Adaptation Process for Ad hoc Routing Protocol

[1]Cholatip Yawut, [2]Béatrice Paillassa, [2]Riadh Dhaou

[1]Faculty of Information Technology, King Mongkut's University of Technology North Bangkok, Bangkok – THAILAND

[1,2]Univeristy of Toulouse, IRIT laboratory – ENSEEIHT, 2 rue Camichel
31071 Toulouse – FRANCE

{cyawut, Beatrice.Paillassa, Riadh.Dhaou}@enseeiht.fr

Abstract: Because of several constraints in ad hoc networks, an adaptive ad hoc routing protocol is increasingly required. In this paper, we propose a synopsis of an adaptation process for an adaptive ad hoc routing protocol. Next, we put into practice the analysis of the process of adaptation to mobility by realizing an adaptive routing protocol: CSR (Cluster Source Routing) which is an extension of a widely used ad hoc routing protocol: DSR (Dynamic source Routing). Mobility and density metrics are considered to CSR<->DSR mode switching, it moves from a flat architecture working in DSR to a virtual hierarchical architecture. With this mode switching, CSR can enhance the scalability of the DSR routing protocol.

1 Introduction

In an ad hoc Network [MM04], a node communicates either through single-hop transmission if the destination is in its transmission range, or by relying through intermediate nodes using a routing protocol. If one of the nodes on the ad hoc network has an internet connection, it is possible to share it with the other nodes on the network, like in the case of a traditional local network. The standard ad hoc routing protocols normalized by IETF (Internet Engineering Task Force) are OLSR (Optimized Link State Routing) [CJ03], TBRPF (Topology Dissemination based on Reverse-Path Forwarding routing protocol) [OTL04], AODV (Ad hoc On-demand Distance Vector) [PBD03] and DSR (Dynamic source Routing) [JMH07].

However, users require an ad hoc routing protocol which performs more adaptive than a standard ad hoc routing protocol because of several basic characteristics in an ad hoc network, for example: a limited battery capacity, limited and varied bandwidth, dynamic topology and mobility. Researches in adaptive ad hoc routing protocols have focused to adapt protocol behaviors according to these constraints.

To easily understand the mechanism of an adaptive ad hoc routing protocol, we propose a synopsis of adaptation process. With this synopsis, we put into practice the analysis of the process of adaptation to mobility by realizing an adaptive routing protocol: CSR (Cluster Source Routing). CSR [JP07] developed by the IRIT laboratory is an example of such an adaptive ad hoc routing protocol. CSR is a cluster-based extension of the DSR protocol, a popular deployed ad hoc routing protocol. CSR improves the scalability of DSR in high-density and low-mobility networks. CSR using a mode adaptation is regarded in the aspect of Cluster Head and Server selection. As CSR protocol defines a mode switching (it moves from a flat architecture working in DSR to a hierarchical architecture).

The rest of the paper is organized as follows. Section 2 describes a protocol adaptation view about related works, adaptation in the protocol stack. Section 3 illustrates CSR fundamental and how it adapts its behavior according to mobility and density metrics.

2 Protocol Adaptation View

2.1 Related Works

An adaptive ad hoc routing protocol uses metrics to adapt its behaviors. In [YBD09], we define the definition of metric; the metric is a measure indicating the state of a node, its neighborhoods, or the entire network, for example, energy level, transmission power, mobility etc... The measure may be a combination of several parameters such as a number of nodes, number of links, energy state...

Many researches use a mobility metric to adapt the system. Adaptive Routing Protocol for Manets (ARPM) [Se06] begins with using the proactive behavior and dynamically eliminates routing tables and switch to reactive behavior whenever the mobility degree exceeds a certain threshold. Adapting to Route Demand and Mobility (ARM) protocol [AS02] uses the rate of neighbor change as mobility metric. Fast-OLSR [BMA02] considers the number of neighbor changes as mobility metric. A node reduces its Hello-Interval when this metric reaches a predefined threshold. In [YPD07], we use link duration metric to improve MPR selection process.

