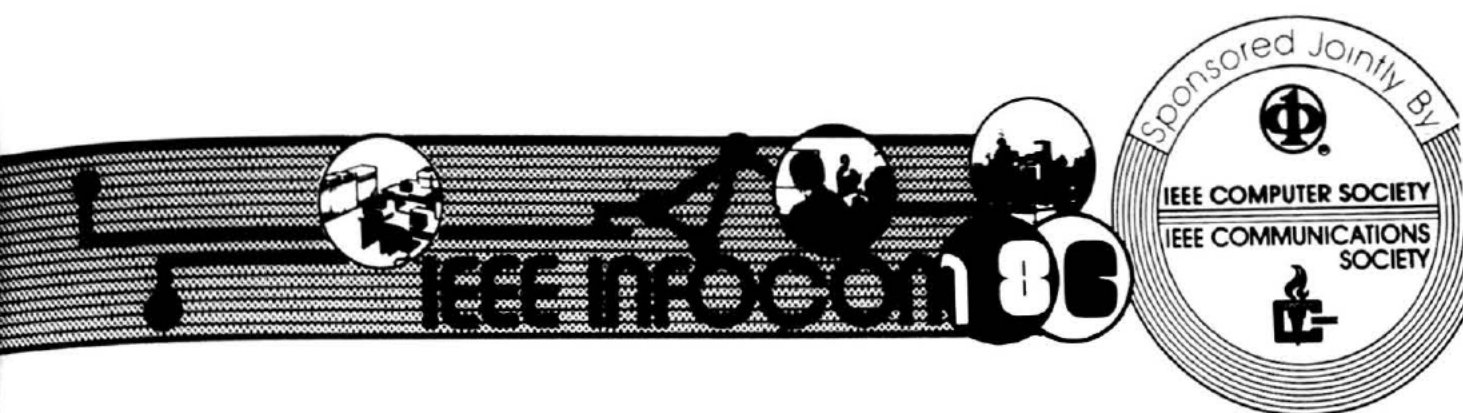


Proceedings

IEEE INFOCOM'86

Fifth Annual Conference

**"Computers and Communications Integration
Design, Analysis, Management"**



Technical Sessions: April 8-10, 1986

CONFERENCE COMMITTEE

General Chair: B.W. Stuck, Teltron, Inc.

Vice Chair: S.J. Yuill, RJO Enterprises, Inc.

Program Chair: B. Gavish, University of Rochester

Sheraton Bal Harbour

Miami, Florida

EXHIBIT

Ex. 1005

 IEEE
COMPUTER SOCIETY

 THE INSTITUTE OF ELECTRICAL
AND ELECTRONICS ENGINEERS, INC.

 IEEE
COMMUNICATIONS SOCIETY

RPX Corporation, IPR2015-00553

ISBN 0-8186-0694-0
IEEE CATALOG NO. 86CH2284-8
LIBRARY OF CONGRESS NO. 86-80342
IEEE COMPUTER SOCIETY ORDER NO. 694

TIME SYNCHRONIZATION AND RANGING FOR MULTIHOP MOBILE RADIO NETWORKS¹

Szu-Lin Su

Victor O. K. Li

Chung Shan Institute of Science and Technology
P. O. Box 1-3-22
Lung-Tan
Taiwan, Republic of China

Communication Sciences Institute
University of Southern California
Department of Electrical Engineering
Los Angeles, CA 90089-0272

ABSTRACT

We propose an efficient algorithm to synchronize geographically distributed nodes in a multihop mobile radio network which uses spread spectrum signals for communication. This synchronism requires the synchronization of the clocks at all nodes and accurate measurements of the distance between them. The distances must be updated periodically to reflect changes in the nodes' locations.

1. INTRODUCTION

A radio network which employs time division multiple access (TDMA) or other time-slotted access schemes (e.g., slotted ALOHA) is required to have all nodes in the system synchronized with a global clock. Furthermore, distributed organization and routing algorithms for multihop mobile radio networks often use TDMA frames to send control messages in order to avoid conflicts. For example, the linked cluster algorithm (LCA) and the link activation algorithm (LAA) in [1, 2, 3] need accurate global timing among the nodes. Therefore, maintaining network synchronization is an important design issue. In addition, in military applications, spread spectrum radios are often used to combat jamming and achieve Low Probability of Intercept (LPI). The acquisition time of the receiver using such communication schemes depends on the uncertainty range (or search window size) of the incoming pseudonoise (PN) phase. If a transmitting node knows the exact PN phase of the receiving node (i.e., it is synchronized with the clock at the receiver) and the distance between them, it can adjust its transmitting phase so that the arriving PN phase will match the local PN phase when the packet reaches the receiver. Under this condition, the acquisition process at the receiver will only take a short time. Therefore, accurate measurement of the distance is essential because the propagation delay may contribute a big PN phase difference between the incoming signal and the locally generated code, and the mobility of nodes will change these distances rather dramatically.

