Hierarchical Model of Guaranteed, Reliable and Secured Broadcast Protocol

Md. Asadul Hoque
Database Administrator,
IT Division
TM International Ltd.
asad1752@ieee.org

S.M.Iftekharul Amin
Senior Programmer,
IT Division
TM International Ltd.
iftekharul.amin@aktel.com

Md. Shohrab Hossain
Department of CSE,
Bangladesh University of Engineering and Technology
mshohrabhossain@cse.buet.ac.bd

Abstract

In modern era of information technology, broadcasting is an integral part of communication engineering. To develop the broadcast technology, a lot of researches have been done in the fields of protocols, scheduling schemes, caching strategies and also using multi-channel communication. In this paper, we have proposed a hierarchical broadcast model of GRSB [1] (Guaranteed, Reliable and Secure Broadcast) protocol, where 1st level broadcast mostly maintains the traditional GRSB model. 2nd level broadcast has two types of nodes where nodes of 1st kind i.e. nodes that are directly connected with the attachment nodes, can be temporarily in 'not alive' condition and hence follow the traditional GRSB model as well. On the other hand, nodes of 2nd kind i.e. nodes that are not directly connected with the attachment nodes, must be 'alive' all the time and have to follow two phases to finish the broadcast.

Keywords: data dissemination, GRSB, attachment nodes, hierarchical broadcast model

1. Introduction

GRSB - Guaranteed, Reliable, Secure Broadcast - is a protocol that provides reliable and secure broadcast / multicast communications. It can be implemented in many types of networks - local area networks, wide area networks, as well as satellite communications.The methodology used in this protocol is surprisingly simple. Three logical nodes are enforced in the network - a Central Retransmiter, a Designated Acknowledger, and a (many when needed) Playback Recorder(s). Through the coordinated service of the three nodes, every user node can be guaranteed to receive all broadcast messages in the correct temporal order. A fourth logical node, the Security Controller, can be added to the protocol to provide security-related services such as user authentication, message encryption, etc.

2. Motivation Behind GRSB Protocol

In general most existing broadcast technology (computer network, radio, TV, etc.) has the following drawbacks:

(1) The sender and receivers must be turned on during the period of broadcast. If a receiver is having hardware problems, busy doing something else or turned off, it will miss the broadcasted messages.

(2) There is no guarantee that a receiver receives all messages correctly. Even if a receiver is turned on and ready to receive, electrical noise could cause loss or distortion of messages.

(3) The sender must broadcast the messages slowly to cater to the slowest receiver. Most computer networks consist of some fast and some slow computers. If the broadcast must cater to the slowest possible computer in the network, the resulting system will not be efficient.

Not only does this GRSB protocol overcome all the above mentioned drawbacks, it also guarantees the reception of all broadcast messages in exactly the same sequence by all nodes in the network. By serializing the broadcast messages, it provides new techniques in distributed database updates. Parallel information retrieval, fail-safe systems and many others. Therefore, the its applications include Backup, Computer Conference, Parallel Information Retrieval, Hot Standby, Distributed Databases and Sales Support Systems

3. Broadcast Model

In Figure 1, three logical nodes are added to an existing network: the Central Retransmitrer, the Designated Acknowledger, and the Playback Recorder. With this arrangement, if user node 2 wants to broadcast to other computers, it first sends the message to the Central Retransmitter on a one-to-one basis. Thus the Central Retransmitter is guaranteed to receive the message correctly.

