A Reliable Totally-Ordered Group Multicast Protocol for Mobile Internet *

Guojun Wang[1,2], Jiannong Cao[1], Keith C. C. Chan[1]

[1]Department of Computing
Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
{csgjwang,csjcao,cskcchan}@comp.polyu.edu.hk

[2]School of Information Science and Engineering
Central South University
Changsha, Hunan, P. R. China, 410083
gordon434@163.net

Abstract

We propose a novel multicast communications model, called RingNet, which is a combination of logical trees and logical rings for multicast communications. Based on RingNet, we propose a reliable totally-ordered group multicast protocol for mobile Internet. The network entities in the top logical ring of the RingNet hierarchy are responsible for totally-ordering messages. All the network entities in all the logical rings are responsible for reliably transmitting multicast messages to their next nodes, children nodes, or their attaching mobile hosts, even in handoffs. The proposed protocol runs in a parallel and distributed way in the sense that each network entity only maintains information about its possible neighbors, and independently decides whether, when, and where to order, forward, and deliver multicast messages. We prove that, compared with the multicast protocol without ordering requirement, our totally-ordered multicast protocol provides the same multicast throughput, only with limited overhead on message latency and buffer sizes.

1. Introduction

Extensive research effort has been made on two kinds of basic communications in traditional Internet, i.e., unicast and multicast. However, multicast-based applications such as distributed file system, distributed database, video conferencing, and distance learning, are relatively immature in comparison with unicast-based applications such as World Wide Web, e-mail and file transfer applications. Recently with the convergence of Internet computing and wireless communications, research on multicast communications in mobile Internet becomes more active and more challenging than that in traditional Internet. In mobile Internet, more concerns should be considered, e.g., (A) *Mobile Hosts* (MHs), such as laptop computers, PDAs, and mobile phones, have severe resource constraints in terms of energy, processing, and storage resources; (B) wireless communications is characterized by its limited bandwidth and high bit error rate; and (C) the number of MHs involved may be very huge and they may be dispersed very widely.

Group Communications Systems (GCSs) provide group communications services among groups of processes. A *group* consists of a set of processes which are called *members* of the group. A process may voluntarily *join* or *leave* a group, or cease to be a member due to *failure*. The *membership* of a group is a list of currently operational processes in the group. One major task of GCSs is *group membership*, which maintains the membership of a group with regard to *Member-Join*, *Member-Leave*, *Member-Failure*, and *Member-Handoff* events. Another major task is *group multicast*, which involves efficiently disseminating information from one source or multiple sources to all the operational processes in the group. Usually group multicast service is implemented on top of group membership service.

Many existing GCSs are designed for generic LAN or WAN environment, which don't explicitly consider MHs as group members. Therefore, there is no guarantee that they can also work well in the presence of MHs. Intrinsic issues

*This research is supported in part by the Hong Kong Polytechnic University Central Research Grant *G-YY41*, and in part by the University Grant Council of Hong Kong under the CERG Grant PolyU *5170/03E*.

Proceedings of the 2004 International Conference on Parallel Processing Workshops (ICPPW'04)
1530-2016/04 $20.00 © 2004 **IEEE**

in traditional Internet like *high message latency*, *frequent connectivity changes*, and *instability due to link failures or congestion* [11], still exist in mobile Internet. More difficult issues, e.g., frequent disconnection, frequent handoff, and frequent failure occurrence, need to be addressed due to introducing MHs. In particular, not only dynamic membership may occur, but also dynamic locations of MHs due to the fact that mobility may occur.

In this paper, we propose a novel multicast communications model, called the *RingNet* model, which is a combination of logical trees and logical rings for multicast communications. Based on RingNet, we propose a reliable totally-ordered group multicast protocol for mobile Internet. The rest of the paper is organized as follows. In Section 2, we introduce some related works. In Section 3, we introduce our RingNet model. In Section 4, we design data structures and algorithms for the proposed protocol. In Section 5, we do some performance analysis of the proposed protocol. We give some concluding remarks in the final section.

2. Related Works

The *Mobile IP* protocol [18], as proposed by the Internet Engineering Task Force (IETF) in RFC 2002, provides an efficient, scalable mechanism for host mobility in mobile Internet while retaining the host's unicast communications. Mobile IP provides two basic extensions to accommodate IP Multicast [9]. One is *Mobile IP Bidirectional Tunnelling* (MIP-BT). Another is *Mobile IP Remote Subscription* (MIP-RS). The main advantage of the MIP-BT scheme is that it hides host mobility from all other members of the group. Therefore, it does not incur any overhead in the multicast tree maintenance. However, it incurs a high handoff latency as the MH moves far away from its home network. On the contrary, the main advantage of the MIP-RS scheme is that the multicast packets are always delivered on the shortest paths. However, the overhead is the cost of reconstructing the delivery tree while a handoff occurs.