In other ways, some protocols use a density metric to adapt their behavior. Density adaptive routing protocol (DAR) [Li08] utilizes the local network density to determine the packet forwarding zone; in dense areas, it narrows the forwarding range to reduce the total number of participants in flooding; in sparse area, it enlarges the forwarding scope to enclose enough nodes for packet relaying. LAKER [LM03], a LAR-based protocol, utilizes population density distribution and other knowledge for route guiding and passing around the void area.

2.2 Adaptation in the Protocol Stack

Our work focuses on a metric strategy to improve an adaptive routing protocol. First, it is necessary to understand how adaptation process works and what it consists of. In this step, we propose a synopsis of an adaptation process (Figure 1) by considering three sets of element: network environment, which is perceptible through metric, behavior to adapt, i.e. the algorithms to be applied to depend on metric values, and performance which we are trying to optimize. Therefore, the elements of the adaptation process consist of :

A) Metric

The metric gives values that can be used to adapt protocol behaviors. Several metrics could be used by the algorithm of adaptation, such as a routing protocol adapts its operation to the network density and mobility [JP07]. An environmental metric can be used to select a route or to establish a routing structure in an adaptive routing protocol.

B) Types and Policy of Adaptive Algorithms

Adaptive algorithms can be classified in 2 types: Auto and Cross-layer adaptations. The parameters taken into account in metric calculation may come from a single layer called as auto-adaptation, or several layers called as cross-layer or multiple-layers [SM05] adaptation. In this paper, auto-adaptation is evaluated because of its simplicity.

The adaptive algorithms can be also classified in 2 policies: Parameters and Mode adaptations.

Parameter adaptation: a protocol sets its parameters according to the state of its environment. Examples of such adaptation are:

- A transport protocol computes its window according to the emission rate parameter of congestion;
- A routing protocol chooses neighbors, links, or route depending on the stability or on the delay;
- A routing protocol adjusts its broadcast timer (for proactive routing) based on the network mobility: if the network is high mobility, the broadcast timer is short duration while it is important in low mobility case [QK06].

Mode adaptation: protocol changes its behavior depending on its environmental condition. For example:

- A transport protocol stops to increase its retransmission timer if an important level of mobility is detected [CPJ07];
- A routing protocol works in a mode (e.g. reactive, non architecture) and the number of mobile nodes increases, it switched to the other mode (e.g. proactive, with cluster architecture) [JP07].

C) Performances

Last element is the performance which adaptation seeks to optimize. We investigate the conventional performance characteristics of fixed network: throughput, delay, which adds energy.

A synopsis of an adaptation process is illustrated in Figure 1:

- Initially (Figure 1-a), a given node receives information (a set of parameters) coming from a single layer or multiple-layers. This information can be acquired from local source; such as node itself, or global source; such as neighbor nodes or network.
- Next (Figure 1-b), the parameters are taken into account in metric calculation.
- An adaptation strategy uses the values of metric calculation (Figure 1-c), for example the strategy may consider the M1 and M2 Metrics at the same time or only M1 metric or M2 metric beyond M1 metric or etc…
- The results of adaptation strategy are (Figure 1-d) to adapt a protocol behavior.
- After this adaptation, network parameters are changed by a node's mobility, giving new information to the given node to repeat the adaptation process (Figure 1-e).

