2014-09-26

A practical evaluation of smartphone application on mesh networks

International Conference on Wireless Networks, 3., 2014, Bali.
http://www.producao.usp.br/handle/BDPI/48558

SESSION

SENSOR NETWORKS AND APPLICATIONS + RELATED ISSUES

Chair(s)

TBA

A Practical Evaluation of Smartphone Application on Mesh Networks

Dalton M. Tavares[1], **Anelisa P. da Silva**[1], **Stella J. Bachega**[2],
Rafael V. Aroca[3], **Jó Ueyama**[4], **Glauco A. P. Caurin**[5],
and Antônio Carlos de Oliveira Jr[1],
[1]Computer Science Department, Federal University of Goiás, Catalão, Goiás, Brazil
[2]Production Engineering Department / Federal University of Goiás, Catalão, Goiás, Brazil
[3]Mechanical Engineering Department / Federal University of São Carlos, São Carlos, São Paulo, Brazil
[4]Computer and Mathematical Science Institute / University of São Paulo, São Carlos, São Paulo, Brazil
[5]Mechanical Engineering Department / University of São Paulo, São Carlos, São Paulo, Brazil

Abstract— *This paper presents a mesh architecture proposal called Mobile mEsh Network to Aid in CountEring drug TRAffiCKing (M.E.N.A.C.E-TRACK). This project was born from the hypothesis we could establish a covert network channel independent of the cell phone companies infrastructures. Therefore, law enforcement agencies could establish connection with field personnel, in a fault tolerant fashion allowing the transmission of multimedia data (instead of only voice). The main contribution for this paper is the strategies involved to configure smartphones on the MANET side of this system. We present the main difficulties and one possible solution to implement ad hoc mode on our testbed so we can enable a MANET organization on M.E.N.A.C.E-TRACK.*

Keywords: OLSR, Android, Mesh networks

1. Introduction

The research project Mobile mEsh Network to Aid in CountEring drug TRAffiCKing (M.E.N.A.C.E-TRACK) proposes the creation of a dynamic mesh network, intended to interconnect field personnel (e.g. in vehicles or on foot) to a base of operations (e.g. a police station) whenever possible. M.E.N.A.C.E-TRACK is organized as a Wireless Mesh Network (WMN), with a particular multi hop ad hoc network consisting of a mesh backbone and mesh clients. The stationary wireless mesh routers (MR) interconnects through single or multi hop wireless links forming the backbone (i.e. the police station and MRs deployed strategically throughout the city). The MRs should have wired connections and act as the Internet gateway (IGW) to exchange the traffic between the Internet and the WMN clients.

The mobile devices can connect to any MRs in reach to access the Internet via the IGW in a multi hop or through any mesh client in reach forming a route to the IGW. In this sense, the behavior of the mobile devices is that of a Mobile Ad hoc Network (MANET), where routes are dynamically established allowing devices to be grouped without any predefined infrastructure. This should account for the field personnel [1]. The scope of this article will be on the MANET devices, specifically, smartphones.

In this scenario, connectivity is a matter so important that governments acknowledge the ability to "be connected" (possibly with Internet access) as a commodity as important as food, shelter or healthcare; not literally as an indispensable element to sustain life, but as a means of offering quality of life to citizens [2]. Therefore, the fundamental research question is: can smartphones perform a more decisive part on the connectivity infrastructure? Considering the number of smartphones in use, we formulate as a hypothesis that it could be used as part of a greater network infrastructure, as an active traffic router, in order to extend the reach of traditional network technologies (e.g. a cellphone tower, an access point etc). The number of devices responsible for connectivity may be an improvement considering the increase in routing diversity and fault tolerance. Therefore, we evaluate the use of smartphones, as potential MANET nodes.

The contributions of this paper relies on the proposal for a method that could be made generic enough for the practical application of MANET routing protocols using smartphones. We also describe the details of our testbed implementation and experience gained during deployment. The rest of this paper covers M.E.N.A.C.E-TRACK application scenario (section 2), related work (section 3), the details of our testbed implementation (section 4) and the conclusion and future work (section 5).

2. Understanding the Scenario

Project M.E.N.A.C.E-TRACK is intended to create a dynamic mesh network, inherently fault tolerant, so data could be transmitted from law enforcement agencies to field agents. Depending on the density of MRs in a city, this transmission could be made with minimal delay, or considering the lack of connected landlines, with variable delay depending on the number of MANET devices used to achieve a mesh node gateway. This is a particularly interesting alternative considering the lack of wireless network coverage quality in Brazil.

