A Framework for a Multi-mode Routing Protocol for (MANET) Networks

César Santivañez Ioannis Stavrakakis
Department of Electrical & Computer Engineering
Northeastern University
Boston, MA 02115, USA

Abstract— **Traditional routing protocols for MANET networks are usually designed with a particular environment in mind and fail to adapt to the wide range of environments present in a MANET network. Because of the wide diversity of the conditions that may be encountered in a MANET network it seems that it would be difficult to effectively route information by engaging a single type of protocol. Instead, a multi-mode protocol should be developed which applies the appropriate "mode" or protocol that is determined to be effective at a given point in time and for the appropriate subset of the network. In this paper, two algorithm : the limited link state (LLS) and Self-organizing (SO), that adapt to any MANET network condition –from small networks with nodes with low mobility to large networks with highly mobile nodes to an heterogeneous network with different classes of users – in order to make the most efficient routing decisions are discussed. These algorithms employ new metrics that capture the mobility and traffic pattern of (subsets of) the network, along with some attributes considered in the past (e.g. link stability). Once the instantaneous structure of the network is determined, the multi-mode routing protocol may select its 'mode' of operation.**

I. Introduction

A MANET network may be envisioned as a quickly deployable association of mobile nodes whose motion pattern is not well known in advance in an environment lacking supporting infrastructure. Some nodes may have routing capabilities, whereas some others may not. Some nodes may be highly mobile whereas others may be very slow moving or even fixed. Some nodes may have only one interface, whereas other nodes may have several interfaces, allowing them to become gateways between different classes of nodes. The bandwidth available and the power constrains may also vary notably from one node to another. In summary, there may (or may not) be a high degree of heterogeneity in a MANET network.

MANET networks appear in military and disaster recovery operation, or in a freely association of users as for example in a conference room or a university campus. In all these cases what characterizes a MANET network is not a lack of structure but the existence of an instantaneous, dynamic, not known *a priori* structure intimately related to the task being executed.

The above characteristics of a typical MANET network make the problem of effectively routing in a rapidly varying, bandwidth constrained MANET network supporting a wide range of users with different characteristics (mobility, traffic patterns, etc.) a very challenging one.

Traditional link-state and distance vector algorithms would fail in a highly dynamic MANET network environment since they would consume a significant portion of the available bandwidth to maintain routes that may no longer be valid at the time needed (rapid changing environment). Traditional hierarchical approaches are not effective in this environment either, since the topology of the network is changing over time, and traditional hierarchical approaches do not allow for the protocol to adapt to dynamic network conditions.

Some protocols have been proposed [1]-[11] that appear to be adequate or even effective in a small to medium network, but they soon become ineffective as the network size and/or the mobility level increase and/or the available bandwidth decreases. With the exception of ABR [7] and CEDAR [2] (which try to favor "more stable" links) none of these protocols attempt to obtain/exploit the network state (mobility and traffic patterns) and/or structure. These protocols are usually designed with a particular environment in mind and fail to adapt to the wide range of environments present in a MANET network. Because of the wide diversity of the conditions that may be encountered in a MANET network it seems that it would be difficult to effectively route information by engaging a single type of protocol. Such diverse environments cannot be effectively taken into consideration by simply adjusting the parameters of a single protocol. Instead, a multi-mode protocol should be developed which applies the appropriate "mode" or protocol that is determined to be effective at a given point in time and for the appropriate subset of the network.

The objective of this paper is to present a framework for a MANET routing protocol that adapts itself to the present network conditions taking into consideration the mobility levels and patterns, as well as traffic patterns. In order to identify and utilize the network conditions (state information) the multi-mode routing protocol has to rely on some structure-learning/engaging algorithms that extract the network state information (defined in terms of proper metrics) and based on it implement the proper mode of the supported multi-mode routing protocol. The present work identifies parameters (metrics) that define the state of the network. Based on these metrics, structure-learning/engaging algorithms that extract the network state information and enable the implementation of a multi-mode routing protocol may be developed. As a starting point, two complementing structure-learning/engaging algorithms are discussed in sections III

This research was supported in part by the GTE Corporation.

and IV,respectively, along with their associated metrics : the Limited Link State (LLS) and the Self-Organizing (SO) algorithms.