One line of research on mobile multicast is based on MIP-BT and/or MIP-RS, e.g., Mobile Multicast (MoM) [10], MobiCast [21], Multicast Agent (MA) [22], Multicast by Multicast Agent (MMA) [20], and Range-Based Mobile Multicast (RBMoM) [12]. In order to provide IP Multicast services in mobile Internet to the MHs, the easiest way is to combine Mobile IP and IP Multicast. However, such a combination is not satisfactory in terms of multicast efficiency. The major reason is that, the Mobile IP mobility management scheme designed for unicast communications cannot be easily adapted for multicast communications.

In order to differentiate mobility management for unicast from that for multicast, we call the former *unicast mobility management* and the latter *multicast mobility management*. For unicast mobility management, either one of the sender and the receiver hosts will be mobile, or both of them will do. Therefore, it is relatively easy to tackle such a problem. However, for multicast mobility management, a group of hosts will be mobile at the same time. Therefore, it is relatively complicated to tackle such a problem.

Aware of the difference between unicast mobility management and multicast mobility management, another line of research on multicast for mobile Internet has been done. In [1], a two-tier *Host-View* scheme is proposed: the *Mobile Support Station* (MSS) tier and the MH tier. The basic idea is to associate a Host-View with each group. The Host-View consists of a set of MSSs, which represents the aggregate location information of the group. Through tracking a set of MSSs other than tracking each individual member MH, the protocol becomes very simple. For example, in order to deliver a multicast message to a group of MHs, it suffices to send a copy to only those MSSs in the group's Host-View. In addition, through moving most functionalities from MHs to MSSs, the MHs will be relieved from heavy tasks. However, this protocol does not allow dynamic joins or leaves, and does not specify a method for the creation or deletion of multicast groups. In particular, the global updates necessary with every "significant move" make it inefficient and may cause lengthy breaks in service to the MHs.

To deal with problems in the two-tier Host-View scheme, a three-tier *Reliable Multicast* (RelM) scheme is proposed [6]. The bottom tier consists of the MHs which roam between cells. The middle tier consists of MSSs, which provide the MHs with connectivity to the underlying network. The top tier consists of groups of MSSs. Each group of MSSs is controlled by an assigned supervisor machine called the *Supervisor Host* (SH). The SH is part of the wired network and it handles most of the routing and protocol details for MHs. Simulation results show that the RelM scheme uses fewer buffers in virtually any system configuration in comparison with the Host-View scheme. However, the advantage of moving most functionalities from MSSs to SHs will also become its disadvantage. Since the SHs have to do so many tasks such as maintaining connections for MHs, the RelM protocol scales not very well when the number of group members becomes very large.

Another three-tier reliable multicast scheme is proposed in [2], which accommodates three increasingly strong delivery ordering guarantees: *FIFO*, *causal*, and *total*. It assumes a very general system model: (A) incomplete spatial coverage of the wireless network; (B) unreliable wireless communications; and (C) dynamic group membership. The novelty of the scheme is that movements of MHs do not trigger any message transmission in the wired network as *no notion of handoff* is used in the wired network. As a consequence, it is potentially more scalable than the RelM scheme. Our proposed scheme follows the concept of "no notion of handoff" in the wired network.

Besides the above two/three-tier frameworks, another related work is *Logical Ring*-based reliable multicast protocol for mobile Internet in [16]. A logical ring is maintained among all the *Base Stations* (BSs) that handles the multicast traffic of the same multicast group. A token passing protocol enforces a consistent view among all the BSs with respect to the messages that are considered to have been delivered to all the MHs. Furthermore, a handoff protocol is designed to handle the interaction of reliable multicast and handoff events of MHs. Since all the control information has to be rotated along the ring, it may lead to large latency and require large buffers when the ring becomes large. Each logical ring within our proposed RingNet model functions in a similar way, but it deals with only a *local scope* of the whole group. In this way, our proposed protocol can be very simple as logical ring-based protocols.

