Stocking strategy for the routing table in TAP protocol

Cécile Grivaz
email: cecile.grivaz@epfl.ch

Abstract: We study the possible implementation strategies of stocking the routing tables in a logical router, obeying the ant based routing protocol for terminodes, as described in [14]. Logical routers are a collection of node within a certain area. The nodes operate on a common routing table, that is maintained, updated, and queried by all the nodes within a logical router. This is done in the point of view of the messages that should be exchanged in the logical router. Most specifically three approaches are studied: every node holds the entire routing table, few nodes hold it or the table is split between several holder nodes.

1: Introduction

1.1: Ad-Hoc Networks

Mobile Ad Hoc Networks (MANET [1]) are wireless networks. They are self-administrated. They don't rely on any central structure such as fixed base-stations (as in mobile telephony) or wireless access points (as in wireless LAN).

MANETs can be used without static infrastructure and can be quickly set up, making them perfect for temporary connections. They are also cheaper and impossible to restrict, because they release the user of depending on any provider.

They have a dynamic topology and form an arbitrary graph, because each node can move any time, in any direction, and at any speed. Furthermore, as a result of this movement, nodes can leave or enter the network dynamically. Indeed, they can exit the transmission range of any other node, or, in contrary, become inside the range of a network's node.

Since there is no central router, each node of a MANET should be able to act both as host and router, to allow two nodes out of transmission range to communicate, by using one or more other nodes to relay their packets. No protocol designed for wired or centralised networks is suited for this task. Therefore MANETs need new protocols to be developed specifically for them.

Basically, two different approaches of MANETs routing protocols exist which are topology based and do not make use of any possible available location information: proactive and reactive. Proactive protocols require every node to maintain routing informations to every other node (DSDV[7], WRP[8], OLSR[9], CGSR[10]), whereas reactive ones use the source node to demand a route to the destination before sending the packets (AODV[11], DSR[12], TORA[13], LAR[2]).

1.2: Terminodes

Ad-Hoc protocols have been well examined for relatively small networks, but not so much for very large ones. Neither proactive nor reactive protocols are scalable to large MANETs in terms of number of nodes or area. This is because the traffic generated by proactive protocols to maintain the whole routing information in each node would be to high, and the flooding process used to obtain the demanded route in reactive protocols will also be excessive.

Ad-Hoc like networks with a large amount of nodes (up to several millions) are considered for example in the terminodes project [6]. The large number of nodes greatly increases the complexity of the routing protocols. Actual work on terminodes routing tend to use algorithms which rely on the nodes physical position. This reduces the total routing overhead (LAR [2], DREAM[3], GPSR[4], TR[5]). It is usually done by including a GPS receiver on some or all nodes. If a node doesn't have such a device, its location can be calculated from its neighbors using triangulation (a node inside the transmitting range of another is called its neighbor). One has to note that this reduce the reliability and the freedom of the network as the GPS devices have to refer to a fix and proprietary infrastructure (the GPS satellites, under the control of the US Army), which can be shut down at any time.

Each node is addressed by a unique 64 bits identifier and three coordinates specifying its current location. The packets can then be sent using a greedy approach with a fallback mechanism in case of failure. This dramatically reduces the traffic, but is quite ineffective when topography is not convex, i.e. when there is a hole in it. This could be a natural artifact, with few or no nodes, such as a lake in figure 1; or an over-difficult path such as a congested area or poor quality links [14].

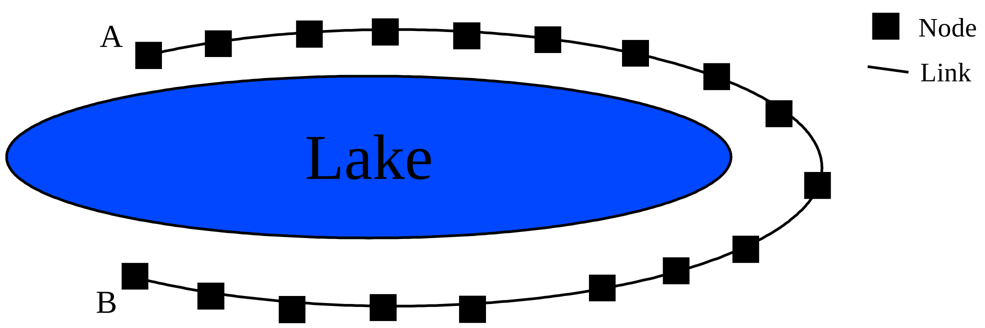

Fig.1 : topology where a greedy algorithm based on location will fail transporting a packet from node A to node B.

