Multicast Routing in Single Rate Loss Networks Based on Markov Decision Process

Ren-Hung Hwang

Department of Computer Science & Information Engineering
National Chung Cheng University
Chia-Yi, Taiwan, R.O.C.

Abstract

In this paper, we propose two multicast routing algorithms for single rate loss networks. Two versions of the multicast routing problem are studied: static and dynamic. In the static version of the multicast routing problem, the identities of all destination nodes is available to the multicast routing algorithm at once. Conversely, in the dynamic version of the multicast problem, the identities of the destination nodes is revealed to the routing algorithm one by one. Based on markov decision process (MDP), we developed two routing algorithms, one each version of the multicast routing problem. Furthermore, the algorithm for the static multicast routing problem is based on the minimum spanning tree (MST) concept while the algorithm for the dynamic version of the multicast problem is based on shortest path (SP) concept. Our simulation results indicate that the algorithm based on the MST concept outperforms the algorithm based on the SP concept.

1 Introduction

In circuit-switched networks, as well as future high-speed networks, reserving certain amount of resources, such as circuits, bandwidth, and buffer space, is required for setting up a call (connection). As a consequence, calls arriving to the network may be rejected (blocked) due to insufficient network resources. A network with calls requiring the same amount of resources is referred to as a single rate loss network. Examples of single rate loss networks include traditional telephony networks and homogeneous VP-based ATM networks [1, 2].

Since the early 1980's, considerable research has focused on the design of point-to-point *adaptive* routing algorithms for circuit-switched networks [3]. Recently, point-to-point adaptive routing algorithms also have been proposed for homogeneous VP-based ATM networks [1, 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 MDP approach, on the other hand, formulates the routing problem as a Markov decision process. Examples of LLR-based routing algorithms include AT&T's Trunk Status Map Routing (TSMR) [4] and Real-Time Network Routing (RTNR) [5], and Bell-Northern Research's Dynamically Controlled Routing (DCR) [6]. On the other hand, MDP-based routing algorithms include Bellcore's State-Dependent Routing (SDR) [7] and Forward-Looking Routing (FLR) [8], and routing algorithms based on INRS-Telecommunications' Markov Decision Process Decomposition (MDPD) approach [9–11].

Recently, many new applications require establishing multicast connections. For example, three-party calls in circuit-switched networks, conference calls in Integrated Service Digital Networks (ISDN), and multimedia, multiparty conferencing services in future high-speed networks all require multicast (point-to-multipoint) communication services. The purpose of this paper is to study adaptive multicast routing algorithms for single rate loss 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 [12] and was shown to be NP-complete [13]. Thus most previous researchers have focused on developing heuristic algorithms that take polynomial time and produce near optimal results (e.g., [14, 15]). 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, 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 [16] or there is a restriction on the number of packet copies per switch [17]. Of the previous work, only Waxman [18, 19] 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.

In this paper, we propose MDP-based multicast routing algorithms for single rate loss networks. Similar to the definition of [18], the multicast routing problems are classified into two categories: static and dynamic. In the static version of the multicast routing problem, the identities of all destination nodes is available to the multicast routing algorithm at once. Conversely, in the dynamic version of the multicast problem, the identities of the destination nodes is revealed to the routing algorithm one by one. Two algorithms are studied in this paper. The first algorithm, referred to as MDP_MST, is designed based on the minimum spanning tree concept and is suitable for the the static multicast routing problem. The other algorithm, referred to as MDP_SP, which is designed for the dynamic multicast routing problem, is based on the shortest path concept. The performance metric proposed in [20] , referred to as *fractional reward loss*, is used to evaluate these two multicast routing algorithms. The performance of MDP-based routing algorithms is also compared to two LLR-based algorithms

This research was supported in part by the National Science Council of Taiwan, R.O.C., under contract NSC-84-2213-E-194-026.

proposed in [20], referred to as LLR_SP and LLR_MST. Our simulation results show that the MDP_MST algorithm yields better performance than the MDP_SP algorithm. We also observe that MDP_SP outperforms LLR_SP slightly, as observed in the literature [21–23]. However, we find that LLR_MST yields very competitive performance as compare to the MDP_MST algorithm.

