Adaptive multicast routing in multirate loss networks

Ren-Hung Hwang

Department of Computer Science and Information Engineering, National Chung Cheng University, Chia-Yi, Taiwan, ROC
E-mail: rhhwang@cs.ccu.edu.tw

Received August 1996; in final form March 1999

In this paper, we study two versions of the multicast routing problem in multirate loss networks: complete and partial. In the complete version of the multicast routing problem, the identities of all destination nodes are available to the multicast routing algorithm at once. Conversely, in the partial version of the multicast problem, the identities of the destination nodes are revealed to the routing algorithm one by one. Although the complete version of the multicast routing problem, also known as the Steiner tree problem, has been well studied in the literature, less attention has been paid for the definition of link costs and evaluating the performance of multicast routing algorithm from the network revenue point of view. Therefore, in this paper, we first propose two approaches, namely, the Markov Decision Process-based (MDP-based) and Least Loaded Routing-based (LLR-based) approaches, for defining link costs. Several heuristic multicast routing algorithms are then proposed for both fully connected networks and sparsely connected networks. We have also proposed a new performance metric, referred to as fractional reward loss, for evaluating the performance of multicast routing algorithms. Our simulation results indicate that algorithms based on partial destination information yield worse performance than those based on complete information. We also found that, for fully connected networks, algorithms that use LLR-based link costs yield very competitive performance as compared to those that use MDP approach. However, for sparsely connected networks, LLR-based algorithms yield significantly worse performance as compared to the MDP-based algorithms.

1. Introduction

In this paper, we investigate the multicast routing problem in Broadband Integrated Services Digital Networks (B-ISDN). B-ISDN are expected to handle a variety of traffic classes, each of which has its own traffic characteristics, such as bandwidth requirement, call arrival rate, call holding time, and reward parameter. More importantly, each connection established requires a certain type of Quality of Service (QOS) guarantee which can be specified by QOS parameters, such as cell loss rate, cell delay variation, and maximum/average cell transfer delay. In order to provide QOS guarantees, call admission control is required when setting up a new connection. A connection is accepted only if the network has sufficient resource to meet this new connection's QOS, while maintaining the agreed-upon QOS for all existing connections. If the connection is accepted and routed on a path, a certain amount of resources, such as bandwidth and buffer space, determined by the requested QOS and other factors, must be reserved, explicitly or implicitly, for the connection on all links along the path. Since determining the amount of bandwidth and buffer space that need to be reserved is very complicated and beyond the scope of this paper, in this paper, we assume that a fixed amount of bandwidth is reserved on each link along the routing path. Examples of fixed bandwidth allocation include peak-rate allocation, constant-bit-rate (CBR) traffic, and the use of "effective bandwidth" [13,14,16,26,27] to estimate the bandwidth requirement of variable-bit-rate traffic. Note that, in ATM networks, the effective bandwidth for a connection could be different at different links.

Since the early 1980's, considerable research has focused on the design of point-to-point *adaptive* routing algorithms for circuit-switched networks [2]. Most of these adaptive routing algorithms can be classified into two categories: *Least Loaded Routing*-based (LLR-based) and *Markov Decision Process*-based (MDP-based). Based on the fully connected topology assumption, the LLR-based algorithms try to route an incoming call to the direct link, the link between the source node and destination node, first. If the call is blocked (because of no free circuits), the "least busy" path is then tried. The least busy path can be defined as the path with the maximum free capacity [3,11,17,31] or the path with the minimum occupancy [33,35]. The MDP approach, on the other hand, formulates the routing problem as a Markov decision process and has been shown to produce a routing policy that is more general and efficient than the LLR approach [37]. The algorithms derived from the MDP approach require the evaluation of different cost functions, at least one of which can be estimated efficiently [37]. More recently, Dziong et al. [9] introduced the maximum revenue criterion and link shadow prices (first introduced by Kelly [25]) into this approach and demonstrated even further improvements.

Both the LLR and MDP approaches have been applied to point-to-point routing for multirate circuit-switched networks successfully. The LLR approach was first applied to the multirate circuit-switched networks in [1]. Later on, many LLR-based algorithms have been developed as the baseline algorithm to compare with MDP-based algorithms [10,11,22]. Applying the MDP approach to multirate circuit-switched networks has been extensively studied by Dziong et al. [10–12]. One of the difficulties of applying the MDP approach to multirate circuit-switched networks is the high computational complexity required to obtain the link shadow prices. In [22], an approximation technique was proposed to approximate the link shadow prices efficiently.

Recently, many new applications require establishing multicast connections. For example, multi-party video conferences, selected video broadcast, and distance learning all require multicast (point-to-multipoint) communication services. Therefore, extending the results from point-to-point routing to multicast routing becomes a subject of research. Thus, in this paper, we study adaptive multicast routing algorithms for multirate networks.

In the past, the multicast routing problem has been formulated as finding a minimum cost multicast tree problem, which is also known as the Steiner tree problem [18] and was shown to be NP-complete [24]. Thus, most previous researchers have focused on developing heuristic algorithms that take polynomial time and produce near optimal results (e.g., [39,41]). In formulating the multicast routing problem as a Steiner tree problem, most researchers have assumed that each link is given a predefined cost. More recently, the multicast routing problem has been formulated as a constrained Steiner tree problem in which either an end-to-end delay bound needs to be satisfied for each path in the tree [29] or there is a restriction on the number of packet copies per switch [42]. Of the previous work, only Waxman [43,44] examined the dynamic multicast routing problem in which destination nodes join or leave the multicast tree dynamically and each join request is associated with a bandwidth requirement.

Our research differs from these previous works in following aspects. First, we classify the multicast routing problems into two categories: complete and partial. In the complete version of the multicast routing problem, the identity of all destination nodes is available to the multicast routing algorithm at once. Conversely, in the partial version of the multicast problem, the identities of the destination nodes are revealed to the routing algorithm one by one. Second, two approaches, referred to as the LLR approach and the MDP approach, are adopted to define link costs. In the LLR approach, the cost of a link is defined as the negative value of the residual capacity of the link. In the MDP approach, on the other hand, each link is modeled as a decomposed Markov decision process and the approximation technique we proposed in [22] is used to estimate the link cost efficiently. Third, a new performance metric, referred to as *fractional reward loss*, is used for evaluating the performance of multicast routing algorithms. In the past, the performance of a heuristic multicast routing algorithm was evaluated by comparing the difference between the cost of the multicast tree found by the heuristic algorithm and the cost of the optimal Steiner tree. However, since the link cost can be defined in many ways, we feel that the results of such a performance evaluation are difficult to interpret. In a multirate loss network, a call carried on the network will produce a certain amount of revenue to the network. Therefore, it is more appropriate to evaluate the performance of multicast routing algorithms from the aspect of network revenue loss. Finally, several multicast routing algorithms are proposed for fully connected networks as well as sparsely connected networks. In ATM networks, Virtual Paths (VPs) are used to facilitate traffic control and network resource management [8]. A VP consists of a bundle of virtual channels and provides certain Quality of Service (QOS) guarantee. By reserving capacity on VPs, a VP can be viewed as a single logical direct link between a source node and a destination node. A fully connected VP-based ATM network can, thus, be constructed in which each pair of nodes is connected by a virtual path [4,17,23]. Therefore, we especially considered the multicast problem in fully connected networks. For fully connected networks, two heuristics used in the point-to-point routing in circuit-switched networks are adopted to develop our new multicast routing algorithms. The first heuristic is to route a call to the direct link first. The second heuristic is to limit the alternate paths to consist of at most two links. These two heuristics are used to reduce the time complexity of our multicast routing algorithms, especially the two algorithms for the complete version of the multicast routing problem. In particular, in these two algorithms, a multicast call with $|D|$ destination nodes is routed to multicast trees with $|D|$ links, referred to as "*direct*" multicast trees, first. (A direct tree is a minimum spanning tree for the source and destination nodes.) If no direct multicast tree can be built, due to some cost consideration or capacity limit, then multicast trees with $|D|+1$ links, referred to as "*alternate*" multicast trees, are then tried. (An alternate tree is a minimum Steiner tree with a single Steiner node.) If no alternate multicast tree can be built, then the call is rejected. As the alternate path for point-to-point routing is limited to consist of at most two links, searching for multicast trees with more than $|D|+1$ links is not necessary in a fully connected network. This argument is verified by our simulation results where we show that, in most cases, a multicast call can be carried by a direct multicast tree, and thus avoid the need to search for alternate multicast trees. Our simulation results also show that, by limiting the alternate multicast trees to consist of at most $|D|+1$ links, our multicast routing algorithms still yield a competitive performance as compared to algorithms that indeed search for a real Steiner minimum cost tree.

The remainder of the paper is organized as follows. In section 2, we describe our network model and define the multicast routing problem. In section 3, we present two approaches for defining the cost of a link, namely, the LLR approach and MDP approach. In section 4, multicast routing algorithms for fully connected networks are proposed. Multicast routing algorithms for general networks are then described in section 5. Performance evaluation of multicast routing algorithms based on simulations is given in section 6. Section 7 concludes our study and discusses some of our future work.

2. Network model and problem definition

We model a multirate loss network as a directed graph $G=(V,E)$ with node set V and unidirectional link (edge) set E, and a function $\mathrm{Cap}: E \to Z_0^+$, where $\mathrm{Cap}(\ell)$ is the capacity of link ℓ. Let the link from node i to node j is denoted by $e(i,j)$. The network handles K traffic classes, labeled $k=1,\ldots,K$. We assume that traffic sources that belong to the same traffic class have the same traffic characteristics and QOS requirements. Furthermore, we assume that, for variable-bit-rate traffic, the "effective bandwidth" required to set up a connection can be computed easily. Assume that the effective bandwidth required for setting up a class k call on a link is b_k units.

A multicast connection request is described by three parameters: (s,D,k), where:

- $s \in V$ is the source node of the connection,
- $D \subset V$ is the set of destination nodes to be connected,
- k is the traffic class of the call.

Let W be the set of all possible (s,D) pairs, $W=\{(s,D) \mid s \in V,\ D \subset V\}$. We assume that multicast connection requests of class k arrive at S–D pair w according to a Poisson process with rate λ_k^w. The call holding time for a class k call is assumed to be exponentially distributed with mean $1/\mu_k$, where $1/\mu_1 = 1$.

To set up a class k multicast connection of S–D pair $w = (s, D)$, a tree $T = (V_T, E_T)$, referred to as a multicast tree, is constructed where V_T is the set of nodes on the tree, $\{s\} \cup D \subseteq V_T$, and E_T is the set of links on the tree. Furthermore, b_k units of bandwidth on each link of the tree, i.e., $\forall \ell \in E_T$, are reserved for the connection. In order to guarantee QOS for established connections, a new connection request may be either accepted or rejected according to the call admission control function.

In the past research on multicast routing, a multicast tree with minimum cost is preferred where the cost is the sum of link costs on the tree. However, the definition of link cost is much less studied. In this paper, we adopt a new performance metric [20] for evaluating multicast routing algorithms. We assume that each class k connection of S–D pair w carried on the network produces r_k^w units of revenue. From another point of view, the network loses r_k^w units of revenue for each class k connection of S–D pair w that is rejected. We assume that there is no additional cost associated with any of the actions of accepting a connection, rejecting a connection, or completing a connection.

