REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEURE ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABOU-BEKR BELKAID – TLEMCEN

THÈSE LMD

Présentée à :

FACULTE DES SCIENCES – DEPARTEMENT D' INFORMATIQUE

Pour l'obtention du diplôme de :

DOCTORAT

Spécialité: *Intelligence Artificiel*

Par :

Mr Bambrik Ilyas

Sur le thème

Etude De La QoS Dans Les Réseaux MANET et WMN

Soutenue publiquement le 9 Juillet 2016 à Tlemcen devant le jury composé de :

Mr Chikh Mohammed Amine	Professeur	Université de Tlemcen	Président
Mme Didi Fedoua	Professeur	Université de Tlemcen	Directeur de thèse
Mme Labiod Houda	MC	ParisTech	Co-Directeur de thèse
Mr Guy Pujolle	Professeur	Université UPMC Paris 6	Examinateur
Mr Khaled Boussetta	MC	Université Paris 13	Examinateur
Mr Abderrahim Mohammed Amine	MCA	Université de Tlemcen	Examinateur
Mr Senouci Sidi Mohammed	Professeur	Université de Bourgogne-France	Examinateur

Laboratoire de Recherche Informatique de Tlemcen (LRIT)
13000 Tlemcen - Algérie

Dedication Note

If this thesis has been possible, it is certainly thanks to God and the support of several people. I would like to take this opportunity to thank the ones that helped me.

I would like to express my gratitude to my family for giving me the motivation to advance in my academic level. Without whom it would have been impossible to go this far. Especially my mother for dedicating here life to meet all my needs. This lifetime is not enough to payback what owe here but, i will devote myself for here the same way she did for me. A special thanks to my aunt Mounia for introducing me to computer science at a very young age. My life would have been different without here support.

I am extremely thankful for having Professor Fedoua Didi as a mentor for the past six years. Always checking my progress and motivating me when difficult times are upon me. Without here support and constant attempts to resolve recurrent issues, it would have been impossible to complete this work. Thank you for everything.

I also want to thank Mme Houda Labiod (Laboratory of INFRES, ParisTech) for here advices and help.

My sincere thanks are also addressed to all the jury members for their appreciated efforts:

Chikh Mohammed Amine and Abderrahim Mohammed Amine, professors at University of Tlemcen. Guy Pujolle, professor at UPMC-Paris 6 University and Khaled Boussetta, professor at Paris 13 University. Senouci Sidi Mohammed, professor at University of Bourgogne-France.

Finally, i wish to express my thanks to all my colleagues from LRIT laboratory and teachers, especially my university professors. I hope that this work can be a proper tribute to those who made me realise that computer science is my true calling.

International Publications

2014 : «Overhead reduction methods for Ad hoc routing protocols», Ilyas Bambrik and Fedoua Didi. Conference «Symposium On Computer Networks and Information Security» , Tunisia, Hammamet. http://www.nngt.org/upload/conference2/p29.pdf

2015: «low overhead routing protocol for ad hoc networks», Ilyas Bambrik and Fedoua Didi. IEEE Conference «3rd International Conference on Control, Engineering & Information Technology», Algeria, Tlemcen. https://doi.org/10.1109/CEIT.2015.7233112

2015: «Novel Proactive Routing Method for Mobile Ad Hoc Networks», Ilyas Bambrik and Fedoua Didi. American Journal of Applied Sciences, Vol. 12, 6th issue, pages: 382.402. http://thescipub.com/abstract/10.3844/ajassp.2015.382.402

2015: «Load Management Algorithm For Wireless Mesh Networks», Ilyas Bambrik and Fedoua Didi. ACM Conference «Conference on Intelligent Information Processing, Security and Advanced Communication». http://dl.acm.org/citation.cfm?id=2816875

Abstract

Wireless Mesh Networks (WMNs) are currently emerging as a promising technology in the field of wireless networks. To offer a more robust solution for Internet access and community networking for mobile clients, this type of network inherits the Mobile Ad hoc Network (MANET) features: (a) auto-configuration, (b) auto-healing, (c) dynamic routing. Thus, unlike the other traditional networks, the WMN uses the MANET faculties to extend the network coverage in areas where there’s no infrastructure within reach. Although the MANET is a core feature in the WMN, this category of network usually contain infrastructure that provides the users with a better Quality of Service (QoS).

The key protocol that enables the flexibility of the topology for the MANET and WMN, and to provide QoS for the users, is the routing protocol. Therefore, this work is focused on studying the different routing protocols designed to operate in dynamic topologies and how they influence the QoS. Afterwards, in order to improve the QoS, we elaborate on different approaches to lower the routing complexity, limit the consumptions of the client nodes resources, improve mobility awareness and to optimize the usage of the WMN infrastructure.