Multicast routing for multirate loss networks has been studied in [24]. Both LLR-based and MDP-based algorithms were studied for the point-to-point routing first. The authors then extend the point-to-point routing algorithms to multicast routing algorithms by treating a multicast call as a collection of multiple independent point-to-point calls. In their numerical results, the authors observed that such approach yields better performance than routing algorithms which try to build a Steiner Tree. This observation is contradictory to the results in [20]. In [20], we found that multicast routing algorithms which treat a multicast call as a collection of multiple independent point-to-point calls yields much worse performance than that of algorithms which construct a Steiner tree for the call. Intuitively, we believe that algorithms which try to find global optimal routing policy by finding a Steiner tree should outperform algorithms which look for local optimal solution by treating a multicast call as a collection of multiple independent point-to-point calls.

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 propose two MDP-based multicast routing algorithms: one for the static version of the multicast routing problem and the other one for the dynamic problem. Performance evaluation of these four multicast routing algorithms based on simulations is given in section 4. Section 5 concludes our study and discusses some of our future work.

2 Network Model and Problem Definition

We model a single rate loss network as a undirected graph $G = (V, E)$ with node set V and bidirectional link (edge) set E, and a function $Cap : E \rightarrow Z_0^+$ where $Cap(\ell)$ is the capacity of link ℓ, measured in units of bandwidth[1]. We assume that the network is fully connected or almost fully connected, as commonly assumed in circuit-switched networks and homogeneous VP-based ATM networks [1,2]. The link between node i, j is denoted by $e(i, j)$.

We define a connection request c to be a pair (s, D) where

- $s \in V$ is the source node of the connection.
- $D \subseteq V$ is the set of destination nodes to be connected.

Connections with the same connection request parameter, (s, D), are then aggregated into one class, $w = (s, D)$, with each class having the same bandwidth requirement. Let W be the set of all possible classes of connections. We assume that connections of class (s, D) arrive at the source node, s, according to a Poisson process with rate $\lambda_{(s,D)}$. The connection holding time for any connection is assumed to be exponentially distributed with mean of unity.

To set up a connection of class (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, $\{s\} \bigcup D \subseteq V_T$, and E_T is the set of links on the tree. Furthermore, a circuit on each link of the tree, i.e., $\forall \ell \in E_T$, is reserved for the connection. Depending on the routing algorithm, a connection may be either accepted or rejected. We assume that each connection of class (s, D) carried on the network produces $r_{(s,D)}$ units of revenue. From another point of view, the network losses $r_{(s,D)}$ units of revenue for each class (s, D) connection that is rejected.

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

$$\text{fractional reward loss} = \frac{\sum_{w \in W} r_w \lambda_w B_w}{\sum_{w \in W} r_w \lambda_w}$$

where B_w is the blocking probability of class w traffic. Minimizing the fractional reward loss is equivalent to maximize 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 (based on some routing rules using current state information). Otherwise, the connection is rejected (even if there is enough network resources to carry it).

Depending on whether the identities of all destination nodes is available to the routing algorithm at once, we can classified the multicast routing problem into two categories: static and dynamic. In the static version of the multicast routing problem, the identities of destination nodes is available to the routing algorithm at once. Most previous research on multicast routing in packet-switched networks has focused on the static version of the multicast routing problem and reduced it to the Steiner tree problem. However, setting up a multicast connection is very different in circuit-switched networks, ISDN, and B-ISDN. For example, according to Q.931 [25], the CCITT recommendation protocol for call setup in ISDN, as well as Q.93B [26], the ITU-TSS draft recommendation protocol for call setup in B-ISDN, a multicast connection is set up by first establishing a point-to-point connection between the source node and the first destination node[2]. After this set up is complete, additional destination nodes can be added sequentially. We refer to this version of the multicast routing problem as dynamic version as not all of the destination nodes are known at once, instead, destination nodes are added to the multicast tree one by one. A similar definition can also be found in [18]. In this paper, different multicast routing algorithms are proposed under both static and dynamic version of multicast routing problems and their performance is compared in our numerical results.

[1] For example, a unit of bandwidth of bandwidth is a circuit in circuit-switched networks. Hereafter, we will refer to the unit of bandwidth as a circuit throughout the paper.

[2] For example, consider the case of dialing a three-party call in current telephony system.

3 MDP approach

Although the routing problem in loss 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 [21, 9]. Thus, numerous studies, e.g., [21, 9], have proposed a link independence assumption and a 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, consequently, the computational complexity. Even with these two assumptions, the state space is still too large for multi-rate loss networks and/or multicast problems for any realistic network [23]. Therefore, in this section, we will focus on developing computational feasible MDP-based multicast algorithms. In the following, we first introduce these two assumptions. We then describe our formulation of the multicast routing problem as a decomposed Markov decision process based on these two assumptions. Based on the result of the decomposed Markov decision process model, we then propose two multicast routing algorithms; one for the dynamic multicast routing problem and the other for the static multicast routing problem.

