1 and go to step 2.
4. For each $S \in (\bar{S})$, set $v^*(S) = v^n(S)$ and calculate the actions by substituting $v^*(S)$ into equations (22) and (23). Stop.

Fig. 3. The value iteration algorithm.

and for state $S = \langle A, B_L, B_P, 2\rangle$, the optimal action would be, from optimality equation (21),

$$a^*\langle A, B_L, B_P, 2\rangle = \begin{cases} 1 & c_s h + c_a \\ & \quad < v^*(\langle A+b, B_L-b, B_P+b, 0\rangle) \\ & \quad -v^*(\langle 0, B_L+B_P-b, b, 0\rangle), \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

This policy will be optimal if the quantities thresholds $v^*(\langle A, B_L, B_P+2b, 0\rangle) - v^*(\langle 0, B_L + B_P + b, b, 0\rangle)$ and $v^*(\langle A+b, B_L-b, B_P+b, 0\rangle) - v^*(\langle 0, B_L+B_P-b, b, 0\rangle)$ are monotone non-increasing which is true and can be proved through induction [10] by utilizing the linearity characteristics of the cost functions. These decisions have a control-limit structure. The values of $v^*(A, B_L, B_P, e)$ can be found by using either value iteration or policy iteration algorithm which are numerical procedures. We first give the value iteration algorithm and then the optimal LSP setup policy.

The value iteration algorithm: There are a number of iteration algorithms [10] available to solve the optimality equations. The value iteration is the most widely used and best understood algorithm for solving discounted Markov decision problems. The algorithm is as shown in Fig. 3.

The optimal LSP setup policy: The optimal policy $\pi^* = \{d^*, d^*, d^*, \ldots\}$ is stationary implying same decision rule at each decision instant and the decision rule is given by

$$d^* = \begin{cases} 0 & S = \langle A, B_L, B_P, 0\rangle, \\ 0 & S = \langle A, B_L, B_P, 1\rangle \\ & \quad \text{for } A \geqslant b, \\ a^*\langle A, B_L, B_P, 1\rangle & S = \langle A, B_L, B_P, 1\rangle \\ & \quad \text{for } A < b, \\ a^*\langle A, B_L, B_P, 2\rangle & S = \langle A, B_L, B_P, 2\rangle, \end{cases} \tag{24}$$

where $a^*\langle A, B_L, B_P, 1\rangle$ and $a^*\langle A, B_L, B_P, 2\rangle$ are given by Eqs. (22) and (23), respectively.

The threshold structure of the optimal policy facilitates the solution of the optimality equations (18)–(21) but still it is difficult to pre-calculate and store the solution because of the large number of possible system states. So, we propose a sub-optimal policy, called the *least one-step cost policy*, that is fast and easy to calculate.

4. The sub-optimal decision policy for LSP setup

The proposed least one-step cost policy is an approximation to the solution of the optimality equations (18)–(21). It minimizes the cost incurred between two decision instants. Instead of going through all the iterations of the value iteration algorithm (given in Fig. 3), if we perform the first iteration with the assumption that $v^0(\langle A, B_L, B_P - b, 0\rangle) = 0$, we obtain

$$v^1(\langle A, B_L, B_P, 0\rangle) = \frac{hc_{\mathrm{b}}(B_L + B_P - b) + h c_{\mathrm{ip}}(B_P - b) + \{c_{\mathrm{ip}} + (h-1)c_{\mathrm{mpls}}\}B_L}{\alpha + \lambda + \mu},$$

$$v^1(\langle A, B_L, B_P, 1\rangle) = \frac{hc_{\mathrm{b}}(B_L + B_P + b) + h c_{\mathrm{ip}} B_P + \{c_{\mathrm{ip}} + (h-1)c_{\mathrm{mpls}}\}(B_L + b)}{\alpha + \lambda + \mu} \quad \text{for } A \geqslant b,$$

$$v^1(\langle A,B_L,B_P,1\rangle) = \min\left\{\frac{hc_b(B_L+B_P+b)+hc_{ip}(B_P+b)+\{c_{ip}+(h-1)c_{mpls}\}B_L}{\alpha+\lambda+\mu}, c_s h + c_a + \frac{hc_b(B_L+B_P+b)+\{c_{ip}+(h-1)c_{mpls}\}\{B_L+B_P+b\}}{\alpha+\lambda+\mu}\right\} \quad \text{for } A<b,$$

$$v^1(\langle A,B_L,B_P,2\rangle) = \min\left\{\frac{hc_b(B_L+B_P-b)+hc_{ip}B_P+\{c_{ip}+(h-1)c_{mpls}\}\{B_L-b\}}{\alpha+\lambda+\mu}, c_s h + c_a + \frac{hc_b(B_L+B_P-b)+\{c_{ip}+(h-1)c_{mpls}\}\{B_L+B_P-b\}}{\alpha+\lambda+\mu}\right\}.$$