II. Routing considerations for MANET networks

In traditional routing protocols (such as the Ideal Link-State (ILS) protocol) a message is generated each time a link's state changes. Consequently, network nodes are aware of the state of links and can precalculate routes to potential destinations nodes (proactive routing protocols). The bandwidth overhead (cost) associated with maintaining precalculated routes is proportional to the frequency of link-state changes or the rate of topological change. The latter is defined as the average number of link-state changes in the entire network per time unit and it is, thus, proportional to the net size and inversely proportional to the mean time to link failure.

In MANET networks link-state changes are mostly due to user mobility. If the rate of topological change is low the proactive ILS protocol would still be effective. On the other hand, if the rate of topological change is moderate to high the ILS protocols would be very inefficient and more effective routing protocols should be employed.

In a diverse MANET network with moderate to high rate of topological change there would typically be links of low rate of change (high stability or stable) and links of moderate to high rate of change (low stability or unstable). An effective routing protocol for such a network should then be able to :

(A) Discover the stable links, properly propagate their state and enable nodes to precalculate routes to destinations which are reachable by using such stable links.

(B) Provide for an effective mechanism to establish routes to the remaining destination, as well as alternative routes to precalculated ones which may be overutilized and become congested.

An example of an algorithm that supports (A) is the Limited Link-State (LLS) algorithm presented in section III. The LLS algorithm will adapt to the network conditions trying to take advantage of the network structure. It will construct routes toward some destination nodes (even if not necessary) at a low cost (bandwidth overhead).

It should be noticed that the stable paths (along stable links) are expected to be a small portion of the possible paths (stable plus unstable) in a typical MANET network. Consequently, these paths may be overutilized and become congested if no alternative paths are available leading to network throughput reduction and delay increase. Thus, a mechanism to identify less stable routes to destinations with precalculated routes as well as to those without precalculated routes needs to be developed.

The traditional approach to identify less stable routes in a highly dynamic MANET network is based or route discovery or flooding algorithms, both requiring a potentially large number of broadcasts before delivering the information to the destination. When flooding is engaged, each packet needs to be broadcasted over the entire network. Route discovery relies primarily on broadcast search for a route to the destination each time a session is to be initiated. Such a search induces some start up delay in addition to consuming bandwidth, unless it is bypassed when a recently utilized route is available and it is selected (at the risk of not being appropriate any longer).

The Self-Organizing (SO) algorithm discussed in section IV is an example of an algorithm supporting (B) above, that tries to reduce the number of broadcasts required by the route discovery or flooding algorithms by providing precalculated routes *toward* some destinations that are *likely* to be involved in new sessions. For those routes to be useful, the cost associated with their maintenance should be less than the expected gain of using them.

The SO algorithm will base its decision on the mobility as well as traffic patterns of the nodes. The SO algorithm will attempt to choose Reference Nodes (RN) and around them Reference Areas (RA) such that the expected number of new sessions having a destination inside the reference area (Gain, G) be maximized. This gain (G) has to be compared against a threshold (the cost of maintaining the routes) to decide whether it is worth to create routes toward a particular reference area.

If all the nodes are assumed to have the same traffic patterns, then the SO algorithm will attempt to find the mobility pattern of the network. Although it is possible that the mobility pattern of a network be totally random, that is not usually the case. Human mobility, for example, is based on groups (forming clouds) or follows some patterns (streets, highway, searching, etc.). Even automata mobility is shaped by the function they are executing and therefore there is some degree of spatial/temporal correlation. The SO algorithm will attempt to find (or select) the mobility 'leaders' (nodes around which others node move). For example, in networks formed by cars in a highway, the cars in the intermediate position would be the best candidates to mobility 'leaders'. However, node mobility is not the only factor to take into account. Even more important is the traffic pattern of the nodes. There is no need to precalculate routes for nodes that are not going to communicate at all, whereas there maybe other nodes that may need to be contacted frequently due to their mission (coordinator, server, etc.) . For the latter nodes it should be highly desirable to have routes readily available saving the network from otherwise almost certain broadcasts.

