A distributed multicast routing algorithm based on bone node set for mobile IP

Ling Zhou*, De Feng Zhang

Department of Computer Science and Technology, Foshan University, Foshan 528000, Guangdong, China

Received 1 March 2014, www.tsi.lv

Abstract

Multicast routing is an important issue in network communication. In order to optimize the multicast routing cost and lessen the transmission delay for mobile IP communication, an idea of bone node set is introduce and the distributed multicast routing algorithm is designed based on the idea firstly. At the same time, the algorithm is implemented according to centre version and distributed version in detail, respectively. Then its necessary data structures, time complexity and message complexity are analysed in theories according to order of sequence for distributed operation. At last, simulation experiments are done in a 7×7 mesh topology and the results show that the designed algorithm can optimize the routing cost for multicast routing and reduce the transmission delay greatly compared to some same type algorithms. The distributed routing algorithm with the simple complexity can be efficiently used in large-scale mobile IP network.

Keywords: Distributed routing, mobile IP, bone node set, performance analysis, simulation

1 Introduction

Multicast is a one-to-many (or many-to-many) data communication method. By multicast communication the sender only needs to send a packet and all the destinations will receive the packet with only one copy of the packet in each link. The key factor to realize multicast is to design a reasonable multicast routing algorithm to construct the data forward tree. Routing cost, time delay and scalability are the three most important parameters to evaluate the multicast routing algorithm. Domestic and foreign scholars have raised some excellent multicast routing algorithm based on traditional IP network and the fixed topology network structure. Recently with the occurrence of mobile nodes (MN), there are some serious challenges to those traditional multicast routing algorithms, which are no longer fitted, to the new mobile Internet environment. Therefore, how to design a multicast routing algorithm for mobile IP (MIP) is becoming a more and more important problem in the current mobile IP research area [1].

IETF has proposed two basic methods to solve the application problem of multicast in mobile IP, that is, the bi-directional tunnel (BT) and remote subscription (RS) [2]. And some scholars have also put forward several improved algorithm according to BT and RS.

With BT algorithm, when the mobile node moves to a foreign subnet, it will acquire a care of address (CoA) from the visiting foreign subnet and register the CoA to its home agent (HA), after that the MN will forward data packet by HA through bi-directional tunnel. Therefore, by BT mode, the MN's movement and handover are transparent to all the other network nodes and it is fit to the current mobile IP architecture. However, BT introduces a tri-angle routing problem and so its routing path is not the most optimization one. In addition, there exists the problem of a longer joint delay due to the long tunnel path, and the focus problem because of different home agents, etc.

With RS algorithm, when a mobile node moves to the visiting foreign subnet, it will apply to join the multicast group by access router (AR) in the foreign subnet and directly receives data packet from AR in foreign subnet. Therefore, in RS method its routing path is the most optimization one. However, RS also introduces some problems. For examples, RS leads to a frequent path reconstruction for multicast routing, and brings about a serious packet loss rate due to high delay in handover.

To reach a compromise between routing path optimization and reconstruction frequency, a range-based mobile multicast (RBMoM) routing algorithmis proposed by Lin C R and Wang K M [3]. The main contribution in RBMoM is that the multicast home agent (MHA) is introduced into mobile IP architecture and the concept of service range is put forward. RBMoM can optimize routing path by the choice of MHA and reduce the frequency of reconstruction by changing the service range. The main disadvantages of RBMoM are the uncertainty of service range and the extra election of MHA.

In order to optimize the handover delay and reduce the mobile IP multicast packet loss rate, a multicast routing algorithm based on mobility support agent (MSA) is proposed by Wu J. MSA is a router in the MN's

***Corresponding author** e-mail: cszhouling@sohu.com

foreign subnet [4], and it is responsible for pre-registration in a visiting foreign subnet when the handover happens. Therefore, MSA can optimize the join delay when handover occurs and reduce packet loss rate. The key work for the algorithm is to predict what time the handover will happen and which direction the MN will move according to.

By trying to solve the problem of packet loss between handover, a multicast routing algorithm called multicast by multicast agent (MMA) is also put forward by Suh Y [5]. The main idea in MMA is to introduce a multicast forwarder (MF) entity, which is a data packet forward router for multicast. When handover happens, MF will buffer the communication data for multicast and later forwards the data to the new subnet. By the collaboration of MF, the packet loss ratio can greatly reduce between the old subnet and the new subnet when the handover occurs.

