Computer Communications 25 (2002) 1774–1781

computer communications

www.elsevier.com/locate/comcom

RPIM-SM: extending PIM-SM for RP relocation

Ying-Dar Lin[a,*], Nai-Bin Hsu[a], Ren-Hung Hwang[b]

[a]Department of Computer and Information Science, National Chiao Tung University, 43 lane 486, Min Hu Road, 30053 Hsinchu, Taiwan, ROC
[b]Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan, ROC

Received 11 July 2001; revised 6 February 2002; accepted 27 February 2002

Abstract

The protocol independent multicast-sparse mode (PIM-SM) protocol establishes core-base tree to forward multicast datagrams in a network. In PIM-SM, the core or rendezvous point (RP) of a group is determined at each multicast router by hashing a group address, i.e. a class-D IP address, to one of the candidate RPs. The hash function is characterized by its ability to evenly and uniquely choose the core for a group and remains insensitive to the geographic distribution of the group members and the sources. However, it may result in a multicast tree with high cost.

This study presents a relocation mechanism, which is extension to PIM-SM, in which RP could be relocated periodically. When a new RP is found, the original RP informs all members to re-join to the new RP. Simulation results indicate that the extended version, RPIM-SM, reduces about 20% tree cost than PIM-SIM when the group size is medium. Moreover, comparing RPIM-SM with the optimal core-based tree reveals that they have less than 5% difference in tree cost. Furthermore, an increase of the number of candidate RPs brings RPIM-SM even closer to the optimal core-based tree. Results in this study demonstrate that relocation improve the performance of PIM-SM. © 2002 Published by Elsevier Science B.V.

Keywords: PIM-SM; Rendezvous point; Relocation; RPIM-SM

1. Introduction

Multicast routing protocols can be categorized as source-based tree and shared-tree protocols. A source-based tree protocol builds separate trees for each (source, group) pair, that is, each source has its own tree that reaches the active group members, such as DVMRP [1], PIM-DM [2], and MOSPF [3]. On the other hand, shared-based tree protocols such as PIM-SM [4] and CBT [5] build distributed trees having a central point (or core) to whom all receivers attached. Typically, a source sends datagrams to the core and core forwards them to all members through the core-based multicast tree. Therefore, a shared-tree router only needs to maintain state information for each group instead of for each (source, group) pair.

In PIM-SM, new members wanting to join a group send *Join* messages to a core, called rendezvous point (RP), of the distribution tree. The RP administers the specific multicast group(s), and facilitates the joining/leaving of the group members. The address of the RP is determined at each multicast router by mapping a group to one of the candidate RPs with a hash function. However, the chosen RP may be inappropriate for its group, for example, causes more tree cost. The hash function is characterized by its ability to evenly and uniquely choose a candidate RP to be a core in order to balance the service load of each candidate RP in the network.

This study proposes an RP relocation mechanism which is extension to the PIM-SM multicast routing protocol. The hash function of PIM-SM is initially used to obtain an RP for the multicast group. As the sources come and go, RP relocates its location periodically afterward. An attempt is made to obtain an appropriate core location by using the estimated tree cost function to evaluate the appropriateness of the candidates. When RP relocation is determined, the original RP multicasts *NEW_RP* message to all members to inform them to re-join to the new RP. Consequently, a new distribution tree is built with the least cost.

The rest of this work is organized as follows. Section 2 presents the issues of PIM-SM protocol and our motivations. Section 3 describes the details of the RP relocation mechanism. Section 4 presents the control messages in

* Corresponding author. Tel.: +886-3-5731899; fax: +886-3-5721490.
E-mail addresses: ydlin@cis.nctu.edu.tw (Y.D. Lin), gis84811@cis.nctu.edu.tw (N.B. Hsu), rhhwang@cs.ccu.edu.tw (R.H. Hwang).