From these single-step cost formulations, we can derive the action decision. For the state $\langle A,B_L,B_P,1\rangle$, we obtain

$$a^1(\langle A,B_L,B_P,1\rangle) = \begin{cases} 1 & B_P+b>B_{Th}, \\ 0 & \text{otherwise,} \end{cases} \quad \text{for } A<b \tag{25}$$

upon comparison of the two terms of $v^1(\langle A,B_L, B_P,1\rangle)$. Similarly, comparing the two terms of $v^1(\langle A,B_L,B_P,2\rangle)$, we get the action decision

$$a^1(\langle A,B_L,B_P,2\rangle) = \begin{cases} 1 & B_P>B_{Th}, \\ 0 & \text{otherwise.} \end{cases} \tag{26}$$

In both Eqs. (25) and (26),

$$B_{Th} = \frac{\{c_s h + c_a\}\{\alpha+\lambda+\mu\}}{(h-1)(c_{ip}-c_{mpls})}. \tag{27}$$

By calculating $v^1(S)$ for all $S\in\overline{S}$, we minimize the one-step cost of the infinite-horizon model. Since $v^n(S)$ in the value iteration algorithm converges to $v^*(S)$, the one-step value $v^1(S)$ is a significant part of $v^*(S)$ and is very easy to calculate.

Least one-step cost LSP setup policy: The one-step optimal policy $\pi^\# = \{d^\#, d^\#, d^\#, \ldots\}$ is stationary implying same decision rule at each decision instant and the decision rule is given by

$$d^\# = \begin{cases} 0 & S=\langle A,B_L,B_P,0\rangle, \\ 0 & S=\langle A,B_L,B_P,1\rangle \\ & \quad \text{for } A\geqslant b, \\ a^1\langle A,B_L,B_P,1\rangle & S=\langle A,B_L,B_P,1\rangle \\ & \quad \text{for } A<b, \\ a^1\langle A,B_L,B_P,2\rangle & S=\langle A,B_L,B_P,2\rangle, \end{cases} \tag{28}$$

where $a^1\langle A,B_L,B_P,1\rangle$ and $a^1\langle A,B_L,B_P,2\rangle$ are given by Eqs. (25) and (26), respectively.

The algorithm given in Fig. 4 can be implemented for our threshold-based sub-optimal least one-step cost policy for LSP setup/re-dimensioning.

5. Numerical results and discussions

5.1. Implementation aspects

Having identified the different parameters involved in the LSP setup policy, we can explain the steps for implementing the policy. For each LSP, during its connection setup phase, the network controller assigns the cost functions based on the signaling load of the network. In our model, the cost functions are assumed to be linear (W_{sign}, W_b, W_{sw} from Eq. (4)) with respect to the bandwidth requirements of the requests. By keeping a history of user requests, the average inter-arrival time and connection duration can be estimated.

Given the input parameters (cost functions and various distributions), the value iteration algorithm (Fig. 3) can be used to determine the optimal policy with the decision rule (24). The optimal policy is then stored in a tabular format. Each entry of the table specifies the optimal decision for the possible events for all possible node pairs of the network. Whenever there is a bandwidth request arrival or departure, the network performs a table lookup at the corresponding node pair entry. Setup/re-dimensioning of the LSP is performed if the traffic not utilizing the LSP exceeds a threshold (Eqs. (22)

At time t_m, $s_m = [A(i,j), B_L(i,j), B_P(i,j)]$ and event e occurs

Case 0: $e =$ **Departure of $b(i,j)$ from $P(i,j)$**
Do not re-dimension $LSP(i,j)$. $s_{m+1} = [A(i,j), B_L(i,j), B_P(i,j) - b(i,j)]$

Case 1: $e =$ **Arrival of $b(i,j)$**
If $LSP(i,j)$ exists and $A(i,j) > b(i,j)$
Do not re-dimension $LSP(i,j)$. $s_{m+1} = [A(i,j) - b(i,j), B_L(i,j) + b(i,j), B_P(i,j)]$
Request accepted and routed on $LSP(i,j)$
Else
If $B_P(i,j) + b(i,j) > B_{Th}(i,j)$
Set-up/re-dimension $LSP(i,j)$. $s_{m+1} = [0, B_L(i,j) + B_P(i,j) + b(i,j), 0]$
Request accepted and routed on $LSP(i,j)$
Else
Do not re-dimension $LSP(i,j)$. $s_{m+1} = [A(i,j), B_L(i,j), B_P(i,j) + b(i,j), 0]$
Request accepted and routed on $P(i,j)$