Handover brings many problems in mobile IP. Beside above strategies to optimize the handover, Tan C L develops an algorithm called MobiCast [6] to handle with it, which is mainly based on the idea of hierarchy, that is, the network is divided into macro layer and micro layer. A macro layer may include several micro layers, and each micro introduces a domain foreign agent (DFA) entity to manage some subnets in the same micro layer. When the MN moves between different subnets (belong to the same micro layer), MobiCast need not to reconstruct the multicast tree and these problems coming from handover can be avoided. When moving between macro layers, DFA is responsible for MN to join the multicast tree, and to reconstruct multicast routing tree.

Above several solutions have been put forward to solve the mobile multicast routing problem by some scholars. Nevertheless, most of all change the architecture of mobile IP by introducing a new network entity to optimize the handover, and few optimizes the network cost and reduces the handover delay according to itself characteristics of multicast [7]. At present, there are still the next important problems to be solved in mobile IP for multicast routing communication. 1) The construction of low cost multicast routing tree. The tree cost computing by the multicast routing algorithm is not the most optimized one and so consumes too much bandwidth to forward data packet; especially in mobile wireless environment, it is difficult to apply. 2) The seamless handover problem. Multicast routing algorithms introduce a relatively large handover delay, which seriously affects the seamless multicast session. 3) The packet loss problem. Due to the high bit error rate of wireless link, the packet loss rate is bigger when MN (or mobile subnet) moves fast; and so a lower multicast session rate will occur. 4) The QoS problem. These algorithms do not consider QoS constraints, for example, delay, delay jitter. 5) The distributed policy. These belong to the centralized routing algorithm, and rarely consider the distributed implementation.

The idea of bone node set is introduced in the paper, and a distributed routing algorithm based on bone node set is designed for mobile IPv6, which is called the bone node set-based multicast routing algorithm (BNSBMR). By BNSBMR algorithm, we can optimize the tree cost for bandwidth management, reduce the handover delay and lessen the packet loss rate. Distributed algorithm is more fitted to employ in large scale of network relative to centralized algorithm and is easier to realize the scalability. Especially in mobile IP environment, the distributed strategy has a unique advantage. The operation of the routing process only relies on the MN, without all the other nodes involved, and which will simplify the operation of mobile IP, optimize the handover, and improve network management performance.

The remainders of this paper are organized by the following way. In the second section, the concept of bone node set is put forward. In the third section, the multicast routing algorithm BNSBMR for mobile IP is described, and the distributed version is implemented in section 4. The distributed data structure, message complexity, and time complexity are analysed in section 5. In section 6 simulation experiments are done to verify the algorithm performance. Finally, in the seventh section we summary the paper and draw some conclusions.

2 Basic conception and model for MIP

A communication network can be modelled by an weighted graph $G=(V, E, W)$, where V is a set of host or router nodes, E is the set of communication links and W is the weight parameter belonged to a specific link. To any links $e \in E$, we can define the cost function as $Cost(e)$: $E \rightarrow R^+$ and the delay function as $Delay(e)$: $E \rightarrow R^+$. Given source node s, destination node set D, the destination node number is $m=|D|$, and the network node number is $n=|V|$.

Definition 1 (path)

Given $G(V, E, W)$, if there exists a node sequence $(s, v_1, v_2, \ldots, v_n, t)$, such that (s, v_1), (v_1, v_2), …, and $(v_n, t) \in E$, so we call those edges as a path, and write the path from s to t as $P(s, t)$.

Definition 2 (Least cost path)

We call the path from u to v a least cost path if the total weighed cost from u to v is the least one and we write the least cost path $p_{lc}(u, v)$.

Definition 3 (Least cost tree)

Given network $G(V, E, W)$, the source node s and a destination set D, if a tree T spans $s \cup D$ and its total cost satisfies the equation:

$$Cost(T) = \min\{Cost(T) = \sum_{v \in D} \sum_{e \in P(s,v)} Cost(e)\}$$

$(\forall v \in D,\ P(s,v) \in T)$,

we call the tree T a least cost tree for $s \cup D$.

Definition 4 (Bone node set)

To mobile IP, the bone node set is a dynamic set of AR (access router) in some selected subnets and written as N_b. The elements in N_b include these subnet's AR where existing static multicast node and these subnet's AR where existing mobile multicast node *i* with the continuous stay time T_i meeting the following condition, $\forall i \in D, T_i > \Delta$, where *i* denotes the mobile node *i*, and Δ is the time boundary. Using *Router*(*i*) to denote the AR of subnet *i*, the bone node set N_b can be expressed as follow,

N_b={*Router*(*i*)|*i includes a static member node*} $\cup$ {*Router*(*i*)|*i includes a mobile member node and* $T_i > \Delta$}.

