Implementing Messaging Services on Ad Hoc Community Networks using Proxy Nodes

Tetsuro Ueda[*], Somprakash Bandyopadhyay[+], Kazuo Hasuike[*]

[*] ATR Adaptive Communication Research Laboratories
2-2-2 Hikaridai, Seika-cho Soraku-gun
Kyoto 619-0288 JAPAN
{teueda, hasuike}@atr.co.jp
[+] MIS Group, Indian Institute of Management Calcutta
Joka, Diamond Harbour Road, Calcutta 700 104 INDIA
Somprakash@iimcal.ac.in

Abstract - Implementing messaging services (both off-line e-mail and on-line instant messaging) are critically important in any community network. However, this is a difficult task in ad hoc community networks with dynamic topology and decentralized control. This paper is an attempt to propose a framework towards implementation of messaging services on ad hoc community network. In an ad hoc community network with decentralized control, there is no known mechanism to ensure delivery of e-mail messages from a member of the community (currently on-line but will go off-line anytime) to another member or a group of members of the community, who are currently off-line (but will join the net eventually). At the same time, implementation of on-line, instant messaging services, which relies on some form of effective connection management protocol to ensure a continuous connectivity among a group of users, are also difficult because of dynamic topology. We have introduced a notion of proxy and implemented it. A proxy-node is an on-line node that would perform some of the task of a node, which is currently off-line. At the same time, we have implemented a presence awareness and connection management protocol to ensure continuous connectivity among a group of on-line users in order to implement the instant messaging services over the network.

I. INTRODUCTION

Internet-based messaging system (both off-line e-mail and on-line instant messaging), whether wireline or wireless, completely rely on wired backbone and fixed infrastructure with central administration. However, sometimes, no wired backbone infrastructure may be available for use. Also, there might be situations in which setting up of fixed access points is not a viable solution due to cost, convenience and performance considerations. Still, the group of mobile users may need to communicate with each other and share information between them. In such situations, an *ad hoc network* can be formed [1,2,3].

Conventionally, ad hoc networks are considered as a specific-purpose, temporary network within a group. However, researchers are also thinking of more general-purpose ad hoc networks to support wider applications [3]. Open community ad hoc network is an example that can be used to serve the communication need of a local community [3]. It is an open network in the sense that it is freely available for any member of a local community for information access and exchange.

In this context of wireless ad hoc community network, it is a challenging task to implement similar messaging services that are currently available with Internet. In an ad hoc community network with decentralized control, there is no known mechanism to ensure delivery of e-mail messages from a member of the community (currently on-line but will go off-line anytime) to another member or a group of members of the community, who are currently off-line (but will join the net eventually). At the same time, implementation of on-line, instant messaging services, which rely on some form of effective connection management protocol to ensure a continuous connectivity among a group of users, are also difficult because of dynamic topology of the network. Implementation of presence service in this context is another issue, since there is no known way in ad hoc networks to maintain the status of a node, when it goes off-line.

The objective of this paper is to propose a decentralized framework for implementing messaging services on mobile, wireless ad hoc networks. We have introduced a notion of proxy and implemented it. A proxy-node is an on-line node that would perform some of the task of a node, which is currently off-line. In Internet based communication, centralized servers hold the information to implement a messaging system. Here, the key idea is to implement a framework such that the *network* holds similar information in a decentralized fashion. At the same time, we have implemented a connection management protocol to ensure continuous connectivity among a group of on-line users in order to implement the instant messaging and presence services over the network.

The paper is organized as follows. We discuss the model and requirement specifications for messaging services over ad hoc network in section 2. In section 3, we discuss a presence awareness and connection management protocol, which would enable a set of nodes to become aware of other nodes and maintain continuous connectivity among a set of nodes in an adaptive fashion for on-line communication. Section 4 discusses the implementation framework for messaging services on ad hoc networks. In section 5, we present some simulation results to show the effectiveness of the framework, followed by concluding remarks in section 6.

II. Messaging Services on Ad Hoc Networks

A. Peer-to-Peer E-mail Messaging

In conventional networks, there is a set of centralized entities to control and coordinate e-mail messaging from a sender to its recipient(s). The lack of these centralized entities makes the situation different in case of ad hoc networks. If X wants to send a message to Y who is currently not on-line or disconnected from X, then X has to buffer this message for a deferred transmission (i.e. X has to wait for Y to become accessible). However, the situation would be complex, if X has to leave the community network or switch itself off and if X still has a desire that Y should get this message as soon as Y joins the network. In other words, X wants Y to receive its message even if X is away from the network. In that case, X has to designate some node as its *proxy* to do some of its unfinished activities in his absence.

