Adaptive Multicast Routing in Single Rate Loss Networks

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 a new approach, based on the Maximum Free Circuit Routing *(MFCR) concept, toward multicast routing in single rate loss networks. Two versions of the multicast routing problems 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. We studied three MFCR-based multicast routing algorithms: one for the static multicast problem and two for the dynamic multicast problem. We also propose a new performance metric, referred to as* fractional reward loss, *for evaluating these three multicast routing algorithms. The performance of these three multicast routing algorithms are evaluated through simulations.*

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]. Among these adaptive routing algorithms, algorithms based on the *Maximum Free Circuit Routing* (MFCR) concept have been shown to be very simple and efficient. Based on the fully connected topology assumption, the MFCR-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 path that has the maximum free circuits, is then tried. Examples of MFCR-based routing algorithms include AT&T's Real-Time Network Routing (RTNR) [4], and Bell-Northern Research's Dynamically Controlled Routing (DCR) [5].

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 [6] and was shown to be NP-complete [7]. Thus most previous researchers have focused on developing heuristic algorithms that take polynomial time and produce near optimal results (e.g., [8,9]). 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 [10] or there is a restriction on the number of packet copies per switch [11]. Of the previous work, only Waxman [12, 13] 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 a new approach toward multicast routing in single rate loss networks. Similar to the definition of [12], 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. Based on the MFCR concept, two algorithms, referred to as IND and SP, are designed for the dynamic multicast routing problem and one algorithm, referred to as MST, is proposed for the static multicast routing problem. In these algorithms, instead of finding a minimum cost multicast tree, a multicast tree with *maximum free circuits* is preferred. Furthermore, a multicast connection will be blocked, even the network has enough resources to carry the connection, if it is expected to block more future calls. A new performance metric, referred to as *fractional reward loss*, is proposed to evaluate these three multicast routing algorithms. Our simulation results show that the MST algorithm yields the best performance while the SP algorithm outperforms the IND algorithm.

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 three MFCR-based multicast routing algorithms. Performance evaluation of these three multicast routing algorithms based on simulations is given in section 4. Section 5 concludes our

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

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 $Cap(e)$ is the capacity of link e, measured in units of bandwidth[1]. We assume that the network is fully connected, i.e., $Cap(e) > 0, \ \forall e$, as commonly assumed in circuit-switched networks and homogeneous VP-based ATM networks [1, 2].

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., $\forall e \in E_T$, is reserved for the connection. [2] 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.

Depending on whether the identities of all destination nodes is available to the routing algorithm at once, we can classify 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 [14], the CCITT recommendation protocol for call setup in ISDN, as well as Q.93B [15], the ITU-T 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[3]. 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 [12]. In the next section, different multicast routing algorithms are proposed under both static and dynamic version of multicast routing problems and their performance compared in our numerical results.

We propose a new performance metric, referred to as *fractional reward loss* [16], for evaluating multicast routing algorithms. In the past, performance of multicast routing algorithms has been evaluated by comparing the costs of the multicast trees found by the multicast routing algorithms to the that of the "optimal solution", i.e., the minimum cost of all possible multicast trees. However, without carefully defining the cost function of each link, we hardly can gain much insight from such comparison. Thus we define a new performance metric for evaluating multicast routing algorithms for loss networks. 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).

3 Multicast Routing Algorithms

Three multicast routing algorithms will be studied in this section. The first two algorithms are developed under the dynamic version of the multicast routing problem. The third algorithm is designed for the static multicast routing problem.

3.1 Algorithms for Dynamic Multicast Routing Problem

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 [12], i.e., routing algorithms are not allowed to modify the existing multicast tree except to add new nodes and links[4]. Based on these constraints, we developed two multicast routing algorithms: IND and SP.

3.1.1 The IND Routing Algorithm

The IND algorithm simply finds a route for each destination independently. The advantage of this algorithm is that it

[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 a point-to-point connection, T becomes a path between the source node and the destination.

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

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

requires almost no change of current point-to-point routing software to be able to route multicast connections. However, as we will see in our numerical results, it yields very high fractional reward loss as compared to other algorithms. To connect a destination node to the source node, following steps are performed: [5]

1. The IND algorithm first tries to connect the source node to the destination node via the direct link.
2. If there is no free circuits available on the direct link, the destination node is connected through an alternate path with maximum free circuits.
3. If there is no alternate path with free circuits to connect the source node to the destination node, the connection is blocked and all circuits that have been reserved are released.