The two commonly made assumptions are [21, 9, 23]:

- **The link independence assumption:**
  This assumption assumes that ([21, 9, 11, 27]):
  1. calls from any class w arrive to 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 [21]. The idea is to assume that the cost for 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 in the tree. Here, the cost can be interpreted can the expected revenue loss due to call loss during the holding time of the call.

Based on these two assumptions, we can decompose the original Markov decision process by forming a Markov decision process on each link. By the separability assumption, the cost for carrying a call on a multicast tree can be obtained by summing the individual cost on each link in the tree. When a multicast call arrives to the network, the near optimal routing policy can then be obtained by finding the multicast tree with minimum cost. The call will be rejected if the resulting cost is larger than the revenue for carrying the call. In the following, we first show how to form a Markov decision process on each link and obtain the cost function efficiently. We then show how to obtain the multicast tree with minimum cost.

3.1 Decomposed Markov Decision Process

When we formulate a Markov decision process on each link, we have two goals. First, in order to reduce the computational complexity, we need to reduce the state space of the Markov decision process. Especially, we would like to have as little classes of traffic as possible. Second, we want to derive simple expressions for computing the cost for adding a call on a link.

In order to reduce the number of traffic classes, calls carried on a link are classified according to their "link reward", instead of their request parameters. The link reward is defined differently under static and dynamic version of the multicast routing problem. For static version of the multicast routing problem, the link reward is defined as follows: when a call with request parameter (s, D) and revenue $r_{(s,D)}$ is carried on a multicast tree with n links, we assume the reward for each of these n links to carry the call is $r_{(s,D)}/n$ and is referred to as the link reward of the call [11]. On the other hand, under the dynamic version of the multicast routing problem, the reward of a multicast call is first decomposed into individual reward for connecting each destination. The individual reward is then evenly distributed to each link on the path that connects the destination to the multicast tree. Therefore, the link reward for a link on the path what connects the destination node to the multicast tree is $r_{(s,D)}/|D|/m$ where $|D|$ is the cardinality of the destination set D and m is the number of links on the path.

Let us focus on a single link ℓ in isolation. Assume that all calls carried on this link can be classified into K classes and let $\mathcal{K} = \{1, ..., K\}$ be the set of all classes. We also assume that calls of class k, with *link reward* r_k, arrive to the link according to a Poisson process with rate λ_k. Without loss of generality, assume that call classes are ordered according to their link rewards such that $r_1 \geq r_2 \geq ... \geq r_K$.

The Markov decision process on link ℓ is then formulated as follows. It has been shown that for systems with finite state and action spaces, there exists an optimal deterministic policy [28]. Thus we focus on the class of deterministic, stationary policies that are allowed to use state information, i.e., policies which take deterministic actions, that depend on the state but not the history of the system. Since all calls require the same amount of bandwidth and class k calls yield more revenues than class k' calls where $k \leq k' \leq K$, thus we further restrict our routing policies to be the set of stationary policies in which if the policy decides to block class k calls at state x, then it would also block any class k' calls. Let Π be the set of all such deterministic, stationary policies.

For a policy $\pi \in \Pi$, define $\pi : X \times \{1, ..., K\} \to \{0, 1\}$ where $X = \{0, 1, ..., Cap(\ell)\}$ is the set of all feasible states and $\pi(x, k)$ is the action taken by policy π when the link is in state x and a class k call arrives. If $\pi(x, k) = 0$, the incoming call is rejected. Otherwise, the call is accepted. Let $q(x, k, \pi(x, k))$ be the revenue loss rate when the link is in state x and action $\pi(x, k)$ has been chosen for an incoming class k call. Define

$$q(x, k, \pi(x, k)) = \begin{cases} r_k\lambda_k & \pi(x, k) = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Let us consider the infinite horizon problem with no reward discounting case only [29]. If policy π is completely ergodic (i.e., the process contains only one recurrent chain), then the expected revenue loss for policy π in the interval of time $(0, t)$ with x as its initial state, $V_x(t, \pi)$, is given by [30]:

$$V_x(t, \pi) = g(\pi)t + v(x, \pi)$$

where $g(\pi)$ is called the loss rate of policy π [21] (also, the "gain" of the policy in [30]) and $v(x, \pi)$ is the relative value of state x under policy π.