Surveys of existing techniques to synchronize geographically dispersed, mobile nodes are given in [6, 7, 11]. There are essentially four main techniques that have been developed, namely, 1) master-slave, 2) independent clocks, 3) external time references, and 4) mutual synchronization. The independent clocks technique has good survivability characteristics since failures of some nodes will not affect the synchronization in the rest of the network. In addition, it is easy to implement [6]. The drawback of the independent clocks (or plesiochronous) technique is the need for highly accurate clocks and frequent updating. (Note that the independent clocks technique with frequent updating is also referred to as time reference distribution in [7, 11].) However, we may get by with moderately accurate crystal clocks and less frequent updating if we allow some time-error in global synchronization. For example, in spread spectrum systems in which synchronism is required to reduce the size of the acquisition search window, some timing error may be tolerable at the expense of longer acquisition time. In this paper we give an efficient distributed updating scheme for a plesiochronous network of mobile, geographically dispersed radio nodes which use spread spectrum signals to communicate with each other. Our proposed scheme will include the measurement of the distances between any pair of adjacent nodes. In the next section, we describe the network model. In section 3, we describe an algorithm for a fully-connected network. An algorithm for geographically dispersed nodes in a multihop network is described in section 4. We conclude in section 5 with suggestions for further research.

2. THE NETWORK MODEL

As in [12], we divide time into two sets of frames. The first set, repeated periodically, is used for network synchronization (NS) and called the NS frames. The rest constitutes the packet (PK) frames. We assume that there is a transceiver at each node and at any instant, it may either transmit or receive, but not both. Spread spectrum signals are used in the system. When we need to broadcast to all neighbors, a common PN code will be used for all message transmissions and receptions. When

¹This work was supported in part by Army Research Office under Contract No. DAAG29-85-K-0016.

we want to exchange information with a specific node, we use the unique PN code of the receiving node. We assume that the processing gain and the Forward Error Correction (FEC) code are powerful enough so that transmissions of the control messages are successful whenever only one transmission with the same code is received at a node. Each node is assigned a unique integer identity (ID). When a node fails to participate in the network synchronization process for a certain time period or when the routing tables show that this node cannot be reached for a long time, the system will consider that this node has been destroyed and the ID will be reused by any newly added node or recovered node. We assume that before each NS process, each node has a unique ID and each node knows the maximum value N of the ID's.

Since timing messages in our synchronization algorithm are sent using the TDMA scheme, the nodes are assumed to be approximately synchronized to begin with. However, since clocks may have drifted and nodes may have moved between two executions of the NS processes, it is necessary to include a guard time in each TDMA slot to ensure that conflicts are avoided. The required guard time corresponds to the maximum anticipated drift in the clock and the maximum anticipated distance covered by a mobile node between two executions of the NS processes. We shall assume that the distance changes due to the mobility of nodes during the execution of the NS process is small and its contributed error is tolerable. Actually, if the speed of the mobile nodes is below 100 kilometers per hour and it takes one second to complete the whole NS process, then the worst error due to this assumption is in the range of 10^{-7} second or about ten chips for 100M chip-rate spread spectrum system. Note that between executions of the NS processes, i.e., during the PK frames, we can track the distance change roughly by piggybacking the timing information on each message transmission and assuming that the free-running clocks are accurate. We further assume that the processing time for the timing message is negligible.

We next define some notations:

$C_i \triangleq$ the clock time of node i

$T_{ij} \triangleq$ the time indicated by node i 's clock when i receives the timing message from node j

$\bar{i}j \triangleq$ the propagation delay between node i and node j

(We assume $\bar{i}j = \bar{j}i$)

$C_{ij} \triangleq$ the difference between the clock time of node i and node $j = C_i - C_j = -C_{ji}$

$M_{ij} \triangleq C_{ij} + \bar{i}j = T_{ij} - C_j$ (M_{ij} is measured

by node i when it learns C_j from the timing message of node j .)