Case 2: $e =$ **Departure of $b(i,j)$ from $LSP(i,j)$**
If $B_P(i,j) > B_{Th}(i,j)$
Re-dimension $LSP(i,j)$. $s_{m+1} = [0, B_L(i,j) + B_P(i,j) - b(i,j), 0]$
Else
If $B_P(i,j) = 0$ and $B_L(i,j) = b(i,j)$
Tear-down $LSP(i,j)$. $s_{m+1} = [0, 0, 0]$
Else
Do not re-dimension $LSP(i,j)$. $s_{m+1} = [A(i,j) + b(i,j), B_L(i,j) - b(i,j), B_P(i,j)]$

Fig. 4. Setup/re-dimensioning policy.

and (23)). The optimal policy table needs to be updated when there are changes in the network topology. The update can, however, be performed off-line.

For networks of considerable size, the storage of the optimal policy for each node pair can be very resource consuming. In such cases the sub-optimal policy, given in Section 4, can be applied. This policy computes the decision upon arrival of each request and does not involve storage of the whole policy.

5.2. Simulation model

In this section, we will present the performances of both the optimal policy (decision rule in Eq. (24)) and the sub-optimal policy (decision rule in Eq. (28)) and then compare them. The performance metric is the *discounted total cost* defined in Eq. (8). Both the optimal and the proposed sub-optimal policy can also be compared with the trivial heuristics where no LSP optimization is performed, or optimization is performed for each event arrival, or optimization is performed on a periodic basis.

For the simulations, we modeled an MPLS network as a non-hierarchical graph G_{ph} shown in Fig. 5. It is a 10-node random graph with a maximum node degree of 3 and 17 edges. Each node represents an LSR and each edge represents a physical link connecting two LSRs. Each link is assumed to have a physical capacity of 1000 c.u.

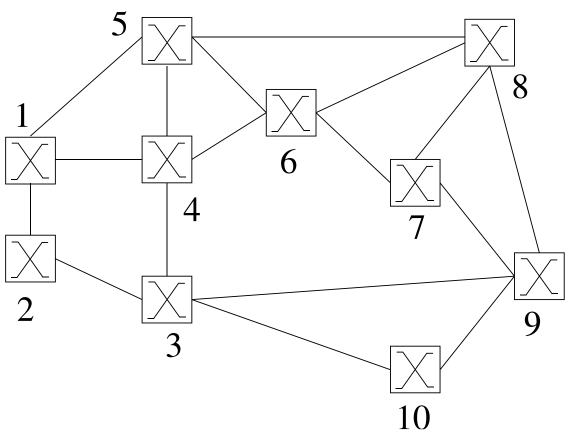


Fig. 5. Physical topology G_{ph}.

Table 2
Cost coefficients

c_s	c_a	c_{b}	c_{ip}	c_{mpls}
15	15	1	2.5	0.5

Based on this network model, we obtain the adjacency matrix of the network as well as the number of links of the shortest path between any two nodes. We assume that the number of links of the shortest path estimated by the source is deterministic. The request duration is assumed to be exponential whereas the request arrival follows a Poisson process.

The values given in Table 2 are assumed for the cost coefficients in Eqs. (5)–(7) which define the cost incurred by the network. With these cost coefficients, the threshold $B_{\mathrm{Th}}(i,j)$, defined in Eq. (27), for the sub-optimal policy (decision rule in Eq. (28)) becomes

$$B_{\mathrm{Th}}(i,j) = \frac{15(h+1)(\alpha+\lambda+\mu)}{2(h-1)} = \frac{7.5(\alpha+\lambda+\mu)(h+1)}{(h-1)}.$$

For different cases, we will vary the values of λ and μ and obtain the $B_{\mathrm{Th}}(i,j)$ independently. In all our simulations, we assume that all user bandwidth requests are for the amount of 1 c.u. Even though both the optimal and sub-optimal policies are independent of the amount of the bandwidth requested, we concentrate on this homogeneous case because the results obtained are representative of the effects the bandwidth requests have on the MPLS network topology. When the bandwidth requests are for 1 c.u., we can get a snapshot of the events and really understand how the events are triggered.

For each source and destination pair, the value iteration algorithm, in Fig. 3, is used to determine the minimum discounted total cost (defined in Eq. (8)) and the optimal policy. For the value iteration algorithm, ε is set to 0.1% of the first-step discounted total cost. The minimum discounted total cost is then averaged over all possible source and destination pairs. For the proposed sub-optimal policy also, the minimum discounted total cost is calculated using the value iteration algorithm. As given in Eqs. (5)–(7), the cost functions are linear with respect to the bandwidth request.

