

Multicast Protocol for Uni-Directional Networks

HUANG Hao (黄 皓), CHEN Guihai (陈贵海), XIE Li (谢 立)
and SUN Zhongxiu (孙钟秀)

*State Key Laboratory for Novel Software Technology
Department of Computer Science and Technology
Nanjing University, Nanjing 210093, P.R. China
E-mail: hhuang@distab.nju.edu.cn*

Received October 8, 1998; revised May 26, 1999.

Abstract A very inexpensive receive-only satellite receiver can receive high bandwidth traffic from a feeder. Therefore the connection between the feeder and the receiver is uni-directional. The existing routing protocols, such as Link-State and Distance-Vector, are designed on the premise that any links are bidirectional and they cannot handle the uni-directional links. In this paper, a dynamic multicast routing protocol is proposed, which can handle uni-directional networks. This protocol can also adapt to the dynamic change of the topology of the network and has good scalability. A formal description of the protocol by Petri net is given. Liveness, deadlock free and other properties of the protocol are proved.

Keywords routing protocol, multicast, unicast, uni-directional network

1 Introduction

Current routing techniques assume that pairs of routers are connected by bi-directional links (BDLs). Uni-Directional Links (UDLs), i.e., links with zero return bandwidth, are nonetheless emerging. For example, to achieve the ultimate goal of "any where any time" access to communications and the global Internet, we must exploit the satellites. Satellite communication systems vary substantially in data rate, ground station size, power requirement, and portability. There are small hand-held or briefcase-sized systems that can send and receive data in a few Kbps, and there are also large dish (several meters in diameter) terminals that can send and receive data in several hundred Mbps. Among many different systems, a digital direct broadcast system (DBS) offers a good trade-off between portability and functionality of a user terminal. In a DBS system, as shown in Fig.1^[1], information is collected at an uplink center that is denoted by *A*. The message is transmitted from uplink center to a broadcast satellite, and is retransmitted back to a terminal on the earth surface, which is denoted with *B*, *C*, and *D*. The data rate is as high as dozens of megabit per second. The DBS system has the most inexpensive and portable high data rate terminals allowed by today's technologies. But the terminals on the earth surface are receive-only on the satellite link and have no uplink capability. Being receive-only on the satellite link, they operate at very low power consumption and are good for remote rural operations. The antenna is a dish of a fraction of a meter in diameter, which is highly portable and easy to deploy. Because the terminal is receive-only on the satellite link and has no uplink capability, the uni-directional link occurs. For example, the links from point *A* to points *B*, *C* and *D* are uni-directional links, though nodes *B*, *C* and *D* can send message on other links, such as PSTN or cellular network, to Internet. In Fig.1, IAP for Internet Access Point. PSTN for Public Switched Telephone Network and CDPD for Cellular Digital Packet Data.

It is supported by the National Natural Science Foundation of China for Young Scientists (No.89803005), and Telecommunication Advances Organization (TOA) of Japan.

The techniques used by current routing protocols in Internet, such as link-state (LS)^[2,3], distance-vector (DV)^[4,5], and path vectors (PV)^[6], are designed on the premise that links are bi-directional, but they do not work in networks with uni-directional links. Routers connected to an "outgoing" uni-directional link (UDL) cannot be aware of the existence of networks reachable through this UDL. Routing protocols based on LS mechanism, such as OSPF^[2] and MOSPF^[3], are not applicable in uni-directional networks because their routing message must be "acknowledged" but the "acknowledgement" cannot be sent back to the routing message sender along the other direction of the link on which the routing message is received. Routing protocols based on DV, such as RIP^[4] and DVMP^[5], are also not applicable in uni-directional networks because one node cannot deduce its own distance vector from all its predecessors' routing information. PIM-SM^[7] is a multicast routing protocol and it also stands on the fact that all links are bi-directional and so it is also not applicable in the uni-directional networks.