In MIPv6 architecture, two entities are mainly introduced: mobile node MN and home agent HA [8]. The MIP multicast model can be described as Fig. 1. Server source is the multicast source. Subnet1 is the home subnet of mobile node MN1, and HA is its home agent. MN1 is visiting subnet2 currently, and F is its AR. When MN1 moves from subnet2 to subnet3, handover happens. As MN1 moves into subnet3, it will obtain a care of address (CoA) from router G (that is, subnet3's AR) so that it can re-join multicast tree in subnet3 or accept continuously multicast data from its HA. It can be drawn from the model that the multicast routing problem for mobile IP mainly includes two aspects. The first is how to construct the optimization multicast routing tree. The second is how to deal with handover. The paper will mainly solve the first problem based on bone node set, and also consider handover in some degree.

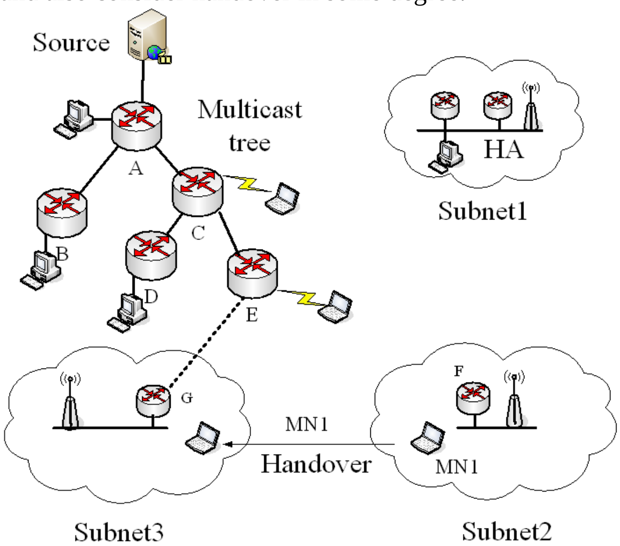


FIGURE 1 Multicast model for mobile IP.

3 Algorithm description

3.1 BASIC IDEA

The basic idea of BNSBMR algorithm is as follows. When source *s* sponsors a multicast session, BNSBMR starts to compute the multicast tree *T* for all destination nodes $\forall i \in D$ and the bone node set N_b, preparatively. The multicast routing tree *T* is used for forwarding data packet, and bone node set N_b is the basis and foundation to optimize the routing path and reconstruct the multicast tree, when mobile node moves from a subnet into other subnet.

When handover occurs, BNSBMR algorithm will search the bone node set N_b and find a node *j* ($j \in N_b$) which makes the *Cost*(*i*, *j*) is the least cost from *i* to bone node set N_b. By bone node set, the new algorithm not only can reduce the multicast tree cost and optimize routing path but also can find a shortest path and lower the join delay.

At the same time, bone node set is not a stationary sharing tree. To mobile node *i*, If $T_i > \Delta$ is meet, the maintenance process for bone node set will be triggered and change the element of N_b. Therefore, the set of N_b is valid and superior.

Bone node set is the core idea of the algorithm. Therefore, how to maintain the bone nodes set is also a key technology of the algorithm. We can illustrate the generating and maintenance process of N_b combined with Figure 1. Supposed there exist seven subnets A, B, C, D, E, F, and G, and accordingly their ARs are also denoted as A, B, C, D, E, F, and G. Simply, supposed there is one and only one multicast node in each subnet. Among them, the multicast nodes in A, B, D are static and the multicast nodes in C, E, F are mobile nodes. 1) When preparatively computing, because there are static/fixed multicast nodes in subnet A, B, D, so N_b = {A, B, D}. 2) When MN1 moves from subnet F to subnet G, MN1 scan N_b and may select AR D to re-join the multicast tree. 3) After a period of time Δ, if the mobile node doesn't move out subnet C, E, so access router C, E can be regarded as bone node and added to set N_b. that is, N_b = { A, B, C, D, E }. If at this time MN1 moves from subnet F to subnet G, MN1 scan the set N_b and may select AR E to rejoin the multicast tree. 4) Supposed that MN1 moves into G and stay in G for a period of time $T > \Delta$ and the mobile node in subnet C moves out C, AR G will become an element of N_b and AR C will be delete from N_b, so N_b = {A, B, G, D, E}.

