Placement of Repair Proxies to Improve Inter-receiver Delivery Delay Fairness in Hierarchical Reliable Multicast Networks[1]

Sang-Seon Byun and Chuck Yoo
Department of Computer Science and Engineering,
Korea University, Seong-buk gu, An-am dong, Seoul, Korea
{ssbyun, hxy}@os.korea.ac.kr

Abstract

In hierarchical reliable multicast schemes, the number of repair proxies and their locations influence inter-receiver delivery delay fairness. Improving inter-receiver delivery fairness is beneficial to synchronized communication in multi-conference or interactive system. In this paper, we propose a scheme to find the optimal locations of repair proxies that can improve inter-receiver delivery delay fairness maximally in heterogeneous network environments when the number of available proxy is limited. Our evaluation results show that if repair proxies are placed by the proposed scheme, delivery delay fairness can be improved by 0.05 maximally.

1. Introduction

Many hierarchical reliable multicast (HRM) protocols deploy repair proxies that perform local recovery and feedback consolidation. Repair proxy can be set up as an exclusive server [1][2][3] or can be designated among adequate receivers [4][5]. The benefits of HRM are well described in references [6], [7] and [8].

The performance of HRM is evaluated by 1) delivery delay, 2) bandwidth overhead due to local recovery and feedback consolidation, and 3) inter-receiver fairness [6][8][23]. Delivery delay is the time that is required to successfully transmit a packet from the sender to a receiver. Inter-receiver delivery delay fairness is measured to estimate the diversity of each receiver's delivery delay. All these three metrics are affected by the locations of repair proxies. For example, if a proxy is adjacent with a receiver that is susceptible to packet losses, and the proxy is robust with packet losses, the recovery and feedback traffics are limited to the proxy's domain. Additionally, packet recovery time can be reduced, and so improved mean delivery delay and inter-receiver delivery delay fairness can be obtained.

Related with placement of proxies, many researches have focused on minimizing bandwidth overhead caused by recovery and feedback traffics [9][10][11] [14] and load balance among proxies [10][12]. Some researches don't consider optimal placement of proxies [13][14]. Also, they assume that every link has same propagation delay and loss rate. Reference [15] suggests a method to localize proxies to minimize web distribution time using dynamic programming formulation. However, in order to reduce combinatorial complexity, the available location of a proxy is limited to some area of the web distribution tree.

Related with inter-receiver synchronization in multicast environments, references [23] and [24] propose schemes that work at application level.

Improved inter-receiver delivery delay fairness is beneficial to inter-participant synchronization in multi-conference system, like IVS [25]. Additionally, heterogeneous loss rate and propagation delay have big impacts on placement of proxies, and in real world, available number of proxies may be limited.

In this paper, we propose a scheme that can determine locations of repair proxies to improve inter-receiver delivery fairness maximally using dynamic programming formulation when the available number of proxies is limited. The obtained inter-receiver delivery delay fairness is not optimal, but is maximally improved value through proper placement of proxies.

The rest of paper is organized as follows. In section 2, we explain HRM model used in this paper and describe expected delivery delay model. In section 3, we present our dynamic programming formulation for optimal placement of proxies. In section 4, numerical evaluations and comparisons are given. Finally, con-

[1] This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment)

cluding remarks and future work are presented in section 5.

2. HRM and Delivery Delay Model

In this section, we describe HRM and expected delivery delay model. In order to determine locations of proxies to inter-receiver delivery delay fairness, HRM and expected delivery delay model must deal with heterogeneity and locations of proxies.

2.1 HRM Model

It is assumed that the HRM model in our work has the following characteristics (Figure 1).

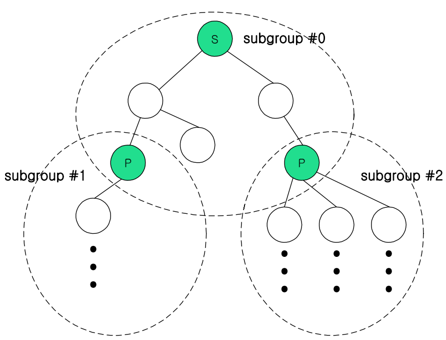

Figure 1. HRM Model.