1.3: Ants-based routing for MANETs [14]

For all reasons quoted above a new type of protocol should be developed. It should not generate too much traffic or overhead, and be proof to the holes in topology. The solution we propose is a multi layer protocol.
To simplify and fix the topology, we use the topology abstracting layer. This is a fixed logical abstraction of the topologically changing network. It uses the

only fixed resource available: the physical area. It is divided into several squares defined by their coordinates, patching the whole area so that each node is inside one and only one square. Each square is called a logical router. It can contain any number of node, and any node can enter or leave it at any time, but its location is known and fixed. All nodes inside a same logical router share a common routing table containing logical links. A logical link is a path to a distant logical router. This algorithm requires that every node knows its position and is able to have a good approximation of the position of the other nodes.

This topology is relative to the routers, where every logical router sees itself as the center of world and logical routers located farther away are aggregated to zones from its point of view. This zones grow larger as farther away they are. This is illustrated in figure 2.

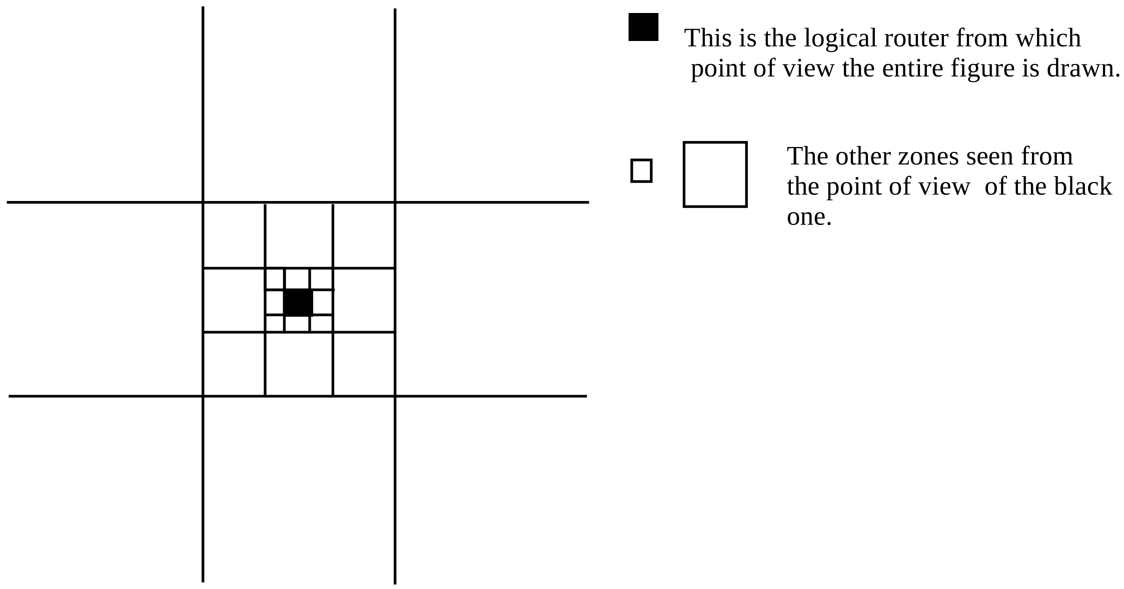

Fig 2: the zones from the point of view of a logical router.

The logical router, to be able to route its packets, needs a routing table. This table is 2D and associates to every logical link and every zone a grade . The lowest grade means that this link never transmitted any packet to this zone . The higher the grade the better the link to route packets to the zone. Those grades are used by the router as a probability to route a packet to the zone using this link.