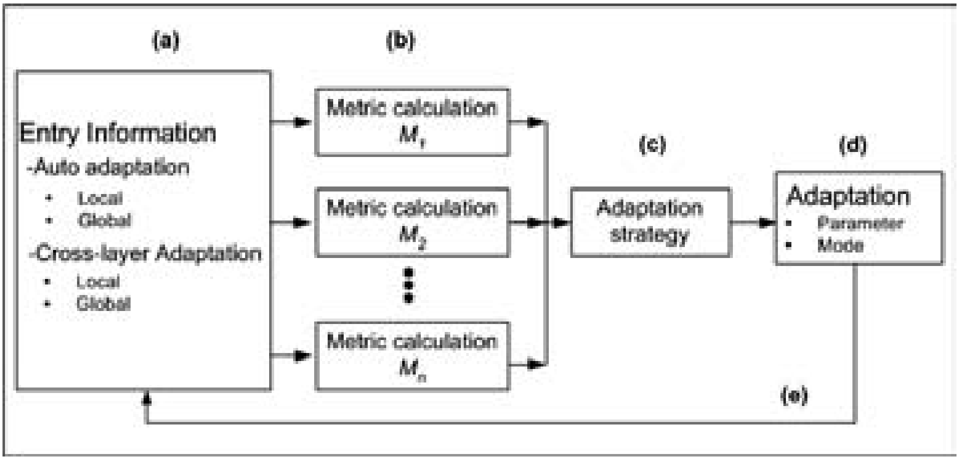

Figure 1: Synopsis of an adaptation process

3 Cluster Source Routing Protocol (CSR)

In this section, we put into practice the analysis of the process of adaptation to mobility by realizing an adaptive routing protocol: CSR (Cluster Source Routing) which is an extension of a widely used ad hoc routing protocol: DSR (Dynamic source Routing).

CSR is a source routing protocol and an architecture routing protocol. This protocol aims to increase the scalability of DSR with regard to network size and node mobility in an adaptive way. CSR is an auto-adaptation considering local information: a number of route errors as mobility metric and a number of neighbors as density metric. These metrics are obtained from node itself. CSR takes a strategy using a combination of these two metrics to change its mode and uses density metric to select a Cluster Head and Server.

Nodes can switch from DSR to CSR (DSR<->CSR) mode if the network stability and the local density are sufficient. DSR<->CSR mode is gainful on dense network configuration. The benefit of DSR-CSR especially grows with node density.

The CSR extension procedures are totally transparent and ensure full compatibility between native DSR and DSR<->CSR nodes. In fact, the DSR packet format is conserved. Native DSR and DSR<->CSR nodes can communicate since CSR integrates the DSR protocol. The CSR procedures are carried out through the DSR option mechanisms. Appropriate option codes are chosen to allow native DSR nodes to treat packets if necessary.

CSR [JP07] aims to increase the scalability of DSR with regard to network size and node mobility in an adaptive way. Nodes can switch from DSR to CSR (DSR<->CSR) mode if the network stability and the local density are sufficient. DSR<->CSR mode is gainful on dense network configuration. The benefit of DSR-CSR especially grows with node density.

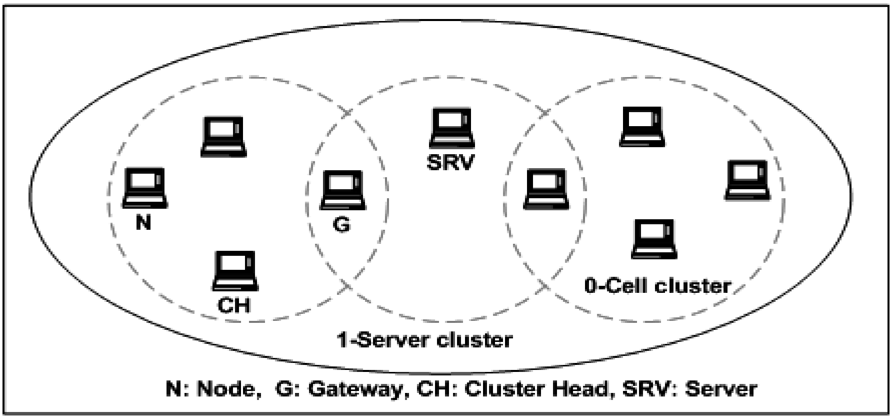

Figure 2: CSR model

The network is partitioned according to a 2-level hierarchical architecture (Figure 2). The lower level of cluster is the cell (0-Cell cluster). Each node within the cell is 1-hop away from the Cluster Head. Communication between 0-Cell clusters is completed through gateway nodes. The upper level of cluster (1-Server cluster) is formed by a set of cells. The associated cluster leader is named Server. Each node has four statuses:

- Undefined: the node has not yet obtained a valid status and is running the native DSR protocol.
- Node: a station which can use the CSR mode.
- Cluster Head: the cluster leader of the 0-Cell cluster.
- Server: the cluster leader of the 1-Server cluster.