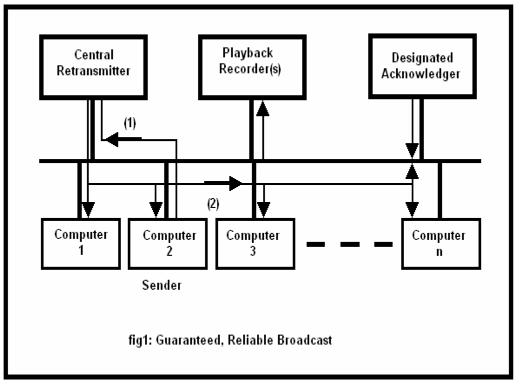

Figure1. Guaranteed, Reliable, Braodcast

The Central Retransmitter adds special information such as global message sequence number, date and time and then broadcasts the message. This broadcasted message is listened to by all nodes. However, only the Designated Acknowledger sends an acknowledgment message back. In effect, every node listens to the one-to-one communication (hence reliable) between the Central Retransmitter and the Designated Acknowledger. The Designated Acknowledger, in addition to acknowledging messages from the Central Retransmitter, also records all broadcasted messages temporarily (a few seconds) for the benefit of the Playback Recorder(s). A Playback Recorder listens to the broadcasted messages and selectively records them (based upon information such as sender identification, "conference number", message type, etc.). A Playback Recorder detects whether it misses any messages by checking the global message sequence number. If it hears messages 1, 2, 3, 5, it knows it has missed message 4. Furthermore, it knows the Designated Acknowledger must have message 4, otherwise, message 5 would not have been sent. Thus it can recover message 4 from the Designated Acknowledger on a one-to-one basis. This is the reason why the Designated Acknowledger needs to store the message for a short period. A Playback Recorder will store information for a much longer period depending on application needs (hours, days, weeks or months). Usually, there is more than one Playback Recorder recording a particular "conference". The concept of "conference" or "group access" is important. With this protocol, we can achieve multicast without using the multicast feature in the Ethernet hardware. The protocol allows a sender to specify which conference the message is meant for. A receiver can filter messages based on the conference number. So it is possible for a sender to send messages to a group of receivers without knowing the individual members. A receiver can also join a conference many days after it has started. All a receiver needs to know is the conference number.

The user computers 1 to n also listen to the broadcasted messages. The detection of missed messages is also via the global message sequence number as in the case of a Playback Recorder. However, the message recovery is via one-to-one communication with a Playback Recorder rather than with the Designated Acknowledger. If a user node was turned off, when it is turned back on again, the software will indicate which message was last received correctly before it was turned off. The software then goes to a Playback Recorder for the missed messages. As the broadcast message contains not just the global message sequence number but also the date and time information, each message is unique. There is no confusion in recovering from a wrong message. This means that the broadcasting need not be slowed down to cater to a few busy, slow malfunctioning or turned off receivers. These receivers do not interact directly with the Central Re-transmitter nor the Designated Acknowledger. They would request the missed messages from one or more Playback Recorders. In most cases, these Playback Recorders can handle these requests without disturbing the broadcast. The Playback Recorders request the Designated Acknowledger to slow down the message acknowledgment only when they themselves start missing messages (e.g. too busy serving the receivers).

A Playback Recorder can serve the archiving function and act as a file server. For example, user node 1, say, is turned off. When it is turned back on, it wants to recover missed messages on conference 7 from user node 2 only. It can go to a Playback Recorder and ask for all missed messages on conference 7 from user node 2 only (treating the Playback Recorder as a file server) until it catches up with the real time broadcast. By enforcing the three logical nodes described above and keeping them functional, we can guarantee reliable broadcast for EVERY node in the network in the correct temporal order. If higher reliability is required, hot backup systems can be provided for these specialized logical nodes.

4. Implementation

In GRSB, in addition to supplying the global message sequence number and the time information, the Central Retransmitter also sends out an "I am alive" message periodically if there were no broadcast messages. Thus a receiver node always knows whether the broadcast system is working. In some situations, a network can become segmented due to failure of a few links or nodes. If a receiver node can still communicate with the

Central Retransmitter and the other logical nodes, it can ignore the failure and just carry on. If a receiver node fails to communicate with the Central Retransmitter, it cannot perform many functions such as broadcasting, computer conferencing nor distributed database updates.

