MANIAC: Mobile Ad-Hoc Networks Interoperability and Cooperation: THE LIVE AND LET LIVE STRATEGY

Ivan KLIMEK, Vladimír SIDIMÁK

Dept. of Computers and Informatics, FEI TU of Košice, Slovak Republic

ivan.klimek@cnl.tuke.sk, vladimir.sidimak@cnl.tuke.sk

***Abstract*—Mobile Ad-Hoc networks are certainly the most flexible computer networks that currently exist. They can be built anytime, anywhere, using any wireless-enabled devices and provide end-to-end connectivity. The benefits of such a decentralized technology are quite clear. That is why there is a vast number of interesting implementations. But why are MANETs not used in everyday life then? This paper describes a solution that combines the strength and simplicity of existing implementations of the Optimized Link State Routing Protocol (OLSR) with a strategy originally presented and awarded with the Strategy Award at the MANIAC Challenge 2007 (Mobile Ad Hoc Networks Interoperability and Cooperation) held in conjunction with IEEE Globecom 2007. This strategy aims to solve the two most important question that hinder MANETs from real-life use, first the minimization of traffic on nodes with limited battery capacity (e.g. mobile phones) and second the detection and isolation of uncooperative nodes while keeping the algorithm as simple and fast as possible so it could be run practically on any device without large overhead. Simplicity was the key factor during the whole developing process that should guarantee easy portability and maximal compatibility.**

***Keywords*—MANET, OLSR, wireless, network**

I. INTRODUCTION

Mobile wireless devices provide a comfortable way of communication for their ease of use and mobility. As a price for this comfort, there is a limiting factor, the battery life. Wireless transfers of information on these devices usually result in huge energy consumption, thus rapidly decrease the time that devices can be used without recharging. Any group of wireless enabled devices should be able to create a MANET, in the most cases a full-mesh topology is used, which is too demanding as for already mentioned power consumption concerns. Presented approach minimizes that by reducing the number of active connections between nodes by a "minimal active neighbor topology calculation" algorithm. Another problem that makes the use of a MANET less attractive, are uncooperative nodes. Those are the nodes that try to use the network for their own data transfers but do not forward traffic destined to, or originating from other nodes. A neighbor forwarding traffic sensing mechanism is applied to check if the neighbors are really forwarding the data as they should be. The strategy is called Live and Let Live.

II. THE LET AND LIVE STRATEGY

A. Minimal active neighbor topology calculation

The general idea is that no node in any network really wants to forward the data of other nodes at its own expenses. If every element would treat others this way, the network would not work. On the other hand, every node needs to communicate. That is why it became part of the network. As MANETs are decentralized, in order to keep the network working, some nodes have to forward foreign data too. Our minimal active neighbor topology algorithm reduces the number of communication links between the nodes while keeping them all connected and being able to communicate with each other. The first step is to find the best next-hop for forwarding the local node's data (e.g. best next hop to the gateway). Then the direct connectivity to all other one-hop neighbors is temporarily terminated. After the changes in topology propagate, the local node checks if its one-hop neighbors are reachable thru the only enabled node, the best next hop. If not, one of the original neighbors is randomly chosen and enabled. The check for accessibility of other nodes is performed again. If they have any other way to access the network except of the local node, the local node will be able to hear of them thru one of the enabled nodes in their routing updates. This algorithm continues until all of the original nodes are accessible again. In each iteration the nodes that become reachable as a result of the actions during that iteration are deleted from the list of nodes that need to be checked. Algorithm is rerun whenever a topology change is detected. Topology changes are propagated extremely fast thanks to the OLSR MPR concept. The minimal neighbor topology calculation is only applicable to battery powered mobile devices. Other nodes with no power saving concern act as "center of topology", ideally a next-hop node for several mobile devices. Such approach creates a topology in which every battery powered device minimizes its number of connections and traffic load, but also acts as an entry point for all other nodes that have no other means of

connecting to the network, keeping the network operational.