Mobility and density metrics can be considered in an individual or combined way. In CSR, the mobility and the density metrics are used, by default, separately in order to perform the mode switching (Figure 3). The number of Route Errors is selected as the mobility metric and the number of neighbors in the route cache provides the density metric. Metrics are periodically computed. The values of the thresholds are as follows: M1 (low) = 2, M2 (high) = 4, D1 (low) = 2 and D2 (high) = 5.

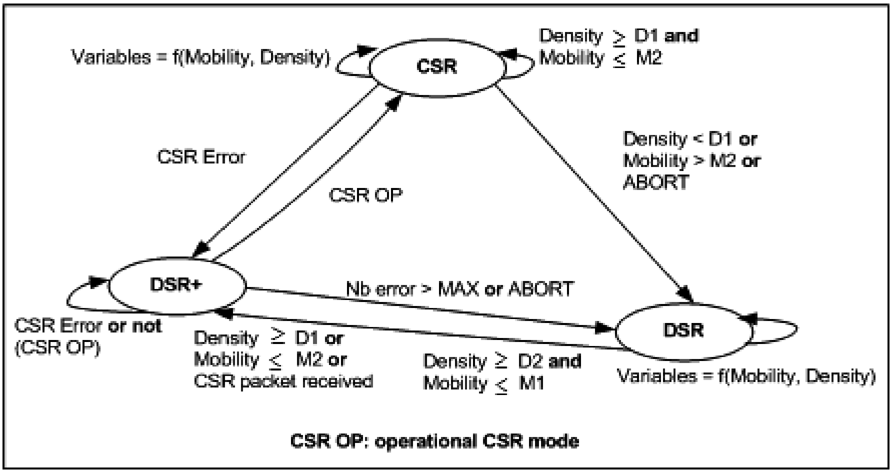

Figure 3: CSR states

A node switches to DSR mode if it experiences more CSR errors than the predefined MAX value. The MAX value is set to 3. CSR errors are caused by failures in setting up the architecture. On receiving an ABORT packet from the Server, a node also switches to DSR mode. Server sends an ABORT packet when it is about to give up its role.

Mobility and density metrics can be considered in an individual or combined way. In CSR default, the mobility and density metrics are separately used to perform the mode switching. The number of Route Errors is selected as the mobility metric and the number of neighbors in the route cache provides the density metric. Metrics are periodically computed. The computation timer value is set to 6s. The mode change is detailed below. Each node which runs DSR-CSR protocol could experience three states (Figure 5):

- DSR: The node uses DSR Route Discovery and DSR Route Maintenance. If network dynamics are favorable (high density and low mobility), the node enters DSR+ state. Two thresholds of mode switching are defined for both the Mobility metric (M1<M2) and the Density metric (D1 < D2):
  - Mobility > M2 or Density < D1: the node stays in DSR mode (high mobility and/or low density).
  - Mobility ≤ M2 and Density ≥ D1: the node changes from DSR to DSR+ mode if it receives a CSR packet (average mobility and density).
  - Mobility ≤ M1 and Density ≥ D2: the node switches from DSR to DSR+ mode (low mobility and high density).
- DSR+: The node uses DSR Route Discovery and Route Maintenance. However, clustering procedures are used to set up or to recover the CSR architecture. If clustering procedures succeed, the node enters CSR state. Else, it goes into DSR state. After its election, Server sets a timer and waits for Cluster Heads Registration to obtain its routing information. On timer expiration, Server is operational. When Server sends back a Registration Reply, it indicates to Cluster Head whether the CSR architecture is active or not. During DSR+ state, Server could stop CSR mode

by broadcasting an ABORT packet in the network (for example, few registered Cluster Heads indicating a low global density) and pass into DSR mode.