According to the Brazilian Network Information Center (NIC.br), none of the Brazilian 3G operators passed quality assurance tests (e.g. TCP/UDP throughput, jitter, latency, packet loss etc) conducted in December 2012 [3]. Preliminary results shows most operators had a packet loss rate per connection 2 percent above the limit when the quality of the Internet connection starts getting compromised. In terms of latency, all major operators in Brazil had values above 200 milliseconds (considered a standard) [4]. Considering the major players in the 3G market today, we have 64.6% of the cities covered, which represents 182,471,019 users and 90.8% of the population [5].

Although the cellphone network could be an alternative to M.E.N.A.C.E-TRACK, this network infrastructure may not survive a devastating earthquake or another natural disaster. Even if the infrastructure stays in place, there can be added overload due to extraordinary circumstances that may render the cellular network useless (e.g. calls from refugees and their families, international aid workers who arrived in the aftermath caused by some unfortunate event etc). Josh "m0nk" Thomas and Jeff "stoker" Robble, both working at Mitre, saw this problem and created a working prototype backup network using only the Wi-Fi chips on Android smartphones [6]. This would be a good staging area for testing MANETs using the approach proposed by Thomas and Robble.

The choice to use mesh networks instead of the traditional cellular or wireless local area networks (WLAN), for the purposes of the proposed system, is based on the following facts [7]: there is no main node, therefore we achieve a certain degree of redundancy innate to the system; it is possible to reach any other node by traversing a number of intermediate nodes, which favors interconnectivity among nodes and a bigger range in some cases; all nodes are equal so there is no centralized control. Therefore, each node participate in networking and as a source or sink of traffic as well; rather than a single hop to a base, multi-hopping/relaying amongst nodes must be a common capability enabling the creation of a new network or the expansion of an existing one. This allows to cope with distance and obstacles by hopping around obstructions; ability to work without infrastructure (e.g. a base station).

3. The Role of Ad hoc Networks on M.E.N.A.C.E-TRACK

The term ad hoc comes from latin and means "for a particular purpose only". Therefore, an ad hoc network represents a network with its purpose defined for a temporary time frame, such that some network devices can be a part of the network topology only while they are in range or during the communication.

Some applications of mobile computing do not depend on a pre-existing infrastructure and can utilize a MANET. A MANET by definition is a wireless network that does not need a rigid infrastructure and its topology is self-configuring for connected mobile devices [8]. Because of self-configuring and self-organizing characteristics, MANETs can be deployed quickly. There is no infrastructure defined in the network, therefore all of the participating nodes relay packets for other nodes and perform routing if necessary. Because of the limitations in wireless transmission range, communication links could be multi-hop [9]. Although routing protocols are the most important element of a MANET, our discussion will be limited to the establishment of a flexible infrastructure so we can test a broad range of protocols and their features in mobile devices.

3.1 Related Work

For quite some time, universities and research centers have developed and widely used mesh networks as access networks for their users. In this section we present related works that is not new but very similar in nature to our proposal. Some examples that inspired our research are projects RoofNet [10] at the Massachusetts Institute of Technology (MIT), Vmesh [11], [12] at University of Thessaly, Greece, MeshNet [13] at University of California, Santa Barbara (UCSB) etc.

The RoofNet project refers to the deployment of mesh nodes deployed over an area of about four square kilometers in Cambridge, Massachusetts, using volunteer users. These volunteers installed a Roofnet kit at home (PC, an 802.11b card, and a roof-mounted omni-directional antenna) and shared (a fraction) of their DSL lines [14].

VMesh is a low cost and inherent flexible deployment in terms of building a prototype wireless router using an embedded Linux. This was one of the first mesh network projects to use OpenWrt. It also adapted the network configuration for the mesh setup so it would take little to no human intervention. This in turn, was exploited to support the dynamic addition, removal and mobility of network elements. Its network architecture model is also very similar to the one proposed for M.E.N.A.C.E-TRACK and therefore is considered the very inspiration for our proposal.

MeshNet is a 30-node wireless mesh testbed implemented at UCSB. The authors present their experience in designing, deploying and using their mesh network. They also present UCSB MeshNet architecture and discussed the challenges regarding management, nonintrusive and distributed monitoring, and node status visualization. Their implementation were also based on OpenWrt for the mesh nodes [13].

