HLMP: High Level MANET Protocol

Juan Rodríguez-Covili, Sergio F. Ochoa, José A. Pino
Department of Computer Science
Universidad de Chile
Santiago, Chile
{jrodrigu, sochoa, jpino}@dcc.uchile.cl

***Abstract*—Wireless communication infrastructure dependence is not always a suitable way to achieve mobile collaboration and communication processes for various scenarios. In those situations a MANET can take advantage of the independent signal range of every mobile device, in order to create communication channels between mobile users. This technical report presents an application level routing protocol designed to automatically perform MANET formation and message routing procedures. It is intended to assist high level groupware development with a structured high level communication system.**

Keywords-MANET; routing protocol; mobile collaborative work; nomadic work

I. INTRODUCTION

Infrastructure elements support communication systems in many mobile technology applications. Examples of these elements are mobile phone cells antennas, Wi-Fi or Bluetooth access points, radio signal boosters and amplifiers. However, there are many scenarios where such infrastructure dependence is not possible , due to the high cost of the hardware elements or set-up processes, or the unfavorable users' movements to and from remote places, far away from wireless signal ranges. Another situation in which the infrastructure dependence is impossible occurs when the system collapses under massive connectivity events. On the other hand, the independent wireless signal range of every mobile device allows the set-up of a Mobile Ad-Hoc Network (MANET) [14]. This alternative may be convenient to use in the scenarios mentioned above.

A MANET is an autonomous peer to peer communication mesh supporting mobile group collaboration. It can be formed by different types of mobile devices which are also free to move (e.g. installed in land vehicles, ships, or transported by people). These devices are equipped with wireless network signal transmitters and receptors, usually Wi-Fi or Bluetooth, allowing them to communicate without making use of any kind of fixed infrastructure element. Previous studies show the usefulness of MANETs in scenarios of collaborative mobile work, e.g. catastrophe assistance or coordination in common emergencies, construction sites inspection, industrial or commercial applications, military activities, and search and rescue operations [2][14].

This network can be modeled as a graph G = (V, E), where V is the set of nodes representing the mobile devices and E is the set of arcs modeling the communicational range intersections between two or more devices [4]. Fig. 1 shows how a set of devices and their respective groups of direct communication connections are represented in this model. However, native ad-hoc wireless networks do not allow communication with devices that are outside the respective wireless signal range. Therefore, in order to enhance the collaboration and interaction possibilities, and create message exchange channels among all possible users inside a MANET graph, each node has to find suitable paths and routing methods to transmit messages to remote devices which are not adjacent neighbors. To make this mechanism possible, the intermediate nodes must re-transmit data packets which are not necessarily of their own interest. Moreover, the protocol used to support this behavior has to take into account the dynamics of the graph definition. The graph can change in an unpredictable way at any moment of time, because of the users' mobility while carrying the devices, the places where they move, or the wireless signal with respect to strength variation and environmental interference.

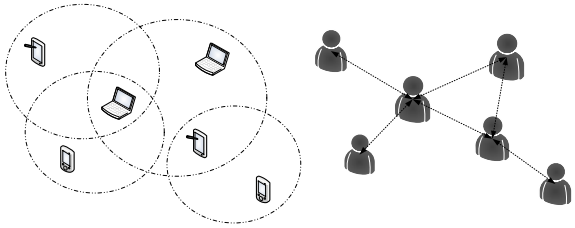

Figure 1. MANET model

This technical report introduces the High Level MANET Protocol (HLMP), an application level routing protocol supporting all communication functionalities required to automatically assemble a MANET structure, using operating system implementation routines and data transfer protocols such as UDP or TCP. Next section presents related work. Section IV presents the High Level MANET Protocol and its specification. Section V describes experimental results of the protocol on mobile shared workspace applications. Finally, Section 6 presents the conclusions and future work.

II. RELATED WORK

Several studies and initiatives have published, and continue researching routing protocols specifications in order to create and communicate MANET systems [1][5][6][11][17][17]. However, their complex and low level characteristic (i.e. accomplishment at IP layer) makes them hard to implement, adapt or reuse when trying to use different kinds of mobile devices or operative systems. Despite this fact, they are useful for hard networking processes and laboratory experimentation. They also create a comparison line for MANET works in progress.

Moreover, many publications have created standard terminology, problem definitions and solutions for several network topology issues, related to MANETs [3][9][14][15][18]. However, some of them are protocol specific solutions, and they are not totally reusable for other types of routing specifications.