IETF (Internet Engineering Task Force) has formed a UDLR (Uni-directional Link Routing) working group to determine a short-term workaround solution. Two approaches have been proposed: modifying routing protocols^[8-10] or creating virtual bi-directional tunnels^[11,12]. They are all designed on the basis of an external return path. That is, the receiver (the receiving node of the two nodes connected by a uni-directional link) knows a path to the feeder (the sending node) in advance or by the unicast routing. If the rest of the network is bi-directional, the external return path can be set up properly. But if the external return path must pass another uni-directional network, there may be a routing bootstrap problem — the external return path cannot be resolved until after that of the dynamic routing of another uni-directional network being resolved - a possible routing deadlock scenario.

In another paper,^[13] we have proposed a unicast routing protocol for uni-directional network. It is not needed to know the external return path from the receiver to the feeder in advance and it can adapt to the dynamic change of network topology. In this paper, we propose a multicast protocol for uni-directional networks (MPUN) based on the unicast protocol for uni-directional networks and some useful idea of PIM-SM.

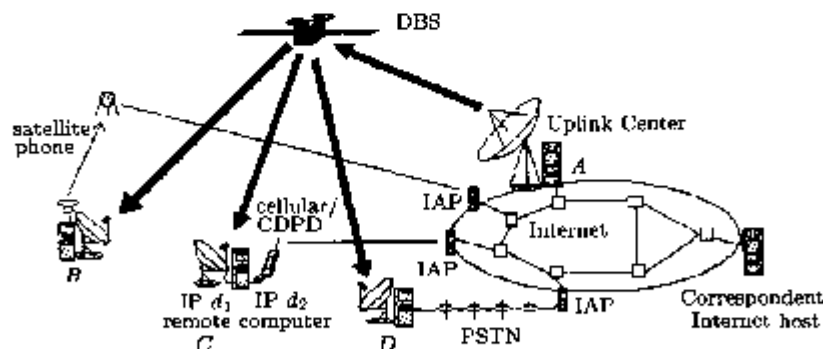


Fig.1. An example of uni-directional network.

2 Overview of the Multicast Protocol for Uni-directional Network

PIM-SM^[7] is an efficient multicast routing protocol, which uses IGMP (Internet Group Management Protocol) and unicast routing table to establish a multicast distribution tree. Because IGMP and PIM-SM are designed on the premise that all links are bi-directional and protocol control message arriving at an interface is responded on the same interface, IGMP and PIM-SM cannot work properly on the uni-directional networks.

Our MPUN does not exploit the IGMP. It is based on the unicast routing table, which

is established by unicast protocol for uni-directional networks and some useful ideas from PIM-SM.

All routers running protocol MPUN form an MPUN domain. In an MPUN domain, there is a unique bootstrap router (BSR) which is elected dynamically and there is a unique Rendezvous Point (RP). All routers work as a candidate BSR (C-BSR) as well as a candidate Rendezvous Point (C-RP). It sends periodic C-RP advertisement to the current BSR so that BSR stores it to its local pool of C-RP and broadcasts these RPs to all MPUN routers later. Each router stores a C-RP set when it receives a proper bootstrap message. When a router wants to participate in a group to receive packets from the group or send packet to the group, it uses prescribed hash function to calculate an RP for the group from the stored RP set. Then it sends periodic JOIN message to the RP to join an RP rooted distribution tree for the group. When the RP receives JOIN message, it sends JOIN ACK message to the JOIN message sender. Through the JOIN and JOIN ACK message the RP rooted multicast distribution tree is constructed. It can then send a packet to the RP for the group so that the packet can be distributed on the RP rooted tree to all group members and receive packet from the group. The construction of multicast distribution tree is triggered by JOIN message sent by the group members.

When MPUN constructs the RP rooted distribution tree, it exploits unicast routing table for uni-directional networks. To be clear, in Subsection 3.1 we will give the outline of our unicast protocol for uni-directional networks^[13] which is referred to as UPUN. In UPUN, each node maintains two tables, incoming and outgoing tables, consisting of several entries. MPUN uses these two tables when it constructs a multicast distribution tree.