5.3. Results

In the following simulations, we show the performance of the two policies. We show how high traffic volume leads to LSP setup/re-dimensioning whereas for less volume, the LSPs are not modified. We show how the MPLS network topology is modified according to varying bandwidth requests. We show some cases where the results of the two policies are different and then compare their performance.

For case I, we simulate the arrival of requests in G_{ph} with the λ/μ values from Table 3 and apply the optimal policy π^*, for which the decision rule is given in Eq. (24). The resulting MPLS network G^*_{I} is given in Fig. 6(c).

Note that since the node pairs 1–9 and 2–8 have a traffic load greater than the others, representing a focused overload scenario, the corresponding LSPs have been established. Instead, if the proposed sub-optimal policy $\pi^{\#}$ (decision rule in Eq. (28)) is applied, the resulting network $G^{\#}_{\mathrm{I}}$ coincides with G^*_{I}, demonstrating the efficiency of the sub-optimal policy. In Figs. 6(a) and (b) we show, for comparison, $G_{\min}$ and $G_{\max}$ that would result if the two simple heuristic decision policies $\pi_{\min}$ and $\pi_{\max}$ were applied, respectively. $\pi_{\min}$ is the policy to never establish non-default LSPs whereas $\pi_{\max}$ is the policy to adapt the LSP to each occurring event. We found that the discounted total cost

Table 3
λ/μ for case I

Node pair	1–7	1–8	1–10	2–7	2–9	2–10	1–9	2–8	Others
λ/μ	5	5	5	5	5	5	30	30	0

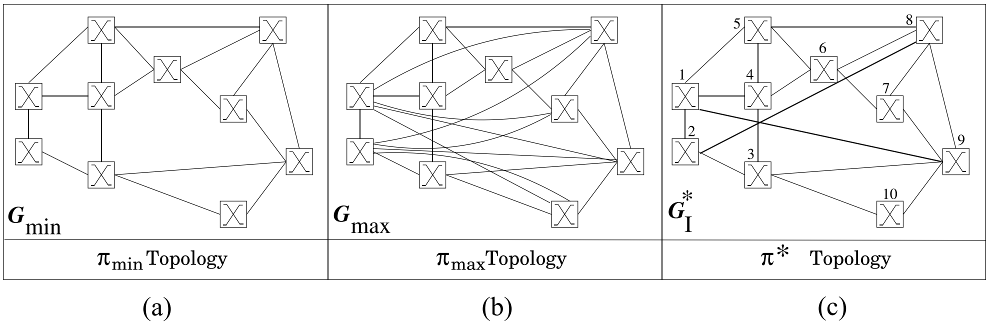


Fig. 6. Topologies and costs for (a) $\pi_{\min}$, (b) $\pi_{\max}$ and (c) π^*.

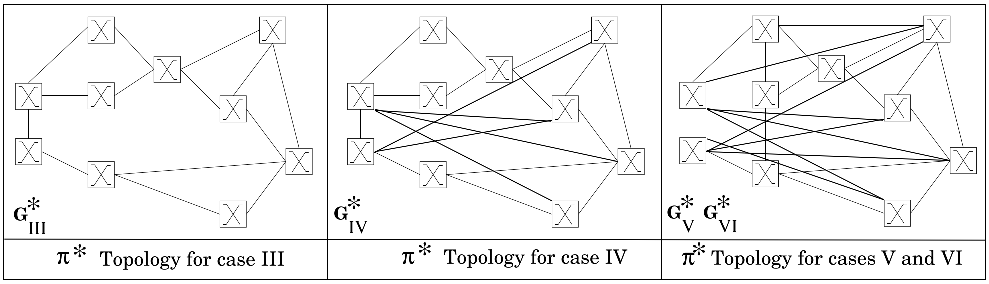
(defined in Eq. (8)) for G_{I}^* is 45% lower than $G_{\min}$ and 77% lower than $G_{\max}$.

In case II, we aim to verify the on-line adaptability of the optimal policy π^* (decision rule in Eq. (24)) when a traffic variation occurs. The node pairs with non-zero traffic requests are kept the same as in case I. All of them but pair 1–10 have $\lambda/\mu = 5$, for which $\lambda/\mu = 30$. If the optimal policy is applied starting from the initial state represented by G_{I}^*, the result of case I, the final topology consists of an added LSP(1,10) to G_{I}^*. The old non-default LSPs are not torn down because they are utilized by the traffic as they provide reduced switching cost (Eq. (7)) without the overhead of the signaling cost (Eq. (5)). We see that the topology has changed from G_{I}^* to better fit the new traffic pattern. On the other hand, if we start from G_{ph}, the obtained network topology will just add the LSP(1,10) to G_{ph}. So, the resulting topologies in the two cases differ and highlight the capability of the optimal policy to adjust to the traffic variation. The same results are obtained upon application of the sub-optimal policy $\pi^{\#}$ (decision rule in Eq. (28)).