1. The root of a multicast tree is the unique source, all leaves are receivers, and all intermediate nodes can be proxy [1][2][3].
2. The topology of control tree is identical to that of its underlying multicast tree (IP multicast tree), and loss probabilities and propagation delays at the links of the control tree are given. References [17], [18] and [19] describe a way of establishing a control tree that is identical to its underlying multicast tree and how to collect link loss statistics.
3. The control tree is partitioned into subtrees that form a hierarchy rooted at the source. All nodes in a subtree are combined into a subgroup, and each subgroup has a proxy located at the root of its subtree. The source is a proxy itself. This feature is deployed in [1], [2], [3], [5], [9] and [20].
4. A proxy multicasts the original data to its own subgroup. Each receiver sends feedback (NACK) to its proxy when a packet loss is detected, and the proxy retransmits the lost packet to the whole subgroup. Neither flow control nor congestion control is considered. All feedback packets are delivered via an out-of-band channel, so all feedback packets are delivered safely to proxies.
5. Feedbacks and transmissions/retransmissions are limited only between a proxy and the receivers of its subgroup and they do not reach receivers/proxies of any other subgroup. For this purpose, a new multicast address per subgroup is assigned [2], or TTL (Time to live) may be used to scope subgroup [4]. Additionally subcasting and TTL scoping can be used simultaneously [5].

2.2 Expected Delivery Delay Model

Table 1. Notations for expected delivery delay model

Value	Description
$L_{a,b}$	Summation of propagation delays on all links between node *a* and node *b*
$E(N_{a,b})$	When node *a* acts as a proxy of node *b*, the expected inter-arrival time of two consecutive packets in node *b*. (their sequence numbers are consecutive)
$\pi(a,b)$	Set of all proxies located on the path between node *a* and node *b*.
$\theta(a,b)$	Number of elements in set $\pi(a,b)$
$E(D_{S,R(w)})$	Expected delivery delay from sender node *s* and receiver node *R*(*w*)
$\psi(a,b)$	Set of all links between node *a* and node *b*.

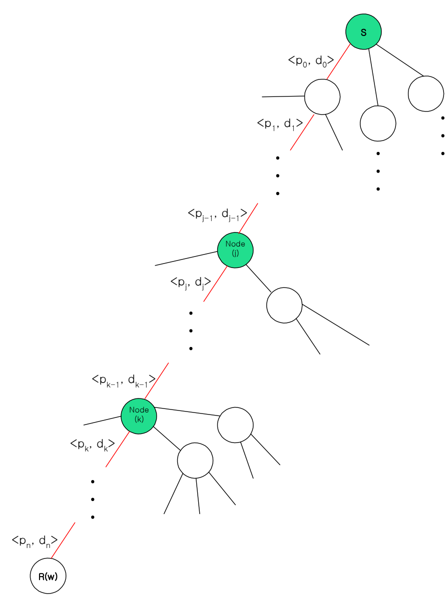

Figure 2. Expected delivery delay model reflecting heterogeneity and locations of proxies. <p, d> on each link presents <loss rate, propagation delay>.

In this section, we describe expected delivery delay model that reflects heterogeneity and locations of proxies. No queuing and transmission delay is considered in this expected delivery delay model and neither flow control nor congestion control is considered. It is assumed that packet losses at each link are independent [13]. Figure 2 shows expected delivery delay model that reflects heterogeneity of propagation delay and loss rate and locations of proxies. A summary of the used notations is given in Table 1.

E(Na,b) can be written as follows, where *t* is inter-packet delay(gap):

$$E(N_{a,b}) = \frac{1-\prod_{i\in\psi(a,b)}(1-p_i)}{\prod_{i\in\psi(a,b)}(1-p_i)} \times t + L_{a,b} \tag{1}$$

$E(D_{S,R(w)})$can be written in two cases as follows:

(1) In case $\pi(S, R(w)) = \{S\}$ (*S* is unique proxy between *S* and *R(w)*):

$$\begin{aligned} E(D_{S,R(w)}) &= \left[\prod_{i\in\psi(S,R(w))}(1-p_i)\right] L_{S,R(w)} + \\ &\left[1-\prod_{i\in\psi(S,R(w))}(1-p_i)\right]\left(E(N_{S,R(w)}) + E(D_{S,R(w)})\right) \end{aligned} \tag{2}$$