3. The RingNet Multicast Model

Many mobile Internet architectures have been proposed by researchers, such as *Unified Wireless Networks Architecture* [13], *System Architecture for Mobile Communications Systems* [17], *All-IP Wireless/Mobile Network Architecture* [23], and *FIT-MIP Global System Architecture* [14]. Based on these architectures, we propose a multicast communications model called a *RingNet* hierarchy shown in Figure 1. The four tiers of the RingNet hierarchy are *Border Router Tier* (BRT), *Access Gateway Tier* (AGT), *Access Proxy Tier* (APT), and *Mobile Host Tier* (MHT). The higher two tiers are organized into logical rings. Each logical ring has a leader node that is also responsible for interacting with upper tiers. *Access Proxies* (APs) are the *Network Entities* (NEs) that communicate directly with the *Mobile Hosts* (MHs). *Access Gateways* (AGs) are the NEs that communicate either between different wireless networks or between wireless network and wired network. *Border Routers* (BRs) are the NEs that communicate among administrative domains. Notice that only those NEs that are configured to run the proposed protocol will be involved in the hierarchy.

In order to form such a hierarchy, we require each AP, AG, and BR to have some knowledge of its *candidate contactors*, either some *candidate neighboring nodes* through which it can join a logical ring, or some *candidate parent nodes* through which it can attach to an existing hierarchy. For each AP, it is configured with one or several candidate AGs. For each AG, it is configured with one or several neighboring AGs for joining the logical rings where these AGs reside, and/or configured one or several candidate parent BRs for attaching to these BRs. In the RingNet hierarchy shown in Figure 1, there is only one BR logical ring. However, when considering more complicated scenarios where sub-tiers of the AGT and BRT tiers are allowed, then it is also necessary to configure candidate neighboring nodes and/or candidate parent nodes for each BR. Notice that at most one of these candidate contactors will be enabled to function as its specific role for any AP, AG, and BR at any specific time.

Multicast communications using the RingNet hierarchy is simple: *Multicast Senders* (MSs) send multicast messages to any of the BRs at the top logical ring. Then the multicast messages are transmitted along each logical ring, and downward to all the children nodes. Finally the MHs will receive multicast messages from their attached APs. For efficient multicast communications, we borrowed ideas from the MRP approach [7] and some multicast-based smooth handoff schemes [19, 15]. We introduce *Multicast Mobility Agents* (MMAs) in each micromobility domain. Our MMAs are similar to MRPs in the sense that a list of entries is maintained in each MMA and the list will be searched for each downlink packet. The differences are that each entry is *group*-oriented in the MMA and that *multiple entries* for the same group may exist in each MMA. The basic idea of using multiple entries comes from the multicast-based smooth handoff schemes. We extend their schemes as follows. When an MH handoffs to a new AP and the AP currently cannot receive multicast messages, it starts to build a multicast path toward one of its candidate AGs. At the same time it notifies its nearby APs to do multicast path reservation between APs and AGs. In most cases, when an MH handoffs, it can immediately receive multicast messages because either some other members have already been there, or some reserved path has already been set up in advance.

In [3], the authors defined a term called *distribution vehicle* for multicast communications. A distribution vehicle is a collection of hardware settings in the network which makes it efficient for one or multiple sources to multicast messages to a group of receivers. In traditional Internet, it is usually a *multicast tree* [9]. In the infrastructure-less mobile ad-hoc networks, it is usually a *multicast mesh* [4], which is more reliable than a multicast tree. Another commonly used distribution vehicle is *logical ring* [16].

Loosely speaking, the proposed RingNet hierarchy is a novel distribution vehicle which combines advantages of both the logical tree and the logical ring. If we consider each logical ring as *one* node, then the RingNet hierarchy becomes a tree. Since each logical ring is organized according to some criteria such as locality/proximity criterion, we can apply some localized mechanism to greatly simplify the protocol design. Therefore, the multicast protocol based on the RingNet hierarchy is potentially simple, efficient, scalable and reliable. More interestingly, the hierarchy is self-organizable in the sense that it is all the way trying to form a larger hierarchy whenever possible.

In the proposed multicast protocol, it relies on the underlying membership protocol to propagate membership information along the RingNet hierarchy and to maintain the

Proceedings of the 2004 International Conference on Parallel Processing Workshops (ICPPW'04)
1530-2016/04 $20.00 © 2004 **IEEE**

topology of the hierarchy. For membership propagation, each MH can join or leave a group at will, or fail at any time. The membership change message is firstly captured by the MH's attached AP node, then propagated to the AP's parent AG node. If the AG happens to be a leader node, then it propagates such message to its parent node; if not, then it propagates such message to the leader node in the logical ring where it resides. This process continues until the leader node in the top logical ring is reached. To propagate membership information more efficiently, some batched update scheme can be used. For topology maintenance, the RingNet hierarchy will dynamically change due to movements of MHs and/or failures occurring within the hierarchy. We shall omit the details of the membership protocol for sake of brevity.