The performance metric, referred to as *fractional reward loss*, we proposed in [20] is adopted for evaluating multicast routing algorithms. Formally, the fractional reward loss of a network is defined as

$$\text{fractional reward loss} = \frac{\sum_{w \in W} \sum_{k=1}^{K} r_k^w \lambda_k^w B_k^w}{\sum_{w \in W} \sum_{k=1}^{K} r_k^w \lambda_k^w},$$

where B_k^w is the blocking probability of class k traffic of S–D pair w. Minimizing the fractional reward loss is equivalent to maximizing the expected revenues produced by the network.

We are interested in adaptive multicast routing algorithms, i.e., those algorithms that use network state information to make routing decisions. We assume that global network state information is available instantaneously whenever it is needed by the routing algorithms. The objective of a routing algorithm is to minimize the fractional reward loss. Thus, a connection is carried on the most "efficient" multicast tree, among all possible trees, based on some routing rules using current state information. Otherwise, the connection is rejected (even if there is enough network resources to carry it).

In order to minimize the fractional reward loss, the cost of a link must be carefully defined. In the following section, we propose two approaches for defining the cost of a link. Multicast routing algorithms are then proposed to find a multicast tree with minimum cost based on the link costs we proposed. In order to reflect the status of the network, the link costs are dependent on the link state. Furthermore, the call admission control function is also performed by the routing algorithm. For example, for MDP-based algorithms, a new arrived connection request is accepted and routed if the cost of the multicast tree found by the routing algorithm is less than the reward earned by carrying the new connection. Otherwise, the connection request is rejected.

Therefore, the proposed multicast routing algorithm not only find the multicast tree to route the connection, but also decide whether to accept the connection or not.

3. Link cost

In most of the past research on multicast routing problem, the link costs had been assumed to be given. However, the definition of link costs affects the performance of a multicast routing algorithm. Therefore, before studying the multicast routing problem, the issue of how to define link costs must be examined first. Since fractional reward loss is used as the performance metric to evaluate multicast routing algorithms, a scheme of defining link costs is considered as a good scheme if multicast routing algorithms adopting this scheme are able to yield lower fractional reward loss. Two approaches, the MDP approach and the LLR approach, have been proposed for defining link costs in circuit-switched networks [20]. In this session, we describe how to define link costs based on these two approaches for the multicast routing problem in multirate loss networks.

3.1. The MDP approach

The MDP approach formulates the routing problem as a Markov decision process and obtains the cost for adding a connection to the network according to the Markov decision theory. Although the routing problem in circuit-switched networks can be formulated as a Markov decision process, researchers have found that solving the Markov decision process to obtain the optimal routing policy is computationally infeasible due to the huge state space of the Markov decision process [9,37]. Thus, numerous studies, e.g., [9,32,37], have proposed a link independence assumption and consequent decomposition of the path cost into a set of separable link costs, referred to as a path cost separability assumption, to reduce the state space and the computational complexity. Even with these two assumptions, the state space is still too large for multirate loss networks for any realistic network. Therefore, in [22], the authors proposed a simplified link model to further reduce the state space and the computational complexity. In this paper, we will define the cost of adding a multicast connection on a link based on the approximation technique developed in [22]. In the following, we first briefly review the approximation technique proposed in [22]. We then describe how to define the link costs based on this technique.

The two commonly made assumptions to simplify the MDP are [9,22,32,37]:

- *The link independence assumption.* This assumption assumes that [9,10,30,37]:
  1. Calls from any class k arrive at any link ℓ according to independent Poisson processes.
  2. A call carried on an n-link path behaves like n independent calls, i.e., the link holding times for the call on each of the n single links are statistically independent.

- *The path cost separability assumption.* This assumption was first introduced in [37] and verified in [15] recently. The idea is to assume that the cost[1] of adding a call on a set of links to the network is *separable*. For example, if the call is to be carried on the multicast tree T, $T = (V_T, E_T)$, then the cost of adding the call is the sum of the cost of adding the call on each link of the tree. The cost of adding a call to a link, referred to as the link cost in this paper, is referred to as the shadow price [25], which is state-dependent.

Based on these two assumptions, we can decompose the original Markov decision process by forming a Markov decision process on each link. Let us consider a single link $\ell \in E$ in isolation. Since the network handles K classes of traffic, the state space for the decomposed Markov decision process on each link is still too large for any realistic networks. Therefore, in [22], we proposed a simplified link model to further reduce the state space and, consequently, the computational complexity. In [22], the state of a link in a multirate loss network is described by its link occupancy, i.e., the number of busy circuits. The steady state distribution of the link can then be approximated by the steady state distribution of the following birth–death process [5,45]. Let λ_k^ℓ be the arrival rate of class k calls at link ℓ and $\rho_k^\ell = \lambda_k^\ell/\mu_k$. The birth–death process has state space $0, \ldots, C_\ell$, where C_ℓ is the capacity of the link. When in state i, the process has a death rate of i and a birth rate of $\bar{\lambda}_i^\ell = \xi^2/\sigma^2 + i(1 - \xi/\sigma^2)$, where $\xi = \sum_{k=1}^K b_k \rho_k^\ell$, and $\sigma^2 = \sum_{k=1}^K b_k^2 \rho_k^\ell$.

By forming a Markov decision process on the birth–death process, we can obtain the cost of adding a class k call to link ℓ at link state i, referred to as the state-dependent link cost for class k calls and denoted by $p_k^\ell(i)$, as follows [22]. First, the difference of *relative values* $\upsilon^\ell(i) - \upsilon^\ell(i-1)$, $1 \leqslant i \leqslant C_\ell$, can be computed by the following set of simple expressions:

$$\upsilon^\ell(C_\ell) - \upsilon^\ell(C_\ell - 1) = \frac{\sum_{j=1}^K r_j^\ell \lambda_j^\ell}{\bar{\lambda}_{C_\ell - 1}^\ell} \frac{E(\bar{\lambda}^\ell, C_\ell)}{E(\bar{\lambda}^\ell, C_\ell - 1)}, \tag{1}$$

$$\upsilon^\ell(i) - \upsilon^\ell(i-1) = \frac{g}{\bar{\lambda}_{i-1}^\ell E(\bar{\lambda}^\ell, i-1)}, \quad 1 \leqslant i < C_\ell, \tag{2}$$

where

$$E(\bar{\lambda}^\ell, i) = \frac{(1/i!) \prod_{j=0}^{i-1} \bar{\lambda}_j^\ell}{\sum_{n=0}^{i} (1/n!) \prod_{j=0}^{n-1} \bar{\lambda}_j^\ell} \tag{3}$$

and

$$g = \sum_{j=1}^K r_j^\ell \lambda_j^\ell - C_\ell\big(\upsilon^\ell(C_\ell) - \upsilon^\ell(C_\ell - 1)\big). \tag{4}$$

[1] Here, the cost for adding a call to the network can be interpreted as the expected revenue loss due to call loss as a result of adding this call.

The parameter r_k^ℓ is referred to as the link reward for class k calls. In section 2, we have defined a reward parameter r_k^w for a class k multicast connection of S–D pair w. However, how to allocate this reward among the links of the multicast tree that carries this call? In this paper, the link reward is defined as follows: when a class k call of S–D pair w with reward r_k^w is carried on a multicast tree with n links, we assume that the reward is evenly allocated among these links, i.e., the reward allocated to each link, referred to as the link reward of the call [10], is r_k^w/n. This allocation scheme has been shown to be simple but efficient in [10,19].

The arrival rate of class k traffic, λ_k^ℓ, can be measured as follows: as suggested in [9,30], we first measure the carried load of each class of traffic at regular time intervals of length Δ_t. Specifically, the measured carried load of class k traffic, $\tilde{\lambda}_k^\ell$, is the average number of class k calls carried on link ℓ per unit of time. Based on the exponential smoothing model, the predicted arrival rate of class k calls on link ℓ is given by

$$\lambda_{k,\text{new}}^\ell = (1-\alpha)\lambda_{k,\text{old}}^\ell + \alpha\frac{\tilde{\lambda}_k^\ell}{1-B_k^\ell},$$

where $\lambda_{k,\text{old}}^\ell$ is the predicted arrival rate of the previous update, B_k^ℓ is the time blocking probability of class k traffic, i.e., the proportion of time the class k traffic is blocked, and α is a constant from $(0,1)$. Since the arrival rates are updated periodically, the link shadow prices are also updated adaptively, according to the traffic loads.

Based the differences of relative values obtained from equations (1) and (2), the state-dependent link costs for class k calls, $p_k^\ell(i)$, can then computed by

$$p_k^\ell(i) = \frac{\upsilon^\ell(i+b_k) - \upsilon^\ell(i)}{\mu_k}.$$

That is, based on the MDP approach, we define the cost of carrying a class k call on link ℓ when the link occupancy is i as $p_k^\ell(i)$ which is obtained by solving equations (1) and (2).

Given that we have defined the link costs, heuristic algorithms proposed in the literature for the Steiner tree problem can be adopted to find the multicast tree with minimum cost, where the cost for carrying a class k call on a multicast tree can be obtained by summing the individual link cost $p_k^\ell(i)$ of each link of the tree. In the next section, we will propose several heuristic algorithms for finding the multicast tree with minimum cost.

3.2. The LLR approach

Among the adaptive routing algorithms proposed in circuit-switched networks, algorithms based on the *Least Loaded Path Routing* (LLR) concept have been shown to be very simple and efficient. Based on the fully connected topology assumption, the LLR-based algorithms try to route an incoming call to the direct link first. If the call is blocked (because of no free circuits), the *least busy* alternate path is then tried. There are many ways to define the least busy path. However, in most research, the path with maximum free residual capacity (MFC) is considered as the least busy path, where the residual capacity of a path is defined as the minimum of the residual capacity of each link on the path. Although LLR is based on such a simple heuristic, many researches have found that it yields a very good performance both in point-to-point routing as well as in multicast routing [20]. Therefore, in this paper, we also study how to define the link cost based on the residual capacity.

Based on the LLR concept, the cost of adding a class k call on link ℓ is defined as following:

$$p_\ell^k(i) = -\big(\mathrm{Cap}(\ell) - x_\ell - b_k\big),$$

where x_ℓ is the current link occupancy.

Note that, according to the LLR concept, when computing the cost of a multicast tree, instead of adding the cost of each link on the tree, the cost of a multicast tree is defined as the maximum of the link costs of the tree.

4. Multicast routing algorithms for fully connected networks

In the last section, we have proposed two approaches for defining a cost function $p: E \times K \times X \rightarrow \mathbb{R}$, where X is the set of possible link states, K is the set of traffic classes, and $\mathbb{R}$ is the set of real numbers. Let us denote a directed graph with its corresponding cost function p by $G = (V, E, p)$. When a multicast connection request with parameters (s, D, k) arrives at the source node s of the directed graph $G = (V, E, p)$, the objective of a multicast routing algorithm is to find a multicast tree, rooted at the source node s and connected to all of the destination nodes in D, with minimum cost.