III. HIGH LEVEL MANET PROTOCOL

The main goal of this protocol is to establish high level procedures of automation for creating a Wi-Fi MANET, so that mobile devices are able to enter a network and collaborate, sending messages to any other node inside the mesh. The key concept stems on the constant emission of a datagram known as "I'm Alive" message. This packet contains information about the sender node and its arcs set. It is the core functionality to create a MANET graph into every node's memory; it represents the vision and knowledge of the network at a certain moment of time. This idea makes the finding of optimal paths into the mesh feasible, in order to send and receive reliable messages.

The type of mobile networks we are studying constantly change behavior. Therefore, HLMP recalculates the path of a message in every node where the packet is transmitted, based on the current MANET graph knowledge and not on data analysis or statistically gathered information. HLMP delegates to the operating system the low level functionalities and it establishes the high level logic and procedures. It decides also the kind of implementation protocol suitable to use in order to provide the communication functionalities between two neighbor nodes (i.e. UDP or TCP).

A. The Three Wireless Signal Layers

Wireless signal communication behavior has been tested in previous studies, concerning protocols performance, and environmental error factors [7][8][10]. However, the authors have empirically found that the wireless signal emitted by a mobile device can be modeled as three main layers, as shown in Fig 2.

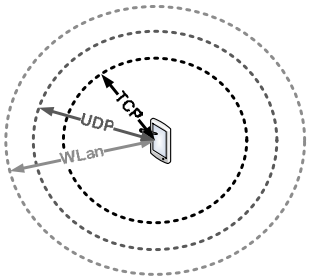

Figure 2. Wireless signal ranges

The WLAN layer is defined as the distance the device can use to create an ad-hoc network with another neighboring device; it is usually longer than the distance necessary to create TCP or UDP efficient procedures. The UDP layer is defined as the distance the device can use to send UDP multicast messages with a reasonable data loss rate; it is usually longer that the distance necessary to create fast TCP connections. Finally, the TCP layer is the distance needed to create TCP links in order to connect two devices with a reliable bridge.

HLMP uses this model to separate the process and functionalities. The WLAN layer is used to perform connection procedures and establish network IP address identification. The UDP layer is used to perform the peer detection mechanism and the MANET graph creation. Finally the TCP layer is used to establish direct paths between the nodes in order to send and rout reliable messages.

B. Connection Procedure

When a new device wishes to access an HLMP MANET, it has to perform a connection procedure ensuring the whole basic system structure. Fig. 3 shows the three macro-components of this process: WLAN Ad-Hoc connection, IP address self-configuration, and TCP and UDP services start.

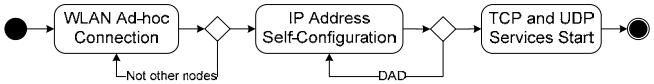

Figure 3. Network connection procedure

1) WLAN Ad-Hoc Connection

The node must delegate the emission of a wireless network profile to the operating system, using as SSID a common word selected by the upper application layers using this protocol, in order to create a WLAN when another device is detected and it emits the same profile. This profile has to be transmitted also with the Independent Basic Service Set modality defined by the IEEE 802.11 standard, which allow direct links between devices (ad-hoc behavior) without using any kind of access points [12]. Most operating systems use an XML profile specification for this mechanism.

2) IP Address Self-Configuration

IP address auto configuration is a desirable requirement in a MANET, because the mobile collaboration processes are on demand and usually new unknown devices need to enter or to go out of the network, and a unique network address must be automatically settled for each one of them. HLMP defines a random selection of the IP address structure and a fixed sub-net mask which defines the number of possible nodes inside the MANET. After the IP random configuration sequence, the devices have to also perform a Duplicate Address Detection process (DAD) in two stages: strong DAD and weak DAD [13]. Strong DAD is delegated to the operating system; this process can detect IP address duplications at the very moment of the conformation of the WLAN. Consequently, it can only detect device addresses belonging to the closest devices set. Weak DAD is managed by the protocol and consists in a verification process, constantly executed when receiving any kind of message. It checks the original IP address of the sender, comparing it with their own IP address, in order to detect duplicate addresses of devices belonging to adjacent or amalgam WLANs. If any duplicate address is detected then the device has to go back on the procedure, and perform the random IP process again.