3.2 Routing Protocols for Mesh Networks

Routing can be roughly divided in topology based, location based or energy aware. Topology based has traditionally used the knowledge of instantaneous connectivity of the network with emphasis on the state of network links. Location based uses information related to the physical position of nodes. Energy aware routing, uses information regarding the remaining battery in mobile devices in order to produce paths that comprise nodes with high value of remaining lifetime, as well as to help them adjust their transmission power so that each node keeps the energy required to accomplish the routing task at the minimum [15].

The discussion will be limited to topology based protocols. In this category, the associated routing protocols can be classified as proactive, reactive and hybrid. In proactive protocols all nodes calculate all possible paths to all destinations independently of their effective use such that when a packet needs to be forwarded, the route is already known, eliminating routing delays. The main drawback is the periodic broadcasts sent taking some time to converge (i.e. to create the routing table at each node). In reactive protocols, the network is evaluated if needed. Routes are created only if there is the need to carry data traffic. This protocol exempts the creation of a routing table, scaling well for large populations. Hybrid routing mixes the features of proactive and reactive protocols and is used when there is a set of circumstances where neither protocol perform well [15]. As our testbed is very simple, we chose to use a proactive routing protocol as it works efficiently for a small scale mesh network with high mobility [16].

3.2.1 Optimized Link State Routing Protocol (OLSR)

According to [17], OLSR is a routing protocol for proactive ad hoc networks, developed by Institut National de Recherche en Informatique et en Automatique (INRIA) and standardized by the Internet Engineering Task Force (IETF) in RFC 3626 [18] as an experimental protocol. Its goal is to calculate and maintain routes for every node in a wireless network using a mesh topology. OLSR is able to do that by executing in each node a process that keeps track of network paths for every other node. Therefore, each node fills a routing table that indicates how he can reach every other node and so the algorithm converges.

Each node regularly exchanges information with each other, updating every routing table detecting the insertion and removal of mesh nodes. Usually, in an ad hoc networks, as a node receives the routing update information, he sends a broadcast message, retransmitting information to every neighbor (a.k.a. flooding). This flood of routing information is performed many times, meaning a node can receive the same packet time and time again unnecessarily, generating an undesired overhead in the network.

A powerful and efficient policy to control these many broadcast messages that are flooded across the network is a mechanism called multipoint relay (MPR). This technique is the main difference between OLSR and other proactive protocols. MPR is an optimization that regards the election of some nodes that are able to broadcast update messages. Therefore, a controlled flooding is achieved avoiding the replication of update messages in the network.

OLSR presents some clear advantages when used in the M.E.N.A.C.E-TRACK MANET environment. These concern its nature as intended to be used in high density networks, greatly due to the MPR approach. The bigger and more cluttered the network is, better the route optimization provided. OLSR was developed to work in a completely distributed fashion, avoiding any dependency with a central entity. It also does not need to transmit reliable control messages using TCP. All the communication is done using UDP port 698 for the transmission of periodical unreliable messages. The loss of some messages is of no consequence to its operation. OLSR messages do not have to be delivered in sequence, since each message contains a sequence number. Therefore, the destination can control the sequence of the messages delivered, and in case of loss, request the retransmission of the missing part. It also supports IPv4 and IPv6 [18].

OLSR does not change the TCP/IP protocol suite in any way. It only interacts with layer 2 management tables. OLSR networks support IP addressing to identify each node. The use of multiple interfaces is also supported, although a preferred IP must be chosen for routing. Each node in an ad hoc OLSR network has a direct and bidirectional (i.e. symmetric) relationship. The uncertainties about the propagation of the radio signal may cause some communications to be restricted to unidirectional links. Nevertheless, each communication must be verified in both directions so that one link can be considered valid. To accomplish that, each node periodically sends a HELLO message that contains the information of neighbors (link sensing, neighborhood detection and MPR selection signalling), and are transmitted in broadcast mode [18].

A HELLO message contains a list of neighbor addresses which have a valid bidirectional connection and a list of the neighbor addresses that are listened by this node, but whose link is still not valid as bidirectional. If a node has its own address in a HELLO message, the link is considered bidirectional for the sender node. The HELLO messages transmitted by a node are received by each of its neighbors with a distance of one hop, but they are not retransmitted by them.