4.1. Algorithms for the partial multicast problem

The partial version of the multicast routing problem assumes that the identities of some destination nodes of a multicast connection request are revealed to the routing algorithm after some other destination nodes have already been added to the multicast tree. Thus, the routing algorithm needs to build a multicast tree with partial information. We further assume that the multicast tree is nonrearrangeable [43], i.e., routing algorithms are not allowed to modify the existing multicast tree except to add new nodes and links.

The routing algorithm we proposed is based on the shortest path (SP) concept. The basic idea is to connect a destination node to a partially-formed multicast tree by choosing the shortest path with minimum cost from the node to the partially-formed tree. If the link cost is defined based on MDP, the cost of a path is the sum of link costs on the path. On the other hand, for LLR-based link cost, the cost of a path is the maximum of all link costs on the path. A detailed description of the SP-based algorithm is given as follows.

Let us consider the establishment of a multicast connection request with parameters (s, D, k). Let T be the multicast tree to be built and cost_T be cost of the multicast tree. For each destination d, $d \in D$, to be added to the multicast tree, the user will issue an ADD_PARTY request to the routing algorithm. Upon receiving an ADD_PARTY request, following steps are performed by the SP algorithm:

1. *Free ride*. If d is already on the partially-formed multicast tree T then go to step 4. Otherwise, go to the next step.

2. *Try direct links*. Among all of the links that connect d to T, find the link ℓ with the minimum link cost $p_\ell^k(x_\ell)$. For MDP-based link costs, the link is *admissible* if $p_k^\ell(x_\ell) + \text{cost}_T < r_k^w$. For LLR-based link costs, the link is *admissible* if $p_k^\ell(x_\ell) \leqslant 0$. If the link is admissible, then node d and link ℓ are added to the multicast tree T and, for MDP-based link costs, cost_T is updated by $\text{cost}_T + p_k^\ell(x_\ell)$. The algorithm then goes to step 4. Otherwise, d cannot be added to the multicast tree via single link and next step is then performed.

3. *Try two-link path*. Among all of the two-link paths that connect T to d, find the path with the minimum path cost. Let ℓ_1 and ℓ_2 be the links on the path with link costs $p_k^{\ell_1}(x_{\ell_1})$ and $p_k^{\ell_2}(x_{\ell_2})$, respectively, and d' be the intermediate node on the path. For MDP-based link costs, the path is *admissible* if $p_k^{\ell_1}(x_{\ell_1}) + p_k^{\ell_2}(x_{\ell_2}) + \text{cost}_T < r_k^w$. For LLR-based link costs, the path is admissible if $p_k^{\ell_1}(x_{\ell_1}) + t_{\ell_1,k}^w \leqslant 0$ and $p_k^{\ell_2}(x_{\ell_2}) + t_{\ell_2,k}^w \leqslant 0$, where $t_{\ell,k}^w$ is the trunk reservation level for class k traffic of S–D pair w on links of alternate paths which is to be defined. If the path is admissible, then nodes d, d' and links ℓ_1, ℓ_2 are added to the multicast tree T and, for MDP-based link costs, cost_T is updated by $\text{cost}_T + p_k^{\ell_1}(x_{\ell_1}) + p_k^{\ell_2}(x_{\ell_2})$. The algorithm then goes to step 4. Otherwise, the multicast request is blocked and all resources previously reserved on the multicast tree are released.

4. *Get next or accept*. If the multicast request is not blocked, get the next ADD_PARTY request and go to step 1. If there is no additional ADD_PARTY request, then the multicast connection has been successfully set up.

Step 3 in the SP algorithm can be extended such that paths with more than two links can be tried. However, from empirical experience of point-to-point routing in the circuit-switched networks, we suggest to limit the length of alternate paths to be at most two.

The purpose of introducing trunk reservation levels at step 3 is to avoid network performance degradation under high traffic load. In original LLR-based point-to-point routing algorithms, a call blocked on the direct link is routed to an alternate path which consist of, in most cases, two links. However, under a high traffic load, such a routing method may lead to an unstable, and inefficient network state in which most calls are carried on two-link paths. Therefore, trunk reservation was introduced to guard against such performance degradation. With the same reasoning, trunk reservation is also used in our LLR-based multicast routing algorithms. On each link of a two-link path, a connection of class k traffic of S–D pair w can be accepted only if the residual capacity of this link is larger than $t^w_{\ell,k}$. In [21], we demonstrated that, in a single rate loss network, setting appropriate trunk reservation levels can reduce the fractional reward loss significantly.

The optimal trunk reservation problem in single rate loss networks has been well studied in the literature [20,28,36,40]. In [28], a set of simple expressions for computing the optimal trunk reservation levels was derived based on the Markov decision theory. We have also derived a different set of simple expressions for computing the optimal trunk reservation levels for LLR-based multicast routing algorithms in [20]. However, extending these results to multiple classes of traffic will encounter extremely high computational complexity. Therefore, in our simulations, the trunk reservation levels are set based on link shadow prices obtained from equations (1) and (2). Specifically, $t^w_{\ell,k}$ is set to the largest integer value such that $p^\ell_k(C_\ell - t^w_{\ell,k}) > r^w_k/2$. The readers are referred to [20] for the derivation of trunk reservation levels.

4.2. Algorithms for the complete multicast problem

The complete version of the multicast routing problem assumes that the routing algorithm is able to know all the destination nodes before building the multicast tree. In this section, we propose a new approach toward the complete multicast routing problem based on the minimum spanning tree (MST) algorithm.

The basic idea of the proposed algorithm is to find the minimum cost tree with $|D|$ links first where $|D|$ is the cardinality of the destination set. If the minimum cost tree is not "*admissible*", then it tries to find the minimum cost tree with $|D|+1$ links. Details of the algorithm is given as follows: upon receiving a multicast request of a class k connection of S–D pair $w = (s, D)$ which has reward r^w_k, the following steps are performed by the MST algorithm:

1. Construct a subgraph $G' = (D', E')$ from G, where $D' = \{s\} \cup D$ and E' is a subset of E and $\ell \in E'$ if and only if the end-nodes of ℓ are in the set D'.
2. Construct the minimum spanning tree $T = (D', E_T)$ from G'. Let cost_T be the cost of tree T.[2]
3. For MDP-based link costs, the tree is admissible if $\text{cost}_T < r^w_k$. For LLR-based link costs, the tree is admissible if $\text{cost}_T \leqslant 0$. If the tree is admissible, then b_k circuits are reserved on each link of the multicast tree T and the multicast connection has been successfully set up. Otherwise, go to the next step.
4. For each node which is neither the source node nor one of the destination nodes, i.e., $\forall v \in V - D - \{s\}$, construct a subgraph $G_v = (D_v, E_v)$ from G, where $D_v = \{s, v\} \cup D$ and E_v is a subset of E and $\ell \in E_v$ if and only if the end-nodes of ℓ are in the set D_v.

[2] Recall that if the link cost is defined based on MPD, the cost of a tree is the sum of link costs of the tree. On the other hand, for LLR-based link costs, the cost of a tree is the maximum of all link costs of the tree.

5. From each graph G_v, construct the minimum cost tree $T_v = (D_v, E_{T_v})$. Let cost_{T_v} be the cost of tree T_v.

6. Among those minimum cost trees T_v, $\forall v \in V - D - \{s\}$, find the tree, referred to as T_*, with the minimum cost, i.e., $\text{cost}_{T_*} \leqslant \text{cost}_{T_v}$, $\forall v \in V - D - \{s\}$.

7. For MDP-based link costs, the tree is admissible if $\text{cost}_{T_*} < r_k^w$. For LLR-based link costs, the tree is admissible if $(\text{cost}_{T_*} + t_{\ell,k}^w) \leqslant 0$, where $t_{\ell,k}^w$ is the trunk reservation level for class k traffic of S–D pair w at link ℓ. If the tree is admissible, then b_k circuits are reserved on each link of the multicast tree T_* and the multicast connection has been successfully set up. Otherwise, the multicast request is blocked.

There exists many efficient algorithms for finding minimum spanning tree in the literature. In our numerical results, Prim's minimum spanning tree algorithm [38] is adopted. The time complexity of the first three steps of the MDP_MST and LLR_MST algorithms is $O(|E'|\log|D'|)$ and is $O((|V| - |D| - 1)|E_v|\log|D_v|)$ for the last four steps.[3]

As in the SP-based algorithm, the MST-based algorithm can be extended to construct a multicast tree with more than $|D| + 1$ links. However, as we can observe in the algorithm, the number of subgraphs, as well as the minimum spanning trees, to be constructed increases exponentially as we increase the size of the multicast tree.[4] Furthermore, we expect that, for most multicast connections, a multicast tree with $|D|$ links can be constructed successfully. Therefore, by finding a multicast tree with $|D|$ links first and limiting the number of links of a multicast tree to be at most $|D| + 1$ links, we are able to reduce the computation time significantly. In our numerical results, we will show that by only searching multicast trees with at most $|D| + 1$ links, the MDP_MST algorithm yields almost the same performance as algorithm that searches for the real Steiner minimum cost tree.

5. Multicast routing algorithms for general networks

The routing algorithms we proposed for fully connected networks in the last section are based on two important heuristics: try the direct tree first, if blocked, then try an alternate tree. However, these heuristics will not hold in general networks. Therefore, in this section, we propose new algorithms for general networks. In the following, we propose one algorithm, based on the shortest path concept, for the partial version of the multicast routing problem and two algorithms, one based on shortest path and the other one based on genetic algorithm (GA), for the complete version of the multicast routing problem.

[3] Assume Prim's algorithm, implemented with binary heap, is used to find the minimum spanning tree. The complexity can be further improved if we know the graph is fully connected (see also [34]), however it is beyond the scope of this paper.

[4] Recall that many point-to-point routing algorithms for circuit-switched networks will only allow an alternate path to consist of at most two links!

5.1. *Algorithms for the partial multicast problem*

The SP algorithm proposed for fully connected networks can be easily extended to general networks. Let us consider the establishment of a multicast connection request with parameters (s, D, k). Let T be the multicast tree to be built and cost_T be cost of the multicast tree. Upon receiving an ADD_PARTY request, following steps are performed by the SP algorithm:

1. *Free ride*. If d is already on the partially-formed multicast tree T, then go to step 3. Otherwise, go to the next step.
2. *Find the shortest path*. Among all of the paths that connect T to d, find the path with the minimum path cost. Let cost_P be the cost of the path. For MDP-based link costs, cost_T is updated by $\text{cost}_T + \text{cost}_P$ and go to the next step. For the LLR-based link costs, if $\text{cost}_P > 0$, then the connection is rejected. Otherwise, go to the next step.
3. *Get next*. If the multicast request is not blocked, get the next ADD_PARTY request and repeat step 1 and 2 until all destinations have been added to the multicast tree. For LLR-based link costs, if no blocking at step 2, then the multicast connection has been successfully set up. For MDP-based link costs, the connection is accepted if $\text{cost}_T < r_k^w$. Otherwise, the connection is rejected.