By eliminating $E(D_{S,R(w)})$ at the right side of (2), we obtain

$$E(D_{S,R(w)}) = L_{S,R(w)} + \frac{1-\prod_{i\in\psi(S,R(w))}(1-p_i)}{\prod_{i\in\psi(S,R(w))}(1-p_i)} E(N_{S,R(w)}). \tag{3}$$

(2) In case $\theta(S,R(w)) \geq 2$ and node(j)$\in\pi(S,R(w))$:

$$\begin{aligned} E(D_{S,R(w)}) &= \left[\prod_{i\in\psi(S,node(j))}(1-p_i)\right]\left(L_{S,node(j)} + E(D_{node(j),R(w)})\right) \\ &+ \left[1-\prod_{i\in\psi(S,node(j))}(1-p_i)\right]\left(E(N_{S,node(j)}) + E(D_{S,R(w)})\right) \end{aligned} \tag{4}$$

$$E(D_{S,R(w)}) = L_{S,node(j)} + \frac{1-\prod_{i\in\psi(S,node(j))}(1-p_i)}{\prod_{i\in\psi(S,node(j))}(1-p_i)} E(N_{S,node(j)}) + E(D_{node(j),R(w)}) \tag{5}$$

By definition of $E(D_{S,R(w)})$, we obtain

$$L_{S,node(j)} + \frac{1-\prod_{i\in\psi(S,node(j))}(1-p_i)}{\prod_{i\in\psi(S,node(j))}(1-p_i)} E(N_{S,node(j)}) \equiv E(D_{S,node(j)}), \tag{6}$$

so $E(D_{S,R(w)})$ can be written as

$$E(D_{S,R(w)}) = E(D_{S,node(j)}) + E(D_{node(j),R(w)}), \tag{7}$$

and if $node(k)\in\pi(node(j),R(w))$, we obtain

$$E(D_{node(j),R(w)}) = E(D_{node(j),node(k)}) + E(D_{node(k),R(w)}). \tag{8}$$

Thus by setting $\pi(S,R)=\{S, proxy_0, proxy_1, \ldots, proxy_z\}$, we obtain an recursive form as follows:

$$\begin{aligned} &E(D_{S,R}) = E(D_{S,proxy0}) + E(D_{proxy0,R}) \\ &E(D_{proxy0,R}) = E(D_{proxy0,proxy1}) + E(D_{proxy1,R}) \\ &\ldots\ldots \\ &E(D_{proxyz-1,R}) = E(D_{proxyz-1,proxyz}) + E(D_{proxyz,R}) \end{aligned} \tag{9}$$

So using this recursive form, we can compute expected delivery delays of all receivers if set of proxies is made.

3. Optimal Placement of Proxies

Table 2. Notations for the dynamic programming formulations

Value	Description
$\mathbf{T}_u$	Set of nodes placed in subtree rooted at node *u*
$\mathbf{R}_u$	Set of receivers placed in subtree rooted at node *u*
$\mathbf{P}_u$	Set of proxies placed in subtree rooted at node *u*
$n(\mathbf{R}_u)$	Size of $\mathbf{R}_u$
$F[u][\theta]$	$F[u][\theta]$is composed of three elements as follows: *nu*: numerator part of inter-receiver delivery delay fairness index of all receivers in $\mathbf{R}_u$ *de*: denominator part of optimal inter-receiver delivery delay fairness index of all receivers in $\mathbf{R}_u$ *v*: newly selected proxy node *n*: $n(\mathbf{P}_v)$, size of set of proxies placed in subtree rooted at newly added proxy node *v* Each element in $F[u][\theta]$ can be accessed by $F[u][\theta]$<*nu*>, $F[u][\theta]$<*de*>, $F[u][\theta]$<*v*> and $F[u][\theta]$<*n*> respectively.

As the network size grows, the number of ways of selecting proxies increases dramatically. So, to find optimal locations of proxies among these numerous ways becomes a combinatorial problem with large computational cost. In this paper, we deploy dynamic programming formulation [16] to alleviate computational cost. When select *m* proxies among *k* candidate nodes, process of improving inter-receiver delivery delay fairness can be performed using $O(k\times 4m)$ space.