3 Multicast Protocol over Uni-directional Network

In this section, we describe the multicast protocol for uni-directional networks. The algorithm uses the existing unicast routing table to establish multicast distribution tree. Subsection 3.1 gives the outline of the unicast protocol for uni-directional networks. Subsection 3.2 defines the broadcast, which is used to distribute bootstrap message to all MPUN routers. Subsection 3.3 describes the bootstrap method, which is used to dynamically select a Bootstrap Router. Subsection 3.4 describes the method of constructing an RP rooted distribution tree. Subsection 3.5 gives the method of distributing packets on the RP rooted distribution tree.

3.1 Construction of Unicast Routing Table

For clarity, we suppose the local router is A in this subsection. In this subsection the method of constructing the incoming table and the outgoing table will be presented. The incoming table and outgoing table both contain several entries. Each entry of the incoming table consists of five fields which can be represented as a vector (NODE, DIST, NEXT, IFN, IFIN). This entry indicates that from the router NODE to the local router A there is a path, the distance of the path is DIST, the NODE is the first router in this path and the second router in this path is NEXT, the NODE can send message on interface IFN to the second router NEXT, and a packet from NODE to the local router through this path will arrive at the local router on the interface IFIN. Each entry of the outgoing table consists of four fields which can be represented as a vector (NODE, DIST, NEXT, IF). This entry indicates that there is a path from the local router A to router NODE, the distance of this path is DIST and a packet can be sent on the interface IF through this path to NODE. $NODE(e)$, $DIST(e)$ and $NEXT(e)$ represent the NODE field, DIST field and NEXT field of an entry e in the incoming table or the outgoing table. $IFN(e)$ and $IFIN(e)$ denote the IFN field and IFIN field of an entry of the incoming table respectively. $IF(e)$ is the IF field of an entry e

of the outgoing table. For simplicity, we suppose that the distance of each link is 1.

3.1.1 Construction of Incoming Table

- I. The initial state of the incoming table is empty.
 - II. Each router periodically sends on all its outgoing interfaces a packet containing the outgoing interface name and its incoming table even though the incoming table is empty.
 - III. When the local router A receives a packet from its parent router containing an interface J , on which its parent router sends this packet, and an incoming table T on its interface I in A , it will check for each entry in the received incoming table T to see whether there exists an entry in the incoming table of the local router A so that the NODE fields of the two entries are equal. If there exist two such entries, they are referred to as matched entries.
 - IV. If the entry e_1 in the received incoming table T and the entry e_2 in the entry of the local incoming table are two matched entries and $\text{DIST}(e_1) + 1 < \text{DIST}(e_2)$, then the entry e_2 in the local incoming table is replaced by the entry $(\text{NODE}(e_1), \text{DIST}(e_1) + 1, \text{NEXT}(e_1), \text{IFN}(e_1), I)$. Restart the entry timer of this entry.
 - V. When the local router A receives a packet from its parent router P containing an interface J , on which its parent router sends this packet, and an incoming table T on its interface I in A , it checks whether there exists an entry in the incoming table of the local router A whose NODE field is the same as the sender P . If there is not such an entry in the incoming table of the local router A , then it adds the entry $(P, 1, A, J, I)$ to the incoming table of the local router and starts the entry timer for this entry. If there is such an entry and it is the same as $(P, 1, A, J, I)$, then restart the entry timer of this entry. Note that the interface J in the IFN field of the entry $(P, 1, A, J, I)$ is the same interface included in the packet from its parent router P .
 - VI. Each router checks each entry timer of the incoming table periodically. If an entry timer is timeout, the entry is deleted from the incoming table.
- If the sender of a packet P is S and there is an entry e in the local incoming table whose NODE field is the same as S , then $\text{IFIN}(e)$, the IFIN field of e , is denoted as $\text{iif}(S)$ or $\text{iif}(P)$.