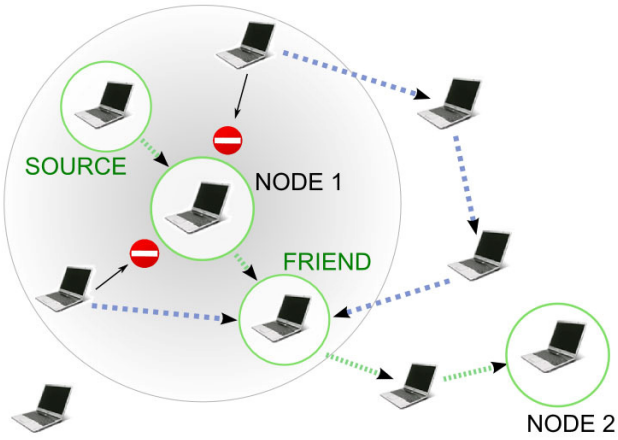

Fig. 1. An example MANET topology. Node 1 has chosen the next hop (FRIEND) for delivering the traffic to its destination (Node 2). Two other nodes that are in the range are blocked as there is an alternate route to the network for their traffic. All nodes keep connectivity to each other.

B. *Unicasting Multicasts and ARP filtering*

To minimize the number of connections, the combination of ARP filtering and unicasting of the originally multicast OLSR update messages is used, so that the nodes are visible only to the nodes they want to be visible to and want to create links with them. By using this approach a logical topology inside the physical topology is created. Despite the fact that the nodes have mutual Layer 1 connectivity (e.g. they are in range of each other wireless adapter) they won't know of each other as of a direct neighbor unless the Minimal active neighbor topology calculation doesn't indicate otherwise.

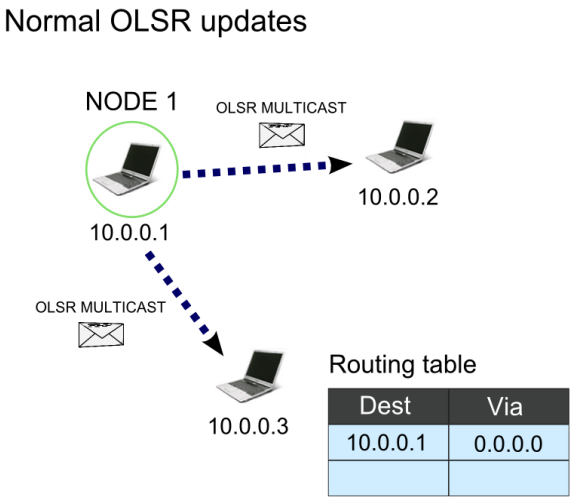

Fig. 2. OLSR updates as they are normally propagated to all nodes in range.

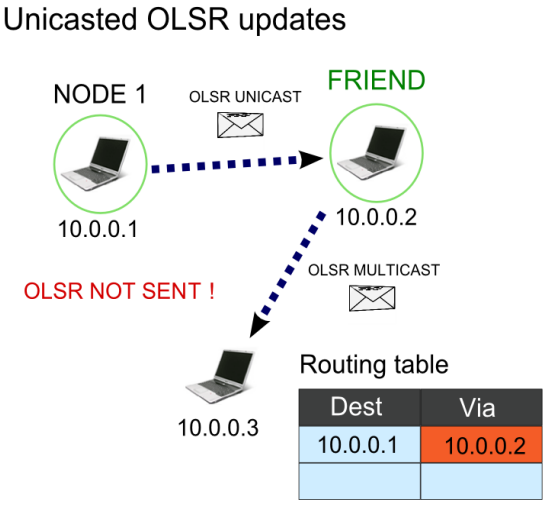

Fig. 3. OLSR updates unicast to chosen locations. Excluded nodes do not recognize NODE 1 as a neighbor node, even though it is in their range.