If we consider a tree rooted at node *u*, inter-receiver delivery delay fairness is estimated using fairness index and is written as

$$fi_{\mathbf{T}_u} = \frac{\left(\sum_{\forall Ri \in \mathbf{R}_u} E(D_{u,Ri}) \right)^2}{n(\mathbf{R}_u) \sum_{\forall Ri \in \mathbf{R}_u} E(D_{u,Ri})^2}. \tag{10}$$

In order to utilize dynamic programming characteristics, we separate the fairness index into its numerator part and denominator part, and store them respectively, and use dynamic programming formulation to configure proxy set **P** to improve inter-receiver delivery delay fairness index maximally as (11) with notations in Table 2. The proof of the dynamic programming formula for computing $F[u][\theta]<nu>$ and $F[u][\theta]<de>$ is given in Appendix.

As done in usual dynamic programming, we compute $F[u][\theta]<nu>$ recursively using $F[u][\theta- n(\mathbf{P}_v)]<nu>$ and $F[u][n(\mathbf{P}_v)]<nu>$. $F[u][\theta]<de>$ is computed by the same way.

First, $F[u][1]$ is computed and mark u as proxy. After that, we compute $F[u][2]$. The stage of computing $F[u][2]$ selects a proxy v with $v \in \mathbf{T}_u$ that is not marked as proxy and improve inter-receiver delivery delay fairness in $\mathbf{R}_u$ maximally. In this case $n(\mathbf{P}_v)=1$, so $F[u][2]$ can be computed from $F[u][1]$. The numerator and denominator part of computed fairness index and the selected proxy are stored at $F[u][\theta]<nu>$, $F[u][\theta]<de>$ and $F[u][2]<v>$ respectively. $n(\mathbf{P}_v)$ is stored at $F[u][2]<n>$.

Like this way, we can obtain $F[u][\theta]$ recursively. The fourth element $n(\mathbf{P}_v)$ is needed when configure $\mathbf{P}_u$ using the matrix F. In equation (11),

$$F[u][\theta] = \begin{cases} \left(\sum_{\forall Ri \in \mathbf{R}_u} E(D_{u,Ri}) \right)^2, \\ n(\mathbf{R}_u) \sum_{\forall Ri \in \mathbf{R}_u} E(D_{u,Ri})^2,\ u,\ 1 & \theta = 1 \\ \underset{v \in \mathbf{T}_u, v \notin \mathbf{P}_u}{\text{Best}} [\sqrt{F[u][\theta - n(\mathbf{P}_v)] < nu >} \\ \quad - \sum_{\substack{\forall Ri \in \mathbf{R}v \text{ with} \\ \pi(u,Ri)-\{v\}}} E(D_{u,Ri}) \\ \quad + \sqrt{F[v][n(\mathbf{P}_v)] < nu >} \\ \quad + n(\mathbf{R}_v) E(D_{u,v}))]^2, \\ \underset{v \in \mathbf{T}_u, v \notin \mathbf{P}_u}{\text{Best}} [F[u][\theta - n(\mathbf{P}_v)] < de > & \theta \geq 2 \\ \quad - n(\mathbf{R}_u)\{ \sum_{\substack{\forall Ri \in \mathbf{R}v \text{ with} \\ \pi(u,Ri)-\{v\}}} E(D_{u,Ri})^2 \\ \quad - \frac{F[u][\theta - n(\mathbf{P}_v)] < de >}{n(\mathbf{R}_v)} \\ - n(\mathbf{R}_v)(2E(D_{u,v})\sqrt{F[v][n(\mathbf{P}_v)] < nu >} \\ \quad + E(D_{u,v})^2)\}],\ v,\ n(\mathbf{P}_v) \end{cases} \tag{11}$$

'Best[…]' means that we choose the one which is closest to 1 among all candidate values since the fairness index is better when it is more close to 1.

```
Output: Proxyset Pu
Inputs : F
Proxyset Make_Proxy_Set(node u, size m) {
   integer i;
   Pu += {u};
   for (i = 2; i <= m; i++) {
      Pu += F[u][i]<v>;
      if ((F[u][i]<n> > 1) && (Pu∩Tv≠Pv)) {
         Pv = Make_Proxy_Set(v, F[u][i]<n>);
         Pu = (Pu-Tv)∪Pv;
      }
   }
   return Pu;
}
```

Figure 3. Pseudo code for configuring proxy set P_u. *Make_Proxy_Set()* is called recursively.