0140-3664/02/$ - see front matter © 2002 Published by Elsevier Science B.V.
PII: S0140-3664(02)00090-7

Table 1
List of symbol definitions

V: set of routers	*hashRP*: initial hashed RP of G
E: set of links	*currentRP*: current RP of G
S: set of sources	*newRP*: RP to be migrated
R: set of members	*rFlag*: relocation flag of G
G: multicast group	P_c, P_e: Probability functions
M: hash mask	*RPset*: set of candidate RPs
m: group member	C_i: ith candidate RP, $C_i \in$ RPset
t_r: relocation timer	C_k: min cost candidate, $C_k \in$ Rpset
TC: cost of multicast tree	k: current # of members in G
u, v: networks nodes, u, $v \in V$	q: cost reduction threshold
n: # of nodes in the network	*dist*: distance or hop count
n_d: # duplicate dist. node in S	*deg*: node degree or connectivity

addition to PIM-SM. Next, Section 5 provides the simulation models and results. Conclusions are finally drawn in Section 6.

2. Issues and motivations

PIM-SM is a commonly used multicast routing protocol that provides efficient communication for multicast groups with sparsely distributed members. The designers observed that several hosts wanting to participate in a multicast conference do not justify having their group's multicast traffic periodically broadcast across the entire network. To eliminate the scaling problem, PIM-SM is designed to limit multicast traffic so that only routers interested in receiving traffic for a particular group will receive it. By unicast routing, the source router knows how to reach and forward the traffic to the RP. Then, RP distributes the traffic to all the members through the RP-based multicast tree.

In order to broadcast the set of candidate RPs to the network nodes, the bootstrap router (BSR) is elected for the domain. BSR originates *Bootstrap* messages to distribute the set of RPs information, which are distributed hop-by-hop throughout the domain. There is only one RP-set per PIM-SM domain. By using a hash function, say *Hash*(), each router can uniquely map a group address G to one of the routers in the RP-set. That is, candidate C_k is chosen as RP if C_k yields highest hash value $Hash(G, M, C_i)$, for all C_i belong to the *Rpset*. The hash mark, M, allows a number of consecutive groups to resolve to the same RP.

The shared tree, although provides better scalability, does not optimize the delivery path through the network. RP for the group is typically designed without respect to its location. Thus, an RP could be located for away from all group members, resulting in inefficient transmission.

A Steiner tree problem [6] is a well-known of finding a multicast tree with minimum cost. However, the sources and the destinations of the multicast tree $T(S, R)$ in PIM-SM, the set of source, S, and the set of members, R, are dynamically changed. Therefore, after sources and/or receivers come and go, the least cost multicast tree is different. Therefore, considering a single group in which no member switches over to the source-based shortest path tree (SPT) [4], there is a sequence of least cost multicast trees, $\{T_0, T_1, T_2, \ldots\}$, where

$$T_i = Tree(S \rightarrow RP_i) \cup Tree(RP_i \rightarrow R) = T_i^S \cup T_i^R.$$

Since the tree T_i is rooted at the RP_i, the problem of finding the sequence $\{T_i\}$ is reduced to the problem of finding the sequence $\{RP_i\}$, where the RP_0 is the initial hash RP.

Ref. [7] provides algorithms of finding a good RP in distributed fashion. However, RP relocation causes overhead including the cost of exchanging the protocol control messages, finding a good RP, and the retransmissions due to loss of data. Obviously, this overhead could be small by reducing the frequency of the RP relocation. Algorithms in Ref. [7] consider the sources of the group only, rather than the members. There are several reasons to support that: RP of the PIM-SM is assumed to have the source list only, which is provided by the sources to *register* the RP. In addition, the source list of G is unlikely to be changed very often, no relocation is run as long as the source list unchanged. Finally, global network topology is usually not available for the local domain, it is not practical to compute the total tree cost of G.

Therefore, in this study, assume RP_i is the current RP of G, we propose an RP relocation mechanism which relocates RP when the tree cost reduction, $TC(RP_i) - TC(RP_{i+1})$, if it is larger than a pre-defined threshold, q. In addition, five control messages are needed to migrate to the new RP and reliably maintain the membership of the group. For convenience of our description, Table 1 summarizes the symbols used in this study.