Finally, it was pointed out that a reference area will be created only if it is effective. For networks (or some nodes) with high mobility rate or low traffic demand it may not be effective to create reference areas. To forward packets to those nodes route discovery will be used. If the routes toward the destination are invalidated too quickly, or if the traffic per session is low – say one or two packets – flooding is expected to be more effective.

The LLS and SO algorithms motivated and briefly mentioned above will basically help identify the stable routes in the network as well as clouds of nodes (reference areas) which are worthy to maintain. That is, such algorithms may be viewed as structure-learning/engaging algorithms.

These algorithms are expected to be effective not only in single class MANET networks - as implicitly assumed above - but also when multi-class nodes are present forming a multilevel hierarchy. As an example, mobile nodes may have an additional interface (more power demanding) to communicate with a more powerful base station in case they become isolated. Similarly, some nodes may serve as gateways to a fixed network such as the internet. The reader is referred to [13] for extensions to heterogeneous environments.

III. Limited Link-State (LLS) Algorithm

It is well known that link-state algorithms (referred to here as Ideal Link-State (ILS)) are effective for stable networks. ILS performance degrades enormously as the network topology becomes dynamic and the network stability decreases.

Nodes in a MANET network are expected to have diverse degree of mobility and consequently, some links may be quite stable. Stable links may be associated with nodes with low mobility, with high transmission power or simply with similar mobility pattern (as for example two mobile users walking down the same street or performing a related task). If the link-state information of such stable links is propagated deep into the network (as in ILS), these links would not produce much bandwidth overhead since they do not require frequent update (as unstable links do). This is not the case with unstable links, though, and for this reason their link-state should not be propagated deep into the network at all.

The bandwidth overhead due to link-state propagation is typically a function of the rate of *change* of the particular link and the number of nodes that receive the update. If the depth of the propagation of the link-state information were selected to be inversely proportional to the rate of change of the link (related to the stability of the link), then the bandwidth overhead caused by a link would be independent of the link stability (mobility) and may be bounded. This is the main motivation for the proposed Limited Link-State (LLS) algorithm, where the link stability (defined below) determines the depth of the propagation of link state information. The LLS algorithm may be designed in order to not congest the network under any environment. On the other hand, since the propagation depth is limited to a number that is a function of mobility and not of network size, it is possible that in huge networks even almost fixed link updates will not be available to all the nodes but only to a portion of them. This is not an issue since for huge networks it is likely that a hierarchical approach that facilitates routing (e.g. the SO algorithm) will be employed.

The availability of stable link information to the nodes make it possible for them to detect stable paths over the network if they exist. This way, the nodes will discover some underlying network structure. For example, the existence of some fixed antennas or some high power stations or some highly reliable paths.

LLS's discovering of stable paths allows the forwarding of packets toward certain destination at a low routing overhead cost (because of infrequent link-state updates). Typically, it will be possible for nodes to forward packets to close by destination nodes, since proper link-state information will be available. In general, the closer to a destination a node is, the more information related to that destination it will have. This property is taken advantage of by the SO algorithm.

The LLS algorithm is similar to the link-state algorithm in that they both propagate link-state information, but differ in the information being transmitted (metrics being used) and the depth of the transmission. The LLS algorithm will consider three metrics associated with a link : cost, stability, and quality. The link *cost* is defined as the ratio of the number of nodes accessing the same associated channel over the available bandwidth. Clearly, the cost will increase with the number of neighbors (more interference) and decrease as the bandwidth increases. The link *stability* is defined as the average time the link is active. A link is active if it is not detected absent for a period greater that d. The link *quality* is defined to be the fraction of the active time that the link is *actually* available (A). The link quality may be regarded as an estimate of the probability that a link is available at a particular time given that the link is in state 'active'. It may be seen that the link stability captures the longer term behavior of the link whereas the shorter term behavior is captured by the link quality.