After its election, each Cluster Head registers to the Server. If the Registration procedure fails, Cluster Head will initiate a Server election. On MAX election failures, Cluster Head passes into DSR state. When Cluster Head receives a Registration Reply, it checks whether the CSR mode is operational or not. If so, it enters the CSR state and signals operational CSR mode to its cluster members through each Cell Maintenance packet. Else, it sets a timer and only goes into CSR state on its expiration. On receiving an ABORT, each Cluster Head switches to DSR state.

On receiving a Cell maintenance packet, the node checks whether the CSR is operational (use of CSR Route Discovery) or not (use of DSR Route Discovery). Each Node switches to DSR state if it receives an ABORT.

- CSR: Node uses CSR Route Discovery and Route Maintenance. CSR mode is operational and Cluster maintenance procedures are applied. Server broadcasts an ABORT message when it is about to give up its role because of network dynamics and switches to DSR state. If Server receives a packet from a higher criterion Server, it becomes Cluster Head and enters the DSR+ state. If Server is unreachable, the Cluster Head locally broadcasts a Cell Maintenance packet indicating to its cluster members that CSR architecture is not operational. Then, it applies the Registration procedure and switches to DSR+ mode. On receiving an ABORT, Cluster Heads and Nodes switch to DSR state.

3.1 CSR Procedures

The required procedures to operate the CSR extension are divided into two categories: Routing procedures serve to the discovery and maintenance of routes and Clustering procedures serve to the establishment and maintenance of virtual architecture.

3.1.1 Routing Procedures

Instead of diffusing a route request in the entire network, a route request in CSR mode is directly managed to the Server (transparently to the nodes, a Cluster Head manages this request). When a node has to send a packet, its Route is firstly searched for a route reaching to the destination. If such a route is not discovered, a Route Discovery (Figure 4) is launched using the non-propagating Route Request of DSR (Time To Live=1) by locally broadcasting a Route Request in its cell. If such a route exists in the cache of the Cluster Head of the cell, it is replied to the source node. If no route is known, the Route Request is transmitted to the Server (the path is known based on the periodic Topology Discovery of the Server). The Server verifies whether the destination survives. If so, the route between the source and the destination is created using its topology knowledge and a Route Reply is sends it back to the source node. Otherwise, Server asks all the Cluster Heads to find the required destination. Each Cluster Head investigates the destination in its cell using a non-propagating Route Request. A positive reply is send back to the Server, if a Cluster Head localizes the destination, The Server modernizes its topology information and replies to the source node. If it cannot indicate the destination, it sends back a Route Error packet (informing the unreachable destination) to the source. On receiving this Route Error from the Server, the node reinitiates a DSR Route Discovery.

In case of successive failures (maintenance clustering phase) route request is broadcasts through the network (DSR or DSR+ mode). CSR routing is completely transparent to the nodes: a DSR node without this extension can operate in the network.