The specific algorithm and pseudo code are analysed in the following sub-section.

3.2 PREPARATION COMPUTING

At the beginning of the multicast session, source node *s* scans all multicast destination nodes, generates bone node set N_b and initializes routing tree *T*. The procedure is described as shown below.

Input: *G*, *D*, *s*
Output: *T*, N_b
Pre_Cal_Tree(*G*, *D*, *s*)
1. $N_b \leftarrow s$, $T \leftarrow s$
2. $Q \leftarrow D$ // *Q is a list table to record the destination node set D*
3. While (*Q* is not *Null*) Do
4. Select the destination *i* and a node $j \in N_b$ *so that* $P_{lc}(i, j)$ is minimum among $P_{lc}(i, \forall j \in N_b)$

```
  If i is a static node then
    Mark Router(i) as a bone node
    N_b ← N_b ∪ i
    T ← T ∪ P_lc(i, j)
    Q ← Q-i}
  End if
  If i is a mobile node then
    T ← T ∪ Path(i, j)
    Q ← Q-i}
  End if
End while
Return T, N_b
```

3.3 PROCESS OF MAINTENANCE FOR BONE NODE SET

Below we consider the process of bone node set maintenance.

On one hand, if a mobile node *i* stays in one subnet for a period of time $T_i > \Delta$, we can regard it as a stable node and take this subnet's AR as an element of N_b. On the contrary, it isn't regarded as an element of N_b. There is a timer in each mobile node, when handover occurs the timer begins to count the time. If $T_i > \Delta$, the below process Add() will be triggered, and this subnet's AR is labelled as an element of N_b.

Add()

```
If T_i > Δ then
  Mark Router(i) as a node of bone tree
  N_b ← N_b ∪ Router(i)
End if
```

On the other hand, the below process Delete() will be run when mobile node *i* moves out its visiting subnet.

Delete()

```
If there doesn't exist other multicast node connecting to Router(i) then
  Select a parent node for its son nodes
  Mark Router(i) as a non-bone node
  N_b ← N_b-Router(i)
End if
If there exists a loop then
  Remove the loop by changing the node's parent
End if
```

3.4 PROCESS OF ROUTING FOR MOBILE NODE

When mobile node *i* switches to a new subnet from the current subnet, it firstly inquires whether there is multicast members in the new subnet. If there exists, it receives multicast information directly from new subnet's AR and do nothing in reconstructing multicast routing tree. Vice versa, if no multicast members, it starts to run the bellow Cal_Tree process, and scans bone node set N_b to find a node $j \in N_b$, which has the shortest path to itself. After that it grafts into the multicast tree by node *j* and receives multicast data from node *j*.

Input: T, N_b, i

Output: T

Cal_Tree(T, N_b, i)

```
For each node in N_b do
  Find node j ∈ N_b which has the shortest path to i
  T ← T ∪ P_lc(i, j)
  If there exists a loop then
    Remove the loop by changing the parent node
  End if
Return T.
```

It is clear that the new algorithm optimizes the routing path originating from RS method and does not rely on mobile nodes HA. Compared with BT, there are smaller join delay and information transmission delay in the new algorithm. Because BNSBMR optimizes the routing tree cost by bone node set, so compared to other RS method algorithms it has better cost performance and transmission delay.

4 Distributed implementation

According to the scalability and mobility, distributed methods are more suitable for implementing the routing algorithm. In this section BNSBMR algorithm is realized in distributed way. The algorithm idea and the variable symbols are the same as the previous centralized description. Distributed policies mainly include two parts: the distributed maintenance for bone node to renew N_b and distributed routing for mobile node to join the multicast tree.

4.1 TOPOLOGY INFORMATION

Supposed that each node *i* (including MN and router) stores a vector table *Routable*(*j*) and uses the table to record routing information form node *i* to node *j*. In *Routable*(*j*) there are $|V|$-1 entries and each entry is corresponding to one node *j*, including the node index *j*, the shortest path $P_{lc}(i, j)$ and its cost value $Cost(P_{lc}(i, j))$, the next hop node *nh*. Generally, this information can be drawn from the network topology graph.

4.2 ROUTING MESSAGES

The below three kinds of messages is essential to implement the distributed method for the algorithm: Message_BNS_modify(), Message_join(), and Message_setup().

Message_BNS_modify() is used to transmit message among element of N_b to maintain the bone node set, and its parameters include type (Add or Delete), the index of bone node *i* that is generating the message, the index of multicast session *ID*.