In the IND algorithm, the alternate paths between any source node s and destination node d are the paths connecting these two nodes. In this paper, we limit all alternate paths to have at most two links. The number of free circuits of an alternate path is defined as the minimum number of free circuits over the path's links. The number of free circuits of a link on an alternate path is given by $Cap(e) - x_e - t(e)$ where x_e is the number of busy circuits on the link and $t(e)$ is the trunk reservation level of the link. Note that, because each destination node is connected independently, a link may be added more than once to multicast tree. In such case, the algorithm simply reserves multiple circuits on this link.

Previous studies on point-to-point routing in circuit-switched networks have shown that the trunk reservation level affects network performance significantly [17, 18], especially when traffic load is very high. The optimal trunk reservation problem in single rate loss networks has been well studied in the literature [19–21]. In the appendix, we solve the optimal trunk reservation problem for our MFCR-based multicast routing algorithms based on the approach proposed in [19].

3.1.2 The SP Routing Algorithm

In this section, we propose a more complicated, but more efficient, multicast routing algorithm based on the shortest path concept where the measurement of the length of a path is the number of links on the path. The basic idea is to connect a destination node to a partially-formed multicast tree by choosing the shortest path with maximum free circuits from the node to the partially-formed tree. Figure 1 gives the detailed description for the SP algorithm.

The inner *for loop* iterates on the length of paths that connects a destination node d to the existing multicast tree T so that shorter paths will be tried first. For paths with the same length, a path with maximum free circuits is preferred (tie is broken randomly). Free circuits of a path is defined similarly as in the IND algorithm except the number of free circuits of a link on a path of length i is define by $Cap(e) - x_e - t(e, i)$ where $t(e, i)$ is the trunk reservation level for path of length i. The trunk reservation level is zero for direct-link path, i.e., $t(e, 1) = 0$, and is an increasing function in i. As we limit the length of an alternate path to be at most two in the IND algorithm, we also limit the length of a path adding one of the destination nodes to the existing multicast tree to be at most 2, i.e., $\ell = 2$, in this paper. The choice of $t(e, 2)$ is further discussed in section 4.

```
Remark:  given a connection request (s, D)
T := {{s}, ∅}
While D ≠ ∅
    remove d ∈ D
    if d already in T
        continue /* skip the rest of the while loop */
    end if
    For i = 1 to ℓ do
        let p(T, d) be the path connecting d to T
            with maximum free circuits among all
            paths of length i connecting d to T
        if the number of free circuits on path
            p(T, d) > 0
            T := T + p(T, d)
            break   /* leaves the for loop */
        end if
    end for
    if i > ℓ
        block the connection
        free all circuits have been reserved
        abort the call setup procedure
    end if
end While
```

Figure 1: The SP routing algorithm.

3.2 Algorithms for Static Multicast Routing Problem

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 MFCR concept.

3.2.1 The MST Routing Algorithm

The MST routing algorithm is based on the minimum spanning tree algorithm. As the direct link is always tried first in point-to-point routing, for a connection request (s, D), the MST routing algorithm first tries to build a multicast tree with exactly $|D|$ edges. Equivalently, the MST algorithm tries to find a tree $T = (\{s\} \bigcup D, E_T)$ within a subgraph $G' = (\{s\} \bigcup D, E')$, where E' is the set of links that connects any two nodes in $\{s\} \bigcup D$, such that $cost(T) = \sum_{e \in E_T} cost(e)$ is minimized. The cost function of a link, $cost(e)$, is defined as the negative value of the number of free circuits, i.e., the current residual capacity, of the link. An additional constraint needed on finding the minimum cost tree T is that a link can be added to the tree only if there exists at least one free circuit on the link. Intuitively, finding the minimum cost tree is equivalent to find a multicast tree with maximum number of free circuits. In our implementation, Prim's minimum spanning tree algorithm [22] with modification for considering the capacity

[5] The order of the identities of the destination nodes presented to the routing algorithm is specified by each connection request.

constraint is used to find the minimum cost tree T.