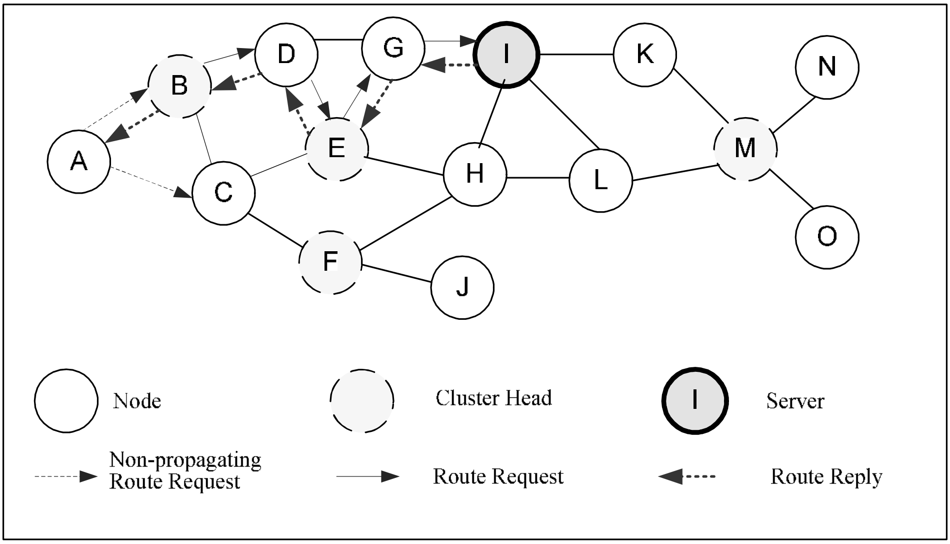

Figure 4: CSR Route Discovery

3.1.2 Clustering Procedures

In CSR, each node must obtain a status (Node or Cluster Head) to proceed with the CSR mode set up. Each Node is managed by a Cluster Head which forwards its Route Requests to the Server.

The set up of cells is based on the highest-connectivity degree algorithm; we can say that it is based on a density (D) metric. When a node enters CSR routing mode, it initiates the GetStatus procedure. Nodes which do not belong to a cluster are called uncovered nodes. To obtain a status, each uncovered node locally broadcasts a Route Request which contains its election criterion and indicates its undefined status (a Status packet). A specific option code is used to prevent neighboring native DSR nodes from processing the packet [JMH07]. Once the Status packet is broadcasted, the node waits for a GetStatus period. If a packet from a Cluster Head is received before GetStatus expires, the node initializes its status to Node. Else, on receiving a Status packet, the node checks its routing mode: CSR mode*:* it compares the packet election criterion with its own, DSR mode: it checks its adaptation criterion. If its criterion is suitable enough to switch to CSR mode, the node starts the GetStatus procedure and native DSR: it just discards the packet (unknown option code).

If the node has the local highest criterion (the lowest ID is preferred in case of tie), it sets up its status to Cluster Head and broadcasts a Cell Maintenance packet indicating its status. Thus, its neighbors take the Node status and stop their GetStatus procedure. If a node does not have the local highest criterion and does not hear any Cluster Head, it becomes itself Cluster Head at the end of the procedure.

Periodically, each Cluster Head locally broadcasts a Cell Maintenance packet to maintain its cell. If Node has not heard any Cluster Head during a Status period, it applies again the GetStatus procedure. The selected Cluster Head revocation algorithm is LCC (Least Cluster Change) in order to control the number of Cluster Heads [Ch97]: when two Cluster Heads are within transmission range, the lower-criterion one gives up its role and becomes Node. Thus, Cluster Heads are at least 2-hop away.

3.2 Server Selection Algorithm

The Server is elected among Cluster Heads and selected on the election criterion. At the beginning of the procedure, Cluster Heads initialize their candidate criterion variable with their election criterion value and the candidate address variable with their own address. Each Cluster Head which initiates an election broadcasts an Election packet in its 3-hop neighborhood. This Election packet is a DSR Route Request which contains the election criterion of the Cluster Head.

4 Conclusion

In this paper, the synopsis of an adaptation process is proposed to easily comprehend how an adaptive ad hoc routing protocol works. With this synopsis, we put into practice the analysis of the adaptation process to mobility by realizing in an adaptive ad hoc routing protocol: CSR. Cluster Source Routing adapts the DSR routing protocol to various conditions of mobility and density in ad hoc networks. The strategy of CSR <-> DSR mode switching is a combination of the number of Route Errors and the number of neighbor nodes as mobility and density metric respectively. These metrics are considered because it is simple; readily available the studied protocol, but efficient; signaling minimizing and no extra modification.

Improving the scalability of the DSR routing protocol can be achieved by realizing on a 2-level hierarchical scheme (0-Cell and 1-Server clusters) in CSR. Route Requests are transmitted to the 1-Server leader, considered as an upper level of Route Cache, to prevent network flooding. Then, data are transferred based on native DSR. Clustering procedures are specified to set up and maintain the CSR architecture. Each station separately adapts its routing mode (DSR or CSR) based on the mobility and density metrics. Computation methods of adaptation criteria are illustrated to authorize the change between modes and to adapt the routing variables.