The OLSR protocol is considered an optimization of the link state protocol adapted to MANETs because it reduces the size of control messages. Instead of stating all the links, he states only a subset of the neighbors' relationships. As a consequence, OLSR minimizes the flooding, controlling the traffic using only selected MPRs to broadcast HELLO messages from the second hop onwards. Only the MPRs related to a given node retransmit its broadcast messages. Therefore, MPRs minimize the overhead of HELLO messages which otherwise would be coming from all the active nodes in a mesh network, avoiding the broadcast of redundant information. To select MPRs, each node in the mesh network selects a set of symmetric nodes a hop away. The premise is a node must reach every second order node using the fewer MPRs possible, allowing a source node to reach any other node at a distance of two hops.

Neighboring nodes to a given S, that are not MPRs, receive the broadcast message but do not relay it. Each node chooses a neighbor to be its MPR considering this is a symmetric (i.e. bidirectional) hop. This selection is performed so the coverage of the radio link of all symmetric nodes are at 2 hops distance. S is also known as a selector node given it is choosing its MPRs. Each node elected to be MPR to S (MPR(S)) keep information about the set of neighboring nodes belonging to MPR(S). The node's set of MPR nodes chosen by S is known as Multipoint Relay Selector Set (MPRSS). A node discovers the MPRSS from periodic information received from its neighbors. A broadcast message destined to be sent in the network, from any MPR(S) is assumed to be relayed back to S, in case S has still not received the message. The set can change over time (i.e. when a selector node pick another MPR). This is indicated by the HELLO message sent from the selector node. The premise here is that the node will only retransmit an OLSR packet if it is chosen as MPR by the last node that retransmitted the message and if the packet TTL is major than zero.

Each OLSR node keeps the information about the network topology. This information is acquired from TC messages and used to calculate the routing tables. A node keeps a routing table that allows finding a path to other network nodes. This routing table is created from the local link information base. The local link information base stores the information about the paths to neighboring nodes. If any of these paths is modified, the routing table is recalculated to update the routing information about any node destiny in the network. The routing entries are defined in [18] as presented in Listing 1.

Listing 1: Routing table format ([18] pg 46).

```
1.  R_dest_addr    R_next_addr    R_dist
    R_iface_addr
2.  R_dest_addr    R_next_addr    R_dist
    R_iface_addr
3.      ,,             ,,            ,,          ,,
...
```

Each entry in the table consists in R_dest_addr, R_next_addr, R_dist, e R_iface_addr, where R_dest_addr distance is estimated in R_dist hops from the local node, its symmetric neighboring node is R_next_addr, which is the next hop in the route to R_dest_addr and its local interface has the address R_iface_addr. These entries are recorded in the routing table for each network destiny for which a route is known. For every destiny, when a route is broken or just partially known, this entry is not registered in the table.

MANETs are usually isolated, however, there are situations where there is the need to access other networks. The HNA is the solution presented to this situation. It works as a host in the mesh network identifying itself as a gateway to other network and it can present its services by means of Host and Network Association (HNA) messages. When a node receives HNA messages from other node, it adds the transmitter as a gateway to other network. The address to this other network is obtained reading the fields `Network Address` and `Netmask`. In general, if the transmitter is an Internet gateway the field `Network Address` and `Netmask` will both have the value 0.0.0.0.

4. Case Study: M.E.N.A.C.E-TRACK – MANET side

The research project M.E.N.A.C.E-TRACK proposes the creation of a dynamic mesh network, intended to interconnect field personnel to a base of operations whenever possible. This type of network accepts the dynamic disconnection and reconnection in case a node or group of nodes leaves or returns to the main base. Some important features for M.E.N.A.C.E-TRACK when considering field personal are high bandwidth (if available), end to end communication with the MRs, all mobile network nodes are also traffic routers forwarding packets until they reach the destiny using some kind of topology control for the deliberate adjustment of certain system parameters (e.g.antenna direction, transmission power, routing protocols etc) to form a particular and more adequate network topology, end to end IP support, data transfer, audio and video streaming, geographic positioning with or without the use of GPS (depending on the accuracy needed) and support to mobility and scalability [19], [20].

The steps to build the M.E.N.A.C.E-TRACK infrastructure consist in defining the devices that can be used as nodes, defining the operating system for such devices and defining the best routing protocol(s) to provide routing adjustments considering mobile nodes with varying ranges. This paper will focus mainly in the first and second steps. We chose a group of mobile devices for testing (notebooks and smartphones) and the most suitable operating system for each one. OLSR will be used as the routing protocol 1 in every device.