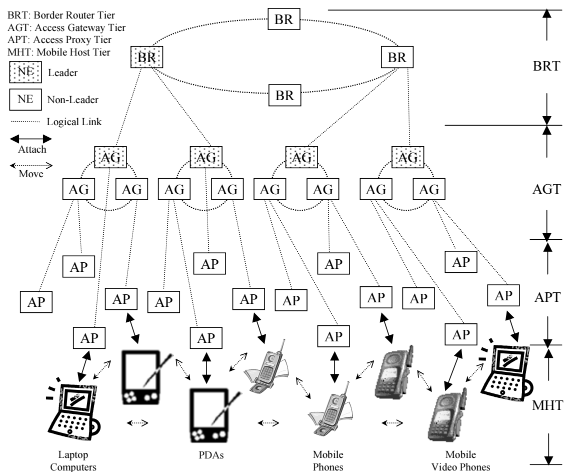

Figure 1. The RingNet Hierarchy

4. The RingNet Multicast Protocol

Informally, the proposed protocol deals with multicast communications as follows: (A) if any message source needs to multicast to the group, it will use some interface mechanism to contact with a corresponding node in the top logical ring of the hierarchy and then send messages to the corresponding node; (B) the hierarchy will be used to order, forward, and deliver messages starting from the top logical ring, then through all other logical rings, finally to all the MHs in a simple and regular way as follows.

- All the NEs in the top logical ring of the hierarchy are responsible for totally-ordering messages by running the *Message-Ordering* algorithm.

- All the NEs in all the logical rings including the top one are responsible for reliably transmitting multicast messages by running the *Message-Forwarding* algorithm to forward messages to their *next* nodes and by running the *Message-Delivering* algorithm to deliver messages to their possible *children* nodes, or their attaching MHs, even in handoffs.

The Message-Ordering, Message-Forwarding, and Message-Delivering algorithms run in a parallel and distributed way in the sense that each NE in the hierarchy only maintains information about its possible *leader*, *previous*, *next*, *parent*, and *children* neighbors, and that each NE independently decides whether, when, and where to order, forward, and deliver multicast messages.

In this section, we firstly describe data structures of MHs, NEs, and the ordering token which circulates along the top logical ring, finally present the Message-Ordering, Message-Forwarding, and Message-Delivering algorithms.

4.1. The Data Structures of MHs, NEs and Tokens

Data Structure of MHs. Once an MH becomes a member of a group, it will start to receive/deliver multicast messages based on the following data structure.

- GID: GroupID. Group identity of some group addressing scheme, e.g. Class D address in IP Multicast [8].

- AP: NodeID. Node identity of the attached AP.

- GUID, LUID: UniqueID. Globally/locally unique identity of MH from some unique identity scheme, e.g. Mobile IP *Home Address/Care-of Address* [18].

- MQ: MessageQueue. We assume using sequential storage allocation scheme to keep multicast messages. MQ has the following attributes: *Data* stands for the allocated storage spaces for keeping multicast messages; *MaxNo* stands for the maximal number of allocated storage spaces; *Rear* stands for the pointer to the most recently received multicast message; *Front* stands for the pointer to the most recently delivered multicast message; *ValidFront* stands for the pointer to the oldest delivered multicast message which is still kept in MQ, which is reserved for APs/AGs/BRs only. For each message stored within the range of MQ, there are at least the following attributes: *Received* to indicate whether the message is really received or not; *Waiting* to indicate whether the message is still waiting to be retransmitted or not; *Delivered* to indicate whether the message has been delivered or not; *SourceNode* to indicate where the message comes from; *LocalSeqNo* to indicate the local sequence number assigned to the message by its SourceNode; *OrderingNode* to indicate where the message is ordered; *GlobalSeqNo* to indicate the global sequence number assigned to the message by its OrderingNode; and *Payload*, the original message from its SourceNode.

Data Structure of NEs. Each NE maintains information about its possible *leader*, *previous*, *next*, *parent*, and *children* neighbors. A similar message queue MQ as that of MH is used. In addition, some assisted working queues/tables are defined. Details of the data structure are as follows.