3. RP relocation algorithm

Of priority concern in a distribution tree, the lower the tree costs implies better multicast routing paths. Tree cost is defined as

$$\sum_{(u,v) \in T} \mathrm{cost}(u, v),$$

i.e. the sum of the costs of all links of the multicast tree. Our mechanism attempts to reduce the tree cost by relocating the RP for a multicast group. To map an appropriate RP in a group, this study uses an estimated function which is inspired by Ref. [7] to obtain a RP with the minimum tree cost from the *RPset* of a group. As in Definition 1, Eq. (1) calculates the distribution tree cost if the candidate C_i is taken as the RP, $TC(C_i)$, by taking the average of Eqs. (2) and (3), i.e. the maximum and minimum bounds on tree cost. These equations use the distance (i.e. $cost : E \mapsto \mathbf{R}^+$) for each possible destination as the metric. Notably, the distance information is already available to routers.

Definition 1. Given a multicast group G, the sources S and the group of members R, the tree cost of the multicast tree as

```
RPIM-SM(G)
set_of_sources S;
set_of_members R;
member m ∈ R;
relocation_flag rFlag;
relocation_timer t_r;
tree_cost_function TC;
Begin
  while (true) do
    Things to do ...
    case hashRP:
      if received Join from a new member m
      // hashRP found by Hash function
        if (G.rFlag == true) then
          unicasts NEW_RP(currentRP) to m
          //m send Join to currentRP
          //m send Prune to hashRP
        endif
      endif
    case currentRP:
      if (t_r expired and S changed) then
        Query TC(C_i) from each C_i of RPset,
        newRP ← Select C_k where TC(C_k) is minimum
        if reduction of TC(C_k) > q then
          multicasts NEW_RP(currentRP, newRP)
            to {hashRP, sources, members} of G
        endif
      endif
  endwhile
End
```

Fig. 1. The RP relocation of group G.

C_i is RP of G is defined as

$$TC(C_i) = \frac{TC_{\min}(C_i) + TC_{\max}(C_i)}{2}, \qquad (1)$$

where the minimum tree cost

$$TC_{\min}(C_i) = \max_{u \in S} \mathrm{dist}(C_i, u) + n_d, \qquad (2)$$

and the maximum tree cost

$$TC_{\max}(C_i) = \begin{cases} \sum_{u \in S} \mathrm{dist}(C_i, u) & \text{if } |S| \le \deg(C_i), \\ \left(\sum_{u \in S} \mathrm{dist}(C_i, u) \right) - (|S| - \deg(C_i)) & \text{otherwise} \end{cases} \qquad (3)$$

C_i denotes a candidate RP, i.e. $C_i \in \mathrm{RP}set$; $\deg(C_i)$ is the node degree of C_i; u is one of the source routers; and n_d is number of sources which use the same path to the C_i.

The best-case tree is linear if a lower bound on the cost, $TC_{\min}$, of a tree rooted at some node C_i is obtained. The cost of the tree is simply the maximum distance from RP to any sender. When using hop count (i.e. cost:$E \mapsto 1$) as the metric, the function can obtain a tighter bound by adding the number of senders that are at an equal distance. This bound is owing to that the distribution tree cannot be completely linear, but must have at least an additional link.

Opposite to the lower bound of the cost, the upper bound on the cost, $TC_{\max}$, of a tree rooted at some node C_i is that no links are shared among the paths to each sender. Therefore, the maximum tree cost is the sum of the sender distances. Also, if the number of group senders is greater than the degree of C_i, the bound is tightened by subtracting the difference to calculate the cost for the sharing links. The estimation function is thus defined as the average of the sum of the minimum cost and maximum cost.

```
ChangeRP(currentRP, newRP, G)
set_of_members R;
member m ∈ R;
relocation_flag rFlag;
Begin
  if (newRP == hashRP) then
    G.rFlag ← false /* return back to initial RP */
  else
    G.rFlag ← true
  endif
  G.oldRP ← G.RP
  G.RP ← newRP

  case newRP:
    send I_AM_RP to hashRP
  case source:
    re-register to the newRP
  case member:
    send Join to newRP
    send Prune to currentRP
End
```

Fig. 2. The RP migration of group G.