4.1 Testbed Preparation

We used as our MANET testbed two Macbook Air 11" and two smartphones Galaxy S3 GT-I9300. The notebooks used OS X 10.7.5 and the smartphones, a modified version of Android OS (version 4.2.2). All of the were configured to behave like mesh clients. One problem that immediately arose when configuring the devices as mesh clients was that not all of then could work in ad hoc mode. The notebooks did not present any problem to be configured in ad hoc mode (just a matter of using the OS X GUI and configure a new

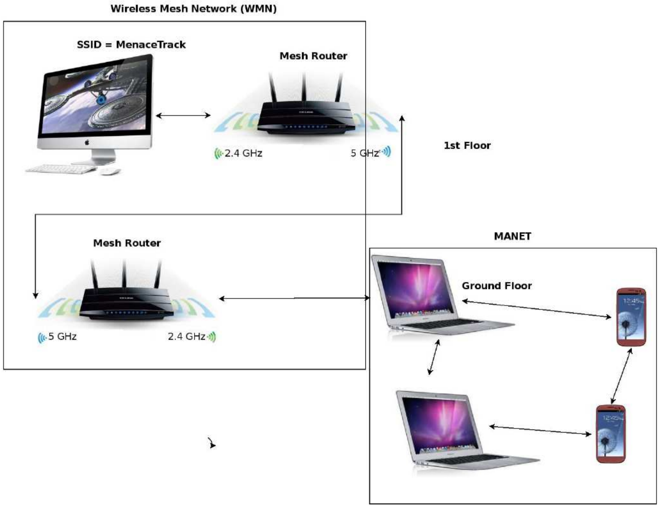

Fig. 1: Proposed testbed for the case study (MANET side).

network). For the smartphones (Galaxy S3 GT-I9300) it was not so easy.

First of all, Android OS does not support ad hoc mode for its mainstream version (i.e. factory release). The need for Ad hoc or Independent Basic Service Set (IBSS) support is not something new and the users and developers community is asking for a solution for more than five years [21]. Users usually understand by “ad hoc” mode the ability to share 3G via W-Fi (i.e. to create a hotspot) [22]. As presented in section 3, the definition of ad hoc is much more broad and complex than this understanding of ad hoc. This capability won’t be shipped on new smartphones anytime soon, but it is a really interesting trend and for some an Android project to which they might contribute [6].

In order to create true ad hoc mode in the smartphones, one will have to completely reinstall the operating system. This procedure includes rooting the device and choosing a version of Android that supports ad hoc mode. As of now, this modified versions of Android are ver scarce (e.g. Galaxy Nexus, Nexus 7, Nexus S, Samsung Epic 4G). Although, Thinktube [23] did not have an image for the Galaxy S3 GT-I9300, the documentation available in the site and the personal answers provided by Mr. Bruno Randolf helped a lot understanding the complexities involved in enabling ad hoc mode in Android smartphones.

Basically, Mr. Randolf and the Thinktube fellows [23] created patches to bring the missing Ad-Hoc (IBSS) mode to Android for the aforementioned devices, in a way that is fully integrated into the Android system API and user-interface. Therefore, it is possible to create and connect to Ad-Hoc networks from the standard user interface (Settings - WiFi) and Applications have an API to configure their "own" ad hoc network. Although this is a commendable effort, it opens some question regarding some official ad hoc support for Android OS provided by Google and even, a timeframe for this support to be available. Otherwise, we are in a situation were one would have to adapt the OS for particular devices in order to gain the possibility to use this feature.

The problem is not the hardware support, but simply a matter of software limitations introduced in the Android OS in order to block the use of the ad hoc mode. In more detail, in order to enable the ad hoc support, one would have to modify the WLAN driver “`bcmdhd`”, extend the Android framework, the public Android API, and add the missing additional hooks to the “Settings” application. It is really a shame the other great mainstream version of Android OS (codename cyanogenmod [24]) also does not support ad hoc mode.

Fortunately while researching alternatives for ad hoc mode on the project smartphones we stumbled on the Smart Phone Ad-Hoc Networks (SPAN) project [25]. This project reconfigures the onboard Wi-Fi chip of a smartphone to act as a Wi-Fi router with other nearby similarly configured smartphones, creating an ad-hoc mesh network. These smartphones can then communicate with one another without an operational carrier network. A modified version of Android OS was created to root specific models of Android smartphones in order to expose and harness the ad-hoc routing features of the onboard Wi-Fi chip [6]. Bottom line, the modified version of Android puts the network interface in ad hoc mode with the SSID configured as AndroidAdhoc. A minor discomfort of this modified Android OS is its inability to connect to infrastructure networks (usual APs). Therefore, we cannot download directly any applications.