Our main purpose of this dynamic programming is to configure a proxy set that improve fairness index of all receivers maximally by proper placement of proxies. Figure 3 shows pseudo code for configuring proxy set.

If node s is a sender and we configure $\mathbf{P}_s$ with $n(\mathbf{P}_s)=m$, $\mathbf{P}_s$ contains node s by default. Next we add nodes stored at $F[s][2]<v>,\ldots,F[s][m]<v>$ to $\mathbf{P}_s$. If $F[s][k]<v>$ is V_1 where $2 < k < m$, $\mathbf{P}_s$ may contain nodes in $\mathbf{T}V_1$. Put in other words, $\mathbf{P}_s$ may have nodes that are already in the subtree rooted at V_1. In this case, if $\mathbf{P}_s \cap \mathbf{T}V_1 = \mathbf{P}V_1$, we have only to add V_1 to $\mathbf{P}_s$ simply. However if $\mathbf{P}_s \cap \mathbf{T}V_1 \neq \mathbf{P}V_1$, we firstly remove nodes that satisfies $\mathbf{P}_s \cap \mathbf{T}V_1$ from $\mathbf{P}_s$ and next add nodes in $\mathbf{P}V_1$ to $\mathbf{P}_s$, which can be written as $\mathbf{P}_s = (\mathbf{P}_s - \mathbf{T}V_1) \cup \mathbf{P}V_1$. $\mathbf{P}V_1$ can be also configured recursively. Also, in this case, $F[s][k]<nu>$ and $F[s][k]<de>$ can not be computed from $F[s][k-1]<nu>$ and $F[s][k]<de>$ because the proxy obtained at the stage $F[s][k-1]$ is not an element of $\mathbf{P}_s$ any more at the stage of $F[s][k]$. Hence $F[s][k]<nu>$ must be computed using $F[V_1][n(\mathbf{P}V_1)]<nu>$ and $F[V_1][k-n(\mathbf{P}V_1)]<nu>$. Also, $F[s][k]<de>$ must be computed using $F[V_1][n(\mathbf{P}V_1)]<de>$ and $F[V_1][k-n(\mathbf{P}V_1)]<de>$.

4. Numerical Examples and Comparisons

In this section, the performance of our proposal and the random placement method are compared using the derived analytical model (equation (9)). A topology is generated by Inet topology generator [21], and the number of nodes in the topology is 13000. Propagation delays and loss rates of all links are assigned heterogeneously. The multicast delivery tree is constituted using Dijikstra Algorithm [22] in order to minimize total propagation delay of all source-receiver pairs. In the multicast delivery tree, 1200 nodes are selected as proxy candidate and 8000 nodes are selected as receivers. We assume that inter-packet gap is 25ms.

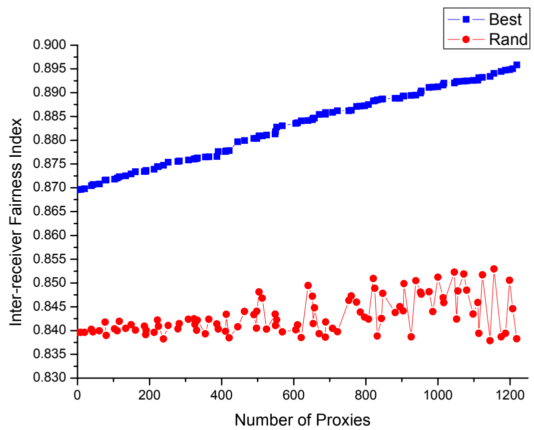

Figure 4. Inter-receiver delivery delay fairness index with respect to the number of proxies

In this section, the performance of our proposal and the random placement method are compared using the derived analytical model (equation (9)). A topology is generated by Inet topology generator [21], and the number of nodes in the topology is 13000. Propagation delays and loss rates of all links are assigned heterogeneously. The multicast delivery tree is constituted using Dijikstra Algorithm [22] in order to minimize total propagation delay of all source-receiver pairs. In the multicast delivery tree, 1200 nodes are selected as proxy candidate and 8000 nodes are selected as receivers. We assume that inter-packet gap is 25ms.