Message_join() is originated by the mobile node to select a bone node to join the multicast tree when handover occurs. Its parameters include the index of mobile node *i*, the selected bone node *j*, multicast session index *ID*.

Message_setup() is used to establish a path hop by hop between MN *i* and the selected bone node *j* with the path $P(i, j)$ being the least cost path $P_{lc}(i, j)$. Its parameters include multicast session index *ID*, node *k* which is on the path from *i* to *j*, and *k*'s next hop *next_hop*.

4.3 MAINTAINING BONE NODE SET

Each network node $i \in N_b$ runs the following two processes to add into or exit from bone node set N_b.

Add()

1. Mark *Router*(*i*) as a bone node

2. Add the router *i* to bone node list: $N_b \leftarrow N_b \cup i$

Delete()

1. Mark *Router*(*i*) as a non-bone node

2. Delete the router *i* from bone node list: $N_b \leftarrow N_b - i$

4.4 SELECTING ROUTING PATH

Each member node *i* runs the following process to select a bone node to join multicast tree.

Join()

1. Scan the bone node list to find node $j \in N_b$ which has a shortest path to *i*

2. *next_hop* = *Routable*(*j*). *nh*

3. *Router*(*i*) joins the *path*(*i*, *j*)

4. Return *j*, *next_hop*

Each network node *i* runs the following process to choose the next-hop route and establish routing path hop by hop.

Setup()

1. *Router*(*i*) joins the *path*(*i*, *j*)

2. *next_hop* = *Routable*(*j*). *nh*

3. If there exists a loop then

4. Remove the loop by changing the parent node

5. End if

6. **Return** *next_hop*

4.5 DISTRIBUTED ROUTING PROCEDURES

Step 1. Preparation computing. Running preparation computing process in source node *s* and obtaining the original multicast tree *T* and bone node set N_b.

Step 2. Adding an element into N_b. To an In-tree node, if its $T_i > \Delta$, it runs the Add() process to mark itself as a bone node, initializes the message Message_BNS_modify(*add*, *i*, *ID*) and sends the message to each node in bone node list. For each bone node which receives Message_BNS_modify(*add*, *i*, *ID*) starts its own Add() process to deal with the message.

Step 3. Deleting an element from N_b. If a node belongs to N_b, it runs the Delete() process to delete itself from the bone node list, initializes the message Message_BNS_modify(*delete*, *i*, *ID*), and sends the message to each node in bone node set. For each bone node which receives the message starts its own Delete() process to deal with the message.

Step 4. Starting to establish routing path. When a MN *i* switches to a new subnet and handover occurs, it runs the Join() process to find a node $j \in N_b$ which has the shortest path to *i*, generates the message Message_join(*i*, *j*, *ID*) , scans the *Routable*() to find the next hop *nh* in *Routable*(*j*) and sent the message Message_join(*i*, *j*, *ID*) to *nh*.

Step 5. Continuing to establish routing path. If one node *k* receives the message Message_join(*i*, *j*, *ID*), it runs the Setup() process, generates the message Message_setup(k, *j*, *ID*), and sent the message to next hop *nh* in *Routable*(*j*). Looping procedure 5, the routing path can be set up hop by hop.

Step 6. Finishing the setup process. When bone node *j* receives the message Message_setup(*k*, *j*, *ID*), it records the node index *i*, and sends an acknowledgement to *i*. If MN *i* receives the acknowledgement correctly, the routing path is finished to establish completely and data can be transmitted along the path bi-directionally.

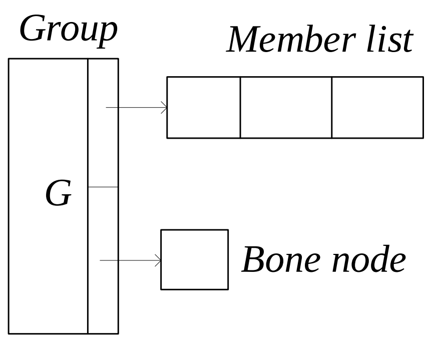
In the routing procedures, there are two possible states for each router: the In-tree state and the Not-in-tree state. In detail, an AR belonging to bone node set may change state in three sub-states: In-tree and bone node state, In-tree and Non-bone node state, and Not-in-tree state. Bone node set maintenance process is operated in each AR. According to the receiving different messages and its current state, each AR may change its states among the three states.