Bibliography

[AS02] Anh, S.: Shankar, A.U.: Adapting to route-demand and mobility in ad hoc network routing. In Computer Networks: The International Journal of Computer and Telecommunications Networking, Volume 38, Issue 6, ISSN:1389-1286, April 2002; p. 745 – 764.

[BMA02] Benzaid, M.: Minet, P.: Alagha, K.: Integrating Fast Mobility in the OLSR Routing Protocol: In Fourth IEEE Conference in Mobile and Wireless Communications Networks, 2002.

[Ch97] Chiang, C. et al.: Routing in Clustered Multihop, Mobile Wireless Network with Fading Channel. In Proc. of IEEE Singapore International Conference On Networks (SICON'97), Singapore, 1997; p. 197-211.

[CJ03] Clausen, T.: Jacquet, P.: Optimized link state routing protocol (olsr), RFC3626, October 2003.

[CPJ07] Charoenpanyasak, S.: Paillassa, B.: Jaddi, F.: Experimental study on tcp enhancement interest in ad hoc networks: International Conference on Wireless and Mobile Communications (ICWMC 2007), Guadeloupe, 04-09, IEEE, Mars 2007.

[JMH07] Johnson, D.: Maltz, D.: Hu, Y.: The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks for IPv4, RFC 4728, February 2007.

[JP07] Jaddi, F.: Paillassa, B.: An Adaptive Hierarchical Extension of the DSR: the Cluster Source Routing. In: Journal of Universal Computer Science, John Wiley and Sons, Special Issue of the Journal of Universal Computer Science, Vol. 13 N. 1, 2007; p. 32-55.

[Li08] Li, Z. et al.: A Density Adaptive Routing Protocol for Large-Scale Ad Hoc Networks: In Wireless Communications and Networking Conference, 2008 (WCNC 2008); p. 2597–2602.

[LM03] Li, J.: Mohapatra, P.: Laker: Location aided knowledge extraction routing for mobile ad hoc networks: In Wireless Communications and Networking, (WCNC), Vol. 2, 2003; p. 1180–1184.

[MM04] Murthy C. S. R.; Manoj, B.: Ad Hoc Wireless Networks : Architectures and Protocols (Prentice Hall Communications Engineering and Emerging Technologies Series): Prentice Hall PTR, Special edition, June 3, 2004.

[OTL04] Ogier, R.: Templin, F.: Lewis, M.: Topology Dissemination based on Reverse-Path Forwarding routing protocol (TBRPF): Network Working Group, RFC3684. February 2004.

[PBD03] Perkins, C.: Belding-Royer, E.: Das, S.: Ad hoc on-demand distance vector (aodv) routing, RFC 3561, July 2003.

[QK06] Qin L.: Kunz, T.: Mobility metrics to enable adaprtive roiting in manet: IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2006 (WiMob'2006), 2006.

[Se06] Seba, H.L.: 2006 ARPM: An Adaptive Routing Protocol for MANETs. In ACS/IEEE International Conference on Pervasive Services, Volume 26-29 June 2006; p. 295 – 298.

[SM05] Srivastava, V.: Motani, M.:_Cross-layer design: a survey and the road ahead: IEEE Communications Magazine, vol. 43, no. 12, December 2005; p. 112-119.

[YBD09] Yawut, C. : Paillassa, B.: Dhaou, R.: Adaptation à la mobilité dans les réseau ad hoc : PhD thesis, INPT-ENSEEIHT, Sep 2009.

[YPD07] Yawut, C.: Paillassa, B.: Dhaou, R.: Mobility versus Density metric for OLSR enhancement: Asian Internet Engineering Conference (AINTEC 2007), Phuket, Thailand, Vol. 4866, Springer, LNCS, November 2007.