- GID: GroupID. See data structure of MHs.
- Current, Leader, Previous, Next, Parent, Children[]: NodeID. Node identity of this node, leader, previous, next, parent, and children nodes, respectively. Notice that Children[] consists of a list of node identities, each of which stands for one child node.
- PreviousOK, NextOK, ParentOK, ChildrenOK[]: Boolean. Status of the previous, next, parent, and children nodes, respectively. Notice that ChildrenOK[] consists of a list of sub-items, each of which describes the status of one child node.
- MQ: MessageQueue. Similar to that of MH. However, meanings of *delivered* are different: for MH, it means that some received messages are *really delivered* to applications running on the MH; for each bottom AP, it means that some received messages are *successfully transmitted* to all the MHs attaching to the AP; for any other network entity in the hierarchy, it means that some received messages are *successfully transmitted* to its children nodes in the hierarchy (if the children nodes exist). Notice that if a message's *Received* value is FALSE (meaning not received) and at the same time *Waiting* value is also FALSE (meaning not waiting for retransmission any longer), then the message is supposed to be *really lost*. Furthermore, such a *really lost* message is also *considered* to be *delivered* by setting its *Delivered* value to TRUE.
- WQ: WorkingQueue. Working queue for keeping messages: (A) some messages come from one or more multicast sources to be totally ordered (we assume at most one source corresponding to each node in the top logical ring of the hierarchy); (B) some messages are forwarded to the current node from its previous node. We design WQ as a list of queues, each of which is used to keep messages from one source, and each of which has similar data structure as MQ but possibly with different meanings. We denote WQ.OrderingNode as the queue which is used to keep messages coming from OrderingNode. Notice that WQ is used only for nodes in the top logical ring.
- OldOrderingToken, NewOrderingToken: TypeOfOrderingToken. Used to keep at most two versions of most recently acquired OrderingToken (See below: *Data Structure of Tokens*). Notice that they are used only for nodes in the top logical ring.
- MinLocalSeqNo, MaxLocalSeqNo: LocalSeqenceNumber. Used to track the minimal and maximal sequence numbers attached to the messages which come from the current node's multicast source and have not been totally-ordered. Notice that they are used only for nodes in the top logical ring.
- WT: WorkingTable. Working table for calculating the maximal global sequence number of the message which has been delivered to either all the children nodes for non-bottom NE or all the attached MHs for bottom NE. Each entry in the table contains at least two items. For an non-bottom NE, one is *node identity* of a child node; another is the maximal global sequence number *MaxGlobalSeqNo* delivered to the child node. For a bottom NE, i.e., an AP, one is *GUID* of an MH; another is the maximal global sequence number *MaxGlobalSeqNo* delivered to the MH.

Data Structure of Tokens. In order to order messages, we use a token called *OrderingToken* that circulates along the top logical ring. It keeps the following information.

- GID: GroupID. See data structure of MHs.
- NextGlobalSeqNo: GlobalSequenceNumber. The global sequence number which will be used to order the next message.
- WTSNP: WorkingTableOfSequenceNumberPairs. A working table to keep pairs of sequence numbers of ordered messages. Each entry in it contains at least the following attributes: *SourceNode* to indicate where the message comes from; *MinLocalSeqNo* to indicate a minimal local sequence number; *MaxLocalSeqNo* to indicate a maximal local sequence number; *OrderingNode* to indicate where the message is ordered; *MinGlobalSeqNo* to indicate the minimal global sequence number associated with MinLocalSeqNo; and *MaxGlobalSeqNo* to indicate the maximal global sequence number associated with MaxLocalSeqNo.

4.2. The Proposed Multicast Algorithms

In this subsection, we design the *Message-Ordering*, *Message-Forwarding*, and *Message-Delivering* algorithms based on the above data structures.

4.2.1 The Message-Ordering Algorithm

We suppose multicast source will select a *corresponding* node using some interface mechanism. Multiple multicast sources may send multicast messages with *local sequence numbers* to their corresponding nodes in the top logical ring.

A token called *OrderingToken* is used to circulate along the top logical ring for ordering messages. The node holding OrderingToken is called *Holder*. Firstly, OrderingToken.WTSNP and OrderingToken.NextGlobalSeqNo will be updated according to Holder.MinLocalSeqNo and Holder.MaxLocalSeqNo; secondly, OrderingToken will be kept in Holder.NewOrderingToken (the original NewOrderingToken is moved to OldOrderingToken); finally, OrderingToken will be reliably transferred to the next node in the top logical ring with some retransmission scheme.