3) TCP and UDP Services Start

Finally, the node has to start the corresponding services in order to initiate the communication mechanisms: A TCP service running at the previous configured IP address, allowing connections and reception of packets under an agreed port. An

UDP service subscribed to an agreed multicast group address is also started using a second agreed port.

C. Message Structure

A Network Message or HLMP datagram is composed of a four bytes header indicating the size of the inner data, and a body containing the message itself, named Communication Message. Fig. 4 (a) shows how a Network Message is structured.

A Communication Message consists of an organized packet of bytes containing data related to a high level message. It is also composed of a header and a body. Header data depends on the required mechanisms to send and route the message. HLMP defines four main mechanisms, named Meta Types.

- Multicast: an attempt is made to send the message to all nodes in the network using the UDP channels.
- Unicast: the message is sent to only one node in the network using the TCP channels.
- Safe Unicast: the message is sent to only one node in the network using the TCP channels, but the delivery of the packet must be confirmed by the receptor.

Fig. 4 (b) shows the header definition for multicast and unicast messages. Information contained in the Meta Type header specification corresponds to:

- Meta Type: the code of the Meta Type of the message.
- Type: the code of the specified type of message contained in the body packet. These codes are established by the specific functionalities required by upper application layers.
- Sub-Protocol: the code of an optional sub-protocol responsible for attending the message. The codes are established in the same way than the message type codes.
- Sender ID: identification code of the sender, which corresponds to a high level code for unique identification of the user at application layers.
- Sender IP: IP address of the sender.
- ID: randomly generated identification code of the message.
- Jumps: number of hosts in which the message has been received and routed.
- Target ID: the identification code of the addressee of the message.
- Target IP: the IP address of the addressee of the message.

(a)

Bytes	Multicast	Unicast
0-3	MetaType	MetaType
4-7	Type	Type
8-11	SubProtocol	SubProtocol
12-27	Sender ID	Sender ID
28-31	Sender IP	Sender IP
32-47	ID	ID
48-51	Jumps	Jumps
52-67		Target ID
68-71		Target IP

(b)

Figure 4. Network Message structure and Meta Type header

Finally, the body of a Communication Message consists of message packet data required by the particular collaboration functionalities.

D. Multicast Transmission Process

This process establishes the functionality required to send and route Multicast messages to be delivered to all users within the MANET. The devices do not require any kind of information about the topology of the network to carry on this procedure. However, a Message ID List is necessary to allow the temporary storage of received messages' identification numbers. The list must be composed of a FIFO queue and hash table. Using this structure makes possible the detection of copied messages that have been received by two or more different paths, in order to avoid message duplication problems.

1) Algorithms

The multicast process is described using Alg. 1 and Alg. 2. Essentially, it consists of the transmission of the message to all possible nodes that are in the UDP multicast group of the device. When a message of this kind is received, it must be re-transmitted again to all possible nodes, like flooding the network with the packet.

```
Send Multicast Message M to Everyone:
01 add M to MessageIdList;
02 send M to multicast group;
03 end;
```

Algorithm 1. Send Multicast message procedure

```
Receive Multicast Message M:
01 if MessageIdList  does not contains M {
02    add M to MessageIdList;
03    send M to multicast group;
04    process M as a received message;
05 }
06 end;
```

Algorithm 2. Receive Multicast message procedure

Sent and received messages ID's are saved into the Message ID List to check and avoid possible collisions. This allows the handling of message duplicates when there are multiple paths to the same nodes.

2) Example

Fig. 5 shows an example of the transmission process of one Multicast message into a simple MANET. The steps are: (a) node A wants to send a Multicast message M, it sends the

message to its multicast group, which correspond only to node C and B; (b) nodes C and B receive and processes the message, resending it to their multicast group; (c) nodes A, B and C detect the duplication of message M, so it is dropped when received, nodes D and E receive and process the message, resending it to their respective multicast group; (d) only node F receives, processes and resends the message to its multicast group, but then, node E will detect the copy of M, dropping it away. Finally, message M has been flooded on the network and all users have received it and processed it just once.