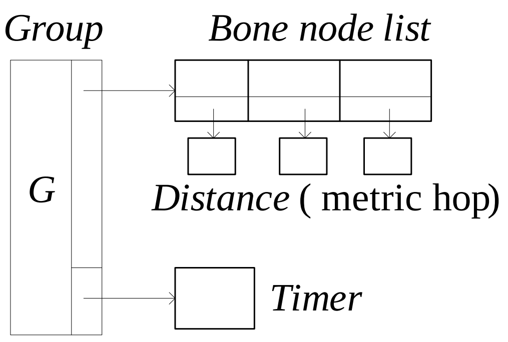
5 Algorithm analysis

5.1 DATA STRUCTURE

It is clear from the above analysis that only the process of preparation computing is run in the multicast source *s*, and the other main parts (including process of maintenance for bone node set and process of routing for mobile node) are run in the mobile entities MN and AR. So we define the data structures for MN and AR as the following Fig. 2. Fig. 2(a) is the data structure for the subnet's AR to deal with mobile multicast routing, where *Group* is used to save the name of multicast group, *G* is a multicast session, *Member list* is used to record *G*'s member nodes, and *Bone node* is boolen and responsible for marking AR whether to be an element in bone node set N_b. Fig. 2(b) is the data structure for MN, where *Group* is used to save the name of multicast group of a mobile node, *Bone node list* records the element of N_b for each multicast group, *Distance* is used to save the shortest distance value/hop from MN to each bone node, and *Timer* is the variable to count the time.

The characteristics of the above data structure are as follows. It maintains a list of bone node in each MN to record the element in N_b for the current multicast. When MN switches from a subnet to other subnet, it scans the bone node list to find the most optimized bone node, which has the shortest distance/least hop to itself to re-join the multicast group. By this method, it can optimize the multicast tree cost and reduce the join delay. In the worst case, its space complexity is $O(n)$, and computational complexity of scanning operations is also $O(n)$, which n is the network scales (that is, the number of network nodes). See the next sub-section time complexity.





a) b)

FIGURE 2 Data structures: a) Data structures for access router AR, b) Data structures for mobile node MN.

5.2 TIME COMPLEXITY

Supposed that the time complexity of generating a message is a constant 1 and the time complexity of transmitting a message is also a constant 1.

In step 4 the routing algorithm scans the bone node list and in the worst case the time complexity is $O(n)$; at the same time the time complexity of generating a Message_join(i, j, ID) is $O(1)$. Step 5 finds the next hop and in the worst case the time complexity is $O(n)$ because the network scale is n. In step 6, bone node returns an acknowledgement message and the time complexity is $O(1)$. So, the time complexity of distributed computing a routing path is $O(n+1+n+1)= O(n)$.

In steps 2 and 3 when a node i is added to or deleted from N_b, it generates a message Message_BNS_modify() with time complexity $O(1)$ and sends the message to each element in N_b. Because the number of the element in N_b is not more than n-1, so the time complexity is $O(1+n - 1)=O(n)$.

So, the total time complexity of the distribute algorithm is $O(n)$.

5.3 MESSAGE COMPLEXITY

Supposed that the number of network is n, in the worst case there may be n nodes and n-1 edges on the path from source s to any one destination node. In this case, the message complexity of new algorithm in step 4 and 5 is $O(n)$. In step 6 message complexity is $O(1)$. Therefore, in the worst case the message complexity is $O(n + 1) = O(n)$.

When a node i is added to or deleted from N_b (steps 2 and 3), it generates a message Message_BNS_modify() and sends the message to each element in N_b. It is clear that the message complexity is also $O(n)$.

So, the total message complexity of the distribute algorithm is $O(n)$.

6 Simulations

We assume that there are 49 subnets (a 7×7 mesh) in the simulation experiment [9, 10]. Each subnet includes a HA, a base station and an AR which can also perform as a multicast router if necessary. Assume the multicast source lies in the centre subnet of the topology and the multicast group members are randomly placed among the 49 subnets. Suppose that the distance count between the adjacent subnets is 1 hop.

In order to simulate mobility of MN, we define the mobility model as a 2-tuple (D, T), where D is the direction to move and T is the sojourn time to stay in the newly visiting subnet [11]. We assume that the mobile member node may randomly roam to each adjacent subnet. In the simulation the sojourn time T is exponentially distributed with the mean value 10 minutes. Besides, we also assume that there is only a multicast group with a single multicast source in the simulation. Initially the multicast tree includes only the source node and then some mobile nodes are randomly selected to add or remove from the multicast tree.

The main parameters and their values see Table 1.