Definition 1. *In an ad hoc community network, an on-line node m is said to be a proxy-node for n, if n designates m to perform some of its role in absence of n.*

It is obvious that node m ceases to be a proxy-node for n, if (i) n rejoins the network, and (ii) m leaves the network. Therefore, (i) when n rejoins the network, m should transfer back its proxy-status to n; and, (ii) when m leaves the network, it should designate some other on-line node as its proxy.

Definition 2. *Nesting of proxy-nodes. If node m, which is a proxy-node for node n, designates p as its proxy node, then node p will act as a proxy node for both m and n.*

However, for an effective implementation, there should be a limit to this nesting. Moreover, there should also be a limit on number of proxy-responsibilities that an on-line node would require to handle. These implementation issues will be discussed in section 4.

B. Instant Messaging and Presence Service

Instant Messaging (IM) is a means for sending small, simple messages that are delivered immediately to online users. A presence service allows users to be aware of other users' status[4]. In the context of ad hoc network, it is difficult to manage uninterrupted connections among a set of nodes in order to implement on-line instant messaging in ad hoc networks. To manage a connection between two nodes for a long span of time, the source node needs to be aware of the changing route status and subsequently of newly available routes. This points to a form of topology awareness [5] that should be incorporated. In the next section, we have addressed the mechanism of managing an uninterrupted stable session between two nodes (through adaptive selection of multiple routes) until the instant messaging session is over. We have proposed a modified link-state based table-driven connection management protocol that captures the approximate network status periodically without generating lot of control traffic. This would help a node not only to find a path to other nodes on a continuous basis, but also to keep track of approximate status information (presence awareness) of other nodes.

III. Connection Management Protocol in WACNet

A. Overview

We have developed a prototype WACNet (Wireless Ad Hoc Community Network) that uses small, low-cost directional antenna, known as ESPAR (Electronically Steerable Passive Array Radiator) antenna, with each user terminal [6,7]. Use of directional antenna in the context of ad hoc wireless networks can drastically improve the medium utilization through overlapping communications in different directions. However, all the existing routing schemes proposed in this context [1,8,9] rely on the use of omni-directional antenna. The use of directional antenna to find out a route and use it in data communication has not been explored properly. We have proposed a modified link-state based table-driven routing protocol that captures the approximate network status periodically without generating lot of control traffic. It uses the directional capability of adaptive antenna for capturing, disseminating and using the network information for directional routing. This will make each node in the system topology-aware. The protocol enables each node to constantly evaluate network conditions and to take decisions on adaptive route selection. This would enable to establish a framework for instant messaging services on WACNet.

B. The Framework

In order to make the directional routing effective, a node should know how to set its transmission direction effectively to transmit a packet to its neighbors. So, each node periodically collects its neighborhood information and forms an Angle-SINR Table (AST). $SINR^{u}_{n,m}(t)$ (Signal -to-Interference and Noise Ratio) is a number associated with each link $l^{u}_{n,m,}$ and is a measurable indicator of the strength of radio connection from node n to node m at an angle u with respect to n and as perceived by m at any point of time t. AST of node n specifies the strength of radio connection of its neighbors with respect to n at a particular direction.

Affinity of node m with respect to node n, $a^{w}_{n,m}(t)$, is a number associated with a link $l^{w}_{n,m}$ at time t , such that $a^{w}_{n,m}(t) = \mathrm{Max}\ [SINR^{u}_{n,m}(t),\ 0< u <360]$. In other words, the transmission angle w with respect to n maximizes the strength of radio connection from n to m, as perceived by m at any point of time. This maximum SINR value is affinity of m with respect to n and this is obtainable when the antenna at n is directed towards m at an angle w with respect to n. Based on this, a Neighborhood-Link-State Table (NLST) at each node is formed. The NLST of node n, at any instant of time, will help us to determine the best possible direction of communication with any of its neighbor.

We have designed a modified link-state protocol to make the nodes in the network topology aware. Our primary aim is to collect all topology-related information from each node in the network and distribute them periodically (as updates) to only one of its neighbor, without flooding the network with topology-update packets. A node maintains a complete (but approximate) topology map of the entire network, called Global Link State Table(GLST). It not only depicts the connectivity between any two nodes but also the strength of connection or affinity value of the connection. This link-metric will help us to determine a stable route as defined as follows:

Path Stability: Given a path p = (i, j, k, …, l, m), Path Stability of p, $\eta^{p}_{sd}(t)$, *at some instant of time t from source s to destination d will be determined by the lowest-affinity link in the path (since this is the bottleneck for the path),and is given by:* $\eta^{p}_{sd}(t) = \min [a^{\bullet}_{i,j}(t)],\ \forall i,j \in p$.