So, the approach we used to put the wireless interface in ad hoc mode using this modified version of Android was to install the MANET Manager app [26] in OTHER smartphone (with an official Android OS version), install Apk Extractor [27] and extract the package file (apk) for MANET Manager, use a USB cable and copy the MANET Manager app to a computer and then, to the target smartphones. Later, we installed the MANET Manager in the target smartphone and start MANET Manager and turn the app on. At this point the MANET Manager app put the wireless interface in ad hoc mode. Now, all that’s left is to create a method to use the smartphone as a mesh client node. In personal conversations with Mr. Jeff “stoker” Robble and Mr. Bruno Randolf it became clear that the best way to put this devices to use as mesh clients would be to explore the structure of Android OS in-depth. The problem is, the authors are not that much well versed in Android OS modification and programming. Therefore, we came with an alternative approach that would allow us to use the smartphones as intended without the need to modify directly the Android OS.

We chose to install a Linux on top of the Android OS. To do so we downloaded an Ubuntu ARM compliant version of Linux (there are other Linux flavors) [28]. After downloading the image file, we unpack and put it on the smartphones (via USB cable). It is recommended to rename the file to `ubuntu.img` and put it in `/sdcard/ubuntu`. After you need a shell script to load the Linux OS [29]. This file have to be put on `/sdcard/ubuntu`. Using a terminal emulator on the smartphone in super user mode will grant `root` access. After, one must issue `sh /sdcard/ubuntu/ubuntu.sh`. This will trigger the Ubuntu OS, mounting the ubuntu.img on /data/local/mnt and chrooting it to /.

We installed Linux on Android so we could have all the flexibility of Linux in Android. Therefore, we need to install some useful tools like `iwconfig`, `ping` and `olsrd`. As we saw earlier, we cannot do that from the target smartphones because they do not have support to infrastructured networks. The best way to accomplish this would be to create an ad hoc gateway or simply install all the packages needed on another standard (rooted) smartphone and copying the directory `/var/cache/apt`. As we already had another smartphone with cyanogenmod installed (from earlier case studies) we installed all the needed packages in it and them, generated a `tar.gz` file and copied it to the target smartphones. There we just issued the appropriate `apt-get` commands and installed all the needed tools (e.g. `apt-get install olsrd`). Although this method is not the best due to the overhead caused by a second operating system running on top of Android OS, this approach was chosen to simplify the case study for mesh networks.

4.2 olrsrd configuration

There is one OLSR implementation that is becoming the standard and most widely used known as OLSRd (old Unik-OLSR). One key advantage observed in OLSRd implementation for Linux is its support for IPv6 (this feature won't be used for this case study). OLSRd is an implementation based on the INRA C code, but has been almost completely rewritten (it's almost GPL). OLSRd also is under continuous development [30].

OLSRd fully complies to the RFC 3626, including support for plugins and an optional GUI. The implementation also has a informative up-to-date web-page with links to mailing lists and papers [31]. The experimental network topology used was as follows: Smartphone A ↔ Smartphone B ↔ Notebook A ↔ Notebook B. One very particular feature of this testbed is the 20 cm reach of the smartphones. This is pretty bad when comparing to a 10 to 15 m of the notebooks but it allows for testing the mesh reconfiguration. For instance, if we move Smartphone B to the reach of Notebook B, the routing to reach Smartphone B would have to be done through Smartphone B.

This first configuration of the file `/sw/etc/olrsrd.conf` (we used `fink` to install it on OS X) was kept pretty much default. The only changes we made were:

- to set `DebugLevel = 2`, kept `IpVersion = 4`;
- enabled `LoadPlugin` "`olsrd_httpinfo.so.0.1`" and put it on port 8080 and available to the localhost and the network range we were using (192.168.0.0/24). Therefore, we can access any of the network nodes from each other;
- for this test we didn't configure any device as an Internet gateway (therefore, no need to set Hna);
- we set each interface appropriately (e.g. Interface "`en1`").