Table 2
Relocation state table

G	*rFlag*	*RP*	*iif*	*nbrRPF*	*oldRP*	*iif*	*nbrRPF*
G_1	*true*	RP_1	i_1	v_1	RP'_1	i'_1	v'_1
G_2	*true*	RP_2	i_2	v_2	RP'_2	i'_2	v'_2
	⋮						
G_n	*false*	RP_n	i_n	v_n			

As shown in the algorithm of Fig. 1, a new member m that joins a group sends a *Join* message to the hashed RP. The *hashRP* verifies whether the RP of this group has been migrated (i.e. the *rFlag* is *true*). If so, *hashRP* redirects m to the new RP by sending the *NEW_RP* message. Thus, m sends a *Join* message and established $(^*, G)$ state along the path towards the new RP. In addition, m sends a *Prune* message to the *hashRP* to discard the previous path. The *currentRP* also confirms whether if the relocation timer t_r has expired and the set of sources of the group has been changed. If it has, The *currentRP* multicasts the query message with source list to the candidate RPs. Tree cost is computed by the above cost functions $TC(C_i)$ in Definition 1 at each candidate RP, then reply the cost to the current RP. The *currentRP*, finds the least cost candidate RP C_k, checks it whether can significantly reduce the tree cost of the group. If it can, it will be the new RP of the group.

Once the new RP with least cost for a group is found, the function **ChangeRP** is invoked, as shown in Fig. 2. The *currentRP* first checks whether the new RP is the *hashRP*. If it is, the *rFlag* is reset to *false* since the *currentRP* returns back to the original hashed RP. To migrate the distribution

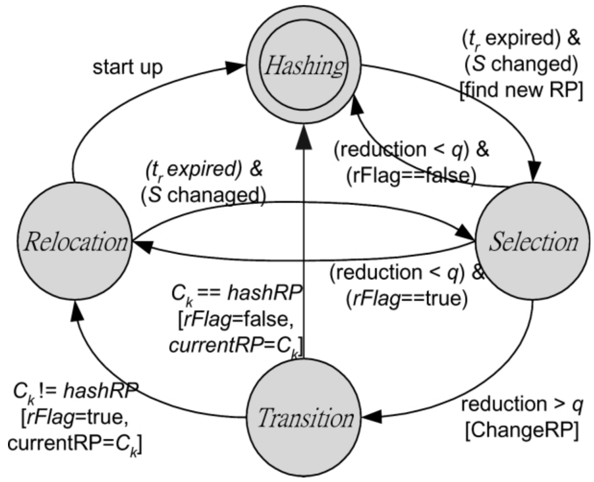

Fig. 3. State transition diagram of the RP relocation.

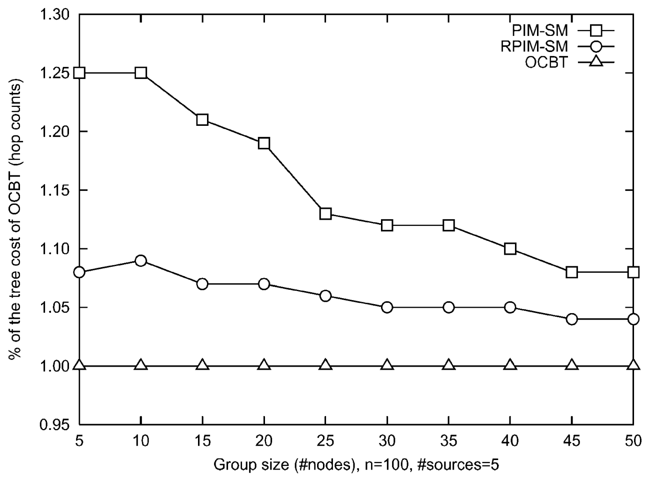

Fig. 4. Tree cost comparison (hop counts).

tree, message *NEW_RP* is advertised by the *currentRP* to inform the *newRP*, *hashRP*, sources, and all members of the group.