Keywords: MANET, WMN, Routing Protocols, QoS

Résumé

Les réseaux Wireless Mesh Networks (WMNs) sont en train d'émerger comme une technologie prometteuse dans le domaine des réseaux sans fil. Pour offrir une solution plus robuste d'accès Internet aux clients mobiles, ce type de réseau hérite les fonctionnalités du réseau Mobile Ad-hoc (MANET): (a) auto-configuration, (b) auto-guérison, (c) routage dynamique. Ainsi, contrairement aux réseaux traditionnels, le WMN utilise les facultés du MANET pour étendre la couverture du réseau dans les zones où il n'y a pas d'infrastructure.

Bien que le WMN hérite les caractéristiques essentielles du MANET, cette catégorie de réseau est équipé d'une infrastructure consacrer à fournir aux utilisateurs une meilleure qualité de service. Le protocole clé qui permet la flexibilité de la topologie MANET / WMN et la promotion de la qualité de service offerte aux utilisateurs, est le protocole de routage. Par conséquence, ce travail se concentre sur l'étude des différents protocoles de routage conçus pour fonctionner dans des topologies dynamiques et leur influence sur la qualité de service. Ensuite, afin d'améliorer la qualité de service, nous détaillerons différentes approches pour : (a) réduire la complexité de routage, (b) limiter les consommations des ressources du noud mobile, (c) diminuer l’effet de la mobilité sur le protocole de routage et (d) d’optimiser l'utilisation de l'infrastructure WMN.

Mot clés : MANET, WMN, Protocole de routage, Qualité de service

الملخص

الشبكات NMW آخذة في الظهور حاليا باعتبارها تكنولوجيا واعدة في مجال الشبكات اللاسلكية. لتقديم حل أفضل للوصول إلى الإنترنت والشبكات المجتمعية لعملاء الجوال، هذا النوع من شبكة يرث ميزات TENAM : (أ) التهيئة التلقائية ,(ب) الشفاء تلقائي, (ت) التوجيه الديناميكي. وهكذا, على عكس الشبكات التقليدية, تستخدم شبكة NMW خصائص TENAM لتوسيع تغطية الشبكة في المناطق التي لا يوجد فيها بنية تحتية. على الرغم من أن TENAM هي آلية أساسية في NMW, هذه الفئة من شبكة عادة ما تحتوي على البنية التحتية التي توفر للمستخدمين نوعية أفضل من الخدمة.

البروتوكول الرئيسي الذي يتيح مرونة طوبولوجية لشبكة TENAM / NMW و تقديم نوعية جيدة من الخدمات هو بروتوكول التوجيه. لذلك, ركزنا على دراسة بروتوكولات التوجيه المختلفة المصممة للعمل في طبولوجية ديناميكية, وكيف أنها تؤثر على نوعية الخدمات. بعد ذلك، من أجل تحسين جودة الخدمة، نقترح أساليب مختلفة لتخفيض تعقيد التوجيه، الحد من استهلاك الموارد عملاء و تحسين الوعي التنقل ولتحسين استخدام بنية NMW التحتية.

الكلمات الرئيسية: بروتوكول التوجيه،جودة الخدمة, NMW, TENAM

General Introduction

Because of their multiple advantages, dynamic wireless networks are considered as a very profitable technology in the field of telecommunication. Though multiple dynamic networks with different characteristics emerged, the random / flexible nature of the topology is the defining commune factor in this category of network. Compared to its traditional wireless networks counterparts, the auto-configuration / auto-healing faculties of the MANET makes it a very suitable solution when the components of the network are moving rapidly (vehicular networks). Moreover, in scenarios where the deployment of a physical infrastructure is difficult (natural disaster) or unwanted (military application), the properties of the MANET occur perfectly.

Additionally, the self-governing feature makes this type of network highly flexible with a negligible deployment / maintenance cost. Hence, over the last few years, many studies have been directed to utilise the MANET for various applications such as the military field (Choudhary et al., 2004; Mariann et al., 2014), healthcare (Hussain, 2015), Internet access (Khalid et al., 2015), Pear to Pear (P2P) networks (Che-Liang et al., 2010; Afzal and Hossam, 2011), emergency situations (Kanchana et al., 2007; Koichi et al., 2010; Patricia et al., 2015) and most commonly, for vehicular communication / safe navigation (Li-Der et al., 2010; Demin et al., 2012; Sofiane et al., 2014; José et al., 2014; Timo et al., 2015).