5.2. *Algorithms for the complete multicast problem*

The complete version of the multicast routing problem for general networks is known as the Steiner tree problem except that, in most of the past research, the Steiner tree problem was defined and solved for undirected graphs. We refer to the Steiner tree problem in directed graphs as the directed Steiner tree problem. The Steiner tree problem had been shown to be a NP-complete problem and many heuristic algorithms had been proposed. In this section, we study two heuristic algorithms for the directed Steiner tree problem. The first algorithm, referred to as the TMR algorithm, is a revised version of the algorithm proposed by Takahashi and Matsuyama [41]. The second algorithm, referred to as the GA algorithm, is a heuristic algorithm we developed based on genetic algorithms.

5.2.1. *The TMR algorithm*

Denote a tree $T = (V_T, E_T, p_T)$ as a subgraph of G such that $V_T \subset V$, $E_T \subset E$, and $p_T(e) = p(e)\ \forall e \in E_T$, where $p(e)$ is the cost of link e. Let $d(v, T)$ be the minimum cost for connecting node v to the tree T. Specifically,

$$d(v, T) = \min_{v' \in V_T} \text{mincost}\big(v', v\big),$$

where $v' \in V_T$ and $\text{mincost}(v', v)$ is the minimum cost connecting v' to v. When a multicast connection request with parameters (s, D, k) arrives at the network, the TMR algorithm proceeds as follows.

Step 1. Let $T_0 = (V_0, \emptyset)$, where $V_0 = \{s\}$.

Step 2. For $i = 1$ to $|D|$ do find a node $v_i \in D \backslash V_{i-1}$ such that $d(v_i, T_{i-1})$ is minimal. Construct the tree $T_i = (V_i, E_i)$ by adding to T_{i-1} the nodes and edges of the shortest path that connects T_{i-1} to v_i.

Step 3. Let the resulting tree be $T = T_{|D|} = (V_T, E_T, p_T)$. Find the minimum spanning tree of the subgraph of G induced by V_T and prune the resulting tree to ensure that all leaves are destination nodes.

The step 3 was suggested in [39] to improve the performance.

5.2.2. The GA algorithm

The Genetic Algorithms (GAs) are used for solving optimization problems based on the principle of evolution. A population of candidate solutions, called chromosomes, are maintained at each iteration of the evolution. Each chromosome consists of linearly arranged genes which are represented by binary strings. Three basic operations, namely, reproduction, crossover, and mutation, are used in the evolution to generate new offspring. Reproduction is based on the Darwinian survival of the fittest among strings generated. The samples (represented as bit strings) with larger fitness function values are selected to generate new offspring bit strings by crossover operations and convert the offspring to new parameter solutions. Intuitively, a bit string with a larger fitness function value should have a higher probability of contributing one or more offspring bit strings in the next generation and vice versa. Crossover is used to cut two parent bit strings into two or more segments and then combine the segments to generate two offspring bit strings. Crossover can produce offspring that are radically different from their parents. Suppose the crossover operation is performed on the two bit strings “01110001” and “10011011”, and they are split at the second bit, then two new bit strings “01011011” and “10110001” are generated. There are other ways for implementing the crossover operation, e.g., arithmetic crossover [6].

Mutation is to perform random alternation on bit strings by some operations, such as bit shifting, inversion, rotation, etc. The mutation operation will create new offspring bit strings different from those generated by the reproduction and crossover operations. Mutation can extend the scope of the solution space and reduce the possibility of falling into local extremes.

The genetic algorithms are typically implemented as follows:

Step 1. Initialize a population of chromosomes (solutions).

Step 2. Evaluate each chromosome in the population.

Step 3. Create new chromosomes by mating current chromosomes and apply mutation and recombination as the parent chromosomes mate.

Step 4. Delete members of the population to make room for the new chromosomes.

Step 5. Evaluate the new chromosomes and insert them into the population.

Step 6. If the stopping criterion is satisfied, then stop and output the best chromosome (solution); otherwise, go to step 3.

In the following, we present a novel multicast routing algorithm based on the GA [7].

5.2.2.1. Representation of chromosomes

For a directed graph $G = (V, E, c)$, there are $|V| \times (|V| - 1)$ possible source–destination pairs. A source–destination pair can be connected by a set of links, which is called a "route". There are usually many possible routes between any source–destination pair. For example, consider the network of figure 1, the possible routes between v_0 to v_4 include $v_0 - v_4$, $v_0 - v_5 - v_4, \ldots$ and so on.

Our GA-based multicast routing algorithm assumes that a routing table, consist of R possible routes, has been constructed for each source–destination pair. For example, figure 2 shows the routing table for the source–destination pair (v_0, v_4). The size of

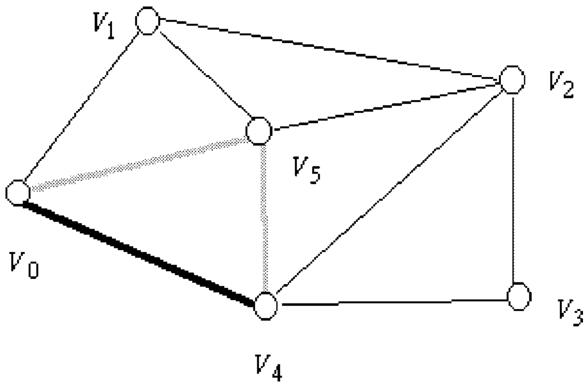

Figure 1. A simple 6-node network.

Routing table of $v_0 \to v_4$	
Route No.	Route lists
0	$v_0 - v_4$
1	$v_0 - v_5 - v_4$
⋮	⋮
R-1	$v_0 - v_5 - v_1 - v_2 - v_3 - v_4$

Figure 2. Routing table for the 6-node network.

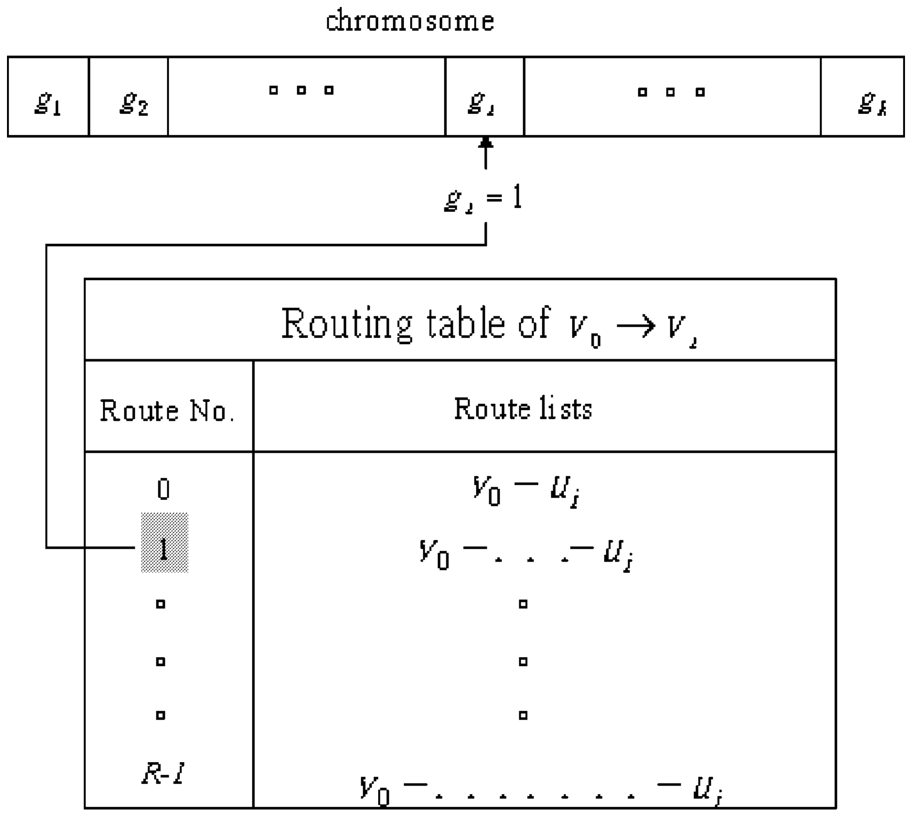

Figure 3. The relationship between chromosome, gene, and routing table.

the routing table R is a parameter of our algorithm. The possible routes in the routing table can be sorted according to their length (i.e., number of links) or delay such that shorter paths are assigned smaller route number.

For a given source node s and destination set $D = \{d_1, d_2, \ldots, d_{|D|}\}$, a chromosome can be presented by a string of integers with length $|D|$. A gene g_i, $1 \leqslant i \leqslant |D|$, of the chromosome is an integer in $0, 1, \ldots, R-1$ which represents a possible route between s and d_i, where $d_i \in D$. The relationship of chromosome, gene, and routing table is explained in figure 3.

5.2.2.2. Description of the algorithm

The GAs maintain a population of chromosomes, each of which has a fitness value. The fitness value defines the quality of the chromosome. Beginning with a set of random chromosomes, a process of evolution is simulated. The main components of this process are crossover, which mimics propagation, and mutation, which mimics the random changes occurring in nature. After a number of generations, highly fit chromosomes will emerge corresponding to good solutions.

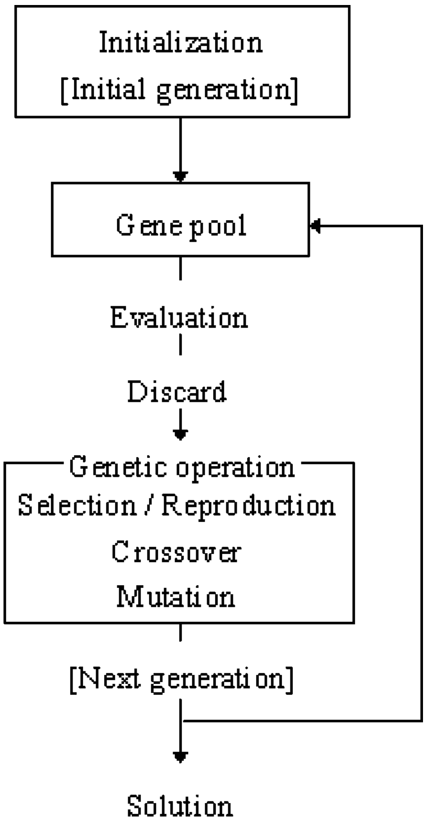

Figure 4. The proposed GA-based multicast routing algorithm.

Figure 4 schematically illustrates the outline of our GA-based multicast routing algorithm. Details of each step is described as follows.

Initialization of chromosomes. Recall that a chromosome consists of a sequence of genes, each corresponding to a specific route of a routing table. The initial procedure generates P different chromosomes at random from the range of $(0, R/4)$.[5] This set of chromosomes is called the gene pool (or population), and P is the size of the gene pool.