For keeping this relocation state, each node with (*, G) state maintains a relocation table to record the state of RP relocation, as well as the addresses of the new RP and old RP, as shown in Table 2, *RP* and *oldRP* fields, respectively. The *rFlag* indicates whether if RP of this group has been relocated, i.e. not the original hashed RP. When a new RP is obtained, the previous *newRP* becomes the *oldRP* and its address is moved to the field *oldRP*; the new RP address is then saved at the field *RP*. Notably, the corresponding incoming interface (*iif*) and the Reverse Path Forwarding (*RPF*) neighbor (*nbrRPF*) of the *RP/oldRP* are also saved. State information of the relocating process prevents the chaos caused by *RPF* checking during the transition between the two consecutive multicast trees.

For example, a node with (*,G_1) state continues to distribute packets from the old RP (RP'_1) at the interface i'_1, until this node receives packets from the new RP (RP_1) at the interface i_1. Furthermore, in order to migrate the distribution tree of the group G_1, according to Table 2, the *currentRP* (RP'_1) of G_1 extracts the *newRP* RP_1 from the table and multicasts the encoded *NEW_RP* message towards the *newRP*, all sources and members of group G_1. This message redirects all sources to re-register, all members to re-join to the new RP. Note that in case of G_n, since the *rFlag* is *false*, this entry records *hashRP* of G_n at the *newRP* field.

Fig. 3 shows the RP state transition of a specific group G. At the system start up, the RP is in the *Hashing* state and enters the *Selection* state as t_r timer out and the source list changed. If the selected candidate (C_k) can significantly reduce the current tree cost, **ChangeRP** is invoked and the *Transition* state is entered, and thereafter transits further to the *Relocation* state or *Hashing* state, depending on whether the newly selected RP is the original hashed RP. Returning to the *Hashing* state means that the original hashed RP turns out to be the best RP for G again. On the other hand, if the reduction in tree cost does not exceed the threshold, q, the RP enters the *Relocation* state or *Hashing* state, depending on the value of *rFlag* where *true* means this RP is a relocated, instead of hashed, one. Note that the *RPIM-SM* verifies the source list, S, and the tree cost of G every t_r time unless the system re-startup.

Table 3 shows the state machine in tabular form, including the event conditions and the actions to be triggered.

4. Control messages

These control messages in this study use the existing *Join/Prune* Expiry Timers and thus no additional timer is needed.

4.1. NEW_RP

This message is used to advertise to the group that who is the new RP from now on. When the computation of tree cost in Fig. 1 is done and the decision of the RP relocation is positive, the current RP sends the message *NEW_RP*(G, *newRP*) to the *newRP*, *hashRP*, sources, and multicasts it to all members of the group. Then, the migration process is triggered, i.e. the sources re-register to the *newRP* and members re-join to the *newRP*.

Source routers of the group G put the address of the *G.RP* as the *G.oldRP*, and the address of the *newRP* as the *G.RP*. To eliminate loss of packets due to the RP-migration, source routers of G should continue to send packets to *G.oldRP* for a time period.

4.2. COST_QUERY and COST_REPLY

The current RP uses the COST_QUERY message to trigger each candidate RP to compute the tree cost if this candidate RP becomes the RP of the group. Then the

Table 3
State machine of the RP relocation

condition [*action*]	to *Hashing*	to *Selection*	to *Transition*	to *Relocation*
from *Hashing*	–	t_r expired S changed [find *newRP*]	–	–
from *Selection*	reduction $< q$ $rFlag == false$[reset t_r]	–	reduction $> q$[ChangeRP, reset t_r]	reduction $< q$ $rFlag == true$[reset t_r]
from *Transition*	$C_k == hashRP$ [$rFlag \leftarrow false$, $currentRP \leftarrow C_k$]	–	–	$C_k \neq hashRP$[$rFlag \leftarrow true$, $currentRP \leftarrow C_k$]
from *Relocation*	start up [$rFlag \leftarrow false$, reset state]	t_r expired S changed [find *newRP*]	–	–