3.1.2 Construction of Outgoing Table

Before describing the method of constructing the outgoing table, we first show how to find a path to the parent router. If a router A receives an incoming table which contains an entry e whose NODE field is the same as A , it indicates that there exists a path from local router A to the sender P of the incoming table. As there is a path from the router $\text{NEXT}(e)$ to the router P , there is also an entry whose NODE field is the same as $\text{NEXT}(e)$ in the received incoming table. As a result, it can find serial entries e_1, \dots, e_n , such that $\text{NODE}(e_1) = A$, $\text{NEXT}(e_1) = \text{NODE}(e_2), \dots, \text{NEXT}(e_n) = P$. By the tuple sequence $(\text{NODE}(e_1), \text{IFN}(e_1)), (\text{NODE}(e_2), \text{IFN}(e_2)), \dots, (\text{NODE}(e_n), \text{IFN}(e_n))$, router A can send a packet to its parent router using source routing method, that is, the router $\text{NODE}(e_1)$ sends packet on the interface $\text{IFN}(e_1)$ to $\text{NODE}(e_2)$, router $\text{NODE}(e_2)$ sends packet on the interface $\text{IFN}(e_2)$ to $\text{NODE}(e_3), \dots$, the router $\text{NODE}(e_n)$ sends the packet on the interface $\text{IFN}(e_n)$ to router P . The construction of the outgoing table is as follows:

- I. The initial state of the outgoing table is empty.
- II. When the local router A receives a packet from its parent router containing an interface J , on which its parent router sends this packet, and an incoming table T on its interface I in A , it tries to find a path to its parent router by the method mentioned above. In order to find the path from the local router to its parent router sending the packet containing the incoming table, it only needs to check whether there exists an entry in the received incoming table T whose NODE field equals the local router A . If it finds such an entry e , it sends a packet containing interface J and its outgoing table through the interface $\text{IFN}(e)$ to its parent by source routing method even though its outgoing table is empty.
- III. When the local router A receives a packet from its child router S containing an interface J , on which the local router A can reach the router S , and an outgoing table T of the router S , it

will check each entry in the received outgoing table T_r to see whether there exists an entry in the outgoing table of the local router so that the NODE fields of the two entries are the same. If there exist two such entries, we call to them as matched entries.

IV. If the entry e_1 in the received outgoing table T_r and the entry e_2 in the local outgoing table are two matched entry $\text{DIST}(e_1) + 1 < \text{DIST}(e_2)$, then the entry e_2 in the local outgoing table is replaced by the entry $(\text{NODE}(e_1), \text{DIST}(e_1) + 1, S, J)$. Restart the entry timer of this entry. If the entry e_1 in the received outgoing table and the entry e_2 in the local outgoing table satisfies $\text{NODE}(e_1) = \text{NODE}(e_2)$, $\text{DIST}(e_1) + 1 = \text{DIST}(e_2)$, $\text{IF}(e_2) = J$, then restart the entry timer of the entry e_2 . Note that the interface J is included in the received packet mentioned in III.

V. When a router receives an outgoing table mentioned in III, it checks whether there exists an entry in the outgoing table of the local router whose NODE field is the same as the sender S of this received outgoing table. If there is not such an entry, then add the entry $(S, 1, S, J)$ and start the timer for this entry. If there is such an entry and its IF field is the same as J , then restart the entry timer of this entry. The interface J is what is included in the received packet mentioned in III.

If a router receives a packet from a source router S on the interface J , it will check the entry in the local incoming table whose NODE field is S to see whether the IFIN field equals the interface J . If the IFIN field equals the interface J , we say the packet comes on to the proper interface. Only the packets coming on to the proper interface are processed, while other packets are dropped.

3.2 Broadcast

When a router receives a broadcast packet sent by a source router S it checks whether the incoming interface of this packet matches with the interface $\text{iif}(S)$ of the entry in the incoming unicast routing table. If it matches, this packet is forwarded to all other interfaces except the interface $\text{iif}(S)$. Otherwise, this packet is dropped.