An optimal policy, π^*, is the policy such that $V_x(t, \pi^*) = \min_{\pi}\{V_x(t,\pi)\}$ for all x. That is, an optimal policy is the policy that minimizes the expected loss revenue in any time t with any possible initial state. According to Markov decision theory, an optimal policy can be found by a policy iteration procedure based on the following iteration cycle [30]:

- **Value Determination Step:** for a given policy during the ith iteration, $\pi^{(i)}$, solve the following set of equations

$$\begin{aligned}
g(\pi^{(i)}) &= 0 + \sum_{j=1}^{K} \lambda_j (v(1,\pi^{(i)}) - v(0,\pi^{(i)})),\\
g(\pi^{(i)}) &= 0 + \sum_{j=1}^{K} \lambda_j (v(x+1,\pi^{(i)}) - v(x,\pi^{(i)})) - x(v(x,\pi^{(i)}) - v(x-1,\pi^{(i)})),\\
&\qquad 0 < x < C_K^{\pi^{(i)}},\\
g(\pi^{(i)}) &= \sum_{j=k}^{K} r_j\lambda_j + \sum_{j=1}^{k-1} \lambda_j (v(x+1,\pi^{(i)}) - v(x,\pi^{(i)})) - x(v(x,\pi^{(i)}) - v(x-1,\pi^{(i)})),\\
&\qquad C_k^{\pi^{(i)}} \le x < C_{k-1}^{\pi^{(i)}},\ \ 2 \le k \le K,\\
g(\pi^{(i)}) &= \sum_{j=1}^{K} r_j\lambda_j - C(v(C,\pi^{(i)}) - v(C-1,\pi^{(i)})),
\end{aligned}$$

(where $C_k^{\pi^{(i)}}$ is the state at which policy $\pi^{(i)}$ will block class k' calls, for $k \le k' \le K$), for all relative values $v(x,\pi^{(i)})$ and policy loss rate $g(\pi^{(i)})$.
Solutions to the set of equations (1), in terms of $\mathcal{W}(x) = v(x+1,\pi^{(i)}) - v(x,\pi^{(i)})$, are given as follows. (For convenience, we omit $\pi^{(i)}$ in our notations.)

$$g = E(\bar\lambda, C)\left(\sum_{j=1}^{K} r_j\lambda_j + C\cdot f(C-1)\right), \quad (2)$$

$$\mathcal{W}(x) = \frac{g}{\bar\lambda_x E(\bar\lambda, x)} - f(x), \quad 0 \le x < C, \quad (3)$$

where $\bar\lambda_x = \sum_{j=1}^{k}\lambda_j$ if $C_k \le x < C_{k-1}$ for $2 \le k \le K$,

$$E(\bar\lambda, x) = \frac{\frac{1}{x!}\prod_{j=0}^{x-1}\bar\lambda_j}{\sum_{n=0}^{x}\frac{1}{n!}\prod_{j=0}^{n-1}\bar\lambda_j}, \quad (4)$$

and the function $f(x)$ is defined as follows:

$$f(x) = \begin{cases} 0 & \text{if } x < C_K \\ \dfrac{\sum_{j=k}^{K} r_j\lambda_j + x\cdot f(x-1)}{\sum_{j=1}^{k-1}\lambda_j} & \text{if } C_k \le x < C_{k-1},\ \ 2 \le k \le K \end{cases}$$

- **Policy Improvement Step:** find the policy, which will be used for the next iteration, that minimizes the right hand side of equation (1) for each state x using the relative values obtained from the previous step. Equivalently, the improved policy, $\pi^{(i+1)}$, will reject a class k call at state x if $\mathcal{W}(x) \ge r_k$. That is, $\mathcal{W}(x)$ is the cost for adding a call on the link. (Note that the cost for adding a call is independent of its class because every call requires the same amount of bandwidth and has identical call holding time distribution.) Therefore, we are able to compute this cost efficiently by simple expressions.

The iteration procedure iterates until the improved policy converges to the optimal policy, i.e., the policy obtained in the policy improvement step is equivalent to the previous iteration policy. The cost for adding a call, or equivalently reserving a circuit for a time period of call holding time, on link ℓ when the link is in state x_ℓ is given by $\mathcal{W}_\ell(x_\ell)$.