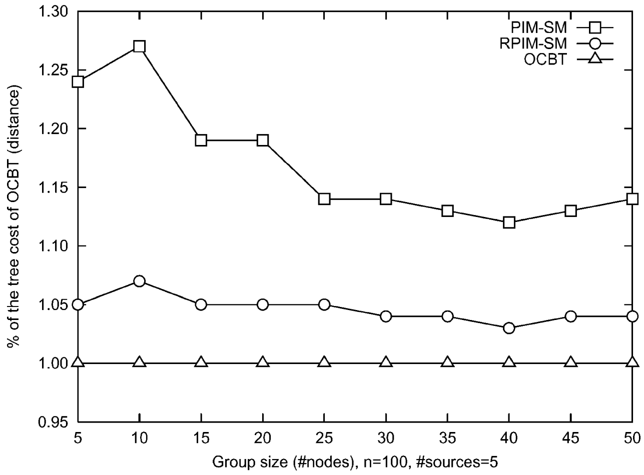

Fig. 5. Tree cost comparison (distance).

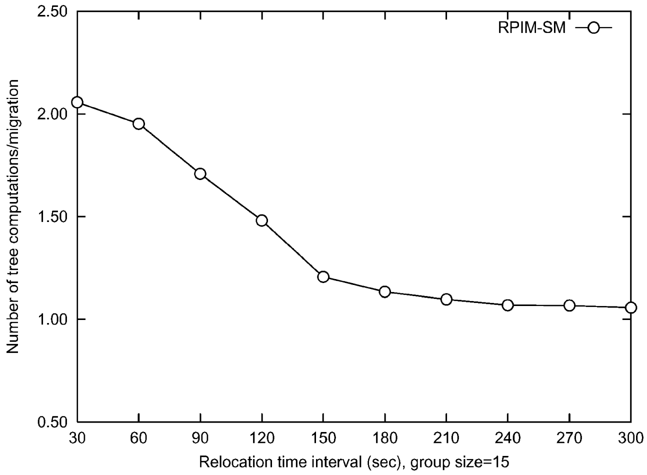

Fig. 7. Effects of relocation time interval.

candidate RP reply the tree cost to the current RP by sending the COST_REPLY (*myAddress*, *cost*) message.

4.3. *I_AM_RP and RP_CONFIRM*

Using this message, *newRP* informs *hashRP* that "I am the RP of the group at this time". Without this information at the *hashRP*, the incoming receivers could not find the current RP to whom the *Join* message send. Upon receiving *I_AM_RP* message, *hashRP* updates the corresponding relocation state of *G*, and acknowledges the *newRP* with the *RP_CONFIRM* message.

5. Simulation model and results

5.1. *Network model*

Simulations are performed to evaluate the performance of the proposed *RPIM-SM* extension to the PIM-SM multicast routing. Random graphs [8,9] are used to simulate network models in order to ensure that the effects of the different routing algorithms are independent of any specific network.

For all possible pairs (u, v) of nodes, the edges can be added to the graph by using the probability function:

$$P_e(u,v) = \beta \exp\left(\frac{-d(u,v)}{\alpha L}\right) \quad (4)$$

where $d(u, v)$ is the *Euclidean* distance between nodes u and v, L is the maximum possible distance between any two nodes in the graph, and α and β are parameters of real numbers in the range (0, 1). Setting the values of parameters α and β to 0.25 and 0.2, respectively, can generate graphs which have the appearance resembling that of geographical maps of major nodes in the Internet [10]. Graphs are generated with an average node degree of 4. Unless otherwise specified, the number of nodes in networks is 100. After each graph has been generated, Prim's or Kruskal's algorithm is used to ensure that the random graph comprises of only one component.

The sequence of events of *join/leave* is generated by a simple probability model [8]. The probability that a adding a node to the multicast group is determined by the probability function is defined as follows:

$$P_c(k) = \frac{\gamma(n-k)}{\gamma(n-k) - (1-\gamma)k} \quad (5)$$

where n is the number of nodes in the network, k is the current number of group members, and γ is a parameter in the range (0, 1) which determines the fraction of nodes in the connection at equilibrium [11]. If the value of γ is set to 0.5; that is, half of the nodes in the graph are in the multicast group in equilibrium. When $\gamma n = k$, the probability $P_c(k) =$ 1/2, means the probabilities of join and leave are equal.