Evaluation of chromosomes. The fitness value of a chromosome is the value of the objective (fitness) function for the solution (e.g., a multicast tree) represented by the chromosome. Since we are finding the multicast tree with minimum cost, our fitness function has an opposite meaning than those found in other GAs, i.e., the smaller the fitness value the better the chromosome. Specifically, given a gene pool

[5] Intuitively, genes corresponding to shorter routes are preferred. Since the routes in routing tables are sorted according to their length, preference for shorter routes is equivalent for preference for genes with small values.

$H = h_0, h_1, \ldots, h_{P-1}$, the fitness value of each chromosome is computed as follows. The fitness value of the chromosome h_i, $F(h_i)$, is the sum of the costs of the links of the graph represented by the chromosome h_i. After evaluating the fitness values of all, chromosomes are then sorted according to their fitness values such that $F(h_0) \leqslant F(h_1) \leqslant \cdots \leqslant F(h_{P-1})$. (That is, the first chromosome in the pool is the best solution found so far.)

Discard the duplicate chromosomes. Applying some of the genetic operations, e.g., crossover, on two duplicate chromosomes will yield the same chromosome. Therefore, too many duplicated chromosomes in the gene pool will reduce the ability of searching. Once this situation occurs, the duplicated chromosomes must be discarded. In our algorithm, they are replaced by new randomly generated chromosomes.

Reproduction. According to the computed fitness values, some of chromosomes are selected to generate more offspring through crossover and mutation operations, and others will be removed from the gene pool. In our algorithm, chromosomes with small fitness values will survive and reproduce more. On the other hand, chromosomes with large fitness values die off. The reproduction process selects certain number of chromosomes with the best fitness values from the current generation for reproduction. Another number of chromosomes, again with the best fitness values, are selected to reproduce offspring through crossover operation. Note that the number of the chromosomes in the gene pool is always restricted to P.

Crossover. Crossover operation is used to exchange genetic information between two chromosomes. In this process, two chromosomes strings with smaller fitness values are picked from the gene pool first. The start point and length of the portion to exchange are randomly selected. Two new offsprings are created and put back into the gene pool.

Mutation. The mutation operation provides an opportunity for a random change in the chromosome. In our algorithm, pointwise mutation is adopted in which one bit in the chromosome string is changed with a probability, referred to as the mutation probability. The mutation operation gives the genetic algorithm an opportunity to search for new more feasible chromosomes in new regions of the solution spaces.

Stopping criterion. The genetic algorithm stops when a pre-defined number of iterations is encountered, which is set to 100 in our simulations. The effect of this number on the performance of the genetic algorithm has been studied in [46].

6. Numerical results

In this section, the performance of the proposed routing algorithms is studied through simulations on fully connected networks as well as sparsely connected networks.

6.1. Fully connected networks

The performance of routing algorithms is studied on symmetric networks as well as asymmetric networks. In symmetric networks, each link has the same capacity which is set to 120 units of bandwidth. On the other hand, in asymmetric networks, links are ordered according to their source and destination id and then divided into five groups. Links in group 1, 2, 3, 4, 5 have capacity of 150, 60, 120, 30, 90, respectively.[6]

Figures 5, 7 and 8 show the performance of the four routing algorithms on 10-node, 20-node, and 30-node fully connected networks under various traffic loads, respectively. The x-axis of these figures is the arrival rate of class 1 calls with one destination node, i.e., the point-to-point connections. Figure 6 shows the performance on 10-node symmetric networks under different arrival rate settings. For 10-node networks, the performance of a "*pseudo-optimal*" multicast routing algorithm,[7] OPT_MST, is also shown in the figures. In OPT_MST, the link costs are obtained as in the MDP_MST algorithm. However, the search for the multicast tree with minimum cost is different. In OPT_MST, no preference is given to multicast trees with smaller number of links. Instead, a real multicast tree with minimum cost is found by an exhaustive search.

The parameters of our simulations are set as follows: two classes of traffic are presented to the network with bandwidth requirement $b_1 = 1$ and $b_2 = 5$. The arrival rate of class 1 traffic is set five times more than the class 2 traffic for each S–D pair, i.e., $\lambda_1^w = 5\lambda_2^w$. The arrival rates for multicast requests with different sizes of destination set are set as follows. Let W_i be the set of all multicast requests which have exactly i destinations. Let λ^{W_i} denote the aggregated arrival rate for connection requests of W_i. For simulations of figures 5, 7 and 8, the aggregated arrival rates for different sizes of destination sets are set to the same, i.e., $\lambda^{W_i} = \lambda^{W_j},\ \forall i, j$. For the simulations of figure 6(a), the ratio between two aggregated arrival rates is given by $\lambda^{W_i}/\lambda^{W_1} = i$, while the ratio is given by $\lambda^{W_i}/\lambda^{W_1} = 1/i$ in figure 6(b). The reward of a class k connection of S–D pair $w = (s, D)$ is set to $b_k|D|$ units of revenue. In these figures, the fractional reward losses yielded by the five routing algorithms are compared under different traffic loads. The vertical lines about each point in these figures indicate the 95% confidence interval. Each simulation point in these figures is observed over 10 independent runs. The length of each run is 2000 units of mean call holding time of class 1 calls. For each run, the initial 10% of the samples were discarded.

From figures 5–8, we can observe that the algorithms based on the minimum spanning tree (MST) approach outperform the algorithms based on the shortest path (SP) approach. Clearly, with the identities of all destination nodes available at once, MST-based algorithms are able to construct a more efficient multicast tree for a mul-

[6] There is no particular reason for such a capacity assignment. It is just to make the topology asymmetric.

[7] It is not the real optimal multicast routing algorithm. It is optimal if the link costs we obtained from the decomposed Markov decision process are exact.

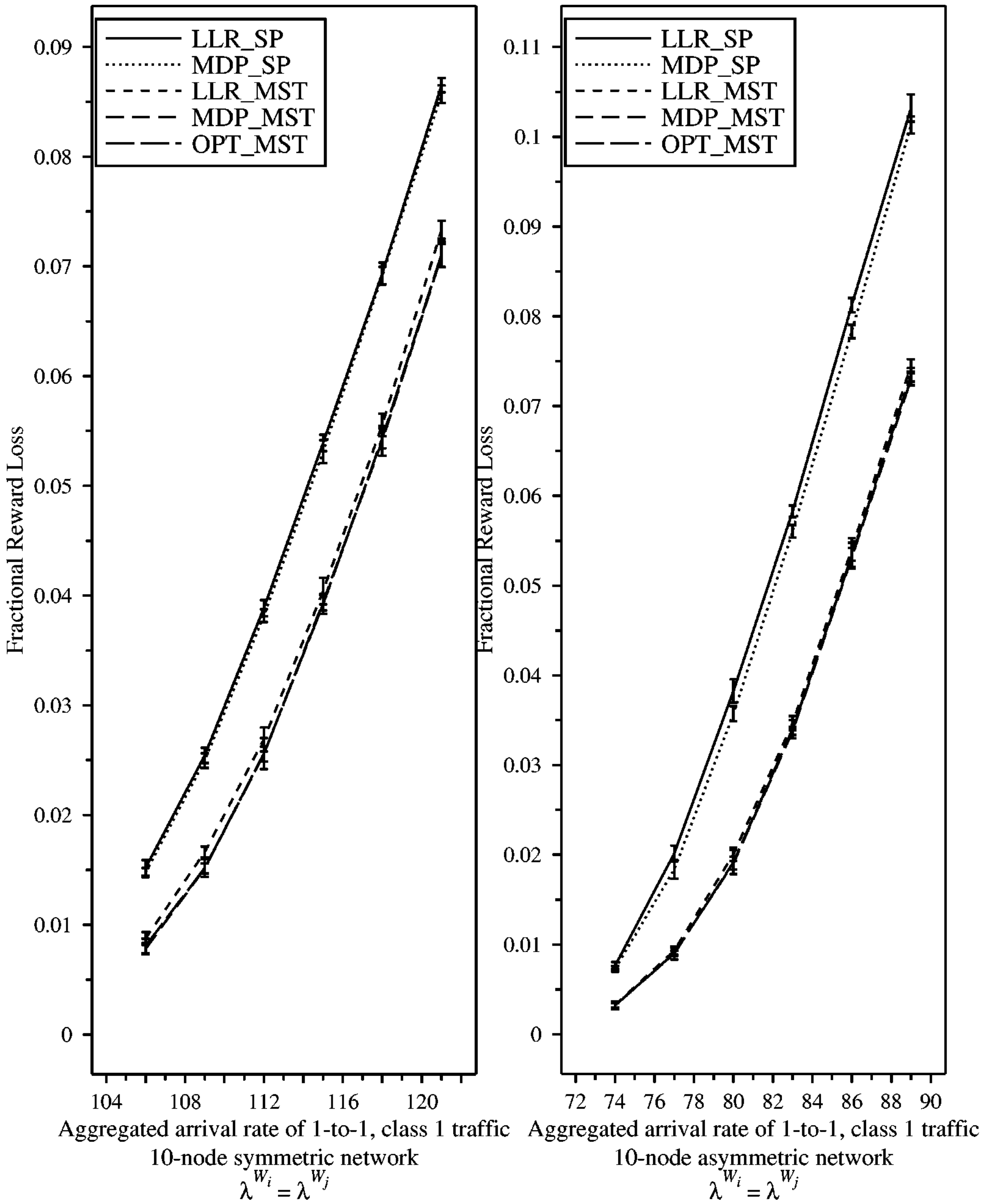

Figure 5. Performance comparison of the proposed multicast routing algorithms on the 10-node network.

ticast connection. Intuitively, this result suggests that as the routing algorithms use more information, they achieve a better performance.