3.3 Bootstrap

In order to obtain the C-RP information, all routers within an MPUN domain collect bootstrap messages. The domain's BSR originates the bootstrap messages and broadcasts to all MPUN routers. Bootstrap message includes fields as follows:

Table 1. Bootstrap Message

BSR address	priority	group no.	group address 1	C-RP set 1	...	group address n	C-RP set n
-------------	----------	-----------	-----------------	------------	-----	-----------------	------------

BSR address field is the IP address of BSR. Priority field is the priority of the current BSR. Each router in an MPUN domain has unique priority. The "group no." indicates the number of groups for which this BSR message advertises its C-RP set. For all i from 1 to n , the "group address i " field and "C-RP set i " field indicate that in this MPUN domain the candidate RP for the group "group address i " is "C-RP set i ". Bootstrap messages are used to carry out a dynamic BSR election and to distribute C-RP set information to all MPUN routers within the domain. The candidate BSR with the highest priority will be elected as active BSR. Each router is configured as a candidate BSR as well as a C-RP. BSR periodically broadcasts bootstrap messages to all MPUN routers. The bootstrap algorithm is as follows:

I. Each router starts a bootstrap-timer after initiating. If the bootstrap-timer expires, the router broadcasts a bootstrap message, which includes its own IP address in the "BSR address" field and its own priority in the "priority" field, and fills in the "group no." field and for all i from 1 to n fills in "group address i " and "C-RP set i " fields according to its own C-RP pool.

II. The BSR broadcasts a bootstrap message periodically. When the BSR originates a bootstrap message, it fills in the "BSR address" field and "priority" field with its own IP address and its own

priority. The BSR fills in the "C-RP set" fields and "group address" fields according to its C-RP pool. Each MPUN router periodically sends C-RP advertisement to the current BSR and the BSR puts it into its C-RP pool so that it can be included in the "C-RP set" field in the following bootstrap messages.

III. When a router receives a bootstrap message, it first checks whether the bootstrap message comes onto the proper interface. If the bootstrap message does not come onto the proper interface, the message is dropped.

IV. If the bootstrap message comes onto the proper interface and the priority of the BSR included in this message is higher than the currently active BSR's and itself's, the receiving router stores the bootstrap message, that is, stores the "BSR address" field of this bootstrap message as the "current BSR address" and the C-RP sets information. It restarts its bootstrap-timer. Then it broadcasts this bootstrap message.

In an MPUN domain, all routers negotiate a Hash function f that takes the group address and candidate RP's IP address as its variables. When a router wants to know the RP for a group G , it selects the router from the C-RP set at which the Hash function obtains its maximum value. If the Hash function obtains its maximum value at more than one C-RP, it selects the one whose IP address is the smallest.

3.4 Creation and Maintenance of Shared RP Rooted Tree

Routers use route entries to maintain a multicast distribution tree. We use the term route entry to refer to the state maintained in a router to represent the distribution tree. A route entry includes such fields as follows:

Table 2. A Route Entry

group address	RP address	lif	oif-list	entry timer	some other useful flags
---------------	------------	-----	----------	-------------	-------------------------

The "group address" field denotes the group that this entry stands for. The "lif" field represents the only interface from which a multicast packet can be accepted. oif-list is the interface set over which a multicast packet for the group must be sent. For each oif, the router containing this entry sets an oif timer for it. We use the RP rooted tree as multicast distribution tree. Any data packet will be distributed to all group members over the RP rooted tree. In other words, the RP rooted distribution tree is shared by all members of a group and is referred to as RPT hereafter. The method of creating the RPT is described as follows.

I. If a router wants to participate in a group G , it creates a route entry, starts an entry-timer for it and sends a JOIN message to the RP for the group G . When the entry-timer expires and it still wants to receive packets from the group G or send packets to the group G , it sends a JOIN message to the RP again and restarts the entry timer.

II. When the RP receives a JOIN message from a router, it will send a JOIN ACK message to the sender of the JOIN message according to the unicast routing table.

III. When the intermediate router along the path from RP to the sender of the JOIN message receives the JOIN ACK message from RP, it checks whether there is a route entry for this group in this intermediate router.

IV. If it has not the existing route entry for this group, then it creates such a route entry, adds the interface, over which it reaches the sender of the JOIN message according to outgoing table, to the oif-list of the newly created route entry and starts the entry timer for this entry and oif timer for this oif. It also fills in the "RP address" field and "lif" field accordingly.