In order to use the directional antenna, a node propagates its perception of the topology-information to only one of its neighbors at a periodic interval. Selection of target neighbor to propagate topology-map based on a criterion termed as least-visited-neighbor-first. Each node monitor a metric called recency of its neighbors to decide which of them has received least number of update messages. The neighboring node that has received least number of update messages so far will be the target node for updating.

It is to be noted that by controlling the periodicity of updates (P_T), it is possible to control the topology-update-traffic in the network and the accuracy of topology map stored in each of the node. So, the network would never get flooded with propagation of topology updates. The total number of update packets moving around the network will be equal to the number of nodes in the network.

Since the propagation of topology updates from different nodes is asynchronous, it becomes imperative to introduce a concept of recency of information. To implement that, every node in the network has a counter that is initialized to 0. When a topology-update-packet leaves a node, it increments that counter by one. We term this counter as *recency*. Thus at any point of time, the magnitude of the *recency* of any node represents the number of times that node has been updated since the commencement of the network. This also implies that if two update-packets have a set of data concerning the same node, say node n, then the update-packet carrying the higher recency token value of node n has more current information about it. At the same time, the recency token has another role to play. Any node, say n, selects a target neighbor to propagate topology-update-packet based on the value of recency of its neighbors. The neighboring node that has received least number of update messages so far, i.e. having lowest value of recency token, will be the target node for updating.

C. Connection Management and Presence Awareness

We are now in a position in which each of the nodes has a local view of the network topology i.e. each node is topology aware. Each node can now determine the most stable route locally and initiate the sending of data packets through it. After a point of time, if the source node finds that the chosen route has attained a low stability (indicating that it would soon cease to exist), the node computes a new, more stable route from the local information cache and redirects data packets through the later. This adaptive route selection facilitates continuous communication through multiple paths in the temporal domain.

In Figure 1, we have shown the instant messaging service between node 14 and node 5. The figure shows the stability of most stable path between two arbitrary nodes 14 and 5, sampled at each 5-second interval of time. As shown in the figure, we are getting a sustained stability between node 14 and 5 through different intermediate nodes. In other words, it is possible for a source to find out a series of multiple paths in temporal domain to complete a instant messaging session.

IV. Implementation of Messaging Services on Wacnet

A. Assumptions

Cooperation : All nodes in the network are willing to cooperate in order to sustain the network and its functionalities.
Connectivity : No node will be *permanently* isolated from the network.
Liveness: If the number of on-line members in the network goes below a certain threshold (say M_{min}), the network will cease to operate. So, it is assumed that the network would have at least M_{min} number of on-line members to ensure the liveness of the network.

B. Implementing off-line messaging Services

Whenever a node is going off-line, it may decide to have a proxy and designate another on-line node to act as a proxy on his behalf. A node has an approximate knowledge about who can be its proxy (to be discussed in Table I). So, the node sends a proxy-request. If the node receiving the proxy-request is willing to honor the request, it will send an acknowledgement. Otherwise, it will redirect this request to another on-line node.

It is to be noted that if the number of proxy-responsibilities per node is large, it will only be a burden on memory and processing capability of a node. So far as communication is concerned, whether a message is delivered by an actual node or by a proxy-node, network traffic will be the same. So, in our current implementation, we have not enforced any limit to this number. However, the number of proxy-responsibilities will be evenly distributed among all on-line nodes so that no node is over-burdened. This will also ensure some uniformity in network traffic flow.

In section 3, we have discussed the formation and propagation of Global Link-State Table (GLST), which would contain the approximate topology information. In the present context, GLST would carry one more information about an on-line node: its proxy-status. Thus, each node, while updating GLST, will also update its own status, which would then be propagated in the network.

Table I shows a modified GLST (MGLST) at node N_1 at some instant of time t. Let us assume that a WACNet consists of eight members (N_1 to N_8), out of which four are currently on-line (N_1 to N_4). N_1 is connected to N_2, N_2 is connected to N_3 and N_3 is connected to N_4. The first four columns indicate the affinity-values of the corresponding links and the fifth column indicate the recency-value of each node. The nodes N5 to N_8 are currently off-line. N_1 is the proxy-node for N_8, N_2 is the proxy-node for N_5 and N_3 is the proxy-node for N_7. N_6 has no proxy at this moment.