Next we still consider the traffic for the same node pairs as before. This is because they represent pairs with two or more hops in between them. Starting with the initial topology G_{ph}, the traffic matrix was homogeneously increased as shown in Table 4 for cases III–VI. The corresponding π^* topologies are shown in Fig. 7. As expected, for larger bandwidth requests, more LSPs are setup because the expected bandwidth and switching costs (Eqs. (6) and (7)) exceed the signaling cost (Eq. (5)) overhead and it becomes economically viable to setup the LSPs. More LSP setup leads to a more connected MPLS network: the network G_{V}^* is more connected than the network G_{IV}^*, which is in turn more connected than the network G_{III}^*. If we apply the sub-optimal policy $\pi^{\#}$ (decision rule in Eq. (28)) to G_{ph} with the traffic from Table 4, slightly different results are obtained. For case III, $G_{\mathrm{III}}^{\#}$ is same as G_{III}^* because the traffic is very less and it is not economically efficient to setup any LSPs. For case IV, the sub-optimal policy does not find it viable to setup any LSPs and hence $G_{\mathrm{IV}}^{\#}$ does not add any new LSPs, i.e., it is the same as G_{III}^*. For case V, the traffic is a little higher and thus, the threshold B_{Th} in Eq. (27) is exceeded for length 3 LSPs but not for length 2 LSPs. Thus, $G_{\mathrm{V}}^{\#}$ is the same as G_{IV}^*. Finally, $G_{\mathrm{VI}}^{\#}$ is the same as G_{VI}^* as the threshold B_{Th} (Eq. (27)) is exceeded even for length

Table 4
Bandwidth requests for cases III–VI

Node pair	1–7	1–8	1–9	1–10	2–7	2–8	2–9	2–10
λ_3/μ_3	5	5	5	5	5	5	5	5
λ_4/μ_4	10	10	10	10	10	10	10	10
λ_5/μ_5	30	30	30	30	30	30	30	30
λ_6/μ_6	50	50	50	50	50	50	50	50



(a) (b) (c)

Fig. 7. Topologies for cases III–VI.

2 LSPs. It can be seen from Eq. (27), the threshold is smaller for longer LSPs as B_{Th} is inversely proportional to $(h-1)$.

As a verification for the results in Fig. 7, we calculate the costs for the topologies G^*_{III}, G^*_{IV} and G^*_{V} for the three cases III, IV and V in Table 4. In Fig. 8, we show the plots for the cost components (switching and signaling costs, Eqs. (7) and (5)) and the discounted total cost for the three cases for topologies G^*_{III}, G^*_{IV} and G^*_{V}. In each figure, the respective minimum discounted total cost corresponds to the topologies shown in Fig. 7. For instance, in Fig. 8(a), the minimum discounted total cost is given for topology G^*_{III}; in Fig. 8(b), the minimum discounted total cost corresponds to topology G^*_{IV}, and in Fig. 8(c), the minimum discounted total cost corresponds to topology G^*_{V}.

Having seen a case where the final topologies obtained by application of policies π^* (decision rule in Eq. (24)) and $\pi^{\#}$ (decision rule in Eq. (28)) are different, we now compare the performance of the two policies. For the initial topology G_{ph}, we plot in Fig. 9 the total discounted cost for different initial states for one node pair with three hops in between. The final state of the system is shown for each initial state and the two policies as the numbers in brackets close to the curves. As the discount rate α (from Eq. (8)) is smaller for Fig. 9(b), the costs are larger in magnitude. We see that the expected costs are identical or marginally close, except for one point in each figure. For the initial state $[1, 5, 10]$, the optimal policy optimizes the LSP immediately whereas the sub-optimal policy does not since the threshold is not exceeded, resulting in the lower expected cost for the optimal policy. On the other hand, for the initial state $[1, 1, 1]$ in Fig. 9(a), only the optimal policy performed the optimization but the costs are equal for both cases. This is because of the discount factor α as events too far in the future have marginal effect on the cost. One point to be observed from the figures is that the final states from the optimal policy have large available bandwidth values. This is because the optimal policy performs LSP optimization very often whereas the sub-optimal policy performs optimization only when the traffic exceeds a threshold which is large. This, in effect, reduces the sensitivity of the decision policy to minor variations in the traffic, i.e., by filtering small fluctuations.