In order to improve efficiency of message ordering, OrderingToken held by the current node will be used to pre-assign global sequence numbers to all the ready-to-be-ordered messages kept in WQ of the current node, which come from the current node's multicast source. The Message-Ordering algorithm doesn't really assign totally-ordered global sequence numbers to those messages, which is done by an *Order-Assignment* algorithm. The Order-Assignment algorithm is also responsible for copying ordered messages from WQ to MQ. For each node in the top logical ring, it periodically checks its WQ, if the ready-to-be-ordered messages in its WQ can be ordered according to either NewOrderingToken or OldOrderingToken, then the messages will be assigned global sequence numbers and further be copied to its MQ.

Now we turn to discuss *Token-Loss* problem in token passing logical rings. Since we suppose to adopt some retransmission scheme for reliably transferring OrderingToken to the next node, in case that no communications node faults and link faults in the top logical ring, then there isn't any Token-Loss problem. However, in case that the top logical ring may be broken due to faults and later be repaired, then the Token-Loss problem may occur.

To solve the Token-Loss problem, we use NewOrderingToken kept in each node along the logical ring. Notice that each time the topology maintenance algorithm of the membership protocol runs, the token will be lost only when the token happens to be held by a faulty node. However, it's unreasonable to require the membership protocol to know the running status of the multicast protocol. Therefore, we only need the membership protocol to send out a *Token-Loss* message to the multicast protocol when running its topology maintenance algorithm.

On receiving such a Token-Loss message by some node, the *Token-Regeneration* algorithm will do the following to solve the Token-Loss problem.

- If the node shows that the Message-Ordering algorithm runs well, it will ignore the Token-Loss message and terminate the Token-Regeneration algorithm. Otherwise, it will originate a *Token-Regeneration* message which encapsulates NewOrderingToken of the node and let the message traverse along the next link of the top logical ring.
- For each node the message traversed, it will do the following: if the current node shows that the Message-Ordering algorithm runs well, then it will destroy the message and terminate the Token-Loss algorithm; if the current node's NewOrderingToken.NextGlobalSeqNo is greater than that encapsulated in the message, then re-encapsulate the message with NewOrderingToken kept in the current node, otherwise, we assume the current node be the starting point to restart the Message-Ordering algorithm with a newly generated OrderingToken, which is NewOrderingToken encapsulated in the message.

With the Token-Loss problem solved, there still exists a *Multiple-Token* problem related to token circulation. Since two or more top logical rings may merge, if not properly handled, multiple OrderingTokens may co-exist. Similar to the solution to the Token-Loss problem, we suppose the membership protocol will send out a *Multiple-Token* message to the multicast protocol during running its topology maintenance algorithm, then the multicast protocol will keep only one OrderingToken alive according to some rule.

4.2.2 The Message-Forwarding Algorithm

Notice that WQ of each node in the top logical ring contains a list of queues, each of which is used to keep messages coming from one corresponding node/multicast source; and that MQ of each node in any logical ring in the hierarchy contains a queue of totally-ordered messages. The Message-Forwarding algorithm runs in each node to independently do the following: (A) reliably forward any message kept in WQ of each node in the top logical ring to that of the next node of the current node (if the next node is not the corresponding node of the message); (B) reliably forward any message kept in MQ of each node in the non-top logical ring to that of the next node of the current node (if the next node is not the leader of the logical ring).

4.2.3 The Message-Delivering Algorithm

The above Message-Forwarding algorithm deals with messages transmitted along *next* link of the logical rings, while the Message-Delivering algorithm here deals with messages transmitted along the *children* links from the parent node to its children nodes in the hierarchy, or along the *wireless* links between the bottom APs and their attaching MHs.

Consequently, the Message-Delivering algorithm does the following: (A) reliably delivering messages from MQ of each node in all the non-bottom logical rings to its children nodes; (B) reliably delivering messages from MQs of the bottom APs to their attaching MHs, even in handoffs.

In [5], the authors classified the reliability property of multicast communications into *strong reliability* and *best-effort reliability*. Our reliable multicast protocol follows the latter case. We divide the RingNet hierarchy into multiple *local scopes* and implement reliable transmission within each local scope. We call such a scheme *local-scope*-based retransmission scheme. If each NE in the hierarchy will reliably transmit multicast messages within some local scope, e.g., the *immediate neighbor* scope, the *single logical ring* scope, or the *multiple neighboring logical rings* scope in a best-effort way, then highly probable reliability can be expected when the network is highly stable.