Considering this parameters for each and every mesh node, and the suggested topology, we just have to configure every node (except the smartphones) to connect to the ad hoc network with SSID set to AndroidAdhoc (set by MANET Manager). All the IPs at this point are configured manually but we are working on a DHCP setup. Therefore, with all the IPs in the range 192.168.0.0/24 (Smartphone A = 192.168.0.100, Smartphone B = 192.168.0.101, Notebook A = 192.168.0.10 and Notebook B = 192.168.0.20) we triggered `olsrd` in all of the nodes.

With the debug level set to 2, we can follow everything that is happening on every node. For each node, the routing table is created and we can ascertain that by issuing `netstat -r`. We can also view the nodes that are accessible by one hop count and two hop counts. At first, Notebook A has nodes 192.168.100, 192.168.101 and 192.168.0.20 as nodes reached by one hop count. Due to the terrible reach of the smartphones, when we get Smartphone B and take it next to Notebook B, the cost to reach Smartphone B from Notebook A turns immediately to INFINITE. Therefore, it is removed from the nodes reached by one hop count. It takes sometime for it to be "reintegrated" to the mesh. After about 3 minutes, the routing tables are slowly updated and we can reach Smartphone B again. This testbed was not completely stable regarding some failures of the smartphones and the slow reintegration when we move them out of reach and into the reach of another node.

5. Final Remarks

This paper presented the bare bones of project M.E.N.A.C.E-TRACK and its intended application context. We also have shown the complexities regarding the MANET side of this project. Although the initial idea of using smartphones as a covert channel for a MAN size mesh network, the lack for ad hoc mode support is worrisome. The demand for this feature exists for at least five years and the main market players haven't issued an answer to date. What we have is an stoic effort from the open source community with individuals going to the extent of modifying the kernel of some smartphone OSs so the community can tinker with the ad hoc mode. Although this is not the best case scenario, as far as research goes, we can still test mesh routing protocols on the proposed testbed. In this paper, we achieved a configuration for the smartphones of the project that will allow us flexibility to configure any Linux supported mesh routing protocol and see its effects on a controlled test environment. The ridiculously short reach of the smartphones is quite interesting considering tests involving the change in topology of an established mesh network. Therefore, the answer for the proposed research question is NO, considering the current state of smartphones, they could not play a key part in M.E.N.A.C.E-TRACK.For our future works, we intend to explore OLSR in depth and test various configuration parameters so we can determine its effectiveness in the aforementioned testbed. We also need to integrate the MANET side of M.E.N.A.C.E-TRACK with the WMN side (i.e. with the OpenWrt APs).

Acknowledgment

The authors would like to thank the Research Assistance Foundation of the State of Goiás (FAPEG – Fundação de Amparo a Pesquisa do Estado de Goiás) for the full sponsorship of this research – edict number 006/2012.

References

[1] J. Wang, B. Xie, and D. P. Agrawal, *Guide to Wireless Mesh Networks*. London UK: Springer, 2009, ch. Chapter 1 - Journey from Mobile Ad Hoc Networks to Wireles Mesh Networks, p. 28.

[2] R. Woods, "Connectivity as commodity," *Latin America*, pp. 36–37, 2010.

[3] F. Tamusiunas. Aferição da qualidade pelo usuário e defesa de seus direitos (evaluation of quality by the user and defense of their own rights). [Online]. Available: http://az545403.vo.msecnd.net/uploads/2013/08/Fabricio-Tamusiunas-.pdf

[4] (2013, January) Nic.br divulga diagnóstico da internet brasileira durante cp 2013 (nic.br discloses diagnosis of brazilian internet for cp 2013). [Online]. Available: http://www.ebc.com.br/tecnologia/2013/01/nicbr-divulga-diagnostico-da-internet-brasileira-durante-cp-2013

[5] Teleco, "3g: 3rd cellular generation in brazil (3g coverage)," April 2014. [Online]. Available: http://www.teleco.com.br/en/en_3g_cobertura.asp

[6] S. M. Patterson, "Previous article next article android phones are connecting without carrier networks," February 2013. [Online]. Available: http://www.networkworld.com/community/blog/android-phones-are-connecting-without-carrier-networks

[7] S. Methley, *Essentials of Wireless Mesh Networking*, ser. Cambridge Wireless Essentials Series. Cambridge University Press, 2009.

[8] X. Zhao, W. N. Hung, Y. Yang, and X. Song, "Optimizing communication in mobile ad hoc network clustering," *Computers in Industry*, vol. 64, no. 7, pp. 849–853, September 2013.

[9] M. Naserian, *Game Theoretic Approach in Routing Protocols for Wireless Mobile Ad Hoc Networks*, ser. Canadian theses, U. of Windsor, Ed. Canada: University of Windsor, 2008, vol. 1.