In order to compute the cost for adding a call, $\mathcal{W}_\ell(x_\ell)$, arrival rate of each class of traffic, λ_k, is required. As suggested in [9, 27], 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, $\tilde\lambda_k$, is the average number of class k connections carried per unit of mean connection holding time. Based on the exponential smoothing model, the predicted arrival rate of class k connections is given by

$$\lambda_k^{new} = (1-\alpha)\lambda_k^{old} + \alpha\cdot\frac{\tilde\lambda_k}{1-b_k},$$

where λ_k^{old} is the predicted arrival rate of the previous update, b_k 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 cost function, $\mathcal{W}_\ell(x_\ell)$, are also updated adaptively, according to the traffic loads.

3.2 The MDP-based Multicast Routing Algorithms

Based on the link cost functions obtained in the previous section, we now develop MDP-based multicast routing algorithms for both dynamic and static multicast routing problems.

3.2.1 MDP-based Dynamic Multicast Routing Algorithm: MDP_SP

The dynamic version of the multicast routing problem assumes that the identities of the destination nodes of a multicast connection request is presented to the routing algorithm one by one. Thus the routing algorithm needs to build a multicast tree with partial information. We further assume that the multicast tree is non-rearrangeable [18], i.e., routing algorithms are not allowed to modify the existing multicast tree except to add new nodes and links[3].

The routing algorithm we proposed, referred to as MDP_SP, is based on the shortest path with minimum path cost 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 where the cost of a path is the sum of link costs on the path and the link costs are computed as shown in the previous section. A detailed description of the MDP_SP algorithm is given as follows.

Let us consider the establishment of a multicast connection request with parameter (s, D) and reward $r_{(s,D)}$. In the MDP_SP algorithm, let T be the multicast tree to be build and $cost_T$ be cost of the multicast tree which is defined as the sum of link costs of all links on T. For each

[3] In this paper, we do not consider remove a destination node from a multicast tree dynamically.

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 MDP_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, $\mathcal{W}_\ell(x_\ell)$ where $\mathcal{W}_\ell(\cdot)$ is given in equation (3) and x_ℓ is the state of the link. If $\mathcal{W}_\ell(x_\ell) + cost_T < r_{(s,D)}$, then node d and link ℓ are added to the multicast tree T and $cost_T$ is updated by $cost_T + \mathcal{W}_\ell(x_\ell)$. 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 d to T, find the path with the minimum path cost where the path cost is defined as the sum of link costs of all links on the path. Let ℓ_1 and ℓ_2 be the links on the path with link costs $\mathcal{W}_{\ell_1}(x_{\ell_1})$ and $\mathcal{W}_{\ell_2}(x_{\ell_2})$ respectively and d' be the intermediate node on the path. If $\mathcal{W}_{\ell_1}(x_{\ell_1}) + \mathcal{W}_{\ell_2}(x_{\ell_2}) + cost_T < r_{(s,D)}$, then nodes d, d' and links ℓ_1, ℓ_2 are added to the multicast tree T and $cost_T$ is updated by $cost_T + \mathcal{W}_{\ell_1}(x_{\ell_1}) + \mathcal{W}_{\ell_2}(x_{\ell_2})$. 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. If the multicast request is not blocked, get the next ADD_PARTY request and go to step 2. If there is no additional ADD_PARTY request, then the multicast connection has been successfully set up.

Step 3 in the MDP_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 telephony networks, we suggest to limit the length of alternate paths to be at most two.

3.2.2 MDP-based Static Multicast Routing Algorithm: MDP_MST

The static version of the multicast routing problem assumes that the routing algorithm is able to know all the destination nodes before building the multicast tree. Although many heuristic algorithms have been proposed in the literature, none of the existing algorithms is designed for the loss networks. In the following, we propose a new approach toward static multicast routing problem based on the MDP concept and minimum spanning tree (MST) algorithm. The algorithm is referred to as the MDP_MST algorithm.

Recall that a single rate loss network is modeled as a undirected graph $G = (V, E)$ in section 2. At the beginning of this section, we also have derived a state-dependent cost function, $\mathcal{W}_\ell(x_\ell)$, for computing the cost of adding a call on link ℓ when the link is in state x_ℓ. In the MDP_MST algorithm, when a minimum cost tree is constructed, the cost of a tree is defined as the sum of link costs of all links on the tree and the link costs are computed by the link cost function $\mathcal{W}_\ell(x_\ell)$.