5. Performance Analysis

We suppose the top logical ring contains r (≥ 2) nodes, and there are s ($\leq r$) multicast sources, each of which sends λ messages each time unit (e.g. each second). We also suppose each node in the top logical ring can transmit messages to its next node and its possible children nodes under the above message loads when ordering is not required. We now present a theorem.

Theorem 5.1 *Without considering overheads of processing/forwarding OrderingToken along the top logical ring and without considering retransmission, compared with the multicast protocol without ordering requirement, our totally-ordered multicast protocol provides the same multicast throughput as $s*\lambda$ messages each time unit, only with limited overhead on message latency and buffer sizes.*

Proof 5.1 *Notice that the Message-Ordering algorithm runs concurrently with the Message-Forwarding and Message-Delivering algorithms for each node in the top logical ring. The Message-Forwarding and Message-Delivering algorithms run in full speed in the sense that, for each node in the logical ring, (A) any received message is immediately forwarded to the next node (if not the corresponding node of the message); (B) any totally-ordered message copied to MQ from WQ of the node, is immediately transmitted to its possible children nodes, and waiting to be tagged delivered.*

Suppose the maximal round-trip time for an OrderingToken circulating the top logical ring is T_{order} time units, then each message received by its corresponding node will be ordered within T_{order} time units. In addition, suppose the maximal round-trip time for any message forwarding along the top logical ring is $T_{transmit}$ time units, and suppose the timer cycle for invoking the Order-Assignment algorithm in each node to check and copy ordered messages into MQ of the node is τ, then the maximal time for any message between being received by its corresponding node and being copied to MQs of all the nodes in the logical ring will be within $Max(T_{order}, T_{transmit})+\tau$ time units.

Without considering retransmission, we then design the sizes of buffers, i.e., WQ and MQ, of each node. Since WQ is a list of queues, each of which is used to buffer received messages from one multicast source waiting to be ordered and be copied to MQ, the size of WQ can be set to $s*\lambda*(Max(T_{order}, T_{transmit})+\tau)$; since MQ is used to buffer totally-ordered messages, and suppose the node can transmit ordered messages from MQ to its possible children nodes, then the size can be set to $s*\lambda*T_{order}$. Therefore, we have proved that all the buffers only need limited sizes.

We then suppose the maximal time for any ordered message kept in MQ to be transmitted and then be tagged as delivered to its possible children nodes is $T_{deliver}$, then the maximal message latency for any message between being received by its corresponding node and being delivered by all the nodes in the top logical ring consists of: (A) the maximal time for any message between being received by its corresponding node and being copied to MQs of all the nodes in the logical ring, which is $Max(T_{order}, T_{transmit})+\tau$ time units; (B) the maximal time for any ordered message kept in MQ to be transmitted and then be tagged as delivered to its possible children nodes, which is $T_{deliver}$. Therefore, we have proved that any message will be ordered, forwarded, and delivered within the message latency bound of $Max(T_{order}, T_{transmit})+\tau+T_{deliver}$.

Since we don't have any additional requirement on multicast sources, e.g., we don't require any multicast source to decrease message transmission rate to accommodate ordering requirement, the same message throughput can be achieved by running the Message-Ordering algorithm compared with message transmission without ordering requirement, which completes the proof.

Notice that, retransmission will occur in unreliable communications environment for reliable message transmission. Therefore, buffer sizes of WQ and MQ of each node may be larger and message latency may be larger to accommodate retransmission. We will do more analysis in our future work regarding retransmission.

6. Concluding Remarks

Remark 1. Within the RingNet hierarchy, APs, AGs and BRs can be or function as traditional routers, but will be augmented by running the RingNet protocol. In fact, we require that only a portion of routers will be configured to run the RingNet protocol. Furthermore, we recommend gradual deployment of RingNet-aware applications because it will be easier to form a single or only several disjoint partitions of the RingNet hierarchy in case that mobile users are not dispersed from each other too much.

Remark 2. Although it is highly expected to dynamically find and configure neighboring network entities and/or logical rings for self-organizing the RingNet hierarchy, it is much easier and more practical to manually and statically configure the candidate neighboring relationship and parent-children relationship between all the network entities and logical rings. In this case, system administrators can have their full control of the hierarchy.

Remark 3. If totally-ordered property is not required, then multicast using the RingNet hierarchy will be more efficient and message latency will decrease due to the fact that ordering operations are not required in the top logical ring.