In an arbitrary, constantly changing topology, the standard routing approach is almost obsolete. Therefore, various protocols were designed to operate in such a dynamic environment. Primarily, two categories of routing methods exist. First, the proactive protocols create and update a route to every reachable destination. Nevertheless, maintaining a complete view of the topology requires a continuous advertisement / processing of routing messages which raises the overhead. Hence, another method called reactive, was proposed to lessen the complexity of the routing operations. Conversely, in the reactive strategy, only the essential paths are built and maintained. On the other hand, after the establishment of a path, this strategy cannot detect the presence of a new route. Furthermore, the failure of the transmission path usually forces the source to restart the path building procedure all over again.

Hybrid protocols merge both previous methods. Rather than maintaining a complete routing table, the nodes create only local routing zones to limit the overhead / routing complexity. Plus, during the reactive session, the destination is located faster. However, the path produced by this approach is not necessarily the shortest. Similarly, to ease the nodes routing duties, the hierarchical protocols elect Cluster Heads (CH) and Gateways (GW) to manage the routing functions. But, this method does not guarantee the best path either and the concentration of traffic load / routing operations on the CHs / GWs can weaken the network performance.

Based on the same MANET core concepts, several similar network technologies emerged such as the Vehicular Ad hoc Network (VANET), Wireless Sensor Network (WSN) and more recently, the WMN. Though all the routing protocols, tailored for the aforementioned wireless network categories, are based on the same notions, each network category favours specific aspects more than others. For example, hierarchical protocols are largely preferred for WSN to maximize the battery lifetime of the sensors (Susmit, 2010; Hoda et al., 2012; Nora et al., 2012; Prabakaran et al., 2015). To make better use of the heterogeneous Wireless Mesh Network components and improve the QoS, hybrid and hierarchical protocols are more appropriate.

In this work, we take special interest in the MANET and WMN routing protocols. First, we study the basic as well as the most recent routing protocols developed for MANETs because this network category is the most primitive incarnation of the dynamic network notion. Then, we compare various routing protocols in different scenarios to verify their adaptability. After illustrating through the results that the routing protocols are pivotal in determining the QoS provided to the users, we detail solutions to two complex routing issues that heavily influence the QoS: (a) the random mobility of the nodes, (b) the complexity of proactive routing.

After discussing in detail the complex routing problem in dynamic networks, we analyse the WMN routing challenges. As a subcategory of the MANET, WMN inherits obviously the same routing issues in addition to the QoS maximisation task. Because, the WMN is designed specifically to provide a better QoS to the mobile clients while having the same flexibility attributes of the MANET. Hence, we propose a load balancing algorithm for WMNs, based on the Genetic Algorithm (GA), to conclude this work.

List of Contributions

Additionally to the studies of the state of the art, analysis and synthesis of the work in the literature we do in every step of our work, the main contributions of this thesis are:

1. We conducted several simulations in order to identify the most effective protocols to ensure QoS for the user data flows, while several important metrics are evaluated: overhead, delivery ratio, delay, throughput and path length (which have a direct influence on jitter), based on various scenarios: network size / density, number of transmission demands and velocity of nodes.
2. We discuss in detail the most efficient solution intended to limit the nodes mobility effect on the protocols performances. By anticipating path failures, this mechanism can be very efficient for packet losses avoidance in a protocol such as AOMDV.
3. We propose a proactive protocol ERBOR (Effective Routing Based on Overhead Reduction) to limit the routing complexity. Moreover, we included a mechanism that enables ERBOR to ignore control messages that contain redundant routing information. Instead of processing all the received control messages like other proactive protocols, ERBOR verifies first the control message generator field to determine if the message is useful.
4. We analyse the WMN routing challenges. In addition to the general dynamic routing problems, the WMN is meant to offer the best possible QoS to the mobile clients. Thus, we propose a load balancing algorithm for WMNs.

Plan of Manuscript

To detail all our contributions and our work approach, this thesis is organized as follows:

In Chapter I, we present a study of the state of the art covering the evolution of the routing solutions developed for MANET. This chapter has several tables summarizing the advantages and disadvantages of different solutions previously proposed.

Chapter II provides a comparative study of seven routing protocols, four of which are reactive (AODV, DSR, AOMDV and LAR) and three proactive (DSDV, OLSR, DREAM) through different scenarios, to determine the best protocols able of promoting QoS for the users data flows.

Chapter III is devoted to the presentation of our main contribution named ERBOR.

Chapter IV includes our proposition in load balancing for WMN after a brief presentation of routing protocols tailored specially for WMN.