C. *Neighbor forwarding traffic sensing mechanism*

There are cases when technology is not used exactly like how it was supposed to be used. In an MANET network, there might be nodes unwilling to participate on common interest, which is reliable network with end-to-end connectivity for all elements. These uncooperative nodes would accept the traffic destined to them, but not forward and drop other traffic in order to conserve their battery. To prevent these rogue nodes from exploiting the network, each node (node A) checks its neighbor whose data it is going to forward (node B) by simulating another node (node C). That node then tries to connect to node whose data the local node is forwarding /A is simulating node C that wants to connect to node B/. Because node A is the only possibility for node B to send its data to the network, node A has to be able to hear node C fake data. This can be done by generating fake traffic (node C traffic) with different source MAC and IP address than the original interface. To make this check mechanism even more realistic and harder to detect, the interface can be temporarily configured with different L1 parameters as for example signal strength. This would effectively avoid possible recognition that node A and node C are the same nodes. If node B fails in any stage of this check, node A simply does not forward node's B traffic because there is a reason to believe node B is a rogue node. Another possible scenario is that the attacker would become the OLSR MPR and would drop other nodes traffic. To avoid that, it is needed that each node monitors its neighbors and finds out how cooperative they are by seeing its own traffic being forwarded by the neighbor by listening for packets with the source MAC address of the forwarding next-hop, and the original source and destination IP addresses and/or other packet identifiers as packet length etc. At this stage of algorithm, the topology is already minimized and most of the traffic is forwarded through the one chosen next-hop node. That fact makes the monitoring process much easier. Uncooperative nodes should be blocked and not included in further topology calculations.

III. Real Life Example

The following figure shows an example of real implementation of the strategy - limitation of the unnecessary traffic and keeping all nodes connected. PC1 and PC2 represent devices with no battery concerns, while NODEs 1 – 3 are mobile battery driven devices. NODE 2 does not need to establish the connection to NODE 1 as it can connect to the network via PC1, which is not limited by battery. On the other hand, NODE3 is fully dependant on NODE 2 and that is why NODE 2 provides the connection.

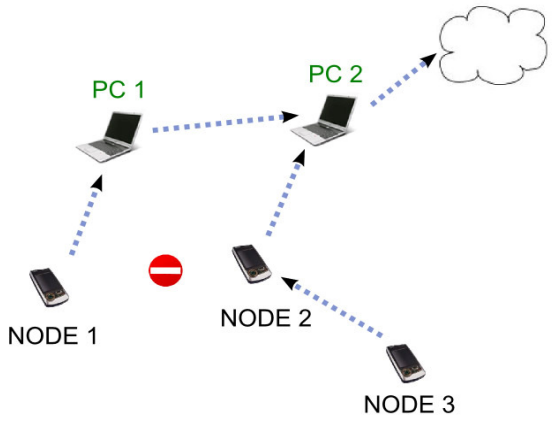

Fig. 4. Real life example for the Live and let live strategy

IV. Conclusion

The presented solution is a simple MANET implementation based on the popular Optimized Link State Routing Protocol which is enhanced in a way that it can provide a fair-use environment for the network users without rogue nodes and optimizes the topology so that the traffic directed thru battery driven nodes is minimized. Parts of this solution are built using the MANIAC Challenge 2007 API, and are still under development. First practical tests with promising results were done during the MANIAC Challenge 2007, which finally led to the winning of the Strategy Award. Simulations and practical measurements are planned for the near future.

References

[1] IETF, "OLSR Routing Protocol (RFC3626)," 2005, [Online; accessed 19-March-2008], [Online], Available: http://www.ietf.org/rfc/rfc3626.txt

[2] Wikipedia, "Optimized Link State Routing protocol," 2008, [Online; accessed 28-March-2008], [Online], Available: http://en.wikipedia.org/wiki/OLSR

[3] MANIAC Challenge, "MANIAC Challenge API," 2007, [Online; accessed 15-March-2008], [Online], Available: http://www.maniacchallenge.org/maniac_release01.tar