In Fig. 10, a stepwise increased homogeneous traffic is offered and we show the percentage setup of LSPs using the sub-optimal policy $\pi^{\#}$ (decision rule in Eq. (28)). For λ/μ values less than 10, no LSP is setup as no threshold is exceeded. A stable configuration of the network is achieved for λ/μ values between $[20, 30]$ where all LSPs with length 3 are setup and those with length 2 are not setup. For λ/μ greater than 45, all the LSPs are always setup and the network reaches its fully connected stable state. For the other values of λ/μ, the LSPs are setup with percentages as shown in Fig. 10, e.g. for λ/μ of 15, the LSPs with length 3 are setup with 80% probability.

5.4. Discussion

In our simulations, we found the value iteration algorithm to be very efficient and stable. The

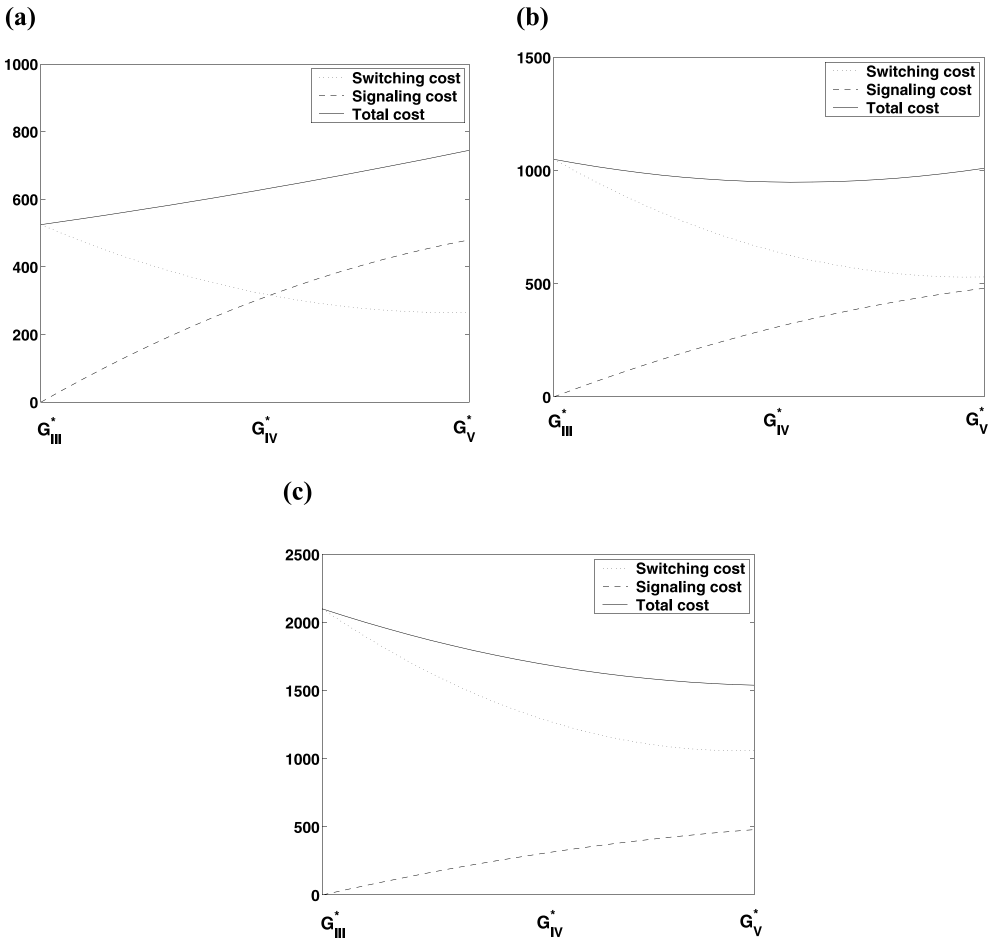


Fig. 8. Discounted total cost and cost components for cases (a) III, (b) IV and (c) V.