The Central Retransmitter supplies both a global message sequence number and the actual date and time information for the broadcast messages. This is because the broadcast system can be thought of as a “journaling” system. A user node can be turned off for weeks (e.g. a user turns off his personal computer and goes on vacation for two weeks). When the user node is turned back on, all the missed messages can be retrieved from Playback Recorders. In a “journaling” system, the actual date and time information is important. Since the time information supplied is from the Central Retransmitter only, clock synchronization is not an issue.

As the Playback Recorder checks the global message sequence number on every broadcast message, any lost or missed messages would be detected on the reception of the next good message. The Playback Recorder then “immediately” requests the missed message from the Designated Acknowledger. Thus the number of messages to be retained by the Designated Acknowledger is small. To further safeguard the possible overwriting of an unrecorded broadcast message, the Designate Acknowledger polls Playback Recorders before overwriting a group of its buffered broadcast messages. When necessary, the Designated Acknowledger can slow down the whole broadcast system by delaying the sending of acknowledgment messages to the Central Retransmitter.

A conference number field in the message header identifies the conference. A sender does not even need to know who are the conference participants or the receiver nodes. If a receiver node is interested in Conference A only, when it is turned on, it will listen to all broadcast messages and filter out all messages except those for conference A. Thus hardware or firmware implementation for message filtering is highly recommended. However, if the user node is turned off, when it is turned on again, it can go to the Playback Recorder and request only conference A messages. It does not need to go through all messages. Thus the Playback Recorder serves as a file server in addition to archiving and journaling.

In GRSB, many-to-many data communication is handled in exactly the same way as one-to-many data communication. Any node can be a sender node. To be more exact, one-to-many is a subset of this broadcast system. The concept of many-to-many data communication can be compared to that of having many people in a conference. In a conference, any participant can address the whole group. In the broadcast system, any computer can address the group for a subset) with guaranteed reliability. No messages will be missed and all messages w ill be “heard” in exactly the same order.

It is important to know when a broadcast message should take effect especially in update type operations where many nodes will be affected. For single message operations, this will not be a problem. For multiple message operations, the technique is to use start, middle and end message types. A receiver is NOT supposed to take any action until the end message is received. It should receive the entire group of messages in its buffers before taking any action. This avoids the situation of starting certain actions and then failing to complete them if the sender node fails.

5. Performances

In GRSB, the message to be broadcasted is sent at least twice. Once a message sent from the sender node to the Central Retransmitter and then from the Central Retransmitter to everyone else. It is true that GRSB will have more overhead as compared with the one-to-one communication technique if the message is to be sent to only one or two receivers. However, GRSB starts to have an edge if communication between three or more nodes is required. Also additional hardware may be needed to do the message filtering. If a personal computer must be interrupted every time a broadcast message is detected, the performance of the personal computer will be greatly affected because a lot of these messages may not be of interest to this personal computer.

Most computer networks consist of some fast and some slow computers. In GRSB, the presence of a few slow receivers will not slow down the broadcast system. When the few slow receivers miss messages, they will get these missed messages from the Playback Recorders on a one-to-one basis. These slow receivers do not communicate directly with the Central Retransmitter nor the Designated Acknowledger. When the Playback Recorders themselves start missing messages (for example, too busy servicing many slow receivers), they will communicate with the Designated Acknowledger to get the missed messages and can then inform the Designated Acknowledger to vary its time delay before sending the acknowledgment messages. Thus, a few slow receivers will not slow down the broadcast system - but a large number will.

For a small network consisting of ten nodes or less, it is possible to have a single physical computer implementing multiple logical functions.

For example, a single computer can be the Central Retransmitter and a Playback Recorder. For networks with hundreds or even thousands of nodes, more than one Playback Recorder node is highly recommended.

The simplest and the least expensive implementation of GRSB is to use existing nodes and add software only. Most of the functions of the four logical nodes except the Designated Acknowledger can be implemented in one physical computer. When a node sends information to this physical computer, this physical computer can put in global message sequence and time information; store the message; broadcast the message; and service receivers' requests for missed messages. One of the receivers will perform the message acknowledgement function. All the above can be done in software without additional hardware. However, the performance is expected to be poor. This configuration is good only for demonstration and for networks of ten nodes or less.