The link stability will determine how far away the link-state information will be propagated by the LLS algorithm. For more stable links the algorithm would propagate the link cost and quality far away whereas for less stable links the link-state information would be transmitted only to the closest nodes. This way, the excessive bandwidth overhead produced by traditional link-state algorithm in highly mobile environments is dramatically reduced. It should be noted that even in a highly dynamic large network, the limited link-state algorithm may obtain stable paths (if they exists) at a low bandwidth overhead cost.

IV. Self-Organizing (SO) Algorithm

A. Motivation

Suppose that a large network is divided into four regions, each associated with one of the four cardinal points (N, S, E, and W). The border of these regions are not well delimited but for most of the nodes it is trivial to say which region they belong to. Consider also a source node desiring to send a message to a destination node in region W. Furthermore, assume that the source knows that the destination is within region W but does not have any route toward this destination node. In this scenario the source node may initiate a route discovery procedure that will result in the broadcast of a REQUEST over the entire network.

Alternatively, the source node could begin forwarding the data packets (not the REQUEST) in the "West" direction, hoping that along the way the packets will be heard by a node that has knowledge of some routes toward the destination. In this case a broadcast will not be necessary. It is obvious that this second alternative is by far more attractive than the first one. To implement this second approach, however, two major issues need to be addressed. First, the nodes do not know which the "West" direction is; and second, the source does not always know in which region the destination currently is.

The first problem can be addressed if one node is chosen inside each region as a "beacon". This node may be for example some node in the center of each region. Each node serving as a "beacon" is referred to as *reference node*. Then, a tracking algorithm (e.g. TORA [5] or some form of geographical routing) may be used to track these four reference nodes. Thanks to the tracking algorithm all the nodes (possible sources) would have downstream links toward each reference node. For example, let N_W be the reference node of the region W. All the nodes (specially the nodes outside W) would have downstream links toward N_W and therefore they have downstream links toward the region W. These downstream links provide each node with a sense of direction towards the region W. Thus the first problem may be solved but at the cost of creating reference nodes and tracking them.

The second problem can also be solved with the inclusion of a location management scheme that takes into account the past history of a node. If a node was in the immediate past inside the W region it is unlikely that it is too far away due to expected spatial/temporal correlation. Thus, the packet could be sent initially in the W region. As the packet crosses the network and gets closer to the destination node, more up-to-date information will be available (by means of the LLS algorithm) and the packet will eventually be routed toward the current location of the destination.

An example of routing using the reference area concept is illustrated in Figure 1. Node S has a packet to send to node D which is inside the reference area R (the reference area is denoted by its reference node). All the nodes (even D) have routes towards node R using some tracking algorithm such as TORA. In this example it is assumed that node S knows in which reference area node D is by means of a location management agent. Therefore, node S sends the packet in the 'direction' of node (reference area) R. The packet will travel the network, following always the direction 'closer to node R' until reaching node I. Node I knows the exact location of node D (by means of the LLS algorithm, or any other approach following the principle : "the closer you are, the more up-to-date information you have") and interrupts the flow of the packet sending it directly to node D (no longer along the direction to node R).

The above approach will reduce the number of broadcasts that other protocols produce. In particular, this algorithm may be seen as an improvement over traditional route discovery/flooding algorithms that with an additional cost dramatically reduces the number of bordercast in a large, dynamic, highly loaded network. It should be noted, however, that for some nodes – due to their high mobility or low use – it may not be cost-effective to maintain reference nodes and tracking their location. For those nodes other alternatives such as route discovery or even flooding may be considered. The proposed protocol also considers these cases by employing route discovery/flooding mechanisms to route packets to these nodes when the cost required to maintain reference nodes to these destinations is greater that the expected cost due to bordercast to these nodes.