TABLE 1 Simulation parameters

Parameters	Description	Value
N	Number of LANs	7×7
M	Number of multicast group	1
n	Number of group members	5-40
s	Sources per multicast group	1
D	The direction to move	Random
T	The mean sojourn time	Exponential distribution
Δ	The lower boundary for a bone node	20 minutes

The objective of the simulation experiments is to test the performance of BNSBMR and compare to three excellent same types of algorithm. Tested parameters include tree cost and the average transmission delay. The two parameters can be defined as follows.

Tree cost: $Cost = \sum_{e \in T} Cost(e)$.

Average transmission delay: $Trans\_delay\_average = (\sum_{i \in D} \sum_{e \in P(i,s)} Delay(e)) / n$.

Parameter performances are tested by the following simulation experiments. Because RS and BT are the two basic multicast routing algorithms for mobile IPv6, and some other algorithms are originated or improved from them, we test and compare BNSBMR with the two algorithms. At the same time, due to RBMoM being a relatively successful same type algorithm in mobile IP area for multicast routing, we also compare BNSBMR's performances with RBMoM. When simulations are done, the number of destination/member nodes in equilibrium varies from 5 to 40, each time increasing 5, and all the member nodes are mobile nodes. We run a set of 1000 randomly generated handovers for each data point and calculated the average value over all runs as the final experiment data.

1) The tree cost. Firstly, the tree cost is tested and the simulation results are shown in Figure 3. The cost performance BNSBMR is the most optimized one in the four algorithms. RS algorithm also has a good cost performance. BT algorithm has the worst cost performance. The cost performance of RBMoM is between RS's and BT's. With the increasing of mobile nodes number from 5 to 40, the cost gap among the four algorithms becomes more and more serious. The reason is that BT relies on home agent HA and has a long transmission tunnel, which makes no any routing optimization. RS re-joins multicast tree from visiting subnet directly, as makes the cost performance is better. Because BNSBMR uses backbone node set to optimize the cost performance and bases on RS mode to reconstruct multicast tree, so it achieves the best cost performance. RBMoM is a compromise of RS and BT. Simulation results also verify the four algorithms' design idea well.

2) The average transmission delay. Average transmission delay is an important aspect for routing algorithm to provide QoS guarantee ability, which is mainly related to the path from the source node to the destination node. Experimental results are shown in Figure 4 according to the relation of the average transmission delay and the number of members. RS has the most optimal performance in transmission delay because a mobile node in RS method reconstructs routing tree directly by external AR, which cause the shortest path and the least transmission delay to the multicast source.

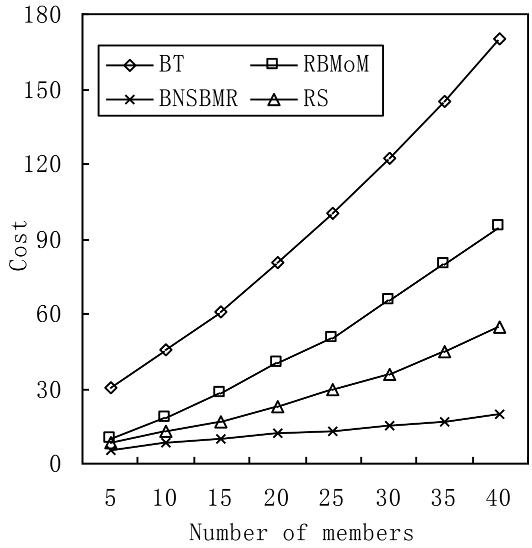


FIGURE 3 Relation of cost versus member number

The transmission delay performance of BNSBMR is only a little worse than RS but much better than RBMoM and BT, thanks to the bone node set used to share the routing path. The transmission delay performance of RBMoM is between BNSBMR's and BT's (supposed that the radius of parameters in RBMoM is R=2). BT has the worst transmission delay value also because it introduces a long delay by bi-direction tunnel in HA.

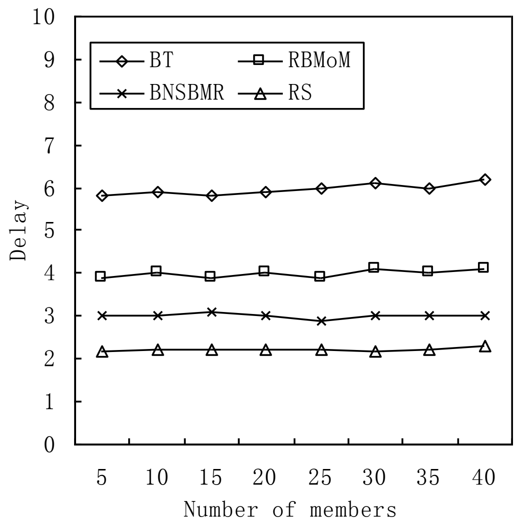


FIGURE 4 Relation of transmission delay versus member number.