Another possible alternative is to combine the Central Retransmitter and the Security Controller into a physical computer. The Designated Acknowledger and the Playback Recorders are left separate. In this arrangement, both confidential and non-confidential messages are sent to this single physical computer. The number of messages on the network will be reduced especially for confidential messages as there will be no need for message exchange between the Central Retransmitter and the Security Controller.

For failsafe systems, special redundancy can be introduced. The Central Retransmitter, Designated Acknowledger, and Security Controller would have hot standbys. However, a redundant Playback Recorder may not be necessary. The same conference can be recorded in more than one Playback Recorder. Furthermore, a duplicate network line may be used, If a duplicate network line is available, the system can benefit by using one of these as a broadcast-only line in normal operations.

6. Organizational Implementations and Applications

Two GRSB applications - Backup and Computer Conference are implemented. The GRSB Backup application provides a network journal, eliminating the need for advanced scheduling. It is implemented by broadcasting the data immediately after the data is written onto the user's disk. The broadcasted data would be recorded in the Central Broadcast Controller which has Write-Once-Read-Many times (WORM) disks. If a user's system or disk develops problems, an alternative system can be brought up-to-date from a previously backed up disk and the "network journal". The GRSB Backup application is transparent to the users in normal operations but guarantees that their data can be recovered if their personal computers or disks develop problems. The overhead occurs only on write operations and should be acceptable with a small number of users.

The GRSB Computer Conference application allows an authorized user to initiate a conference; specify a list of invited writers and/or readers; exchange information and maintain data confidentiality. A conference initiator can specify participants by individual or group names. A new employee may retrieve all historic information on a conference from computer X or its archives. The company uses this application to organize its management information system. Different topics are put into different conferences with the necessary attributes of confidentiality, authorized access, audit trails, etc.

6.1 Large Number of User Nodes

If the Company now expands to hundreds of employees housed in the same building, the new configuration is shown in the following figure 2.

The logical components are now implemented in separate computers and that a special Broadcast Line is used. The use of separate computers allows the number of user nodes to greatly increase (from tens to hundreds). The special broadcast line is used only for sending the broadcast and the acknowledgment messages between the Central Retransmitter and the Designated Acknowledger. There will be no congestion on this Broadcast Line thus eliminating the possibility of catastrophic jamming (Catastrophic jamming occurs if there is a temporary congestion on a network, and additional messages are sent to try to recover the lost messages. These additional messages cause more congestion. The situation deteriorates until no useful messages can be sent. This possibility can occur on a single Ethernet implementing the GRSB technology.) The Broadcast Line can also be Ethernet based so that the system can fall back to the single line mode in case one of the line fails. The GRSB protocol specification caters for this possibility. To further enhance reliability, the Central Retransmitter, the Designated Acknowledger and the Security Controller have conventional "hot standby" configurations. In a hot standby configuration, a Secondary (S) node takes over if the Primary (P) node fails. So long as these GRSB logical nodes are functional, all user nodes effectively have the "cold standby" capabilities via the GRSB Backup application.

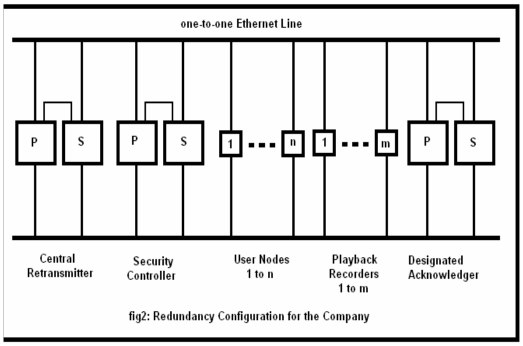

Figure 2. Redundancy Configuration of the Company

To enhance efficiency, an intelligent Ethernet card with a built-in Data Encryption Standard (DES) chip is used in all nodes. This allows message filtering and encryption/decryption to be done in hardware. The performance of the user nodes will not be degraded by unwanted broadcast messages.