3. SYNCHRONIZATION FOR FULLY-CONNECTED NETWORKS

For fully-connected networks, all transmissions and receptions of the timing information during the NS frames use the same PN code and the TDMA scheme is used. We designate node 1 as the reference node and it will broadcast its timing message first. A timing message has three fields, including the preamble, a measured value M and the clock time C . (See Figure 1.) In the event that node 1 is destroyed before the beginning of this NS session, the first time slot will be idle and node 2 will take over its role as the reference node. In other words, if node 2 does not receive the timing signal from node 1 in the first time slot of an NS frame, node 2 will assume node 1 has been destroyed and broadcast its own timing message. If node 2 is also destroyed, node 3 will take over, and so on. Therefore, although this scheme has a centralized reference node, it is nonetheless resilient to failures. However, it is required that the reference node (no matter which node it is) will not be destroyed during the session. We next describe our algorithm.

Algorithm.

Slot 1: Node 1 sends a timing message with $M_{11} = 0$ and C_1 . $M_{11} = 0$ indicates this timing signal is from the reference node. Any other node i calculates $M_{i1} = T_{i1} - C_1$.

Slot 2: Node 2 sends a timing message with M_{21} and C_2 .

After this slot, node 1 can calculate $M_{12} = T_{12} - C_2 = C_{12} + \bar{1}2$ and obtain the difference of the two clocks as $C_{12} = (M_{12} - M_{21})/2$. Node 1 can also estimate the distance $\bar{1}2 = M_{21} + C_{12}$.

Moreover, any other node i can compute $M_{i2} = C_{i2} + \bar{i}2 = T_{i2} - C_2$ and will use this value later in distance estimation.

Slots 3 to N : All other existing nodes will send their timing signals (as node 2 has done) in turn. After these slots node 1 gets the information of the distance $\bar{1}i$ and the time difference C_{1i} for all i . Also, node i , $i=2$ to N , knows M_{ij} for all $j=1$ to N .

Slot $N+1$: Finally, node 1 (the reference node) will broadcast the information it has:

Preamble	I.D.	C_{12}	C_{13}	C_{1N}
----------	------	----------	----------	-------	----------

Upon receipt of this message, node i , $i=2$ to N , can adjust its clock by C_{1i} to synchronize with the clock at node 1. Then it can estimate the distances between itself and all other nodes. For example, node i can obtain $\bar{1i} = M_{i1} + C_{1i}$, and $\bar{ji} = \bar{ij} = M_{ij} + (C_{1i} - C_{1j})$ for all $j \neq 1, i$.

The NS session terminates at this point. We see that this algorithm is very simple, but only good for one hop network. In the next section, we shall modify it for multihop networks.

4. SYNCHRONIZATION FOR MULTIHOP NETWORKS

In a multihop network, since each node cannot hear all other nodes' transmissions directly, a routing algorithm must be specified for the communication between a pair of nodes. Numerous routing strategies have been proposed. (See, for example, [4, 9].) But the inherent vulnerabilities of centralized routing restricts its usefulness in a military environment. There are two types of distributed routing algorithms which have been published for multihop mobile packet radio networks. One is the fully distributed scheme [8, 13], the other is the hierarchical distributed routing scheme [1, 2, 3, 10]. Here, we propose network synchronization processes to be used with these two different types of distributed routing algorithms.

4.1. Synchronization under the Distributed Routing Scheme

We assume that the routing strategy described in [8] is used. The routing tables of a node are shown in Figure 2 and consists of an attached device table, a distant device table and a distant node table. The distant node table at each node, say node i , shows the ID's of all existing nodes, their distances from node i (measured in hops and called the tier value), and their corresponding reporting nodes. A reporting node at node i for node j is the next node in the path from i to j . We assume that the routing tables at the different nodes are complete and consistent just before and during the network synchronization process. The NS process is triggered periodically by a randomly chosen node, known as the ultimate reference node. The procedure is as follows:

Phase 1: Each node, say node i , transmits a message containing its clock value C_i , using its unique PN code. Nodes receive and transmit in increasing order of their tier values to the ultimate reference node, say u . Thus u (tier value zero) will transmit first, at which time nodes one hop away (tier value one) will receive. The tier-one nodes will then transmit to the tier-two nodes, and so on. When receiving, each node tunes to the PN code of its corresponding reporting node to node u . Therefore, even if multiple transmissions are heard, it will only

receive the one sent by its reporting node. After this phase, each node, say i , knows M_{ij} , where j is the reporting node of i .