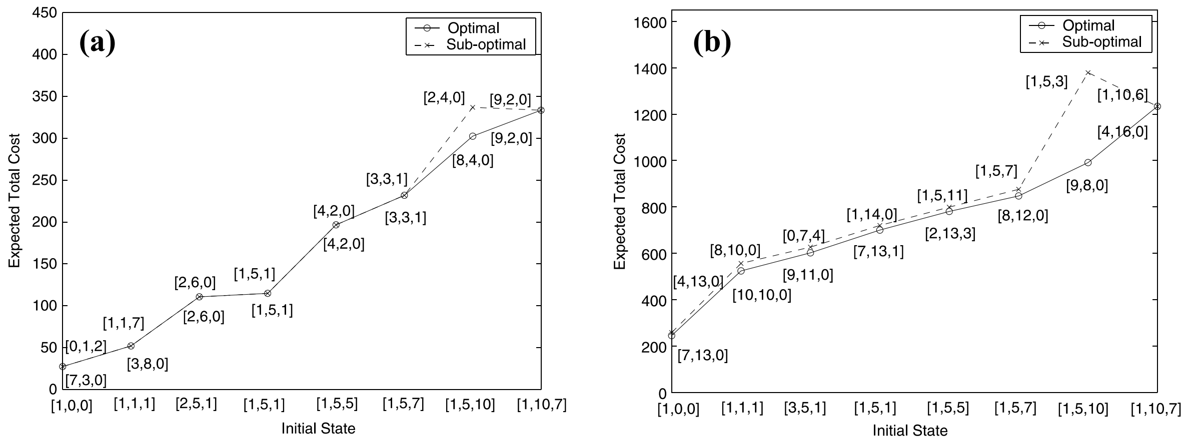


Fig. 9. Total expected cost vs. initial state: (a) $\alpha = 0.5$, $\lambda/\mu = 5$; (b) $\alpha = 0.1$, $\lambda/\mu = 10$.

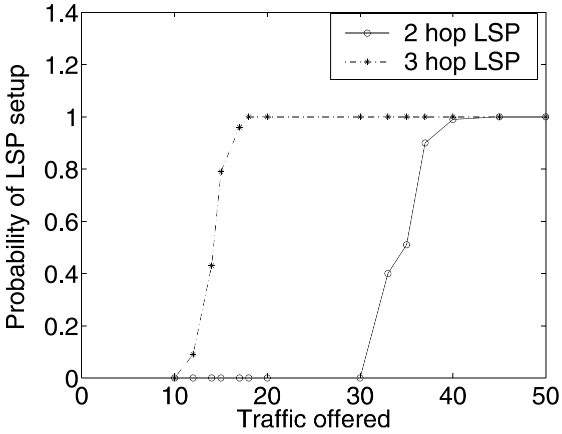


Fig. 10. Percentage setup of LSPs for homogeneous traffic.

convergence is fast resulting in a low number of iterations. In general, the number of iterations does not depend on the cost parameters (c_s, c_a, c_b, c_{ip}, c_{mpls}), but depends on the values of λ and μ. There are other iteration algorithms (e.g., policy iteration [10]) that have a higher rate of convergence but are more intensive computationwise (the policy iteration involves a search through the set of all possible decision policies). The proposed sub-optimal policy is much less computationally intensive (no storage of decision policy) and provides the expected discounted total cost values close to the optimal policy.

6. Conclusions

In this paper, we presented a new optimal decision policy that provides the on-line design of a network topology for the current traffic load and pattern. The proposed policy is used to solve the following issue: a new request for bandwidth reservation between two routers, that are not directly connected by an LSP, arises. In this case, the decision concerning whether or not to setup a new direct LSP, modifying the current MPLS network topology, should be taken. Adding a new direct LSP requires high signaling effort, but improves the switching of packets between the two routers. The LSP optimization problem is formulated as a continuous-time Markov decision process. We have presented the value iteration algorithm which determines the expected discounted total cost and the optimal policy. Under certain conditions, we have shown the existence of an optimal policy which has a threshold structure.

Because of the computational intensiveness of the optimal policy, we have proposed a sub-optimal least one-step cost policy that simplifies the threshold determination and thus the decision rule. This policy is based on the network load, which is part of the defined network state, via a threshold criterion. The threshold calculation takes into account the bandwidth, switching and signaling costs and depends on the cost coefficients. Furthermore, since a given traffic load may just be a temporary phenomenon, our policy also performs filtering in order to avoid oscillations that can be typical in a variable traffic scenario.

The performance of both the optimal and the sub-optimal policy was demonstrated by simulation. Several examples were considered. Significant cases were analyzed in the paper. The results confirm that the proposed policy is effective and improves network performance by reducing the cost incurred. Simulation results also indicate that the total expected cost is similar for both the policies proving the accuracy of the sub-optimal policy.

References

[1] E. Rosen, A. Viswanathan, R. Callon, Multiprotocol label switching architecture, IETF RFC 3031, January 2001, Available from <http://www.ietf.org/rfc/rfc3031>.

[2] D.O. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, RSVP-TE: extensions to RSVP for LSP tunnels, IETF RFC 3209, December 2001, Available from <http://www.ietf.org/rfc/rfc3209>.

[3] D.O. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J. McManus, Requirements for traffic engineering over MPLS, IETF RFC 2702, September 1999, Available from <http://www.ietf.org/rfc/rfc2702>.

[4] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An architecture for differentiated services, IETF RFC 2475, December 1998, Available from <http://www.ietf.org/rfc/rfc2475>.