6.2 Hot Standby Capability

Some applications of the Company require Hot Standby capabilities. That is, if the primary computer fails, a secondary computer can take over with little down time or loss of data. In existing technologies, the primary computer communicates all relevant information to the secondary, keeping it up-to-date. In the GRSB implementation, all relevant information is broadcasted. There can be more than one secondary computer and the secondary computers do not need to be in exact synchronization with the primary computer or with each other all the time. This means a secondary computer can be doing some non-critical tasks and may occasionally be so busy that it misses some broadcast messages in real time. However, these missed messages can be recovered from a Playback Recorder with guaranteed reliability. The GRSB Hot Standby application can be thought of as an improvement of the original GRSB Backup capability.

6.3 Distributed Database Application

For efficiency reasons, the Company has some identical databases at each local site. With traditional systems, there is a concern that these identical databases will get out of synchronization and would not be identical after a certain time. In traditional systems, it is difficult to guarantee that all database nodes receive all update messages correctly in exactly the same sequence [8]. The GRSB technology provides such guarantees. In a GRSB Distributed Database system, an update message must be broadcasted via the Central Retransmitter node. If two user nodes want to update the same piece of data, one update request will be broadcasted before another. In the GRSB technology, the first update request will be honored while the second will be rejected. All distributed database nodes are guaranteed to receive the update messages in the correct temporal order. This means the message serialization feature of GRSB can help to solve the very difficult problem of distributed database updates. Instead of using "remote database locking", two or three phrase commit techniques, GRSB uses the simple message serialization feature.

7. Limitations of GRSB Protocol

1. The requirement of dedicated "multipoint broadcast" line(s) of arbitrary (possibly very large) length in GRSB [1,2] implementation seems impractical due to the limitations of wired-line capacity.
2. Efficiency of GRSB [1,2] module (in quality and quantity) should be in accordance with the effectiveness i.e. whether it is broadcast or group/multicast.

8. Approximate Solutions to these Problems in the Hierarchical Model

Nevertheless, the Hierarchical model discussed in this chapter has the following approximate achievements.

1) Broadcasting in temporarily disconnected network(s) is resolved by the coordinated service of Central Retransmitter and Playback Recorder of GRSB [1,2] as implemented in this modified multilevel GRSB model.

2) For similar reason extra processing overhead of slower receiver node(s) according to the broadcast protocols [3] has also been eliminated by the GRSB [1,2].

3) The length(s) of dedicated "multipoint broadcast" line(s) may be reduced locally and therefore globally, as shown in the following figure. Also the number of points in this line is now in fine form.