Figure 4 shows the inter-receiver delivery delay fairness index. Our proposal shows improvement of fairness index maximally about 0.05 over the random placement.

5 Conclusions and Future Work

In this paper, we propose a scheme to configure repair proxies that can improve inter-receiver delivery delay fairness maximally in heterogeneous network environments if proxy size is limited to some value. We describe expected delivery delay model to reflect heterogeneity and locations of proxies. We apply dynamic programming in order to configure an optimal proxy set in reasonable time. Through numerical evaluations, we show that our proposal yield better performance than the random placement with respect to the inter-receiver delivery delay fairness.

We will consider a number of issues in the future work, including formal evaluations of our algorithm complexity and deployment of GRASP [26] approach to reduce the algorithm complexity and configure a proxy set to minimize expected delivery delay of the slowest receiver. Additionally, how the GRASP can be deployed in order to incorporate dynamic nature of network environment into the proxy configuration method remains a challenge.

6. References

[1] Y. Chawathe, S. McCanne, and E. Brewer, “RMX: Reliable Multicast in Heterogeneous Networks”, In *Proc. of IEEE INFOCOM*, 2000.
[2] M. Hoffman, “A Generic Concept of Large-Scale Multicast”, *International Zurich Seminar on Digital Communications*, Springer, Feb. 1996, vol. LNCS 1044, pp. 95-106.
[3] H. Holbrook, S. Singhal, and D. Cheriton, “Log-Based Receiver-Reliable Multicast for Distributed Interactive Simulation”, In *Proc. of ACM SIGCOMM*, 1995.
[4] S. Floyd, V. Jacobson, C.-G. Liu, S. McCanne, and L. Zhang, “A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing”, *IEEE/ACM Trans. Networking*, vol. 5, pp. 784-803, Dec. 1997.
[5] J. Lin, and S.Paul, “RMTP: A Reliable Multicast Transport Protocol”, In *Proc. of IEEE INFOCOM*, 1998.
[6] J. Nonnenmacher, M. Lacher, M. Jung, E. Biersack, G. Carle, “How bad is Reliable Multicast without Local Recovery?”, In *Proc. of IEEE INFOCOM*, 1998.
[7] M. Lacher, J. Nonnenmacher, E. Biersack, “Performance Comparison of Centralized versus Distributed Error Recovery for Reliable Multicast”, *IEEE/ACM Trans. Networking*, vol. 8, no. 2, pp. 224-238, April 2000.
[8] C. Maihofer, and K. Rothermel, “A Delay Analysis of Tree-based Reliable Multicast Protocols”, In *Proc. of Computer Communications and Networks*, 2001.
[9] A. Markpoulou, and F. Tobagi, “Hierarchical Reliable Multicast: performance analysis and placement of proxies”, In *Proc. of ACM SIGCOMM*, 2000.
[10] H. Lin, and K. Yang, “Placement of Repair Servers to Support Server-based Reliable Multicast”, In *Proc. of ICC*, 2001.
[11] Z. Wan, M. Kadoch, and A. Elhakeem, “Performance Evaluation of Tree-based Reliable Multicast”, In *Proc. of ICCCN*, 2003.
[12] O. Daescu, R. Jothi, B. Raghavachari, and K. Sarac, “Optimal Placement of NAK-Suppressing Agents for Reliable Multicast: A Partial Deployment Case”, In *Proc. of ACM SAC*, 2004.
[13] S. Ratnasamy, and S. McCanne, “Scaling End-to-End Multicast Transports with a Topologically-Sensitive Group Formation Protocol”, In *Proc. of ICNP*, 1999.
[14] B. Li, F. Chen, and L. Yin, “Server Replication and Its Placement for Reliable Multicast”, In *Proc. of ICCCN*, 2000.