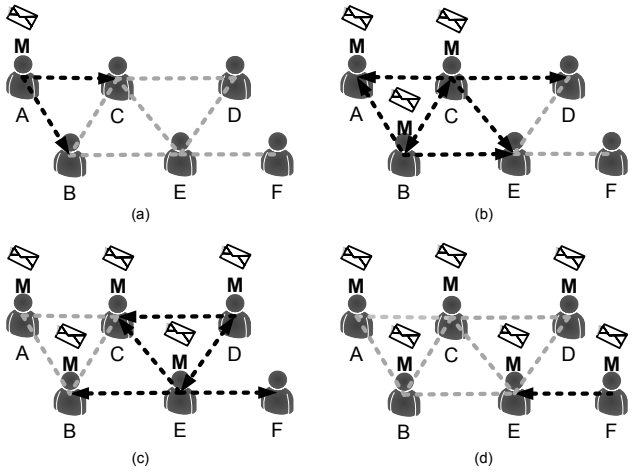

Figure 5. Multicast message transmission process example

3) Peer Detection Mechanism

, The nodes must emit and gather the network data, performing a peer detection activity. This process allows the generation of the MANET graph into each node's memory. This methodology is based on the transmission of a Multicast message named "I'm Alive" message, which contains the set of arcs of the sender, corresponding to the neighborhood created by direct TCP active connections between two nodes.

When connected to the MANET, nodes have to constantly send their "I'm Alive" message every one second. New nodes will send empty messages, and the old ones will send their corresponding neighborhood. Nodes have to process the received "I'm Alive" messages using the Alg. 3. When a sender node is added to the MANET graph, the node information and all its arcs defined in the "I'm Alive" message are kept. When a sender node is added to the neighborhood, this relays on trying to generate a TCP handshaking in order to establish a constant reliable connection between those two nodes. If a TCP connection is not possible, then the node has to wait a time interval shorter than 10 seconds after trying to connect again using Alg. 3. This time interval is a determining value in how fast the system reacts to neighborhood changes. If a TCP connection is dropped later, then the link is just removed from the neighborhood arcs set. The MANET graph obtained using this mechanism in each node is used later to send unicast messages.

4) Nodes Signal Quality

Nodes signal quality is measured in order to detect older information within a MANET graph (i.e. a portion of the graph has not been updated recently), and to detect nodes that have passed to an off line status, or have been moved out of the network.

When a node is added to the MANET graph using Alg. 3, a quality flag is set with value 25. The flag of all users in the graph is reduced by 1 every one second. If the flag of a node reaches a zero value, then it is assumed the node has gone away of the network, and it is deleted from the graph. If the information of a node is updated when performing Alg. 3, then the quality flag is increased by 5. Signal quality is then divided in three main sets. If a node has a flag value between 1 and 10, then the node has a Critical value. If a node has a flag value between 11 and 20, then the node has a Low value. And finally, if the node has a flag value between 21 and 25, then the node has a Normal value. This quality value represents how updated is the information of a specific node into the MANET graph, and it is a helpful information on the determination of unicast messages procedures.

```
Process Received I'm Alive Message M:
01 if graph does not contains sender of M {
02    add sender of M to graph;
03 }
04 else {
05    update sender of M;
06 }
07 if neighborhood does not contains sender of M {
08    if jumps of M is equal to 1 {
09       add sender of M to neighborhood;
10    }
11 }
12 end;
```

Algorithm 3. Process Received I'm Alive message procedure

5) Nodes Traffic State

Node traffic state is a local measure of each node. A flag value is set into each device and it counts how many Unicast or Safe Unicast messages are received per second. The traffic state is also divided in three main sets in order to propagate this information,. If a node has a flag value between 0 and 10, then the node has a Normal value. If a node has a flag value between 11 and 20, then the node has an Overloaded value. Finally, if the node has a flag value greater than 20, then the node has a Critical value. The set value is then attached to every "I'm Alive" message the node sends. This measure represents how much processing delay can have a message when passing by that node, and it is also helpful information while determining unicast messages routing. These values can be changed depending on the processing power of the devices and message sizes.

E. Safe Unicast Transmission Process

This process establishes the functionality required to send and route Safe Unicast messages with the goal to be delivered to only one node within the MANET using the TCP channels. The devices use the knowledge about the MANET obtained by the peer detection mechanism and the quality and traffic state values to generate a path cost matrix. This matrix is used to assign cost weights to the paths on the MANET graph. Fig. 6 shows the matrix table, the horizontal labels indicate the traffic state of a node, and the vertical labels indicate the signal quality value. When performing optimal paths calculation, the

result combination value is set to all paths surrounding the node.