7 Conclusions

In this paper, a distributed multicast routing algorithms BNSBMR is proposed based on bone node set for mobile IP, which can be efficiently used in large-scale mobile IP network and mainly has the following advantages.

1) The distributed algorithm is based entirely on MIPv6 protocol, and there is no extension or change in MIPv6 architecture. Also BNSBMR doesn't introduce any new protocol and entity and only relies on mobile

node MN and access router AR. In the process of running it needs only multicast source, MN, AR without any other irrelevant nodes.

2) BNSBMR optimizes the cost performance of multicast routing tree by bone node set, which is beneficial to the wire/wireless network bandwidth management.
3) The new algorithm optimizes the transmission path of multicast routing tree by bone node set, which reduces the average join delay when handover occurs and lessens the average transmission delay when data is transmitted. It is clear that this is also helpful to realize the smooth handover.
4) BNSBMR is implemented based on distributed method which only depends on the bone node set with computing complexity $O(n)$. It makes the algorithm a good scalability, which is beneficial to realize multicast in large-scale mobile network.

In the future, we will do more experiments and test more parameters to verify the designed algorithm.

References

[1] Gossain H, Cordeiro C M, Agrawal D P 2002 *IEEE Commun. Mag.* **40**(6) 116-23
[2] Kuntz M R, Noel J T 2013 *IEEE Commun. Mag.* **51**(1) 128-35
[3] Lin C R, Wang K M 2002 *ACM Wirel. Netw.*, **8** 27-36
[4] Wu J, Magnire G Q 2002 *IFIP Proc. PWC 2000*, Gdansk.
[5] Suh Y, Shin H, Kwon D 2001 *ACM Wirel. Netw.* **7**(5) 443-53
[6] Tan C L, Pink S Mobicast 2000 *ACM Mobile. Netw. Appl.*, **5** 259-72
[7] Seok J K, Gohar M 2010 *IEEE Commun. Lett.*, **14**(7) 676-78
[8] Zhou L, Sun Y M 2008 *J. Comput. Res. Dev.t*, **45**(7) 1126-32 (*in Chinese*)
[9] Madden S, Levis P 2008 *IEEE Internet Comput*. **12**(4) 9-11
[10] Zhou L, Sun Y M 2006 *IEEE Proc. Int. Conf. Wirel. Commun. Netw. Mobile Comput.* 22-24
[11] Suh Y, Shin H, Kwon D 2001 *ACM Wirel. Netw.* **7**(5) 443-53

Authors

Ling Zhou, born on September, 1972, Yueyang, Hunan, China

Current position, grades: Associate Professor; University of Foshan
University study: Hunan Normal University, Changsha, 1998
Ph.D. degree in computer science and technology, Nanjing University of Science and Technology, Nanjing, 2007.
Research activities: Computer Networks, Wireless Sensor Networks, Data Communication, and Routing Algorithm and Protocol.
Publication: more than 10 papers in journals and international conferences including International Journal of Computer Networks & Communications, Journal of Computer Research and Development, Journal of System Simulation, IEEE proceedings of WiCom'2006, IEEE proceedings of CDCIEM'2012, etc.
Experience: From July 2001 to June 2007, he has been a Lecturer in the Department of Communication Engineering, University of Changsha, Hunan, China. Since July 2007, he has been with Foshan University in the Department of Computer Science and Technology, and is currently an Associate Professor of computer science. Dr. Zhou is a member of CCF.

Defeng Zhang, born on September, 1963, Dalian, Liaoning, China

Current position, grades: Professor, University of Foshan,
University study: Harbin Institute of Technology, Harbin, 1990.
Research activities: Mathematics, Pattern Recognition, Statistical Decision Theory, Adaptive Control, and Computer Networks.
Publication: more than 20 papers in journals and international conferences since 1993, including Journal of System Simulation, Acta of Sunyatseni University, Computer Engineering, etc.
Experience: Since September 1993, he has worked in Foshan University in the Department of Computer Science and Technology, Guangdong, and is currently a Professor of computer science and automatic control. He is also a senior member of CCF.