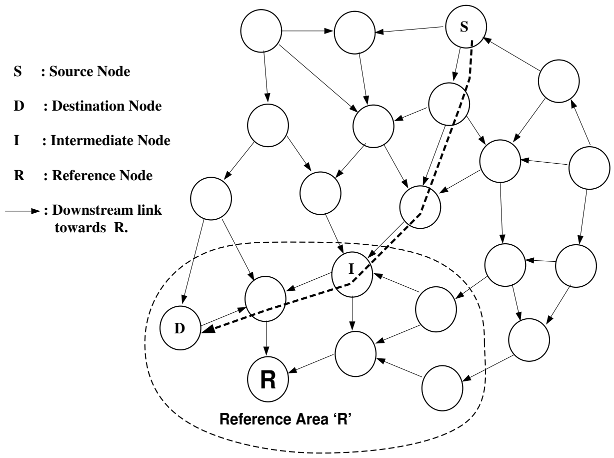

Fig. 1. Routing using the self-organizing (SO) algorithm and the reference area concept

B. Self-Organizing (SO) algorithm overview

The SO algorithm is responsible for choosing the best candidate to be a reference node and define the reference areas. The network will construct routes toward these reference areas in a proactive fashion (before these routes are required).

It is clear that there is a benefit in grouping nodes in a reference area, mainly because the number of broadcasts needed is significantly reduced (gain). On the other hand, there is a bandwidth overhead associated with maintaining a reference area and tracking a reference node (cost).

Route pre-calculation in a highly mobile environment has been generally considered to be inefficient since these route may become obsolete before used. In the SO algorithm, the routes constructed are *likely* to be used because reference areas are created only when it is likely to have packet transport involving them. In other words, the cost of maintaining routes toward the reference area is 'shared' among all the nodes inside it. It may be possible that the SO algorithm decides not to create any reference area (because the gain is lower than the cost), but this decision will depend on the network state and will not be an *a priori* (maybe incorrect) decision of the routing protocol.

The SO algorithm will choose the nodes with the larger expected gain, and if this gain is greater than the expected cost, these nodes will be chosen as reference nodes and define new reference areas around them. The exact calculation of the expected gain of a candidate reference area (node) is not an easy task. In general, that gain would depend on whether the nodes remain inside the reference area (mobility pattern) and whether new sessions having their destination node inside the reference area (requiring otherwise a broadcast) are originated (traffic pattern).

Therefore the gain should be two-dimensional function of mobility and traffic as discussed in the next subsection.

C. Gain function

In [12] the concept of 'footprint size' has been introduced. The footprint size is equal to the number of k-neighbors of a node (nodes that are at a distance of k hops or less). A classification of the nodes may be based on this footprint size. The larger a node's footprint size the larger the region that the node covers and the less likely that a given node will leave that region in the near future. Therefore, a node with higher footprint size would be a better candidate to be a reference node, since it would serve as a beacon for a greater number of nodes. Here, the concept of footprint size is extended to a gain function that will serve to determine which nodes are the most suited to be reference nodes.

The gain function at a level L and at time t for a reference area is proposed to be equal to the expected number of broadcasts saved over the next L seconds. A broadcast will be saved if the destination of a new (or broken) *session* is inside a reference area. Here, the term *session* is applied to a sequence of packets not more than T_s seconds apart. For example two consecutive file transfers to the same destination may be considered as part of the same session, whereas in a transaction that requires a packet to be send every half hour, each packet will be considered to form a new session.

Let A be a potential reference node and let $V(A,t)$ be its reference area at time t; let $G_L(A,t)$ be the gain function of node A at time t and at level L. Each node $i \in V(A,t)$ has two parameters : $S_i(A,t)$ (percentage of the next L seconds that node i will stay inside node A's zone, i.e. within a radius of k from A) and $R_i(t)$ (total expected number of new sessions over the next L seconds). Then, the gain function may be defined as: $G_L(A,t) = \sum_{i \in V(A)} S_i(A,t) R_i(t)$.

Different approaches can be considered to estimate the values of $S_i(A,t)$ and $R_i(t)$, depending on the desired amount of complexity. For $S_i(A,t)$, a first approach could be to set its value to the inverse of the distance (in hops) between nodes A and i elevated to an integer exponent. A better choice may be to consider the percentage of the previous L seconds that the node was inside $V(A,t)$ and assume that this will be repeated. A more complex approach (and likely a more successful one) will be to consider node i's location history over the past L seconds and estimate its trajectory (incoming, outgoing, etc.). Different estimation techniques may be tried to find the more accurate one for a limited complexity implementation.