The routing table is maintained using small packets called ants. Those packets always travel through the network while providing, to the nodes they cross, information on the links they take. The more ants a node receive from a certain zone using a certain link, the better is the link to this zone . This is inspired from the behavior of biological ants which leave a chemical substance called pheromone on their path when it leads toward food ,for instance. The more pheromone there is on a path, the better the path is. This aids the other ants to find a food (a pack of cookies would be an excellent approximation of a far zone. The colon of ants foraging from the

anthill - the logical router - would use a certain path - a logical link - to gather food).
When a packet is sent through a logical link, it is routed in a greedy way called StPF (Straight Packet Forwarding).

2:Routing tables

2.1: Specifications

My job was to determine a good way to store and share the routing information inside a logical router. The purpose wasn't to implement the actual data object of the table, but to develop a protocol of communication related to this table inside the router.

As seen above, the routing table consists in a row for every outgoing logical link and a column for every zone. The intersection of a row and a colon contains the grade of this link to this zone. This is the probability of routing a packet to this zone by this link.

The data of this table can be stored on one or several nodes. It should only be as much as possible consistent and up-to-date. It would also be safer to have redundancy, as a backup not to have to rebuild the whole table from scratch, if the only node with the information fails, move to another router, or is not easily accessible from any node inside the router.

I proposed three different approaches to solve this. The firsts two are obvious:

1) Every node stores the whole information. This is the simplest one. It doesn't need any messages to request a route, since every node knows it. To keep the tables up-to-date, any node receiving an ant has to flood the whole router with the new information. In terms of memory this is far from being optimal since every node needs to store the whole table. Nevertheless, I think that this is a marginal problem, since the actual communication devices usually have plenty of memory.

2) Only one or few central nodes store the information. This reduces the traffic due to flooding, since the new information only need to be forwarded to few nodes. Most of the nodes need to demand the route before transmitting, but this is much more effective than the standard reactive MANET protocols, since it doesn't imply any flooding, but only two messages: the question and the answer. It also needs a specific strategy to change the central node if it leaves the router or fails.

3) Using several nodes to split the table and only store a part of it in some routers. I think that this approach is not as efficient as it seems to be, as I will show later. The nodes need to know where the relevant informations are stored, which could be solved in a location based architecture. For example the node storing the data for one zone could be the closest to this zone. Since more nodes are important this implies more mechanism to change them if they happen to leave the router or fail. The traffic needed to get or give an information would be the same as 2).

2.2:Implementation

To test this I used the glomosim simulator. The implementation is done in PARSEC. I based my code on the AODV algorithm but only roughly.

I first implemented a simulation of the messages that would be needed for the first approach (the one where all nodes store the whole information). This requires a flooding process to be initiated by any node, to transmit any new information. This is done in the following way: the node initiating the process broadcasts the information to all its neighbors. When a node gets this message it broadcasts the message further if it is inside the router and never saw it. Otherwise the message is ignored. This allows every node of the router to get the new information, at least most of the time.

The second approach with only one node holding the whole information was much more complicated. To simplify I ignored the fact that a node could fail and implemented it with only one node holding the table. I will call this node the central node. This implementation is complex for two reasons:

- A node entering the router must be informed of the identity of the current central node. This is done so: when a node enters the router it broadcasts a message to all its neighbors. All of them answer with the identity of the central node and the last time this information was correct for sure. The entering node then chooses the one with the most recent correct information.
- When it leaves the router the central node must choose another node promoted to be the next central node and transfer the table to it. The node which is physically closest to the center of the router will be the next central node. This prevents it from leaving the router too soon. To decide which node is a good candidate, the actual central node broadcasts a message when it leaves the router. This message contains an information called distance, which represents the distance from a node to the center. This is initially set to a very high value. When a node inside the router receives such a message, if it is closer to the center than the distance in the message, it replaces it with its own distance and broadcasts it further. It also sends a copy of this message to the ex central node. Otherwise, the message is broadcasted as is. After a certain amount of time, the closest node to the center from which the central node has received the message will be the next central node. The exiting central node sends the table to the new one, and a message is flooded, containing the identity of the new central node.