Phase 2: Each node, say i , broadcasts a timing message consisting of C_i , j and M_{ij} in increasing order of ID, where j is the reporting node of i . After this phase, the reporting node j can calculate the clock difference C_{ij} and the distance \bar{ij} between i and j . In addition, every node, say k , knows M_{kn} , where n is a neighbor of k .

Phase 3: Each node, say i , sends C_{uk} 's it has learned so far with its unique PN code in increasing order of its tier value, where u is the ultimate reference node and k is the node which uses i as the reporting node. Before its transmission, node i listens to the transmission of its reporting node to get the value of C_{ui} and to compute C_{uk} 's by $C_{uk} = C_{ui} + C_{ik}$.

Phase 4: In increasing order of ID, each node, say i , broadcasts the clock difference C_{iu} and adjusts its clock to synchronize with node u . So each node can calculate the distance between itself and any neighbor by $\bar{ij} = M_{ij} - C_{ij} = M_{ij} - (C_{iu} - C_{ju})$.

At this point, we have finished our procedure. It takes a total of $2 \cdot (N + m)$ slots to complete the NS process, where N is the total number of existing nodes and m is the maximum tier value in the network for a particular ultimate reference node.

4.2. Synchronization under the Hierarchical Routing Scheme

The distributed algorithms proposed in [1, 2, 3] consist of the linked cluster algorithm (LCA), the link activation algorithm (LAA) and the routing algorithm. These algorithms, which are activated periodically, divide the whole system into several clusters and form the backbone network which is made up of the cluster heads and gateways. Usually, the communication route between any pair of nodes will be decided by the cluster heads and the path will go through the links of the backbone network and links between cluster heads and the nodes. Therefore, for normal packet transmissions and receptions during the PK frames, we only need local synchronization. This means that an ordinary node needs to synchronize only with the cluster head, the gateways and the other nodes in its own cluster. The gateway node, however, should be aware of the clock difference between two clusters so that it can interconnect them. Our algorithm

piggybacks timing information on the control messages generated by the LCA. Our NS process is as follows.

Phase 1: During the execution of the LCA, each node, say i , sends two frames of data. We append C_i and M_{ih} to the end of frame 2 of node i , where C_i is the clock time of node i and h is the cluster head of node i . Note that since the cluster head is always the node with the smallest ID in the cluster [1], and these broadcasts of frame 2 messages are in increasing order of ID, node i will hear the timing message from its cluster head and can compute M_{ih} before its turn to transmit.

After this phase, each cluster head can calculate the clock difference and the distance between itself and any node in its cluster. Moreover, any node, say i , knows M_{ij} , where node j is a neighbor of node i .

Phase 2: Each cluster head, say h , transmits with an unique PN code the clock differences C_{hi} 's, where node i belongs to its own cluster. All other nodes listen to the transmission of its own cluster head and adjust their clocks. So after this phase, we get local synchronization and range measurement within each cluster.

Phase 3: The gateway nodes try to synchronize with the neighboring cluster, i.e., to get the clock difference between the two clusters by exchanging timing information (e.g., M_{ij}) with each other or with the neighboring cluster head.

Phase 4: During the normal packet (PK) frames, the chosen ultimate reference node (which may be the cluster head with the smallest ID, for example) floods its clock time along the backbone links to inform all other cluster heads of the global time.

Phase 5: Before the next execution of LCA, cluster heads broadcast the global time to the nodes in its cluster and each node adjusts its clock to synchronize with the global time.

The NS process will be repeated periodically along with the LCA and LAA algorithms. We see that there is little additional overhead due to the NS process (less than 10 time slots for phases 2, 3 and 5).

5. CONCLUSION

We have described a simple and efficient distributed updating algorithm for plesiochronous system to synchronize a multihop radio network. The time needed to complete this process is proportional to the number of existing nodes. As we have mentioned earlier, since each pair of adjacent nodes which may communicate directly have synchronized their clocks, i.e., they know the difference between their clocks, the distance change due to the movement of the nodes can be traced by piggybacking the timing information C_i . The receiver j can estimate the new distance by computing M_{ji} and assuming C_{ij} is fixed. There are some aspects of this problem which require further study.