5.2. *Simulation results*

5.2.1. *Tree cost comparison*

Figs. 4 and 5 compare the tree costs of PIM-SM and *RPIM-SM* with the cost of the optimal center-based tree

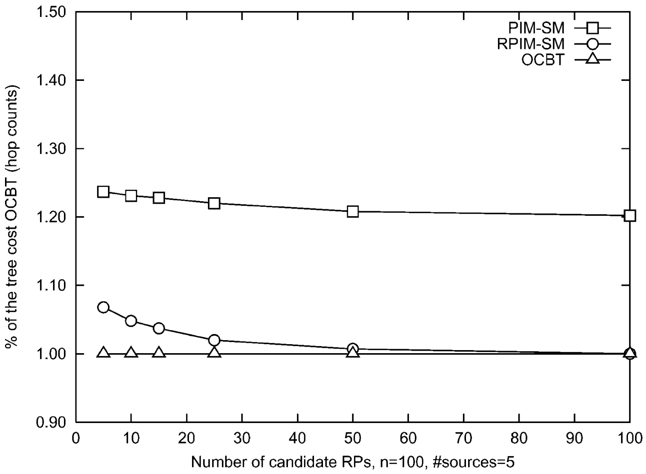

Fig. 6. Tree cost vs. number of candidate RPs.

(OCBT). The tree cost of the OCBT calculates the actual cost of the shortest path tree rooted at each node in the network and the one with the lowest maximum length among all of those with the lowest cost is selected. Fig. 4 shows the average results of 100 simulations in a 100-node network. The number of sources is fixed to five. Notably, the *Y*-axis plots the ratio of tree cost of the PIM-SM (before RP relocation) or *RPIM-SM* (after RP relocation) to the tree cost of the OCBT. The hash function in PIM-SM does not consider the geographic distribution of members, accounting for why the *RPIM-SM* performs better than PIM-SM, i.e. closer to the cost of OCBT, particularly in a small group size. With a larger group size, there is a higher likelihood of the hashed RP near to the center of the group, thereby decreasing the tree costs of both mechanisms. Hence, for various group sizes, *RPIM-SM* performs better than PIM-SM.

Similar to Fig. 4; Fig. 5 compares the tree costs of PIM-SM and *RPIM-SM* using the distance metric. The tree cost direction of the *RPIM-SM* is more stable than the cost of PIM-SM. The behavior resembles that found in Fig. 4. Our results indicate that when the group size is under 10, the relocating RP renders about 20% reduction in tree cost. The curve clearly reveals the merits of *RPIM-SM*.

5.2.2. Number of candidate RPs

Fig. 6 describes the tree costs versus the number of candidate RPs in the group. Both PIM-SM and *RPIM-SM* reduce the trees cost with increase of the number of candidate RPs. However, *RPIM-SM* is more effective on tree cost reduction. In particular, as all network nodes are candidates, tree cost of the *RPIM-SM* is the same as that of OCBT.

5.2.3. Time interval of RP relocation

Obviously, in *RPIM-SM*, computing the RP relocation causes additional overhead. To minimize the additional consumption, how long is the relocation interval t_r should be set must be determined. Fig. 7 shows the ratio of number of relocation computations to the number of RP migrations on average. These statistics depend on the cost reduction threshold, q, which is set to 10% in our experiments. In addition, they also depend on the event probability of sender, which is relevant to the type of multicast application. Here we simply assume the probability is a fraction of Eq. (5), say, $0.2P_c(k)$. According to this figure, as the time interval t_r is small, more than one computation is required per each RP migration. When the time interval exceeds 150 s, the curve approaches to 1.0. This finding suggests that the time interval of relocation was sufficient so that the migration almost took place when relocation computation is executed. Based on our simulation model, the time period t_r can be set at every 150 s to verify whether if the RP of each group needs to be relocated.