For different approaches of defining link costs, we observe that algorithms based on the LLR approach yield almost the same performance as those based on the MDP ap-

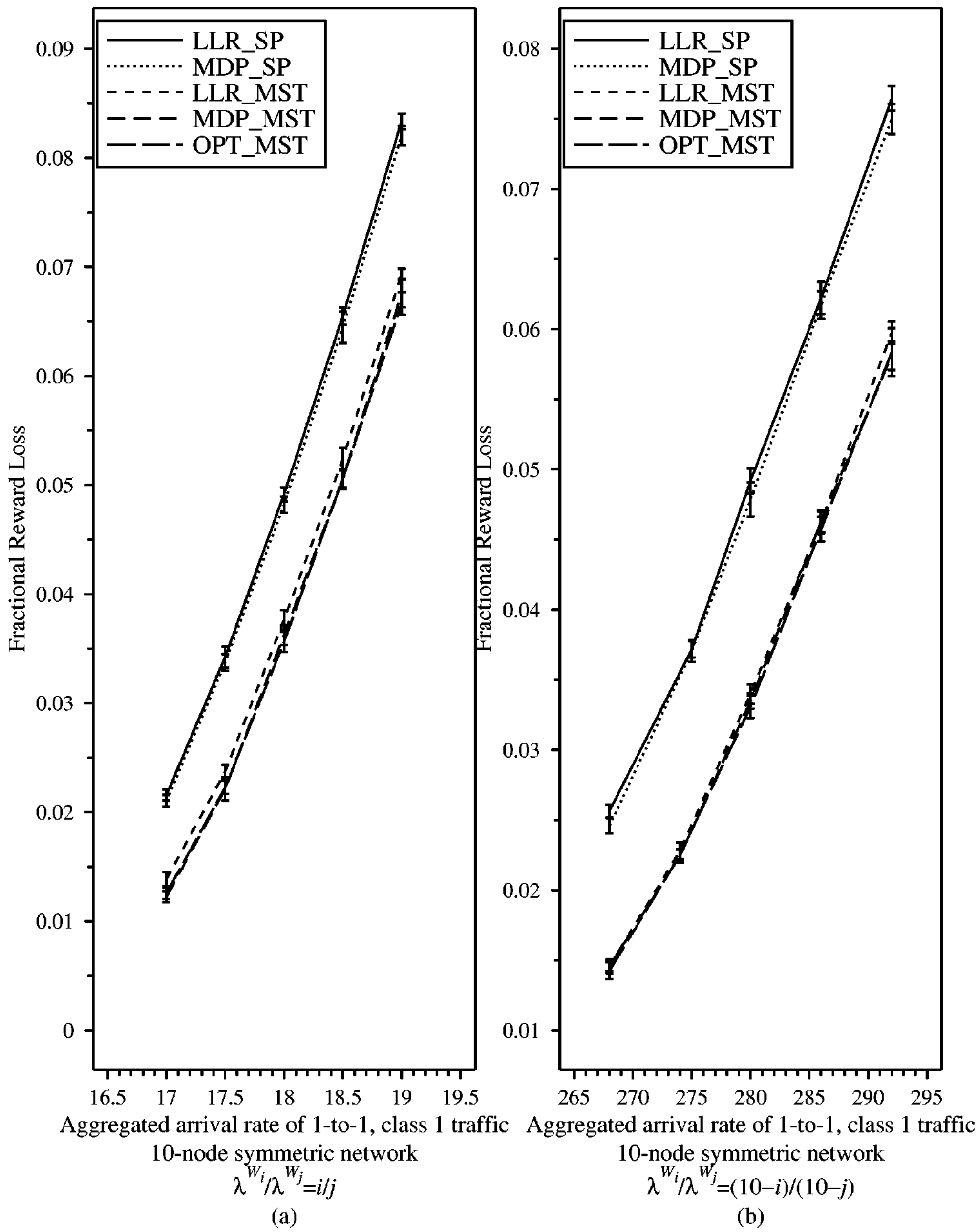

Figure 6. Performance comparison of the proposed multicast routing algorithms on the 10-node network.

proach. Although in symmetric networks, the MDP_MST algorithm performs slightly better than the LLR_MST algorithm, the difference is not significant. Therefore, for fully connected networks, defining link cost based on residual capacity of the link is a simple but efficient mechanism.

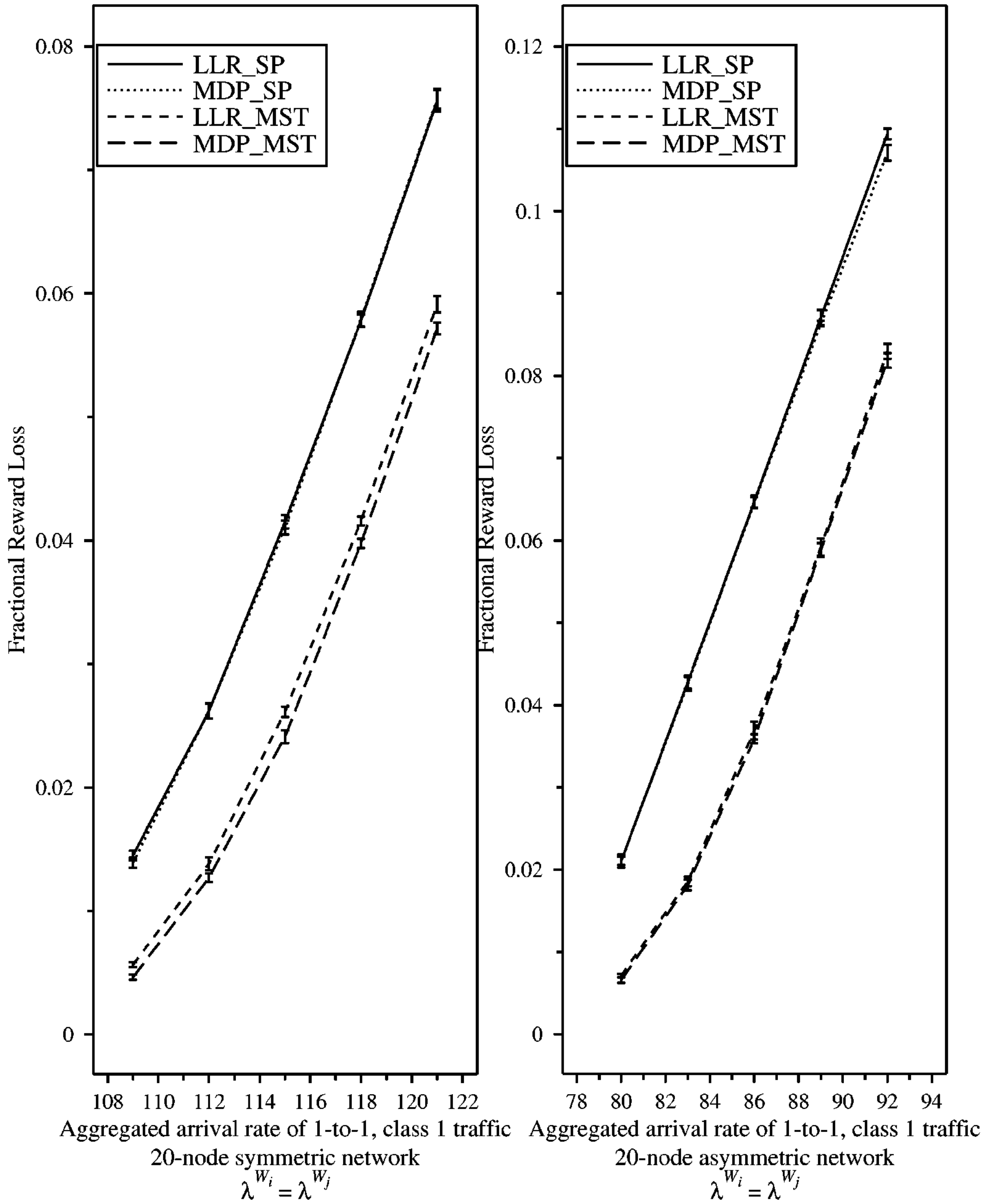

Figure 7. Performance comparison of the proposed multicast routing algorithms on the 20-node network.

From figures 5 and 6, we also observe that MDP_MST and OPT_MST have almost the same performance. A very important conclusion can be made from this observation. For point-to-point routing in circuit-switched networks, researchers have found that giving priority to the direct link and limiting the alternate paths to consist

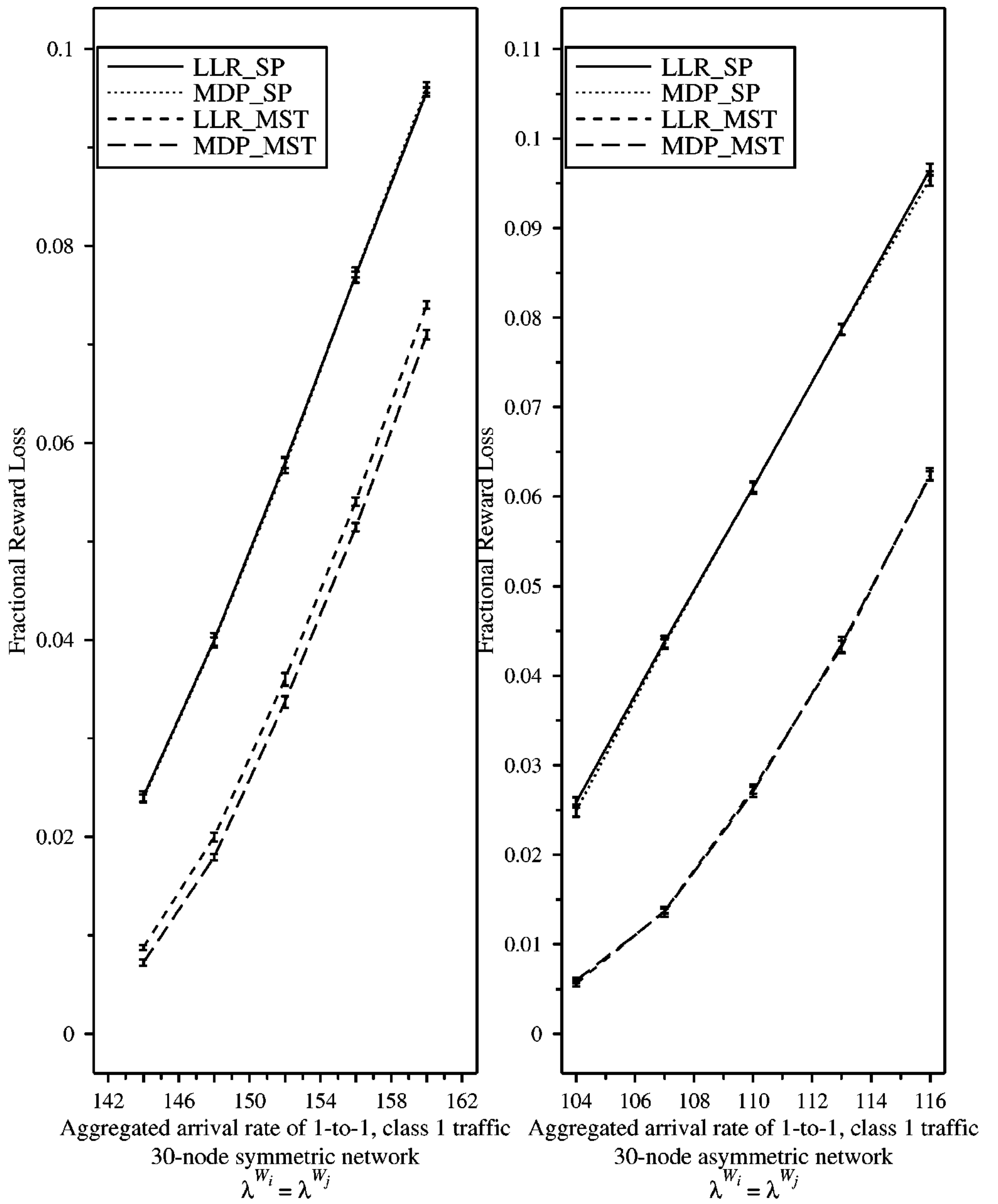

Figure 8. Performance comparison of the proposed multicast routing algorithms on the 30-node network.

of at most two links are two very important heuristics. We believe that this is also the case for multicast routing in multirate loss networks with fully connected topology. Therefore, we limit the search for multicast trees to at most $|D| + 1$ links. Results shown in these figures indicate that there is indeed no need for finding a minimum cost Steiner tree.

Table 1
Percentage of calls that are carried on the direct multicast tree for the MDP_MST algorithm.