For the value of $R_i(t)$ one simple approach would be to associate a value equal to 1 for nodes already in a session (likely to be broken) and a fixed small value for nodes not in a communication. This approach will result in a gain function closely related to the footprint size [12] (that is, nodes with greater footprint size will have greater gain). A more complex approach should take into consideration different traffic patterns for different classes of users. For example in a client-server architecture, it is likely that the server receives constant requests resulting in a large number of new sessions, however these sessions will be short resulting in almost no broken sessions. On the other hand, if the node is handling voice communications it is likely to have a small number of new sessions, but since their duration is greater it is more likely to have broken sessions. Therefore, the class of the user has to be taken into account along with its past history to estimate its traffic behavior.

The established set of reference nodes reveal the structure of the network. In fact, they represent the skeleton of a MANET network. For example, if a network is composed of two group of users performing two different tasks, the network would be composed of two reference nodes (areas). Also, since the reference nodes may be required to perform some extra functionalities (due to their leadership role) it would be appropriate to also weight the computational power and available bandwidth. In multi-class networks the availability of a second, more powerful interface can also be considered as a plus for a candidate reference node (see [13]).

Finally, due to space limitations, the reader is referred to [13] for considerations regarding the cost function, the location management scheme, and a detailed exposition of the clustering algorithm.

References

[1] A. Qayyum, P. Jacquet, and P. Muhlethaler. "*Optimized Link-State Routing Protocol*". Internet draft draft-ietf-manet-olsr-00.txt, (November 1998). Work in progress.

[2] R. Sivakumar, P. Sinha, and V. Bharghavan. "*CEDAR: a Core-Extraction Distributed Ad hoc Routing algorithm*". To appear in INFOCOM'99, New York.

[3] C. Perkins. "*Ad-Hoc On-Demand Distance Vector Routing*". MILCOM'97 panel on Ad-Hoc Networks, Monterey, CA, November 3, 1997.

[4] Z. Haas. "*A New Routing Protocol for the Reconfigurable Wireless Networks*". ICUPC'97, San Diego, CA, Oct. 12, 1997.

[5] V. D. Park, and S. Corson, "*A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks.*", IEEE Infocom '97. Kobe, Japan(1997).

[6] D. B. Johnson and D. Maltz, "*Dynamic Source Routing in Ad Hoc Wireless Networks.*", In Mobile Computing, edited by Tomasz Imielinski and Hank Korth. Kluwer Academic Publishers, 1995.

[7] C-K Toh, "*Associativity Based Routing For Ad Hoc Mobile Networks*". Wireless Personal Communications Journal, Special Issue on Mobile Networking & Computing Systems, Vol. 4, No. 2, March 1997.

[8] T. Chen and M. Gerla. "*Global State Routing: A New Routing Scheme for Ad-hoc Wireless Networks* ". In Proceedings of IEEE ICC '98

[9] B. Das, S. Raghupathy, and B. Vaduvur."*Routing in Ad Hoc Networks Using a Spine*".Proceedings of the 6th International Conference on Computer Communications and Networks. Las Vegas, USA. September 1997.

[10] P. Krishna, N. H. Vaidya, M. Chatterjee, and D. K. Pradham. "*A Cluster-based Approach for Routing in Dynamic Networks*". Proceedings of the Second Usenix Symposium on Mobile and Location-Independent Computing, 1995.

[11] M. Gerla and T.C. Tsai. "*Multicluster, mobile, multimedia radio network*".ACM/Balzer Journal of Wireless Networks, 1995.

[12] I. Stavrakakis and R. Landry. "*Management of Communication Resources in a Hierarchical Mobile Mesh Network*". Technical Report TR-CDSP-98-49, Communications and Digital Signal Processing Center, ECE Dept., Northeastern University. Boston, MA. 1998.

[13] C. Santivanez and I. Stavrakakis. "*SOAP : a Self-Organizing, Adaptive Protocol for routing in large, highly mobile ad-hoc networs*". Technical Report TR-CDSP-99-50, Communications and Digital Signal Processing Center, ECE Dept., Northeastern University. Boston, MA. 1999.