V. If there already exists a route entry but its oif-list does not include the interface, over which it reaches the sender of the JOIN message, then it adds this interface to the oif-list of this route entry and starts the oif-timer for this outgoing interface newly added. If there already exists a route entry and its oif-list includes the interface, over which it reaches the sender of the JOIN message, then it restarts oif-timer for this outgoing interface.

VI. Forward this JOIN ACK message via the next hop to the sender of JOIN message according to outgoing table.

contains tokens $\langle i, j, k \rangle$, $i = 1, \dots, L$, $j = 1, \dots, M_k$, $k = 1, \dots, N$ and tokens $\langle i, k \rangle$, $i = 1, \dots, L$, $k = 1, \dots, N$. If token $\langle i, k \rangle$ is located in position F , it means that router is not a member of group i . If token $\langle i, k, T \rangle$ is located in position A , it means that router sent JOIN message to RP at the moment T and now is a member of group i . If token $\langle i, j, k \rangle$ is located in position F , it means that when router k receives a message for group i it need not send this message onto interface j , otherwise if token $\langle i, j, k, T \rangle$ is located in position A it should send message onto interface j for group i . In this Petri net, there is a time parameter t , whose value always equals the current time. For example, when transition t_1 fires at time T , it puts token $\langle i, j, k, T \rangle$, in which T is the time when transition fires, to position A .

In Fig.3, $p(i, RP, k)$ is the token set $\{\langle i, j, k' \rangle$ node k' is located in the path from RP to node k , j is an interface of node k' through which one can reach the node k according to unicast routing table}. When the transition t_2 or t_5 fires, it sends token set $p(i, RP, k)$ to position IIO or CIO . This means that RP sends JOIN ACK message to the JOIN message sender k , and all the routers along the path from RP to the JOIN message sender k will add the corresponding interface to oif-list of the route entry for group i .

Firing of transition t_1 means the following.

- 1) The router k , which is an intermediate router on the path from RP to a JOIN message sender, receives a JOIN ACK message from RP for the group i .
- 2) The router k can reach the JOIN message sender over interface j according to the unicast routing.
- 3) In router k , there is not an entry for group i or the interface j is not in the oif-list of this entry for group i .
- 4) If there is not such a route entry, then it creates one for group i and adds the interface j to the oif-list of the route entry for the group i . The router adds the interface j to the oif-list of the route entry for the group i .

Firing of transition t_2 means that this router itself wants to participate in the group i and it sends JOIN message to RP for the group i . The conditions $1 \leq i \leq L$ and $1 \leq k \leq N$ should be observed.

Firing of transition t_3 means that at the moment t the timer for sending JOIN message would expire and the router k does not want to participate in group i anymore. As a result, some of the interfaces corresponding to $p(i, RP, k)$ will expire and may be deleted from the oif-list of router k later by timeout mechanism.

Firing of transition t_4 means that the timer of the interface j in the route entry for the group i in router k has expired and the interface j must be deleted from the oif-list of the route entry for group i in router k .

Firing of transition t_5 means that the timer for group i in router k has expired and it still wants to participate in group i . Therefore it sends JOIN message again to the RP.

Firing of transition t_6 means that the router k sends JOIN message to RP again, and the RP sends JOIN ACK message back to it and all intermediate routers from RP to router k reset all timer for corresponding interface in the route entry for group i .

Lemma 1. *No matter what state the Predicate/Transition net Σ is in, it can return to the initial state after serial firings.*

Proof. It is obvious that after transitions t_3 and t_4 fire several times, the place A is empty and place F contains all initial tokens. So the Predicate/Transition net Σ returns to the initial state.

Lemma 2. *In the initial state, any transition is a live.*

Proof. There are 6 transitions in the Predicate/Transition net Σ . It is evident that transitions 1 and 2 are fireable in the initial state. After the transition 1 or 2 is fired, transitions 3, 4, 5 and 6 are fireable. So all transitions are alive.