1. Analysis of the effect of error on the performance of NS process - The error includes the channel error and the measurement error. The former is due to channel noise and jamming interference. The transmissions and receptions of the timing messages cannot be perfect and the decoding error of the data such as M_{ij} and C_{ij} will affect the timing synchronization dramatically. The measurement error comes from the fact that there is a phase error in the spread spectrum PN code synchronization and there is quantization error on the digital clock. For two-node networks, Ghazvinian and Davarian have given a detailed analysis of the error effect [5]. The analysis for multihop network is much more complicated, since the error will propagate throughout the network and the cumulative effect may be large. Of course, such analysis may result in the development of improved distributed NS algorithms.

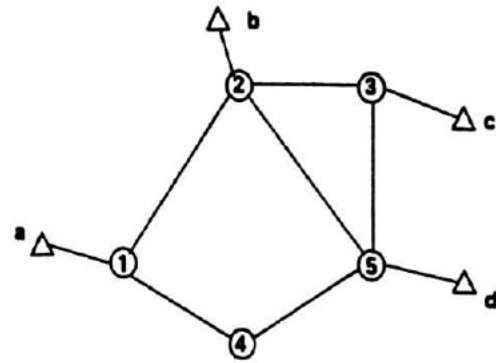
2. In our proposed NS algorithm, we assume that during the synchronization process, the system topology (connection) will not change, i.e., no node (especially the chosen ultimate reference node and cluster heads) is destroyed and all existing links are maintained. We also assume we have consistent and complete routing tables for the network. We would like to investigate: 1) can we relax these assumptions under the described NS process? and if not, 2) how would we modify the proposed scheme to accommodate these changes?

References

1. Baker, D.J. Distributed Control of Broadcast Radio Networks with Changing Topologies. *IEEE INFOCOM*, March, 1983, pp. 49-55.
2. Baker, Dennis J., and Ephremides, Anthony. "The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm." *IEEE Trans. on Commun.* COM-29, 11 (November 1981), 1694-1701.
3. Baker, D. J., Wieselthier, J. and Ephremides, A. A Distributed Algorithm for scheduling the Activation of Links in a Self-Organizing, Mobile, Radio Network. *Proc. IEEE ICC*, June, 1982, pp. 2F.6.1-2F.6.5.
4. Garcia-Luna-Aceves, J. J. and Shacham, N. Analysis of Routing Strategies for Packet Radio Networks. *Proc. IEEE INFOCOM*, March, 1985, pp. 292-302.
5. Ghazvinian, F. and Davarian, F. A Time Synchronization and Ranging System. *Proc. IEEE ICC*, June, 1982, pp. 5H.2.1-5H.2.5.
6. Hagmann, W.C., and Lindsey, W.C. Network Synchronization Techniques. *The Proceedings of the NATO-ASI Conference*, Norwich, England, NATO, August 4-16, 1980.
7. Harrington, E. A. "Issues in Terrestrial/Satellite Network Synchronization." *IEEE Trans. on Commun.* COM-27, 11 (November 1979), 1690-1695.
8. Jubin, J. Current Packet Radio Network Protocols. *Proc. IEEE INFOCOM*, March, 1985, pp. 86-92.
9. Kahn, R., Gronemeyer, S., Burchfiel, and Kunzeman, R. "Advances in Packet Radio Technology." *Proc. IEEE* 66, 11 (November 1978), 1468-1496.
10. Shacham, N. and Tornow, J. D. Future Directions in Packet Radio Technology. *Proc. IEEE INFOCOM*, March, 1985, pp. 93-98.
11. Stover, H.A. "Network Timing/Synchronization for Defense Communications." *IEEE Trans. on Commun.* 28, 8 (August 1980), 1234-1244.
12. Su, S.L., and Li, V.O.K. Performance Analysis of A Slotted Code Division Multiple Access (CDMA) Network. *Proc. IEEE MILCOM*, October, 1984, pp. 23.3.1-23.3.5.
13. Westcott, J. and Jubin, J. A Distributed Routing Design for a Broadcast Environment. *Proc. IEEE MILCOM*, October, 1982, pp. 10.4.1-10.4.5.

Preamble	M_{i1}	C_i
----------	----------	-------

Figure 1: Format of a Timing Message for a fully-connected Network.



(a) Example Network

Attached Device Table	Distant Device Table	
	Device ID	Node ID
a	b	2
	c	3
	d	5

Distant Node Table		
Node ID	Reporting Node ID	Tier Value
1	1	0
2	2	1
3	2	2
4	4	1
5	4	2

(b) Typical Routing Tables at Node 1 for the Network shown in (a)

Figure 2: Routing Tables under the Distributed Routing Protocol.