Cost Matrix	Normal	Overloaded	Critical
Normal	1	10	100
Low	2	20	200
Critical	4	40	400

Figure 6. Path cost matrix

The same messages ID list used in the Multicast message transmission process is necessary to avoid the message duplication problem. It is also required a Unicast acknowledge message, named Ack. This message has the functionality of transporting the ID of a received message in order to confirm its reception.

1) Algorithms

The unicast process to send a message is described using Alg. 4. When trying to send a Safe Unicast message the node selects the best neighbor (first node in the optimal path), and it uses that TCP connection to send the message. The path is not saved in the message, it is recalculated in every node, when the message is received and it is intended to be routed. Then it holds that procedure for a time interval in order to resend the message if the acknowledgement has not been received yet. If the path finding algorithm does not returns any path, then there are not suitable ways to reach the host destination, so the message has to be processed as a failed message using the Alg. 6.

Alg. 5 shows the procedure a node has to execute when a Safe Multicast message has been received. On the one hand, if the node detects the target node of the message is not itself, then it has to perform the path finding algorithm in order to select the best neighbor to route the message. On the other hand, if the target of the message is actually that node, then it has to use the Message ID List to keep track of the reception, because the original node could be sending copies of the message due to times delays, messages lost for disconnection or other causes, and then, it has to send the Ack message, informing the reception of it.

If any node detects that a copy of a Safe Unicast message has been received, then it has to send the Ack message again, because it is unknown which message generated the duplication: a delay or loss of the original message or a delay or loss of the Ack.

Alg. 6 describes the procedure for processing a message targeted as a failed message in Alg. 4 or 5. While performing this operation, the node has to check if the target user still exists in the network, but no TCP connection has been initiated from the MANET to that node. In this case, it is possible to hold the procedure waiting a time interval (for the target node to connect to the network or to go finally out). After this time, the message is sent again, using the corresponding algorithm. If the target node is not detected in the network, then the node has been disconnected, and the message is dropped, or warned if the procedure is performed on the sender node.

This step can also be controlled using a maximum number of times for trying to find a path. If the same message is processed as failed too many times, then it is assumed there is a low probability for a path reaching the node to exist, so the message is finally dropped or warned.

```
Send Safe Unicast Message M to Node N:
01 while Ack of M has not been received {
02    Path <- minimum path to N;
03    if there exist a Path {
04       send M to first node in Path;
05       wait a Time interval;
06    }
07    else {
08       process M as a failed message;
09       end;
10    }
11 }
12 end;
```

Algorithm 4. Send a Safe Unicast message

```
Receive Safe Unicast Message M:
01 if target of M is not myself {
02    Path <- minimum path to target of M;
03    if there exist a Path {
04       send M to first node in Path;
05    }
06    else {
07       process M as a failed message;
08    }
09 }
10 else {
11    if MessageIdList does not contains M {
12       add M to MessageIdList;
13       send Ack of M to source of M;
14       process M as a received message;
15    }
16    else {
17       send Ack of M to source of M;
18    }
19 }
20 end;
```

Algorithm 5. Receive Safe Unicast message

```
Process Failed Safe Unicast Message M:
01 if target of M is in UserList {
02    wait a Time interval;
03    send M again;
04 }
05 else {
06    if source of M is myself {
07       warning failed delivery of M;
08    }
09 }
10 end;
```

Algorithm 6. Process failed Safe Unicast message

2) Example

Fig. 5 shows an example of the transmission process for one Safe Unicast message through a simple MANET (for simplicity, path costs are not displayed), showing disconnection events. The steps are: (a) node A wants to send a Safe Unicast message M to node D, it calculates the optimal path, and sends the message to node C (its best neighbor); (b) node C receives the message and the shape of the network changes, but it recalculates the path and it sends the message to node E; (c) node E receives and routes the message to node D; (d) node D receives and process the message, calculating the path to send the corresponding Ack, and sends that message to

node C; (e) node C routes the Ack message to node A; (f) finally, node A receives the Ack message and the process ends.