If the construction of the multicast tree with exactly D edges is not successful (due to capacity constraint), the MST algorithm then tries to build the multicast tree with exactly $D+1$ edges. The cost function of a link is defined as above except that the free circuits of a link is defined as $Cap(e) - x_e - t(e, |D|)$ where $t(e, |D|)$ is the trunk reservation level for connection requests with $|D|$ destinations. If the construction still fails, the connection is rejected. The MST algorithm is given in Figure 2.

```
Remark:  Given a connection request (s, D)
Construct a fully connected subgraph
    G' = ({s} + D, E')
Construct the minimum cost tree T = ({s} + D, E_T)
    with capacity constraint
if the construction is successful
    done
else
    let T_* := (∅, ∅)
    for each v ∈ V − D do
        Construct a fully connected subgraph
            G_v = ({s, v} + D, E_v)
        Construct the minimum cost tree
            T_v = ({s, v} + D, E_{T_v}) with
            capacity constraint
        if T_* == (∅, ∅)
            T_* := T_v
        else if cost(T_v) < cost(T_*)
            T_* := T_v
        end if
    end for
    if T_* == (∅, ∅)
        block the connection
        abort the call setup procedure
    end if
end if
```

Figure 2: The MST routing algorithm.

4 Numerical Results

In this section, the performance of the three routing algorithms is studied through simulations on a 7-node fully connected symmetric network under variable traffic conditions. In this study, we only consider point-to-point and point-to-two-point (e.g., the three-party calls in telephony networks) connections. Three cases of traffic patterns are studied. In case 1, the point-to-point traffic load is nine times of that of point-to-two-point traffic[6]. In case 2, the point-to-point traffic load is two times of the point-to-two-point traffic load. Finally, in case 3, there is no point-to-point traffic; the point-to-two-point traffic contributes all the network traffic. The reward parameter is set to unity for all point-to-point connections and two units of revenue for point-to-two-point connections. Trunk reservation levels for each algorithm under different traffic load are obtained based on the algorithm presented in the appendix.

[6] Note that, in real networks, we expect point-to-point connections will still be the major traffic in future high-speed networks with multimedia applications.

The performances of the three routing algorithms under the three different traffic configurations are compared in Figure 3–5. In these figures, the fractional reward losses yielded by the three routing algorithms are plotted under different traffic loads. The vertical lines about each point indicate the 95 percent confidence interval. Each simulation point 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 3–5, as we expected, we can observe that the MST algorithm outperforms the rest two algorithms while the SP algorithm outperforms the IND algorithm. Clearly, with the identities of all destination nodes available at once, the MST algorithm is able to construct a more efficient multicast tree for a multicast connection. (Note that the routing rule for point-to-point connections is the same for these three multicast routing algorithms.) Therefore, as we increase the traffic load of multicast connections (from configuration 1 to configuration 3), the difference among the performance of these three routing algorithm also increases. This seems to suggest that multicast routing algorithms based on the shortest path concept (e.g., [11, 13]) may yield much worse performance than MST-based algorithms. On the other hand, practical use of MST-based algorithms in ISDN or B-ISDN is questionable as the multicast connection establishment procedure defined in Q.931 and Q.93B requires setting up the destination nodes sequentially.

5 Summary

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. Based on the maximum free circuits routing concept, two algorithms were proposed for the dynamic multicast routing problems and one algorithm is designed for the static multicast routing problems. A new performance metric, *fractional reward loss*, is proposed for evaluating these three routing algorithms.

Performance of these three routing algorithms was studied on a seven-node, fully connected, symmetric, network. Our simulation results show that the the algorithm designed for the static multicast routing problems yields the best performance.

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, 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.

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

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

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

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

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

[10] 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.

[11] 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.

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

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

[14] 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.

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

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

[17] J. H. Weber, "A Simulation Study of Routing and Control in Communications Networks," *Bell System Technical Journal*, vol. 43, pp. 2639–2676, 1964.

[18] J. M. Akinpelu, "The Overload Performance of Engineered Networks with Nonhierarchical and Hierarchical Routing," in *Tenth International Teletraffic Congress*, June 1983.

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

[20] V. Nguyen, "On the Optimality of Trunk Reservation in Overflow Processes," *Probability in the Engineering and Informational Sciences*, vol. 5, pp. 369–390, 1991.

[21] M. I. Reiman, "Optimal Trunk Reservation for a Critically Loaded Link," in *13th International Teletraffic Congress*, June 1991.

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

[23] 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.

[24] B. L. Miller, "A Queueing Reward System with Several Customer Classes," *Man. Sci.*, vol. 16, pp. 234–245, 1969.

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

[26] 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.