Lemma 3. *The Pr/T net Σ_1 is alive.*

Proof. Due to Property 1 and Property 2, it is evident that the Pr/T net Σ_1 is alive.

From Lemma 3, we can conclude that the routing protocol MPUN is deadlock free.

4.2 Correctness of RPT Tree

Lemma 4. *The group G specific RP rooted distribution tree contains all members of the group G .*

Proof. Let $N = (D, E)$ be the directed graph of uni-directional network we discussed. It is evident that the RP rooted distribution tree for group G contains the router A_n , if and only if $A_1(A_1, A_2)A_2 \cdots A_{n-1}(A_{n-1}, A_n)A_n$ is a path in the graph N , i.e. $A_i \in D$, $i = 1, 2, \dots, n$, $(A_i, A_{i+1}) \in E$, $i = 1, 2, \dots, n-1$, $A_1 = RP$ and there is an entry for the group G in router A_i and the interface from A_i to A_{i+1} is included in the oif-list of this route entry for all i from 1 to $n-1$. Now let A_n be a member of the group G . According to protocol MPUN we described above, the router A_n must periodically send JOIN message to RP for group G and RP also sends JOIN ACK to router A_n . As a result each router A_i along the path from RP to A_n creates a route entry for group G and adds corresponding interface from A_i to A_{i+1} to the oif-list of the route entry for group G or restarts oif-timer of the related interface. In other words, as long as A_n is a member of group G , RP rooted distribution tree contains the router A_n .

4.3 Uniqueness of BSR in an MPUN Domain

Lemma 5. *In the MPUN domain, only one router can be elected as BSR.*

Proof. Suppose router A has the highest priority. When it receives a BSR message, it can detect that the priority of itself is higher so it will not store this BSR message. As a result the bootstrap timer expires and it broadcasts BSR message whose BSR address field is the address of its IP address. Any router in the MPUN domain receiving this BSR message will select it as bootstrap router as the priority of router is higher. In other words, only router A will be elected as bootstrap router.

5 Simulation

A simulation experiment for the MPUN proposed above has been made. A simulator has been created in a Network Of Workstation (NOW) environment in our laboratory and 10 IBM RS-6000 workstations are exploited. The architecture of the simulator is shown as Fig.4.

- The 'Virtual Network' modular provides an interface for upper modular. It provides an interface for upper modular and makes the Ethernet look like a uni-directional network. In other words, we make a simulated uni-directional network. Each node has a text file describing the topology of the uni-directional network. At the very beginning, the modular Virtual Network reads this file and establishes necessary TCP connections between pairs of nodes according to the topology of uni-directional network. Nodes exchange routing messages on these TCP connections. A node only processes messages from the appropriate nodes according to the topology of the uni-directional network.

- UPUN implements the unicast protocol for uni-directional network^[19]. It periodically sends routing incoming routing table and outgoing routing table to neighboring son nodes, to which there is a uni-directional link from this node, on the corresponding sockets created by 'Virtual Network' modular. Using these routing messages, UPUN can establish unicast routing table according to the algorithms in Subsection 3.1.1 and Subsection 3.1.2.

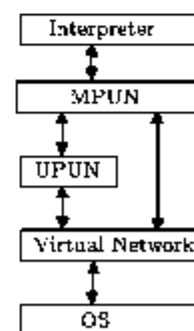


Fig.4. The simulated architecture of MPUN.

- MPUN is a modular that implements the multicast protocol for uni-directional network we described in Section 3. It contains three major parts.
 - ◆ It handles bootstrap message as described in Subsection 3.3. It maintains a table, which maps a group to its current RP.
 - ◆ It contains a module processing attending group request as described in Subsection 3.4. That is, it periodically sends attending request to the RP of the group. When this module receives a quit group request, it stops sending attending request to the RP of the group. As a result, the corresponding routing entry in each intermediate node of the multicast tree for this multicast group will be deleted according to time-out mechanism.
 - ◆ It also contains a module, which listens to the incoming routing messages on incoming interface sockets by selecting system call. Some procedures are waked up by coming multicast routing message to process the multicast routing message. Steps III through VI of the algorithm in Subsection 3.4 describe the method applied by this module.
- The command interpreter explains a few commands. Now it can process the commands such as participating in a group, quitting a group and transmitting a file.