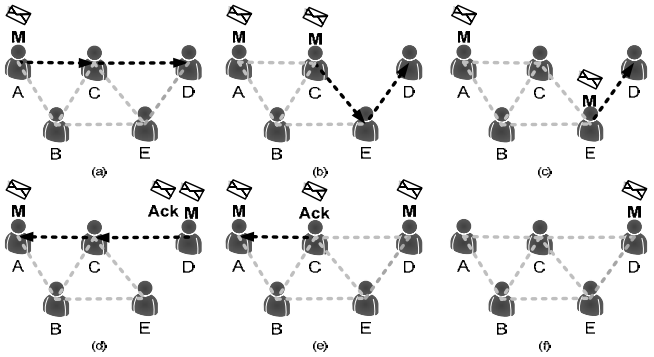

Figure 7. Safe Unicast message transmision process example.

IV. Implementation Results

HLMP has been implemented and tested to be a reliable solution for detecting users and sending messages in mobile contexts inside or outside buildings. It has been implemented for mobile shared workspaces applications in various scenarios such as: construction inspection mobile workspace; mobile map and information assistance for firefighters; secure peer to peer mobile file sharing application. It has been also useful at the development of frameworks as a communication block.

Results show that HLMP uses a fast deploy, and an automatically process for creating a MANET. It provides developers with a trustable communication mechanism they can use in order to create mobile groupware applications.

V. Conclusions and Future Work

HLMP offers a significant communication base to mobile groupware applications that do not have fixed infrastructure dependence. The high level logic allows to port and reuse the implementation of the protocol to different kind of devices and operating system in a fast and easy way.

References

[1] A. Neumann, C. Aichele, M. Lindner, S. Wunderlich: “Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.)”. IETF, Internet-Draft, work in progress, Apr 07, 2008.

[2] A. Neyem, S. Ochoa, J. Pino: “Integrating Service-Oriented Mobile Units to Support Collaboration in Ad-hoc Scenarios”. Journal of Universal Computer Science 14(1), pp. 88-122.

[3] C. Bernados, M. Calderon, H.Moustafa: “Survey of IP addres autoconfiguration mechanisms for MANETs”. IETF Internet Draft, work in progress, October 10, 2007.

[4] C. Perkins: “Mobile Ad Hoc Networking Terminology”. IETF Internet Draft, work In progress, November 1998.

[5] C. Perkins, E. Belding-Royer: “Ad hoc On-Demand Distance Vector (AODV) Routing”. IETF RFC 3561, July 2003.

[6] D. Johnson, Y. Hu, D. Maltz: “The Dynamic Source Routing Protocol (DSR)”. IETF RFC 4728, February 2007.

[7] Dan Duchamp, Neil F. Reynolds: “Measured Performance of a Wireless LAN”. Proceedings of the 17th IEEE Conference on Local Computer Networks, September 1992, pp. 494-499.

[8] David Eckhardt, Peter Steenkiste: “Measurement and analysis of the error characteristics of an in-building wireless network”. Conference proceedings on Applications, technologies, architectures, and protocols for computer communications, 1996, Pages: 243-254.

[9] Gavin Holland, Nitin Vaidya: “Analysis of TCP performance over mobile ad hoc networks”. Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, Pages: 219-230, 1999.

[10] George Xylomenos, George Polyzos: “TCP and UDP Performance over a Wireless LAN”; INFOCOM '99, Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Volume: 2, 21-25 Mar 1999, On pages: 439-446.

[11] I. Chakeres, C. Perkins: “Dynamic MANET On-demand (DYMO) Routing”. IETF Internet-Draft, Work in Progress, March 8, 2009.

[12] IEEE Computer Society. “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”. IEEE Standard for Information technology, Telecommunications and information exchange between systems, Local and metropolitan area networks, Specific requirements, Part 11, IEEE 802.11, 2007.

[13] Nitin H. Vaidya: “Weak duplicate address detection in mobile ad hoc networks”. Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computings, 2002, pages: 206 – 216.

[14] S. Corson, J. Macker: “Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations”. IETF, RFC 2501, January 1999.

[15] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, Jang-Ping Sheu: "The broadcast storm problem in a mobile ad hoc network". Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, Pages: 151-162, 1999.

[16] T. Clausen, C. Dearlove, P. Jacquet: “The Optimized Link State Routing Protocol version 2”. IETF Internet-Draft, work in progress, September 25, 2009.

[17] T. Clausen, P. Jacquet: “Optimized Link State Routing Protocol (OLSR)”. IETF RFC 3626, October 2003.

[18] Thirapon Wongsaardsakul, Kanchana Kanchanasut: “A Structured Mesh Overlay Network for P2P Applications on Mobile Ad Hoc Networks”. Lecture Notes in Computer Science, Volume 4882/2007, pages 67-72.