The basic idea of the MDP_MST algorithm is to find the minimum cost tree with $|D|$ links first where $|D|$ is the cardinality of the destination set. If the cost of the tree is larger than the reward of the multicast connection, then MDP_MST tries to find the minimum cost tree with $|D|+1$ links. Details of the algorithm is given as follows. Upon receiving a multicast connection request with parameter (s, D) and reward $r_{(s,D)}$, following steps are performed by the MDP_MST algorithm:

1. Construct a subgraph $G' = (\{s\} \bigcup D, E')$ from G where E' is a subset of E and $\ell \in E'$ if and only if the end-nodes of ℓ are in the set $\{s\} \bigcup D$.
2. Construct the minimum spanning tree $T = (\{s\} \bigcup D, E_T)$ from G'. Let $cost_T$ be the cost of tree T.
3. If $cost_T < r_{(s,D)}$ then a circuit is reserved on each link in 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., $\forall v \in V - D$, construct a subgraph $G_v = (\{s, v\} \bigcup D, E_v)$ from G where 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 $\{s, v\} \bigcup D$.
5. From each graph G_v, construct the minimum cost tree $T_v = (\{s, v\} + D, E_{T_v})$. Let $cost_{T_v}$ be the cost of tree T_v.
6. Among those minimum cost trees, $T_v, \forall v \in V - D$, find the tree, referred to as T_*, with the minimum cost, i.e., $cost_{T_*} \leq cost_{T_v}, \forall v \in V - D$.
7. If $cost_{T_*} < r_{(s,D)}$ then a circuit is reserved on each link in the multicast tree T_* and the multicast connection has been successfully set up. Otherwise, the multicast request is blocked.

There exists many efficient algorithm for finding minimum spanning tree in the literature. In our numerical results, Prim's minimum spanning tree algorithm [31] is adopted.

As in the MDP_SP algorithm, the MDP_MST 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. 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 time complexity of the MDP_MST algorithm significantly.

4 Numerical Results

In this section, the performance of the two MDP-based routing algorithms and the two LLR-based routing algorithms proposed in [20] is studied through simulations on a 10-node symmetric network as well as a 10-node nonsymmetric network. In the 10-node symmetric network, each link is assumed to have 120 circuits. On the other hand, each link may have different link capacity in the 10-node nonsymmetric network. The detailed link capacity configuration for the 10-node nonsymmetric network is given in Table 1.

The performance of the four routing algorithms under the two network configurations is compared in Figure 1 and Figure 2. In this study, we assume that each class of multicast connections, $w = (s, D)$, has the same arrival rate

Table 1: Link capacity configuration in the 10-node nonsymmetric network

Link ID	Capacity	Link ID	Capacity	Link ID	Capacity
1-9	150	10-18	60	19-27	120
28-36	30	37-45	90		

and the reward of a multicast connection is set to $|D|$ units of revenue. In these figures/tables, the fractional reward losses yielded by the four routing algorithms are compared under different traffic loads. The vertical lines about each point in these figures indicate the 95 percent confidence interval. The 95 percent confidence intervals are also given in the tables. Each simulation point in these figures/tables 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 Figure 1–2, we can observe that the algorithms based on the minimum spanning tree (MST) approach outperforms the algorithms based on the shortest path (SP) approach. Clearly, with the identities of all destination nodes available at once, MST-based algorithms is able to construct a more efficient multicast tree for a multicast connection. Intuitively, this result suggests that the more information the routing algorithms have, the better performance can be achieved.

We are surprisingly to observe that the LLR_MST algorithm yields very competitive performance as compare to the MDP_MST algorithm under both symmetric and nonsymmetric networks. In the past, researchers have found that MDP-based routing algorithms yield better performance than LLR-based algorithms; especially under nonsymmetric networks [21–23]. In Figure 1 and Figure 2, we do observe that MDP_SP slightly outperforms LLR_SP. However, we find that LLR_MST yields almost the same performance as the MDP_MST algorithm. Our conjecture for such observation is that near optimal trunk reservation levels adopted in the LLR_SP and LLR_MST are also obtained based Markov decision theory [20]. Therefore, these two LLR-based routing algorithms have been tuned to yield near optimal performance.

5 Summary and Future Work

In this paper, the multicast routing problem in single rate loss networks was studied. We first classified the multicast routing problems into two categories: static and dynamic. Two algorithms based on the MDP approach were proposed. The performance metric, *fractional reward loss*, proposed in [20] is proposed for evaluating these four routing algorithms.

Performance of these two routing algorithms was studied on both symmetric and nonsymmetric networks. The performance is also compared to that of the two LLR-based multicast routing algorithms proposed in [20]. Our simulation results show that the the algorithms designed for the static multicast routing problem yield much better performance than that of algorithms designed for the dynamic problem. Intuitively, this result suggests that routing algorithms with global information, i.e., with the priori knowledge of identities of all destination nodes, are able to achieve better performance. We also found that, based on the minimum spanning tree approach, LLR-based algorithms yield very competitive performance as compare to the MDP-based algorithms.