[15] B. Li, M. Golin, G. Italiano, X. Deng, and K. Sohraby, “On the Optimal Placement of Web Proxies in the Internet”, In *Proc. of INFOCOM*, 1999.
[16] R. Bellman, “Dynamic Programming”, *Princeton University Press*, 1957.
[17] B. Levine, S. Paul, and J. Garcia-Luna-Aceves. “Organizing Multicast Receivers Deterministically by Packet-Loss Correlation”, In *Proc. of ACM International Conference Multimedia*, 1998.
[18] S. Casner, and A. Thyagarajan, “mtrace(8): Tool to Print Multicast Path form a Source to a Receiver”, *UNIX manual command*.
[19] T. Speakman, J. Crowcroft, J. Gemmell, D. Farinacci, S. Lin, D. Leshchiner, M. Luby, T. Montgomery, L. Rizzo, A. Tweedly, N. Bhaskar, R. Edmonstone, R. Sumanasekera, and L. Vicisano, “PGM Reliable Transport Protocol Specification”, *IETF*, RFC 3208, Dec. 2001.
[20] D. Li, and D.R. Cheriton, “OTERS (On-Tree Efficient Recovery using Subcasting): A Reliable Multicast Protocol”, In *Proc. of ICNP*,1998.
[21] “Inet Topology Generator 3.0”, University of Michigan, *http://topology.eecs.umich.edu/inet/*
[22] S. Deryfus, “An Appraisal of Some Shortest-Path Algorithms”, *Operations Research*, 17(3), 1969.
[23] T. Nunome, and S. Tasaka, “An Application-level QoS Comparisons of Inter-destination Synchronization Schemes for Continuous Media Multicasting”, In *Proc. of IEEE GLOBECOM*, 2003.
[24] A. Benslimane, “A Multimedia Synchronization Protocol for Multicast Groups”, In *Proc. of Euromicro Conference*, Sept. 2000.
[25] T. Turletti, “The INRIA Videoconferencing System (IVS)”, *ConneXions-The Interoperability Report Journal*, vol. 8, no. 10, pp. 20-24, 1994
[26] C. Blum, and A. Roli, “Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison”, *ACM Computing Surveys*, vol. 35, no. 3, pp. 268-308, 2003.

Appendix. Proof of Dynamic Programming Formula for Computing Fairness Index

In this appendix, we prove our dynamic programming formula, $F[u][\theta]<nu>$ and $F[u][\theta]<de>$.

If node u is the sender and $\theta = 1$, inter-receiver delivery delay fairness index of $\mathbf{T}_u$ can be written as follows by the definition of fairness index:

$$\frac{F[u][1]<nu>}{F[u][1]<de>} = \frac{\left(\sum_{\forall Ri \in \mathbf{R}_u} E(D_{u,Ri})\right)^2}{n(\mathbf{R}_u)\sum_{\forall Ri \in \mathbf{R}_u} E(D_{u,Ri})^2}. \tag{12}$$

In order to prove $F[u][\theta]<nu>$ and $F[u][\theta]<de>$ in case $\theta \geq 2$, we consider a multicast delivery tree as in Figure 5. On this tree, we suppose that node v is selected as a new proxy, and hence $n(\mathbf{P}_u)$ becomes θ ($\theta \geq 2$). Additionally, let $E(D_{u,Ry})$ be the expected delivery delay between node u and Ry in case node v is *not* selected as a proxy, and $E^*(D_{u,Ry})$ be the expected delivery delay between node u and Ry in case nodes in $\mathbf{P}_v$ works as a proxy. By the definition of fairness index, $F[u][\theta]<nu>$ and $F[u][\theta]<de>$ can be expressed as

$$\begin{aligned} \frac{F[u][\theta]<nu>}{F[u][\theta]<de>} &= \frac{[\sum_{i=0}^{j} E(D_{u,Ri}) + \sum_{i=j}^{k} E^*(D_{u,Ri}) + \sum_{i=k+1}^{n} E(D_{u,Ri})]^2}{n(\mathbf{R}_u)[\sum_{i=0}^{j} E(D_{u,Ri})^2 + \sum_{i=j}^{k} E^*(D_{u,Ri})^2 + \sum_{i=k+1}^{n} E(D_{u,Ri})^2]} \\ &= \frac{[\sum_{i=0}^{n} E(D_{u,Ri}) - (\sum_{i=j}^{k} E(D_{u,Ri}) - \sum_{i=j}^{k} E^*(D_{u,Ri}))]^2}{n(\mathbf{R}_u)[\sum_{i=0}^{n} E(D_{u,Ri})^2 - (\sum_{i=j}^{k} E(D_{u,Ri})^2 - \sum_{i=j}^{k} E^*(D_{u,Ri})^2)]}. \end{aligned} \tag{13}$$