I didn't have time to try and implement the third approach.

3: Result and analysis:

As stated above, I used the glomosim simulator to test my program. I used the random waypoint, for the simulation of the movements of the nodes.

The first approach has quite good results. Obviously flooding this way is a good method, and every node gets the information. It should be implemented with real messages to see if it's reasonable to send the whole table, or if the message gets too big. If only the new information is flooded then there should be a mechanism for an entering node to get the table. My implementation also ignores how often a ant reaches a node. This should be investigated to know the efficiency of this approach. The special situation of one or several nodes being inside the router but not in communication range of any other node from the router should also be investigated. In this case some nodes, that are not inside the router, should be used to transmit the data.

The second approach has some problems I didn't think of in the first time. There was actually a lot of inaccuracy in the router. Due to the fact that the central node leaves very often the router, and that the flooding process implies some delay, lots of nodes do not have an accurate information about the identity of the central node. The random waypoint moving was probably not a good enough moving choice, and much answers could be found using a better system as the restricted random waypoint [15]. A strategy that would take in account possible failure of nodes and thus being redundant should also be developed .

The third approach could be tested, but I think that this would not be very useful since the problems discovered in the second one would be even worse, as much more nodes are important and have to be replaced if they fail or leave the router. The only positive point would be that not so much traffic would be addressed to the same node, which could decrease the risk of congestion, but since the router is composed of only few nodes, I don't think this is relevant enough.

4: Bibliography

[1] Mobile Ad-hoc Network (MANET) Working Group. Http//www.ietf.org/html.charters/manet-charter.html

[2]Ko Y. and Vaidya N. Location-aided routing in mobile ad-hoc networks. Proceeding of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCOM'98), Dallas, Texas, USA, August 1998.

[3]S.Basagni, I.Chlamtac, V. Syrotiuk and B.Woodward. A Distance Routing Effect Algorithm for Mobility (DREAM). Proceedings of Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98), Dallas, Texas, USA, August 1998.

[4]Brad Karp and H.T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00), August2000.

[5]L. Blazevic, J.-Y. Le Boudec and S. Giordano. A Scalable Routing Method for Irregular Mobile Ad Hoc Networks. EPFL.

[6]J.-P. Hubaux, J.-Y. Le Boudec, S. Giordano, M. Hamdi. The Terminode Project: Towards Mobile Ad-Hoc WANs. EPFL-ICA.

[7]Perkins C.,Bhagwat, P.: Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. Proceedings of the SIGCOM '94 Conference on communications Architecture, protocols and Applications (Aug. 1994) 234-244.
[8]Murthy S., Garcia-Luna-Aceves, J.:An Efficient Routing Protocol for wireless Networks. ACM Mobile Networks and App. J., Special Issue on Routing in Mobile Communication Network (Oct. 1996) 183-197.
[9]Jacquet, P., Muhlethaler, P., Qayyum, A.: Optimized link state routing (OLSR) protocol. Internet Draft, draft-ietf-manet-olsr-06.txt (Jul. 2002).
[10]Chiang, C.-C., Wu H., Liu W., Gerla, M.: Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel Proceedings of IEEE SICON'97 (Apr 1997) 197-211.
[11]Perkins C., Royer, E.: Ad-Hoc On-Demand Distance Vector Routing. Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications (Feb. 1999)
[12]Johnson, D., Maltz, D.:Dynamic Source Routing in Ad-Hoc Wireless Networks. Mobile Computing Systems and Applications (December 1994) 158-163.
[13]Park, V., Corson, M.: A Highly Adaptativ Distributed Routing Algorithm for Mobile wireless Networks. Proceedings of INFOCOM'97 (Apr. 1997).
[14]Heissenbüttel, M., Braun T.: Ants-Based Routing in Large Scale Mobile Ad-Hoc Networks. IAM Bern Univercity, CH.
[15]Blazevic, L.: Scalable Routing Protocols with applications to mobility. Thesis 2517, EPFL, 2002. See section 5.5.1