[10] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11 b mesh network," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 121–132, 2004.

[11] U. of Thessaly, "Vmesh – wireless network testbed at university of thessaly, volos, greece," Volos, Greece. [Online]. Available: http://vmesh.inf.uth.gr/

[12] N. Tsarmpopoulos, I. Kalavros, and S. Lalis, "A low-cost and simple-to-deploy peer-to-peer wireless network based on open source linux routers," in *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2005. Tridentcom 2005. First International Conference on*. IEEE, 2005, pp. 92–97.

[13] H. Lundgren, K. Ramachandran, E. Belding-Royer, K. Almeroth, M. Benny, A. Hewatt, A. Touma, and A. Jardosh, "Experiences from the design, deployment, and usage of the ucsb meshnet testbed," *Wireless Communications, IEEE*, vol. 13, no. 2, pp. 18–29, April 2006.

[14] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *Proceeding of the MobiCom'05*, Cologne, Germany, August 2005.

[15] G. Aggelou, *Wireless Mesh Networking*. McGraw-Hill Communications, 2009.

[16] P. Garnepudi, T. Damarla, J. Gaddipati, and D.Veeraiah, "Proactive, reactive and hybrid multicast routing protocols for wireless mesh networks," in *Proceedings of the 2013 IEEE International Conference on Computational Intelligence and Computing Research*. IEEE, 2013.

[17] M. Bahr, J. Wang, and X. Jia, *Wireless Mesh Networking - Architectures, Protocols and Standards*, ser. Wireless Networks and Mobile Communications. Boca Raton, FL: Auerbach Publications - Taylor & Francis group, 2007, ch. Chapter 4 - ROUTING IN WIRELESS MESH NETWORKS, p. 125.

[18] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626 (Experimental), Internet Engineering Task Force, Oct. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt

[19] M. Natkaniec, "Ad hoc mobile wireless networks: Principles, protocols, and applications (sarkar, s. k. et al.; 2008) [book review]," *Communications Magazine, IEEE*, vol. 47, no. 5, pp. 12–14, May 2009.

[20] H. X. T. Tan and W. K. S. Seah, "Dynamic topology control to reduce interference in manets," Institute for Infocomm, Tech. Rep. NET-2004-AHN-GENERAL-0027, Nov 2004.

[21] Google Groups - android-platform, "Android adhoc ibss wifi support." [Online]. Available: https://groups.google.com/forum/#!msg/android-platform/tLLspmSySbY/fm0zqCFBvQkJ

[22] Google Groups - android, "Issue 82: Support wi-fi ad hoc networking." [Online]. Available: https://code.google.com/p/android/issues/detail?id=82

[23] B. Randolf, "Ad-hoc (ibss) mode support for android 4.2.2/4.3/4.4." [Online]. Available: http://www.thinktube.com/android-tech/46-android-wifi-ibss

[24] cyanogenmod, "Introducing the cyanogenmod installer." [Online]. Available: http://www.cyanogenmod.org/

[25] J. "stoker" Robble, J. "m0nk" Thomas, and D. Doria, "The span project." [Online]. Available: https://github.com/ProjectSPAN

[26] J. "stoker" Robble, "Manet manager." [Online]. Available: https://play.google.com/store/apps/details?id=org.span&hl=en

[27] zxcasd12@gmail.com, "Apk extractor." [Online]. Available: https://play.google.com/store/apps/details?id=net.sylark.apkextractor&hl=en

[28] zacthespack, "Linux-on-android – working to getting a range of linux distros running on android." [Online]. Available: http://sourceforge.net/projects/linuxonandroid/files/

[29] tiborr, "Samsung galaxy s3 with ubuntu." [Online]. Available: http://forum.xda-developers.com/galaxy-s3/general/ubuntu-backtrack-i9300-samsung-galaxy-s3-t1700228

[30] The Linux Tutorial, "Linux optimized link state routing protocol (olsr) ipv6 howto." [Online]. Available: http://www.linux-tutorial.info/modules.php?name=Howto&pagename=OLSR-IPv6-HOWTO/olsrlinux.html

[31] A. Tønnesen, T. Lopatic, H. Gredler, B. Petrovitsch, A. Kaplan, S.-O. Tücke, *et al.*, "olsrd – an ad hoc wireless mesh routing daemon." [Online]. Available: http://www.olsr.org/