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

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

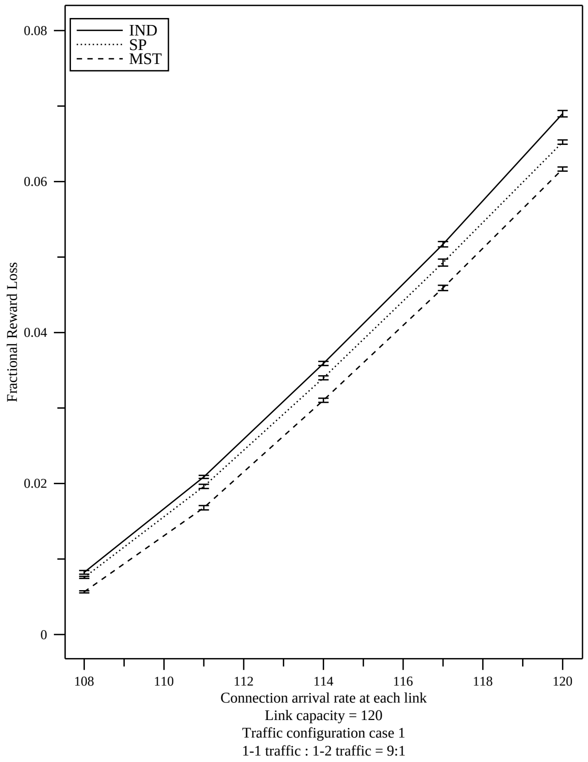

Figure 3: Performance of the three routing algorithms under traffic configuration 1.

Appendix

A Near Optimal Trunk Reservation Level

To avoid network performance degradation under high traffic load, a trunk reservation concept is adopted in MFCR-based routing algorithms [18]. In MFCR-based 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 high traffic load, such 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 MFCR-based multicast routing algorithms.

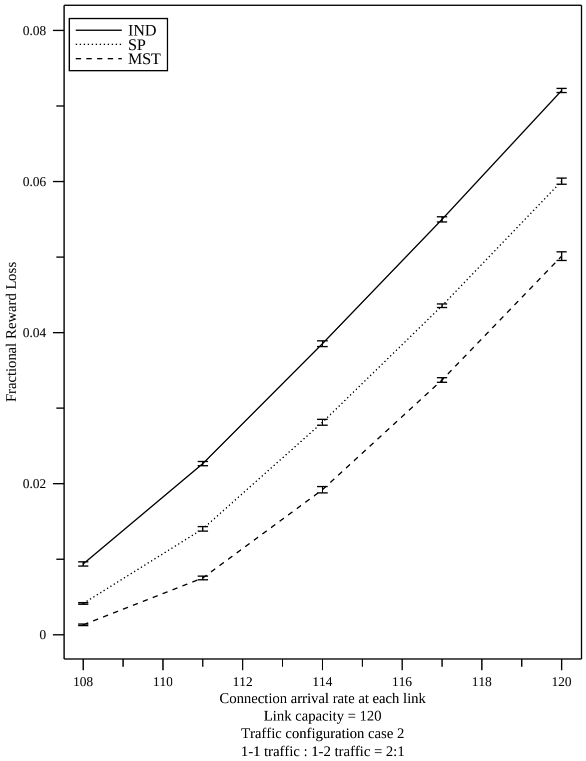

Figure 4: Performance of the three routing algorithms under traffic configuration 2.

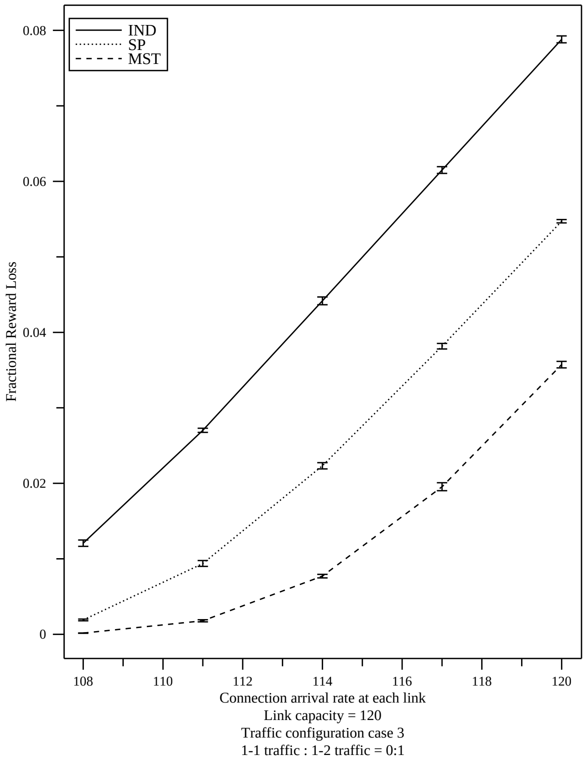

Figure 5: Performance of the three routing algorithms under traffic configuration 3.