TABLE I

A MODIFIED GLST (MGLST) AT NODE N_1 AT t=0

	N_1	N_2	N_3	N_4	recency	Proxy_for
N_1	-	a_{12}	-	-	$Value_1$	N_8
N_2	a_{21}	-	a_{23}	-	$Value_2$	N_5
N_3	-	a_{32}	-	a_{34}	$Value_3$	N_7
N_4	-	-	a_{43}	-	$Value_4$	

As indicated earlier, Table I shows the approximate network status as perceived by N_1 at t=0. N_1's perception about N_2 is more accurate, since N_2 is its immediate neighbor. But N_1's perception about N_4 is much less accurate, since N_4 is its furthest neighbor. So, at that time instant, let us assume that N_4 has just gone off-line (with N3 acting as its proxy), while N_1, from its MGLST, still has a perception that N_4 is on-line.

Now, let us consider the following situations:

Situation 1: N_1 wants to send a message to N_4.

From its MGLST, N_1 assumes that N_4 is on-line and tries to send its message via N_2-N_3 to N_4. However, as mentioned, N_4 has just gone off-line. If there were no designated proxy for N_4, then there would be a delivery failure, which would be intimated back to N_1 by N_3. But in this case, N_3 is acting as proxy of N_4. So, N_3 will accept this message on behalf of N_4 and wait for N_4 to rejoin and get the message.

Situation 2 : N_1 wants to send a message to N_6 and then to go off-line.

From its MGLST, N_1 knows that N_6 is off-line and having no proxy. However, since N_1 also has to go off-line now, it wants to have a proxy. From its own MGLST, it discovers that N_4 is a good candidate since N_4 currently has no proxy responsibility. So N_1 issues a proxy-request to N_4. But, as indicated earlier, N_4 has just gone off-line and only N_3 knows about it (being the nearest neighbor). So, the proxy-request will be terminated at N_3. Now, since N_3 already handling two proxy-responsibilities (N_7 and N_4), it redirects this request to N_2. N_2 accepts it and send proxy acknowledgement to N_1. So, finally N_2 will become the proxy-node for N_1. This is an example of adaptive redirection of proxy-requests.

C. On-line Instant Messaging and Presence Service

The MGLST would contain the status of on-line nodes; off-line status of nodes will be available from their respective proxies. Once again, it is to be noted that this information is approximate information. As indicated in example given in section 4.2.2, the node N_1 has a perception that N_4 is on-line whereas N_4 actually has just gone off-line. There would be a delay in percolating this information and to update the MGLST of N_1. However, this would not cause any serious performance problem, since N_1 can eventually aware of the status of N_4, either pro-actively (attempt to send a message to N_4), or reactively (when the new status information percolates and reaches N_1 after the occurrence of this event). If N_1 wants to know the off-line status information of a node, it has to send an explicit query to its proxy-node.

The connection management protocol discussed in section 3 would provide a framework to send and receive instant messages among a group of nodes by maintaining a continuous connectivity among them.

V. SOME RESULTS

The proposed framework has been evaluated on a simulated environment in which mobile nodes are distributed randomly. We present simulation results for networks with 50 mobile hosts, operating at a transmission range of 250m. The speed of movement of individual node ranges from 5 m/sec to 30 m/sec.

Figure 1 has already been explained. Figure 2 illustrates the use of proxies in off-line message communication. It is assumed that only 20 nodes out of 50 are on-line nodes at any instant of time. To implement this in the simulator, the simulator starts with 20 randomly selected nodes. Rest 30 nodes are off-line nodes. At each simulated time-tick, one node from the pool of off-line nodes are selected randomly and added in the network. At the same time, one on-line node is removed from the network and added in the pool of off-line nodes.

Figure 2 illustrates a scenario where the node s has decided to send a message to all 50 nodes in the community (including s). However, since only 20 nodes (including s) are available at t=0, the message from s is delivered instantly to those 20 online nodes only. Between t=1 and t=22, the message was delivered to 18 other new nodes. However, the source then went off-line with another on-line node (proxy #1) acting as a proxy of s. Proxy #1 could deliver 6 messages to 6 new comers in the network before it went off-line at t=42 giving responsibility of message delivery on behalf of s to another online node proxy #2. Proxy #2 delivered three messages to 3 other new comers and then found s to be on-line again. So, it transferred its proxy-responsibility back to s. s could deliver two messages during its stay in the network (from t=80 to t=112) before it went off-line again nominating another on-line node proxy #3 as its proxy. Proxy #3 delivered the last message at t=120 to the last new comer.