[5] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, Assured forwarding PHB group, IETF RFC 2597, June 1999, Available from <http://www.ietf.org/rfc/rfc2597>.

[6] V. Jacobson, K. Nichols, K. Poduri, An expedited forwarding PHB, IETF RFC 2598, June 1999, Available from <http://www.ietf.org/rfc/rfc2598>.

[7] M. Kodialam, T.V. Lakshman, Minimum interference routing with applications to MPLS traffic engineering, in: Proceedings of IEEE INFOCOM 2000, Tel Aviv, Israel, March 2000.

[8] S. Uhlig, O. Bonaventure, On the cost of using MPLS for interdomain traffic, in: Quality of Future Internet Services, Berlin, Germany, September 2000, pp. 141–152.

[9] C. Scoglio, T. Anjali, J. de Oliveira, I. Akyildiz, G. Uhl, A new threshold-based policy for label switched path set up in MPLS networks, in: Proceedings of 17th International Teletraffic Congress 2001, Salvador, Brazil, September 2001.

[10] M.L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, Wiley, New York, 1994.

Tricha Anjali received the (Integrated) M.Tech. degree in Electrical Engineering from the Indian Institute of Technology, Bombay, in 1998. Currently, she is a Research Assistant in the Broadband and Wireless Networking Laboratory pursuing her Ph.D. degree. She is a student member of the IEEE Communications Society. Her interest is to investigate Quality of Service (QoS) issues in the Next Generation Internet (NGI).

Caterina Scoglio received the Dr. Ing. degree (summa cum laude) in Electronics Engineering from the University of Rome "La Sapienza", Italy, in May 1987. She received a post-graduate degree in "Mathematical Theory and Methods for System Analysis and Control" from the University of Rome "La Sapienza", Italy, in November 1988. From June 1987 to June 2000, she had been with Fondazione Ugo Bordoni, Rome, where she was a research scientist at the TLC Network Department—Network Planning Group. In the period November 1991–August 1992 she had been Visiting Researcher at, Georgia Institute of Technology, College of Computing, in Atlanta, GA, USA. Since September 2000, she has been with the Broadband and Wireless Networking Laboratory of the Georgia Institute of Technology as a Research Engineer. Her research interests include optimal design and management of multi-service networks.

Jaudelice Cavalcante de Oliveira received the B.A.Sc. degree in Electrical Engineering from Universidade Federal do Ceará (UFC), in Fortaleza, Ceará, Brazil, and the M.A.Sc. degree in Electrical Engineering from the Universidade Estadual de Campinas (Unicamp), Campinas, São Paulo, Brazil. She has a Fellowship from the Brazilian Ministry of Education Agency, CAPES. Currently, she is a Research Assistant in the Broadband and Wireless Networking Laboratory pursuing her Ph.D. degree. Her interest is to investigate Quality of Service (QoS) issues in the Next Generation Internet (NGI).

Ian F. Akyildiz received the B.S., M.S., and Ph.D. degrees in computer engineering from the University of Erlangen-Nuernberg, Germany, in 1978, 1981 and 1984, respectively.

He is a Distinguished Chair Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, and Director of the Broadband and Wireless Networking Laboratory. His current research interests include wireless networks, satellite networks, and the Internet.

Dr. Akyildiz is the Editor-in-Chief of Computer Networks (Elsevier) and an Editor for the ACM-Springer Journal for Multimedia Systems, the ACM-Kluwer Journal of Wireless Networks, and the Journal of Cluster Computing. He is a past Editor for IEEE Transactions on Computers (1992–1996) and for IEEE/ACM Transactions on Networking (1996–2001). He was the Program Chair of the Ninth IEEE Computer Communications Workshop, the ACM/IEEE MOBICOM'96, and the IEEE INFOCOM'98 conference. He will be the General Chair for the ACM/IEEE MOBICOM 2002 conference and the Technical Program Chair of the IEEE ICC 2003. He received the Don Federico Santa Maria Medal for his services to the Universidad of Federico Santa Maria in Chile. He served as a National Lecturer for the ACM from 1989 to 1998, and received the ACM Outstanding Distinguished Lecturer Award for 1994. He also received the 1997 IEEE Leonard G. Abraham Prize award for his paper entitled "Multimedia Group Synchronization Protocols for Integrated Services Architecture," published in the IEEE Journal on Selected Areas in Communications in January 1996. He is a Fellow of the ACM.

George Uhl is the Lead Engineer at NASA's Earth Science Data and Information System (ESDIS) Network Prototyping Lab. He directs network research and protoyping activities for ESDIS. Current areas of research include network quality of service and end-to-end performance improvement.