The optimal trunk reservation problem in single rate loss networks has been well studied in the literature [19–21]. Most of previous research has focused on the optimal trunk reservation problem for networks with point-to-point routing only and limiting the length of alternate paths to be at most two. In this paper, we solve the optimal trunk reservation problem based on Markov decision theory, as proposed in [19]. However, we formulate the Markov decision process slightly different from that of [19] and, thus, obtain a different set of expressions for computing the optimal trunk reservation levels.

A.1 A General Model

Let us define the trunk reservation levels first. A number of trunk reservation levels $t_{e,k}$, $0 \leq t_{e,k} \leq Cap(e)$, is defined for each link e such that a type k connection can only use free circuits of link e if there are more than $t_{e,k}$ free circuits available on link e. How connections offered to a link are classified into different types depends on the underlying multicast routing algorithm and is discussed in the next section. Our objective to choose a set of trunk reservation levels for each link such that the fractional reward loss of the network can be minimized. In order to make the problem tractable, following assumptions are made.

1. A connection carried on a multicast tree with n links is assumed to behave like n independent connections, i.e., the link holding times for the connection on each of the n single links are statistically independent [19].
2. The reward for carrying a connection, r, on a multicast tree with n links is evenly distributed to each link, i.e., each link "earns" r/n units of revenue for carrying the connection [23, 16].
3. Connection requests of type k, with *link reward* r_k, arrive to any link e according to a Poisson process with rate $\lambda_{e,k}$.

Based on these assumptions, the optimal trunk reservation problem is formulated as a Markov decision process on each link. Let us focus on link e in isolation and omit the link identifier, e, in our notations. Let $\mathcal{K} = \{1, ..., K\}$ be the set of types of connections offered to this link. Let λ_k be the arrival rate of type k connections and r_k is the link reward parameter (see assumption 2). Without loss of generality, assume that the reward parameters are normalized by the maximum reward parameter and $1 = r_1 \geq r_2 \geq ... \geq r_K$. In the following, the optimal trunk reservation problem is formulated as a Markov decision process.

Focusing on link e still, let π be a trunk reservation policy which will block type k connections if the number of free circuits on the link is less than t_k^π. That is, define $\pi : X \times \{1, ..., K\} \rightarrow \{0, 1\}$ where $X = \{1, ..., Cap(e)\}$ is the set of all feasible states (link occupancy) of link e and $\pi(x, k)$ is the action taken by policy π for an incoming type k connection when the link occupancy is x. If $\pi(x, k) = 0$, the incoming connection is rejected. Otherwise, the connection is accepted. Since π is a trunk reservation policy, $\pi(x, k)$ is given by

$$\pi(x, k) = \begin{cases} 1 & \text{if } x < Cap(e) - t_k^\pi, \\ 0 & \text{otherwise.} \end{cases}$$

It has been shown that, for two types of traffic, the optimal policy (to be defined later) sets $t_1 = 0$ and $t_2 \geq 0$ [24].

Extending this result to more than two types of traffic, we are only interested in policies with trunk reservation levels $0 = t_1 \le t_2 \le ... \le t_K$.

Let $q(x, k, \pi(x, k))$ be the revenue loss rate when the Markov decision process is in state x and action $\pi(x, k)$ has been chosen for an incoming type k connection. Define

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

Now we can define the expected revenue loss for policy π. We only consider the infinite horizon problem with no reward discounting case [19]. Since all trunk reservation policies are stationary, i.e., they take deterministic actions which are independent of time, and completely ergodic, i.e., the resulting Markov decision processes contain only one recurrent chain, the expected revenue loss for a policy π, $V_x(t, \pi)$, is given by [25]:

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

where $g(\pi)$ is called the loss rate of policy π [26] (also, the "gain" of the policy in [25]) 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 [25]:

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

$$\begin{aligned}
g(\pi^{(i)}) &= 0 + \sum_{j=1}^{K}\lambda_j(v(1,\pi^{(i)}) - v(0,\pi^{(i)})),\\
g(\pi^{(i)}) &= 0 + \sum_{j=1}^{K}\lambda_j(v(x+1,\pi^{(i)}) - v(x,\pi^{(i)}))\\
&\quad -i(v(x,\pi^{(i)}) - v(x-1,\pi^{(i)})),\\
&\quad 0 < x < C - t_K^{\pi^{(i)}},\\
g(\pi^{(i)}) &= \sum_{j=k}^{K} r_j\lambda_j + \sum_{j=1}^{k-1}\lambda_j(v(x+1,\pi^{(i)}) - v(x,\pi^{(i)}))\\
&\quad -i(v(x,\pi^{(i)}) - v(x-1,\pi^{(i)})),\\
&\quad C - t_k^{\pi^{(i)}} \le x < C - t_{k-1}^{\pi^{(i)}},\ \ 2 \le k \le K,\\
g(\pi^{(i)}) &= \sum_{j=1}^{K} r_j\lambda_j - C(v(C,\pi^{(i)}) - v(C-1,\pi^{(i)})),
\end{aligned} \tag{1}$$

(where $C = Cap(e)$), 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 $\pi^{(i)}$ and e in our notations.

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

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

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

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

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

$$f(x) = \begin{cases} 0 & \text{if } x < C - t_K \\ \dfrac{\sum_{j=k}^{K} r_j\lambda_j + x\cdot f(x-1)}{\sum_{j=1}^{k-1}\lambda_j} & \text{if } \begin{array}{l} C - t_k \le x < C - t_{k-1}, \\ 2 \le k \le K \end{array} \end{cases}$$

- **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 trunk reservation policy is to set a new set of trunk reservation levels such that

$$\mathcal{W}(Cap(e) - t_k^{\pi^{(i+1)}} - 1) \le r_k < \mathcal{W}(Cap(e) - t_k^{\pi^{(i+1)}})$$

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.

A.2 Traffic Classification, Reward Parameters, and Link-Offered Loads

For both the IND and SP routing algorithms, connections of type (s, D) offered to the network are of two types: direct and alternate, when offered to a link. Recall that a destination is added to the multicast tree sequentially. For adding a destination node, either one link or two links are used for connecting the destination to the multicast tree. If only one link is used, the connection is classified as direct connection on that link with reward parameter $r_{(s,D)}/|D|$ where $r_{(s,D)}$ is the reward for carrying the multicast connection and $|D|$ is the number of destinations. On the other hand, when two links are needed, the connection is classified as alternate connection on that link with reward parameter $r_{(s,D)}/|D|/2$ (because reward is shared by these two links).

Similarly, for the MST routing algorithm, connections of type (s, D) offered to the network are also split into two types. If the connection is carried on a multicast tree with $|D|$ links, then the connection is classified as a direct connection to all links in the tree with reward parameter $r_{(s,D)}/|D|$. However, if the connection is carried on a multicast tree with $|D|+1$ links, the connection is classified as an alternate connection with reward parameter $r_{(s,D)}/(|D|+1)$ (because the whole reward is shared by $|D|+1$ links. (Recall that the multicast tree constructed by the MST algorithm is restricted to have at most $|D| + 1$ links.)

In the computation of near optimal trunk reservation levels, arrival rate of each type of traffic, $\lambda_{e,k}$, at each link, $e \in E$, is required. As suggested in [27,28], we first measure the carried load of each type of traffic at regular time intervals of length Δ_t. Specifically, the measured carried load of type k traffic, $\tilde\lambda_{e,k}$, is the average number of type k connections carried per unit of mean connection holding time.

Based on the exponential smoothing model, the predicted arrival rate of type k connections is given by

$$\lambda_{e,k}^{new} = (1-\alpha) \cdot \lambda_{e,k}^{old} + \alpha \cdot \frac{\tilde{\lambda}_{e,k}}{1 - B_{e,k}},$$

where $\lambda_{e,k}^{old}$ is the predicted arrival rate of the previous update, $B_{e,k}$ is the time blocking probability of type k traffic, i.e., the proportion of time the type k traffic is blocked, and α is a constant from $(0,1)$. Since the arrival rates are updated periodically, the trunk reservation levels are also updated adaptively, according to the traffic loads.