In the future, we are interested in extending the work to ATM networks. In ATM networks, the Virtual Path (VP) concept is proposed for facilitating traffic control and resource management [32]. By reserving capacity to VP's, a fully connected logical VP network can be formed with VP endpoints as nodes and VP's as links [2]. If multiple classes of traffic are supported on a VP, the VP network can be viewed as a multirate loss network [23]. Thus, we are investigating multicast routing algorithms for multirate loss networks.

References

[1] S. Gupta, K. Ross, and M. E. Zarki, "Routing in Virtual Path Based ATM Networks," *GLOBECOM'92*, pp. 571–575, December 1992.

[2] R.-H. Hwang, J. F. Kurose, and D. Towsley, "MDP Routing in ATM Networks Using the Virtual Path Concept," *INFOCOM'94*, pp. 1509–1518, June 1994.

[3] G. R. Ash, R. H. Cardwell, and R. Murray, "Design and Optimization of Networks with Dynamic Routing," *Bell System Technical Journal*, vol. 60, pp. 1787–1820, 1981.

[4] G. R. Ash, "Use of a Trunk Status Map for Real-Time DNHR," in *Eleventh International Teletraffic Congress*, (Kyoto, Japan), September 1985.

[5] G. R. Ash, J. S. Chen, A. E. Frey, and B. D. Huang, "Real Time Network Routing in a Dynamic Class-of-Service Network," in *13th International Teletraffic Congress*, pp. 187–194, 1991.

[6] J. Regnier, P. Blondeau, and W. H. Cameron, "Grade of Service of a Dynamic Call Routing System," in *Tenth International Teletraffic Congress*, June 1983.

[7] K. R. Krishnan, "Markov Decision Algorithms for Dynamic Routing," *IEEE Communications Magazine*, vol. 28, pp. 66–69, October 1990.

[8] K. R. Krishnan and T. J. Ott, "Forward-Looking Routing: A New State-Dependent Routing Scheme," in *Twelfth International Teletraffic Congress*, (Torino), June 1988.

[9] Z. Dziong *et al.*, "On Adaptive Call Routing Strategy for Circuit Switched Networks - Maximum Reward Approach," in *Twelfth International Teletraffic Congress*, (Torino), June 1988.

[10] Z. Dziong, K.-Q. Liao, L. Mason, and N. Tetreault, "Bandwidth Management in ATM Networks," in *13th International Teletraffic Congress*, (Denmark), June 1991.

[11] Z. Dziong and L. Mason, "Control of Multi-Service Loss Networks," in *Proc. of the 28th Conference on Decision and Control*, (Tampa, Florida), December 1989.

[12] S. Hakimi, "Steiner's Problem in Graphs and its Implications," *Networks*, vol. 1, pp. 113–133, 1971.

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

[14] V. J. Rayward-Smith and A. Clare, "On Finding Steiner Vertices," *Networks*, vol. 16, pp. 283–294, 1986.

[15] H. Takahashi and A. Matsuyama, "An Approximate Solution for the Steiner Problem in Graphs," *Math. Japonica*, vol. 6, pp. 573–577, 1980.

[16] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, "Muticast Routing for Multimedia Communication," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 286–292, June 1993.

[17] H. Tode, Y. Sakai, M. Yamamoto, H. Okada, and Y. Tezuka, "Multicasting Routing Algorithm for Nodal Load Balancing," in *INFOCOM'92*, (Florence, Italy), pp. 2086–2095, May 1992.

[18] B. M. Waxman, "Routing of Multipoint Connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 1617–1622, Dec. 1988.

[19] B. M. Waxman, "Performance Evaluation of Multipoint Routing Algorithms," in *INFOCOM'93*, (San Francisco), pp. 980–986, April 1993.

[20] R.-H. Hwang, "Adaptive Multicast Routing in Single Rate Loss Networks," *INFOCOM'95*, pp. 571–578, April 1995.

[21] T. J. Ott and K. R. Krishnan, "State Dependent Routing of Telephone Traffic and the Use of Separable Routing Schemes," in *Eleventh International Teletraffic Congress*, (Kyoto, Japan), September 1985.

[22] 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.

[23] R.-H. Hwang, J. F. Kurose, and D. Towsley, "State Dependent Routing for Multirate Loss Networks," *GLOBECOM'92*, pp. 565–570, December 1992.