As the packet format is insignificant in this case, it is omitted here. The experiment indicates that the MPUN works well on simple uni-directional network.

6 Conclusion

Developing a routing protocol over uni-directional networks is a task of importance. A good routing protocol for uni-directional networks will certainly promote the development of mobile computing and exploitation of satellite resources. The simulation experience shows that the protocol proposed in this paper is an implementable multicast routing protocol for uni-directional network. Before the protocol runs on practical systems, there are still further tasks to finish such as enhancing the performance of the protocol and implementing a prototype. To evaluate its real performance, we will do some simulation in the near future.

References

- [1] Corson S, Macher J. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. Internet Draft, September 1997.
- [2] Moy J. OSPF Version 2. RFC 1583, March 1994.
- [3] Moy J. Multicast Extensions to OSPF. RFC 1584, March 1994.
- [4] Malkin G. RIP Version 2 Carrying Additional Information. RFC 1388, Xylogics, Inc., January 1993.
- [5] Pusateri T. Distance Vector Multicast Routing Protocol. Juniper Networks, February 1997.
- [6] Rekhter Y, Li T. A Border Gateway Protocol 4. RFC 1771, T J Watson Research Center, IBM Corp-Cisco Systems, March 1995.
- [7] Estrin D, Farinacci D, Thaler D *et al.* Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification. Internet Draft, September 9, 1997.
- [8] Duros E, Iuitoma C. Handling of uni-directional links with RIP. Internet-Draft, March 1996.
- [9] Duros E. Handling of uni-directional links with DVMP. Internet-Draft, March 1996.
- [10] Duros E. Handling of uni-directional links with OSPF. Internet-Draft, March 1996.
- [11] Izumiyama H, Tosaka A. Dynamic Tunneling Path Configuration for Uni-directional Link Routing. Internet-Draft, July 1997.
- [12] Izumiyama H, Tosaka A. An IP Tunneling Approach for Uni-Directional Link Routing. Internet-Draft, July 1997.
- [13] Huang H, WU L-F, Chen D-X, Xie L, Sun Z-X. A routing algorithm for network with uni directional links. *Journal of Software* (in Chinese), to be published.
- [14] Zhang Y-G, Dao S. Integrating direct broadcast satellite with wireless local access. In *Proc. of 1st International Workshop on Satellite-Based Information Services*, New York: RYF, 1996, pp.24-29.
- [15] Ernst T, Dabbous W. A circuit-based approach for routing in uni-directional links networks. Research Report of INRIA, October, 1997.

- [16] Deerint S, Edtrin D, Farinacci D, Jacobson V, Liu C-G, Wei L. The PIM architecture for wide-area multicast routing. *IEEE/ACM Transactions on Networking*, Apr. 1996, 4(2): 153-162.
- [17] Wu Z-H. Analysis and implementation of liveness and fairness of bounded Petri nets. *Chinese Computer Technology* (in Chinese), 1989, 4.

HUANG Hao received his B.S. degree in mathematics from Xiamen University in 1982 and Ph.D. degree in computer science from Nanjing University in 1999. His research interests include computer network and distributed system.

CHEN Guihai received his B.S. degree from Nanjing University in 1984, M.S. degree from Southeast University in 1987, and Ph.D. degree from the University of Hong Kong in 1997, all in computer science. He visited Kyushu Institute of Technology, Japan in 1998 as a research fellow. He is currently an Associate Professor of Nanjing University. His main research interests include parallel computing, advanced computer architecture, interconnection networks and Internet computing.

XIE Li received his B.S. degree in mathematics from Nanjing University in 1964. Presently he is the Vice President of Nanjing University and Professor of Department of Computer Science. His research interests include distributed system, distributed database and parallel processing.

SUN Zhongxiu is a Member of Chinese Academy of Sciences and Professor of Nanjing University. His research interests include distributed system and distributed processing.