References

[1] A. Acharya and B.R. Badrinath, "A framework for delivering multicast message in networks with mobile hosts," *Mobile Networks and Applications*, 1(2):199-219, October 1996.

[2] G. Anastasi, A. Bartoli, and F. Spadoni, "A reliable multicast protocol for distributed mobile systems: Design and evaluation," *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1009-1022, October 2001.

[3] J. Auerbach, M. Gopal, M. Kaplan, and S. Kutten, "Multicast group membership management," *IEEE/ACM Transactions on Networking*, 11(1):166-175, February 2003.

[4] S.H. Bae, S.-J. Lee, W. Su, and M. Gerla, "The design, implementation, and performance evaluation of the on-demand multicast routing protocol in multihop wireless networks," *IEEE Network*, 14(1):70-77, January-February 2000.

[5] K.P. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky, "Bimodal multicast," *ACM Transactions on Computer Systems*, 17(2):41-88, May 1999.

[6] K. Brown and S. Singh, "RelM: Reliable multicast for mobile networks," *Journal of Computer Communications*, 21(16):1379-1400, 1998.

[7] A. T. Campbell, J. Gomez, S. Kim, C.-Y. Wan, Z. R. Turanyi and A. G. Valko, "Comparison of IP micromobility protocols," *IEEE Wireless Communications*, 9(1):72-82, February 2002.

[8] S. Deering, "Host extensions for IP multicasting," *IETF RFC 1112*, August 1989.

[9] S. Deering and D. R. Cheriton, "Multicast routing in datagram internetworks and extended LANs," *ACM Transactions on Computer Systems*, 8(2):85-110, May 1990.

[10] T.G. Harrison, C.L. Williamson, W.L. Mackrell and R.B. Bunt, "Mobile multicast (MoM) protocol: Multicast support for mobile hosts," *ACM International Conference on Mobile Computing and Networking (MobiCom 1997)*, pp. 151-160, September 1997.

[11] I. Keidar, J. Sussman, K. Marzullo, and D. Dolev, "Moshe: A group membership service for WANs," *ACM Transactions on Computer Systems*, 20(3):191-238, August 2002.

[12] C.R. Lin and K.-M. Wang, "Scalable multicast protocol in IP-based mobile networks," *ACM Wireless Networks*, 8:27-36, 2002.

[13] W.W. Lu, "Compact multidimensional broadband wireless: The convergence of wireless mobile and access," *IEEE Communications Magazine*, 38(11):119-123, November 2000.

[14] L. Morand and S. Tessier, "Global mobility approach with Mobile IP in all IP networks," *IEEE International Conference on Communications (ICC 2002)*, Vol. 4, pp. 2075-2079, 2002.

[15] J. Mysore and V. Bharghavan, "A new multicasting-based architecture for Internet host mobility," *Proc. 3rd annual ACM/IEEE international conference on Mobile computing and networking (MobiCom 1997)*, pp. 161-172, September 1997.

[16] I. Nikolaidis and J.J. Harms, "A logical ring reliable multicast protocol for mobile nodes," *Proc. Seventh International Conference on Network Protocols (ICNP 1999)*, pp. 106-113, October-November, 1999.

[17] T. Otsu, N. Umeda, and Y. Yamao, "System architecture for mobile communications systems beyond IMT-2000," *IEEE Global Telecommunications Conference (GLOBECOM 2001)*, Vol. 1, pp. 538-542, 2001.

[18] C. Perkins, "IP mobility support," *IETF RFC 2002*, October 1996.

[19] S. Seshan, H. Balakrishnan and R. H. Katz, "Handoffs in cellular wireless networks: The Daedalus implementation and experience," *ACM Wireless Personal Communications*, 4:141-162, 1997.

[20] Y.-J. Suh, H.-S. Shin and D.-H. Kwon, "An efficient multicast routing protocol in wireless mobile networks," *ACM Wireless Networks*, 7(5):443-453, September 2001.

[21] C.L. Tan and S. Pink, "MobiCast: A multicast scheme for wireless networks," *Mobile Networks and Applications*, 5(4):259-271, December 2000.

[22] Y. Wang and W. Chen, "Supporting IP multicast for mobile hosts," *ACM Mobile Networks and Applications*, 6:57-66, 2001.

[23] T.B. Zahariadis, K.G. Vaxevanakis, C.P. Tsantilas, N.A. Zervos, and N.A. Nikolaou, "Global roaming in next-generation networks," *IEEE Communications Magazine*, 40(2):145-151, February 2002.