[24] A. Maunder and P. S. Min, "Analysis and Development of Routing Schemes for Multi-rate, Multi-point Traffic," *28th annual conference on Information Sciences and System*, March 1994.

[25] ITU-T Recommendation I.451:, "ISDN user-network interface layer 3 specification for basic call control, published in the Q-series as Recommendation Q.931." Geneva, 1989.

[26] ITU-T draft Recommendation Q.93B:, "B-ISDN user-network interface layer 3 specification for basic call/bearer control," May 1993.

[27] K. R. Krishnan, "Adaptive State-Dependent Traffic Routing Using On-Line Trunk-Group Measurements," in *13th International Teletraffic Congress*, (Denmark), June 1991.

[28] V. E. Beneš, "Programming and Control Problems Arising from Optimal Routing in Telephone Networks," *Bell Systems Technical Journal*, vol. 45, pp. 1373–1439, 1966.

[29] P. B. Key, "Optimal Control and Trunk Reservation in Loss Networks," *Probability in the Engineering and Informational Sciences*, vol. 4, pp. 203–242, 1990.

[30] R. A. Howard, *Dynamic Programming and Markov Processes*. John Wiley & Sons, Inc., 1960.

[31] R. C. Prim, "Shortest Connection Networks and some Generalizations," *Bell System Technical Journal*, vol. 36, pp. 1389–1401, 1957.

[32] ITU-T SG. XVIII, "Draft Recommendation I.311: B-ISDN General Network Aspects." Geneva, June 1990.

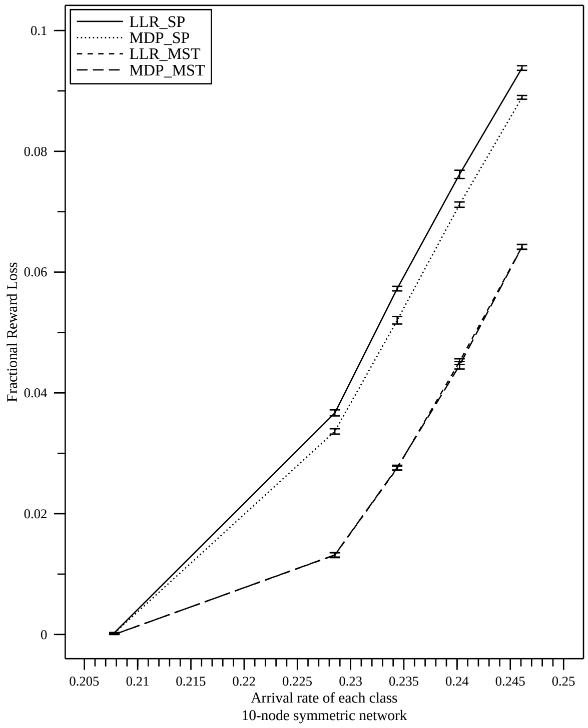

Figure 1: Performance comparison of the proposed four multicast routing algorithms under the 10-node symmetric network

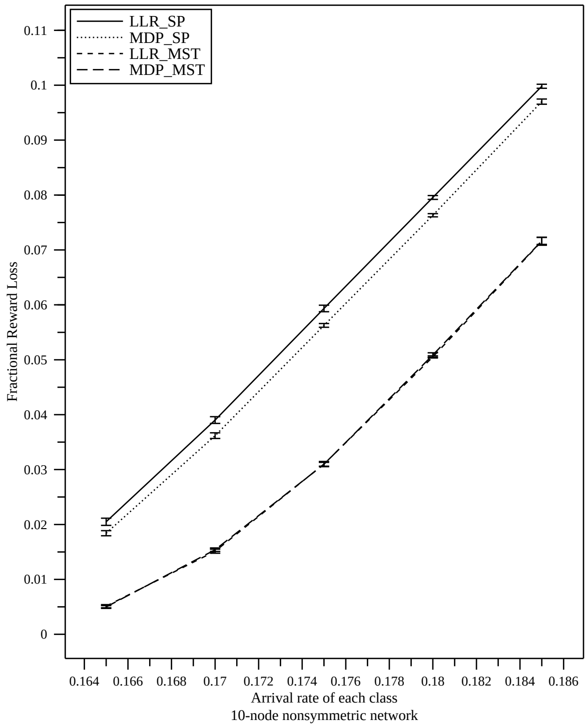

Figure 2: Performance comparison of the proposed four multicast routing algorithms under the 10-node nonsymmetric network