VI. Conclusion

We have proposed a framework towards implementation of messaging services on ad hoc open community network. If the partitioning of network can be avoided with properly selected transmission range and if the network is alive (i.e. operational with number of on-line nodes > M_{min}), it is possible to implement an effective messaging system in spite of mobility using the proposed decentralized framework.

References

[1] E. M. Royer and C-K Toh, "A Review of Current Routing Protocols for Ad hoc Wireless Networks", IEEE Personal Communication, April 1999, pp. 46-55.

[2] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols,' *Proc. ACM/IEEE Mobile Comput. and Network.*, Dallas, TX, Oct. 1998.

[3] K. Mase, M. Sengoku and S. Shinoda, "A Perspective on Next Generation Ad Hoc Networks: A Proposal for Open Community Network", IEICE Trans. Fundamentals, Vo. E84-A, No.1, January 2001.

[4] M. Day, J. Rosenberg and H. Sugano, "A Model for Presence and Instant Messaging", Network Working Group, Request for Comments: 2778, February 2000.

[5] S. Bandyopadhyay, Romit RoyChouldhury and Krishna Paul, "Topology Discovery in Ad Hoc Wireless Networks Using Mobile Agents", Proc, of the Second International Workshop on Mobile Agents for Telecommunication Applications, Paris, France, September 18-20, 2000.

[6] T. Ohira and K.Gyoda, "Electronically Steerable Passive Array Radiator (ESPAR) Antennas for Low-cost Adaptive Beam forming", IEEE International Conference on Phased Array Systems, Dana Point, CA May 2000

[7] S. Bandyopadhyay, K. Hasuike, S. Horisawa, "An Adaptive MAC Protocol for Ad Hoc Networks with Directional Antenna, 7th Asia-Pasific Conference on Communications (APCC 2001), Tokyo, Japan,17-20th September, 2001.

[8] S R Das, Charles Perkins and Elizabeth Royer, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks", Proc. Of the IEEE INFOCOM 2000, Tel-Aviv, 26-30 March, 2000.

[9] Guangyu Pei, Mario Gerla and Tsu-Wei Chen, "Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks" in Proc of the IEEE International Conference on Communication, New Orleans, LA, June 2000.

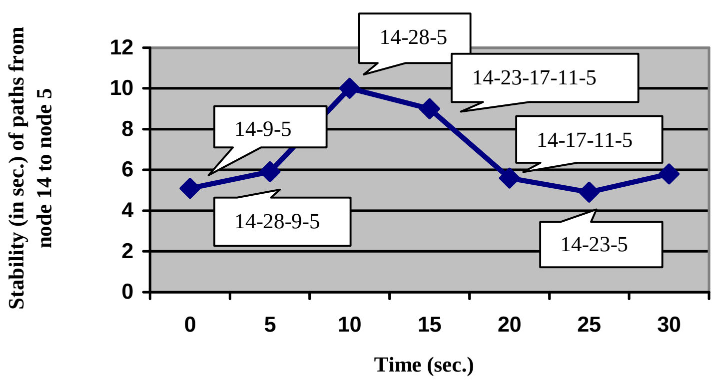

Fig.1. Continuous Connectivity Management between node 14 and node 5 for Instant Messaging Service between them

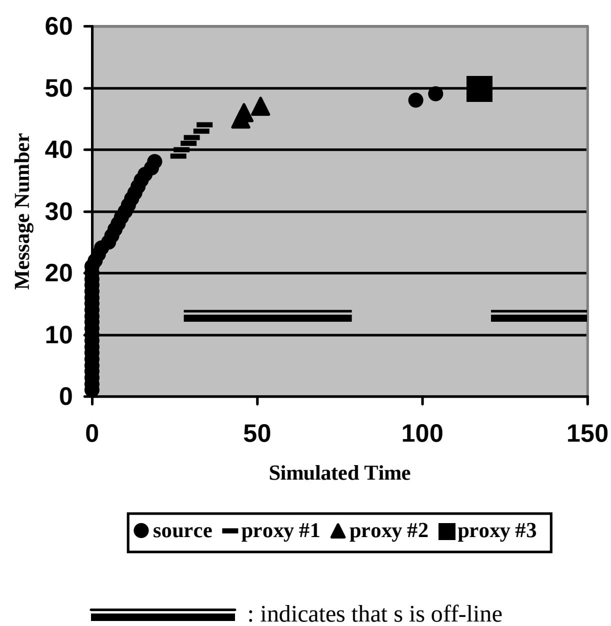

Fig. 2. Delivery of e-mail messages from a source and its proxy to all other nodes in the community

its