6. Conclusions

This paper proposes a dynamic mechanism, *RPIM-SM*, which is an extension to the PIM-SM multicast routing protocol. The original hashed RP location may be inappropriate for the group. Therefore, according to the estimated tree cost, the PIM-SM is extended by relocating the RP periodically. In addition, the candidate RP with minimum tree cost is selected to be the new RP of the group. To migrate to the new RP, an additional message, *NEW_RP*, is needed to notify sources to re-register, and members to rejoin the new RP.

Simulation results indicate that *RPIM-SM* reduces about 20% in tree cost of PIM-SM when group size is around 10. Moreover, when we compare *RPIM-SM* with the optimal core-based tree, *RPIM-SM* has less than 5% difference in tree cost. When the number of candidate RPs increases, *RPIM-SM* is even closer to the optimal core-based tree. In addition, under the network size of 100 nodes and the dynamic group membership, simulation results further indicate that 120–150 s would be appropriate as the time interval of possible RP relocation.

References

[1] D. Waitzman, C. Partridge, S. Deering, Distance Vector Multicast Routing Protocol, RFC 1075, IETF, November, 1988.
[2] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, A. Helmy, L. Wei, Protocol Independent Multicast Version 2, Dense Mode Specification, Internet Draft, draft-ietf-idmr-pim-dm-05.txt.
[3] J. Moy, Multicast Extensions to OSPF, RFC, 1584, IETF, March, 1994.
[4] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, L. Wei, Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification, RFC, 2362, IETF, June, 1998.
[5] A. Ballardie, Core Based Trees (CBT) Multicast Routing Architecture, RFC 2201, IETF, September, 1997.
[6] S.L. Hakimi, Steiner problem in graphs and its implications, Networks 1 (1971) 113–133.
[7] D.G. Thaler, C.V. Ravishankar, Distributed center-location algorithms, IEEE Journal on Selected Areas in Communications 15 (3) (1997) 291–303.
[8] B.M. Waxman, Routing of multipoint connections, IEEE Journal on Selected Areas in Communications 6 (9) (1988) 1617–1622.
[9] B.M. Waxman, Performance Evaluation of Multipoint Routing Algorithms, Proceedings of Infocom'93, vol. 3, IEEE Press, New York, 1993, pp. 980–986.
[10] J.M.S. Doar, Multicasting in the Asynchronous Transfer Mode Environment, Computer Laboratory Technical Report, No. 298, University Cambridge, April, 1993.
[11] D. Estrin, M. Handley, A. Helmy, P. Huang, A Dynamic Bootstrap Mechanism for Rendezvous-based Multicast Routing, Proceedings of Infocom'99, IEEE Press, New York, 1999, pp. 1090–1098.

Ying-Dar Lin received the BS degree in Computer Science and Information Engineering from National Taiwan University in 1988, and the MS and PhD degrees in Computer Science from the University of California, Los Angeles in 1990 and 1993, respectively. He joined the faculty of the Department of Computer and Information Science at National Chiao Tung University in August 1993 and is Professor since 1999. His research interests include design, analysis, and implementation of network protocols and algorithms, wire-speed switching and routing, quality of services, and intranet servers.

Ren-Hung Hwang received the BS degree in Computer Science and Information Engineering from National Taiwan University in 1985, and the MS and PhD degrees in Computer Science from the University of Massachusetts at Amherst in 1989 and 1993, respectively. He joined the faculty of the Department of Computer Science and Information Engineering, at National Chung Cheng University, Chiayi, Taiwan in August 1993. His research interests include design, analysis, and implementation of Internet Technology, Video-On-Demand, Multimedia-On-Demand, Intelligent Networks.

Nai-Bin Hsu received the BS degree from National Taiwan Institute of Technology and the MS degree in Computer Science and Information Engineering from National Chiao Tung University (NCTU), Taiwan, in 1984 and 1990, respectively. He joined Telecom. Lab., Chungli, Taiwan in 1984, where he worked on the project of National Information Infrastructure. He is currently working toward the PhD degree in Computer Information Science from NCTU. His research interests include of QoS routing, and performance evaluation.