10-node symmetric network					
Arrival rate	106	109	112	115	118
Fractional reward loss	0.0078	0.0151	0.0256	0.0394	0.0541
Class 1	99.983	99.966	99.947	99.929	99.905
Class 2	97.747	95.812	93.072	89.542	86.766
10-node asymmetric network					
Arrival rate	74	77	80	83	86
Fractional reward loss	0.0032	0.0090	0.0191	0.0338	0.0531
Class 1	99.983	99.971	99.949	99.940	99.918
Class 2	98.165	95.781	92.552	88.122	84.119
20-node symmetric network					
Arrival rate	109	112	115	118	121
Fractional reward loss	0.0046	0.0127	0.0241	0.0398	0.0572
Class 1	99.990	99.975	99.958	99.942	99.930
Class 2	98.218	95.137	91.237	86.197	81.003
20-node asymmetric network					
Arrival rate	80	83	86	89	92
Fractional reward loss	0.0066	0.0181	0.0359	0.0588	0.0818
Class 1	99.990	99.984	99.975	99.969	99.939
Class 2	95.450	90.943	84.568	78.388	73.966
30-node symmetric network					
Arrival rate	144	148	152	156	160
Fractional reward loss	0.0072	0.0179	0.0337	0.0514	0.0710
Class 1	99.985	99.966	99.942	99.927	99.920
Class 2	96.850	92.686	87.520	82.522	77.190
30-node asymmetric network					
Arrival rate	104	107	110	113	116
Fractional reward loss	0.0060	0.0136	0.0270	0.0435	0.0623
Class 1	99.988	99.981	99.965	99.948	99.963
Class 2	96.368	96.478	93.455	89.879	79.311

Giving priority to multicast trees with smaller number of links also reduces the complexity of the MDP_MST and LLR_MST algorithms significantly. A multicast tree that is built by only executing the first three steps is referred to as a *direct multicast tree* since it consists of exactly $|D|$ links. Similarly, a multicast tree is referred to as an *alternate multicast tree* if it is built by the last four steps. A call is rejected only if all six steps are executed and no multicast tree can be built. Although the time complexity to build an alternate multicast tree or reject a call request is very high, only the first three steps will be executed for most of the call requests, as shown in table 1 which shows the percentage of call requests that are carried on the *direct* multicast trees (among all call requests) for the MDP_MST algorithm under different traffic conditions and network models. These results indicate that with very high probability, a multicast tree can be found through the first three steps. In other words, although

Table 2
Blocking probability for class 1 traffic under different routing algorithms.

Arrival rate	106	109	112	115	118
MDP_SP	0.00104	0.00185	0.00291	0.00395	0.00511
LLR_SP	0.00107	0.00186	0.00287	0.00390	0.00500
MDP_MST	0.00004	0.00008	0.00013	0.00018	0.00023
LLR_MST	0.00003	0.00006	0.00009	0.00014	0.00018
OPT_MST	0.00004	0.00008	0.00013	0.00018	0.00024

Table 3
Blocking probability for class 2 traffic under different routing algorithms.

Arrival rate	106	109	112	115	118
MDP_SP	0.02848	0.04808	0.07350	0.10228	0.13323
LLR_SP	0.02931	0.04903	0.07486	0.10243	0.13371
MDP_MST	0.01560	0.03019	0.05101	0.07862	0.10795
LLR_MST	0.01748	0.03320	0.05372	0.08066	0.11089
OPT_MST	0.01600	0.03020	0.05128	0.07821	0.10794

$|V| - |D| - 1$ minimum spanning trees need to be built at the last four steps of the MDP_MST and LLR_MST algorithms, the probability that these steps will be executed is very low when blocking probability is low.

Tables 2 and 3 show the detailed blocking probabilities for the two classes of traffic under the 10-node symmetric network and the aggregated arrival rate for multicast requests of different sizes of destination sets are set to the same. Since class 2 traffic requires more bandwidth, therefore, it has a higher blocking probability than class 1. The MDP-based policies slightly outperform LLR-based algorithms by reducing the blocking probability of class 2 while only slightly increasing the blocking probability of class 1. The fairness problem among the blocking probabilities for calls of different classes needs further study and is beyond the scope of this paper.

6.2. General networks

In this section, we study the performance of the proposed multicast routing algorithms on general networks. The simulation results presented in this section are carried on a random graph in which $|V|$ nodes are placed randomly on a $|V| \times |V|$ lattice points in the plane and the probability of a link existing between nodes depends on the distance between them [43] (see [42] for the details). In our simulations, a 20-node random graph was constructed as shown in figure 9. Each link is assumed to have 100 units of bandwidth.

The following assumptions are made in the simulations. As in the fully connected networks, the network is assumed handle two classes of traffic with bandwidth requirements $b_1 = 1$ and $b_2 = 5$, respectively. The arrival rate of class 1 traffic is set five times more than the class 2 traffic for each S–D pair. Recall that λ^{W_i} denotes the

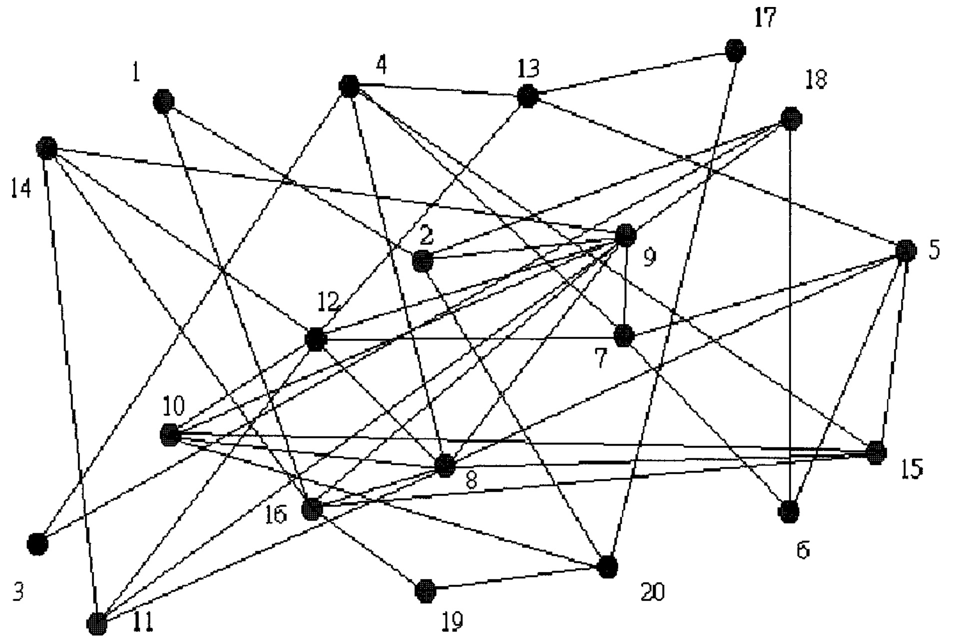

Figure 9. A 20-node random graph.

aggregated arrival rate for connection requests of W_i. In this section, the ratio between two aggregated arrival rates is given by $\lambda^{W_i}/\lambda^{W_1} = 1/i$. All connections are assumed to have an exponentially distributed holding time with the same mean. The reward for carrying a multicast connection with parameter (s, D, k) is set to $|D| \times b_k/\mu_k$.

Six multicast routing algorithms are studied in this section. Three of them use the MDP approach for defining link costs: MDP_SP, MDP_TMR, and MDP_GA. On the other hand, LLR_SP, LLR_TMR, and LLR_GA, use the LLR approach for defining link costs. For the three MDP-based algorithms, the multicast routing schemes adopted are SP, TMR, and GA, respectively. So are the LLR-based algorithms. Table 4 shows the fractional reward loss of each of these six routing algorithms under various traffic loads. The arrival rate shown in table 4 is the aggregated arrival rate for class 1, point-to-point calls. The 95% confidence intervals are also given in table 4. Based on the results in table 4, following observations can be made.

1. Algorithms that estimate link costs based on MDP yield significantly better performance than algorithms that use the LLR approach. In other words, MDP is a better methodology for estimating link costs. This observation is contradictory to what we had observed in fully connected networks. The reason is quite intuitive. In fully connected networks, there is a direct link between every source–destination pair and the routing paths are limited to those with at most two links. Therefore, there is no need for routing algorithms to take the path length of a routing path into consideration. However, in a general network, length of routing paths may differ from each other significantly. Routing a connection on a long path, or a

Table 4
Performance of multicast routing algorithms under general network topology.

Arrival rate	LLR_SP	LLR_TMR	LLR_GA
80	0.2055 ± 0.0006	0.2843 ± 0.0008	0.1712 ± 0.0022
85	0.2283 ± 0.0009	0.3076 ± 0.0009	0.2097 ± 0.0017
90	0.2507 ± 0.0009	0.3303 ± 0.0009	0.2466 ± 0.0017
95	0.2689 ± 0.0009	0.3517 ± 0.0009	0.2838 ± 0.0015
100	0.2881 ± 0.0009	0.3707 ± 0.0008	0.3135 ± 0.0019
Arrival rate	MDP_SP	MDP_TMR	MDP_GA
80	0.0124 ± 0.0006	0.0108 ± 0.0004	0.0108 ± 0.0003
85	0.0262 ± 0.0006	0.0253 ± 0.0006	0.0222 ± 0.0007
90	0.0434 ± 0.0008	0.0388 ± 0.0008	0.0370 ± 0.0009
95	0.0664 ± 0.0008	0.0581 ± 0.0008	0.0563 ± 0.0011
100	0.0907 ± 0.0010	0.0803 ± 0.0010	0.0774 ± 0.0009

large multicast tree, requires more resources, such as bandwidth, than on a shorter one. The LLR approach does not provide any length information to the routing algorithm while the MDP does. Furthermore, in fully connected networks, trunk reservation levels are estimated according to equations (1) and (2) (see [20]) and set on each link to prevent too many calls carried on alternate paths (trees). However, the results of [20] cannot be applied to networks with general topology. Therefore, no trunk reservation is made in the three LLR-based algorithms proposed for general networks. Finally, the MDP approach also provides a better call admission control function by comparing the estimated cost of carrying the incoming call to the expected reward for carrying that call.

2. As in fully connected networks, the MDP-based routing algorithms of the complete version of multicast problem yield better performance than that of partial multicast problem. Between the two routing algorithms for complete version of the multicast problem, the GA algorithm yields slightly better performance than the TMR algorithm. However, the computational complexity is much higher for the GA algorithm. In order words, the TMR algorithm is a simple but efficient heuristic algorithm.

3. Among the three LLR-based algorithms, the LLR_TMR yields the worst performance. Since the LLR approach does not take the length of a multicast tree into consideration, it is not obvious to us why one algorithm would perform better than another one. In other words, the TMR algorithm has been shown to outperform the SP algorithm for the Steiner tree problem. However, it is not necessary true when link cost is set based on LLR approach and fractional reward loss is used for performance evaluation. This results indicate the importance of using good link costs.