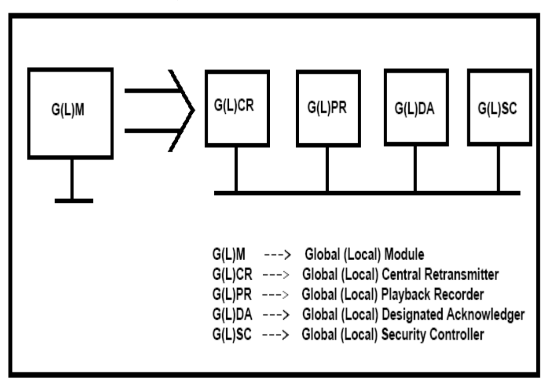

Figure 3: Global (Local) Module

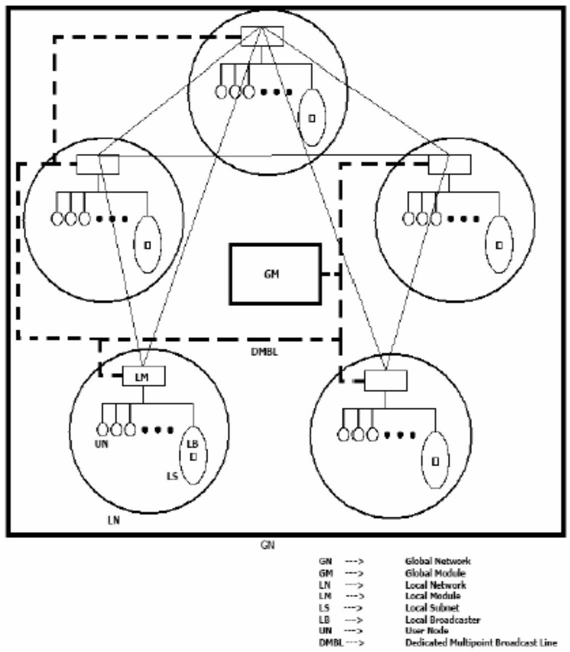

Figure 4: Multilevel GRSB Model

9. Modified Algorithms for the Attachment Nodes

In this hierarchical model, the Designated Acknowledger, Playback Recorder and Security Controller (possibly optional) play almost the same role(s) as in the GRSB [1,2] model. But in addition to these, the residual Global (Local) Central Retransmitter adds Source Local Network Identifier (SrcNetID) with the main broadcast message ***m*** and requires some 'not the least' modification(s) as described in the following.

9.1 Algorithm for GCR (for level 2 implementation)

*Step 1: Wait until **m** (sent from any LCR_i) is available*

*Step 2:Broadcast **m** to all active LCR_js other than sender LCR_i*

*Wait and Tell GPR to store **m** until all $AckLCR_j$(**m**) (for all LN_js other than corresponding LN_i) are received*

*Send AckGCR(**m**) to the sender LCR_i (to let start local broadcast in corresponding LN_i)*

9.2 Algorithm for LCRi (for level 1 implementation)

*Step 1: Wait until **m** is available*

Step 2:

*If SrcNetID(**m**) = **nil** then*

*SrcNetID(**m**) ← NetID(**LCR_i**)*

*Send **m** to GCR*

*Wait until AckGCR(**m**) is received*

*Broadcast **m** to all UN_js, where $UN_j \in LN_i$*

*EnQueue **m** in **Queue(LPR_i)** and tell LPR_i to store **m** at least **k(D+2)T** times, where **k** is QueuePos(**m**) in **Queue(LPR_i)***

*Also tell LDA_i to send $AckLCR_j$(**m**) to GCR if SrcNetID(**m**) ≠ NetID(**LCR_i**)*

9.3 Algorithm for LPRi (for level 1 implementation):

*If **Queue(LPR_i)** ≠ **empty** then*

m ← **DeQueue(**Queue(LPR_i)**)**

*and send **m** to LB_i to broadcast **m** using RESPONSE Protocol [3] almost as fast as hardware speed*

Wait for **(D+2)T** times

9.4 Algorithm for LBi (for the implementation of RESPONSE Protocol [3] in Local Subnet):

*Step 1: Wait until **m** is available*

Step 2:

*If SrcNetID(**m**) = **nil** then*

*select **m** (one of **m**s in case of collision) for higher level i.e. LCR_i's signature for broadcast*

Else

*select **m** for broadcast according to the RESPONSE Protocol [3]*

9.5 Algorithm for each Node in LSi (for the implementation of RESPONSE Protocol):

*Step 1: Wait until **m** is available.*

Step 2:

*If SrcNetID(**m**) = **nil** then*

*select **m** (one of **m**'s in case of collision) to possibly forward upto LB_i*

Else

*select **m** (even in case of collision) to possibly broadcast according to the RESPONSE Protocol [3]*

10. References

[1] Tseung L.C.N. and Yu K.C., " The Implementation of Guaranteed, Reliable, Secure Broadcast Networks ".

[2] Tseung L.C.N. and Yu K.C., " Implication of the Guaranteed, Reliable, Secure Broadcast Technology to Office Information Systems ".

[3] Gopal A., Gopal I. And Kutten S., " Fast Broadcast in High-Speed Networks".

[4] Hoque M.A., Amin S.M.I, "Efficient Broadcasts in High Speed Networks",B.Sc. Engineering Thesis, Dept. of CSE, Bangladesh University of Engineering and Technology