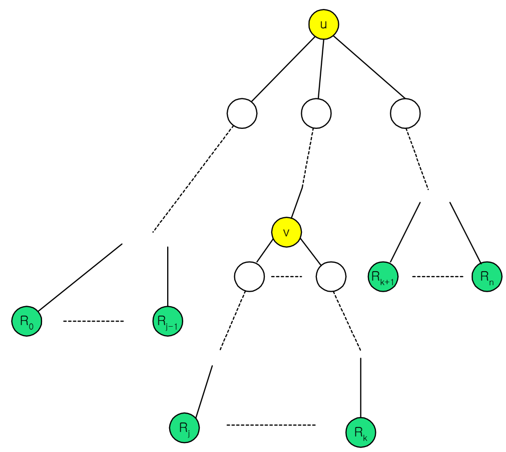

Figure 5. A multicast delivery tree. Node *v* is newly selected proxy.

From (9), we obtain

$$\sum_{i=j}^{k} E^*(D_{u,Ri}) = \sum_{i=j}^{k} (E(D_{v,Ri}) + E(D_{u,v})), \tag{14}$$

so (13) can be rewritten as

$$\frac{F[u][\theta]<nu>}{F[u][\theta]<de>} = \frac{[\sum_{i=0}^{n} E(D_{u,Ri}) - \{\sum_{i=j}^{k} E(D_{u,Ri}) - \sum_{i=j}^{k} (E(D_{v,Ri}) + E(D_{u,v}))\}]^2}{n(\mathbf{R}_u)[\sum_{i=0}^{n} E(D_{u,Ri})^2 - \{\sum_{i=j}^{k} E(D_{u,Ri})^2 - \sum_{i=j}^{k} (E(D_{v,Ri}) + E(D_{u,v}))^2\}]}. \tag{15}$$

By definition of fairness index and $n(\mathbf{R}_v) = k - j + 1$, we obtain

$$\sum_{i=0}^{n} E(D_{u,R_i}) = \sqrt{F[u][\theta - n(\mathbf{P}_v)] < nu >}, \tag{16}$$

$$\sum_{i=j}^{k} \{E(D_{v,R_i}) + E(D_{u,v})\} = \sqrt{F[v][n(\mathbf{P}_v)] < nu >} + n(\mathbf{R}_v)E(D_{u,v}), \tag{17}$$

$$n(\mathbf{R}_u)\sum_{i=0}^{n} E(D_{u,R_i})^2 = F[u][\theta - n(\mathbf{P}_v)] < de > \tag{18}$$

and

$$\begin{aligned} &\sum_{i=j}^{k} \{E(D_{v,R_i})^2 + 2E(D_{v,R_i})E(D_{u,v}) + E(D_{u,v})^2\} \\ &= \frac{F[u][\theta - n(\mathbf{P}_v)] < de >}{n(\mathbf{R}_v)} \\ &\quad + n(\mathbf{R}_v)\{2E(D_{u,v})\sqrt{F[v][n(\mathbf{P}_v)] < nu >} + E(D_{u,v})^2\}. \end{aligned} \tag{19}$$

Hence, from (15) and (16), we obtain

$$\begin{aligned} &F[u][\theta] < nu > = [\sqrt{F[u][\theta - n(\mathbf{P}_v)] < nu >} \\ &- \sum_{i=j}^{k} E(D_{u,R_i}) + \sqrt{F[v][n(\mathbf{P}_v)] < nu >} + n(\mathbf{R}_v)E(D_{u,v})]^2 \end{aligned} \tag{20}$$

and

$$\begin{aligned} &F[u][\theta] < de > = F[u][\theta - n(\mathbf{P}_v)] < de > - n(\mathbf{R}_u)[\sum_{i=j}^{k} E(D_{u,R_i})^2 \\ &- \frac{F[u][\theta - n(\mathbf{P}_v)] < de >}{n(\mathbf{R}_v)} - n(\mathbf{R}_v)(2E(D_{u,v})\sqrt{F[v][n(\mathbf{P}_v)] < nu >} \\ &\qquad + E(D_{u,v})^2)]. \end{aligned} \tag{21}$$