In conclusion, we believe that a good multicast routing technique should not only emphasize on a good design of routing algorithm, but also a good scheme for defining link costs and efficient implementation of the multicast routing algorithm.

7. Summary and future work

In this paper, two versions of the multicast routing problem, complete and partial, in multirate loss networks are studied. The network is assumed to handle multiple classes of multicast connections. To carry a multicast connection, a fixed amount of bandwidth is reserved on each link of the multicast tree. Each connection carried by the network is assumed to bring a certain amount of revenues and the objective of multicast routing algorithms is to minimize the *fractional reward loss* due to call blocking.

One of the contributions of this paper is to study the ways of defining link costs. In most of previous research on multicast routing, the link costs were assumed to be given. However, we believe that the link cost must be defined first before developing multicast routing algorithms. Therefore, in this paper, we propose two approaches, namely, MDP and LLR, for defining link costs. Our numerical results show that, for sparsely connected networks, algorithms that use MDP-based link costs yield significantly better performance than those that use the LLR approach. In other words, it may be more important to study how to define the link costs than to develop a complicated heuristic algorithm.

Several multicast routing algorithms are proposed for both fully connected networks and sparsely connected networks. For sparsely connected networks, we propose a shortest-path based algorithm for partial multicast problem and two heuristic algorithms, TMR and GA, for the complete multicast problem. For fully connected networks, two important heuristics from point-to-point routing in circuit-switched networks are used in the development of our multicast routing algorithms, namely multicast trees with smaller number of links are preferred and all candidate multicast trees consist of at most $|D|+1$ links where $|D|$ is the size of the destination set. These two heuristics are especially important for multicast routing algorithms for the complete version of the multicast problem. Since multicast trees with smaller number of links are preferred, our algorithms search for the multicast tree with $|D|$ links first, using the minimum spanning tree algorithm. Multicast trees with $|D|+1$ links are searched only when the first step failed. For well dimensioned, fully connected networks, the probability that a multicast tree can be found at first step should be very high. Therefore, with very high probability, a multicast tree can be found by executing the minimum spanning tree algorithm once.

From our simulation results, we observe that the algorithms designed for the complete multicast routing problem yield a much better performance than that of algorithms designed for the partial multicast problem. Intuitively, this result suggests that routing algorithms with global information, i.e., with the *a priori* knowledge of identities of all destination nodes, are able to achieve a better performance. For fully connected networks, we observe that defining link costs based on MDP or LLR has little effect on the performance of multicast routing algorithms. The reason is that the multicast trees considered are limited to at most $|D| + 1$ links. However, for sparsely connected networks, we find that algorithms that use MDP-based link costs yield significantly better performance than algorithms that use the LLR approach. Therefore, for general networks, it becomes very important to define the link costs correctly.

We are currently working on several possible extensions of our algorithms. In this paper, we have assumed that a multicast request is rejected if any one of its destinations cannot be connected. For some applications, this is indeed the case, for example, for a multi-party conference, if a decision can be made only if all parties are present, then a multicast connection is meaningful only if all parties can be connected. On the other hand, some other applications may be satisfied with connecting partial destination nodes. For example, an informal multi-party conference. In this case, the routing algorithm should do its best to connect as many of the destination nodes as possible, if not all the destination nodes can be connected. We are currently extending our algorithms to accommodate such applications.

References

[1] G.R. Ash, B.M. Blake and S.D. Schwartz, Integrated network routing and design, in: *Twelfth Internat. Teletraffic Congress*, Torino (June 1988) pp. 640–647.

[2] G.R. Ash, R.H. Cardwell and R. Murray, Design and optimization of networks with dynamic routing, Bell System Technical Journal 60 (1981) 1787–1820.

[3] W.H. Cameron, J. Regnier, P. Galloy and A.M. Savoie, Dynamic routing for intercity telephone networks, in: *Tenth Internat. Teletraffic Congress* (June 1983).

[4] H.-W. Chu and D.H.K. Tsang, Dynamic routing algorithms in VP-based ATM networks, in: *GLOBECOM '95* (December 1995) pp. 1364–1368.

[5] S.-P. Chung and K.W. Ross, Reduced load approximations for multirate loss networks, IEEE Transactions on Communications 41 (August 1993) 1222–1231.

[6] L. Davies, ed., *Genetic Algorithms and Simulated Annealing* (Morgan Kaufmann, Los Altos, CA, 1987).

[7] W.-Y. Do and R.-H. Hwang, Multicast routing based on genetic algorithms, in: *Proc. of Internat. Conf. on Networking and Multimedia* (December 1996) pp. 211–218.

[8] Draft recommendation I.371: Traffic control and congestion control in B-ISDN, ITU-T SG. XVIII, Geneva (June 1992).

[9] Z. Dziong et al., On adaptive call routing strategy for circuit switched networks – maximum reward approach, in: *Twelfth Internat. Teletraffic Congress*, Torino (June 1988) pp. 998–1005.

[10] Z. Dziong and L. Mason, Control of multi-service loss networks, in: *Proc. of the 28th Conf. on Decision and Control*, Tampa, FL (December 1989) pp. 1099–1104.

[11] Z. Dziong and L. Mason, An analysis of near optimal call admission and routing model for multi-service loss networks, in: *INFOCOM '92*, Florence, Italy (May 1992) pp. 141–152.

[12] Z. Dziong and L.G. Mason, Call admission and routing in multi-service loss networks, IEEE Transactions on Communications 42 (1994) 2011–2022.

[13] A.I. Elwalid and D. Mitra, Effective bandwidth of general Markovian traffic sources and admission control of high speed networks, IEEE/ACM Transactions on Networking 1 (June 1993) 329–343.

[14] R.J. Gibbens and P.J. Hunt, Effective bandwidths for the multi-type UAS channel, Queueing Systems 9 (1991) 17–28.

[15] A. Girard and G. Rosenberg, A unified framework for network design with generalized connections, in: *ITC '15* (June 1997) pp. 319–328.

[16] R. Guérin, H. Ahmadi and M. Naghshineh, Equivalent capacity and its application to bandwidth allocation in high-speed networks, IEEE Journal on Selected Areas in Communications 9 (September 1991) 968–981.

[17] S. Gupta, K. Ross and M.E. Zarki, Routing in virtual path-based ATM networks, in: *GLOBECOM '92* (December 1992) pp. 571–575.

[18] S. Hakimi, Steiner's problem in graphs and its implications, Networks 1 (1971) 113–133.

[19] R.-H. Hwang, Routing in high-speed networks, Technical Report 93-43, Department of Computer Science, University of Massachusetts at Amherst (1993).

[20] R.-H. Hwang, Adaptive multicast routing in single rate loss networks, in: *INFOCOM '95* (April 1995) pp. 571–578.

[21] R.-H. Hwang, Multicast routing in single rate loss networks, NSC Report No. NSC-84-2213-E-194-026, National Science Council, Taiwan, R.O.C. (July 1995).

[22] R.-H. Hwang, J.F. Kurose and D. Towsley, State-dependent routing for multirate loss networks, in: *GLOBECOM '92* (December 1992) pp. 565–570.

[23] R.-H. Hwang, J.F. Kurose and D. Towsley, MDP routing in ATM networks using the virtual path concept, in: *INFOCOM '94* (June 1994) pp. 1509–1518.

[24] R. Karp, Reducibility among combinatorial problems, in: *Complexity of Computer Computations* (Plenum, New York, 1972) pp. 85–103.

[25] F.P. Kelly, Routing in circuit switched networks: Optimization, shadow prices and decentralization, Advances in Applied Probability 20 (1988) 112–144.

[26] F.P. Kelly, Effective bandwidths at multi-type queues, Queueing Systems 9 (1991) 5–15.

[27] G. Kesidis, J. Walrand and C.-S. Chang, Effective bandwidths for multiclass Markov fluids and other ATM sources, IEEE/ACM Transactions on Networking (August 1993) 424–428.

[28] P.B. Key, Optimal control and trunk reservation in loss networks, Probability in the Engineering and Informational Sciences 4 (1990) 203–242.

[29] V.P. Kompella, J.C. Pasquale and G.C. Polyzos, Muticast routing for multimedia communication, IEEE/ACM Transactions on Networking 1 (June 1993) 286–292.

[30] K.R. Krishnan, Adaptive state-dependent traffic routing using on-line trunk-group measurements, in: *Thirteenth Internat. Teletraffic Congress*, Denmark (June 1991) pp. 407–411.

[31] K.R. Krishnan and T.J. Ott, State-dependent routing of telephone traffic: Theory and results, in: *Proc. of 25th CDC*, Athens, Greece (December 1986) pp. 2124–2128.

[32] P.M. Lin, B.J. Leon and C.R. Stewart, Analysis of circuit-switched networks employing originating-office control with spill-forward, IEEE Transactions on Communications 26 (June 1978) 754–765.

[33] A. Maunder and P.S. Min, Routing for multi-rate traffic with multiple qualities of service, in: *ICCCN '94* (June 1994) pp. 104–108.

[34] K. Mehlhorn, *Data Structures and Algorithms 2: Graph Algorithms and NP-Completeness* (Springer, New York, 1984).

[35] D. Mitra, R.J. Gibbens and B.D. Huang, Analysis and optimal design of aggregated least-busy-alternative routing on symmetric loss networks with trunk reservations, in: *Thirteenth Internat. Teletraffic Congress* (1991) pp. 477–481.

[36] V. Nguyen, On the optimality of trunk reservation in overflow processes, Probability in the Engineering and Informational Sciences 5 (1991) 369–390.

[37] T.J. Ott and K.R. Krishnan, State-dependent routing of telephone traffic and the use of separable routing schemes, in: *Eleventh Internat. Teletraffic Congress*, Kyoto, Japan (September 1985) pp. 867–872.

[38] R.C. Prim, Shortest connection networks and some generalizations, Bell System Technical Journal 36 (1957) 1389–1401.
[39] V.J. Rayward-Smith and A. Clare, On finding Steiner vertices, Networks 16 (1986) 283–294.
[40] M.I. Reiman, Optimal trunk reservation for a critically loaded link, in: *Thirteenth Internat. Teletraffic Congress* (June 1991) pp. 247–252.
[41] H. Takahashi and A. Matsuyama, An approximate solution for the Steiner problem in graphs, Math. Japonica 6 (1980) 573–577.
[42] H. Tode, Y. Sakai, M. Yamamoto, H. Okada and Y. Tezuka, Multicasting routing algorithm for nodal load balancing, in: *INFOCOM '92*, Florence, Italy (May 1992) pp. 2086–2095.
[43] B.M. Waxman, Routing of multipoint connections, IEEE Journal on Selected Areas in Communications 6 (December 1988) 1617–1622.
[44] B.M. Waxman, Performance evaluation of multipoint routing algorithms, in: *INFOCOM '93*, San Francisco, CA (April 1993) pp. 980–986.
[45] R.I. Wilkinson, Theories of toll traffic engineering in the USA, Bell System Technical Journal 40 (1956) 421–514.
[46] S.-C. Yang, Adaptive multicast routing in ATM networks, Master's thesis, Department of Computer Science and Information Engineering, National Chung Cheng University (June 1997).