Finally, we present a general conclusion obtained of all our work and the prospects of this thesis.

Summary

Chapter I: Routing In Dynamic Networks 6
I.I. Introduction 7
I.II. Proactive protocols 9
I.III. Reactive protocols 15
I.IV. Hybrid and hierarchical protocols 19
I.V. Recent advances 22
I.V.1. QoS Oriented Protocols 23
I.V.2. Energy Aware Protocols 24
I.V.3. Mobility Aware Protocols 25
I.V.4. Security Oriented Protocols 27
I.VI. Conclusion 29
Chapter II: Comparison of the MANET Routing Protocols 32
II.I. Introduction 33
II.II. Results and Discussion 33
II.II.1 The network growth 34
II.II.2 The increase of the network density 41
II.II.3 Number of transmission connections 43
II.II.4 Node velocity 47
II.III. Conclusion 50
Chapter III: Proposition for two Extended Protocols 54
III.I. Introduction 55
III.II. Mobility aware extension for the reactive protocols 56
III.III. Lowering the overhead and complexity of the proactive protocols 58
III.II.1 Routing messages treatment algorithm 61
III.II.2 Path update treatment 62
III.II.3 Deletion updates treatment 64
III.II.4 The routing information filter algorithm 66
III.II.5 Protocol testing 68
III.IV. Results and discussion 76
III.III.1 The network growth influence 77
III.III.2 The network density influence 79

III.III.3 Network members movement 82
III.V. Conclusion 85
Chapter IV: Proposal Of A Load Balancing Mechanism For Wireless Mesh Networks 88
IV.I. Introduction 89
IV.II. Routing in WMN: State of the Art 90
IV.III. Load balancing: State of the Art 94
IV.IV. A Load Balancing Algorithm For WMN Based On the Genetic Algorithm 96
IV.V. Overview on evolutionary computing 97
IV.V.I. Ant Colony Optimization 97
IV.V.II. Genetic Algorithm 98
IV.V.III. Chemical Reaction Optimization 101
IV.VI. Our Proposed Algorithm 102
IV.VII. Conclusion 106

List of figures

Fig 1. Virtual hierarchical organization for the MANET. 8
Fig 2. Example of the MPR selection. 10
Fig 3. Fisheye state routing. (Gerla et al., 2002). 11
Fig 4. OGMs dissemination. 12
Fig 5. The path maintenance process. 15
Fig 6. Dissemination of route requests. 16
Fig 7. Route reply. 16
Fig 8. Zone Routing Protocol. 20
Fig 9. Zone based hierarchical link state. 20
Fig 10. The recorded overhead during the network growth experiment. 34
Fig 11. The processed routing information during the network growth experiment. 36
Fig 12. The average routing packets sizes during the network density experiment. 37
Fig 13. The recorded delivery ratio during the network growth experiment. 38
Fig 14. The recorded throughput during the network growth experiment. 38
Fig 15. The recorded overhead during the network density experiment. 41
Fig 16. The recorded delivery ratio during the network density experiment. 42
Fig 17. The recorded throughput during the network density experiment. 42
Fig 18. The recorded overhead during the transmission connections experiment. 44
Fig 19. The recorded throughput during the transmission connections experiment. 45
Fig 20. The recorded delivery ratio during the transmission connections experiment. 45
Fig 21. The recorded overhead during the nodes velocity experiment. 47
Fig 22. The recorded throughput during the nodes velocity experiment. 49
Fig 23. The recorded delivery ratio during the nodes velocity experiment. 49
Fig 24. Overall performance ranking based on the simulation results. 52
Fig 25. The unawareness of the newly formed paths. 55
Fig 26. Path failure prediction through signal strength measurement. 56
Fig 27. Mobility aware algorithm to predict imminent path failures. 57
Fig 28. The network density effect on the proactive protocol. 59
Fig 29. Update message processing algorithm. 62
Fig 30. Path update treatment. 63
Fig 31. Processing a received update for an existent path. 63
Fig 32. Processing a received update for a new path. 64
Fig 33. Processing a deletion update. 65
Fig 34. Removal update scenario. 65
Fig 35. Redundant information treatment. 66

Fig 36. Routing information filter. 67
Fig 37. Class diagram for the visual components....69
Fig 38. Class diagram for the protocol layer. 70
Fig 39. The topology generator. 71
Fig 40. The various visual interfaces accessible for the user. 71
Fig 41. Path visualization. 72
Fig 42. The overhead measurement screen 73
Fig 43. Routing information processed. 74
Fig 44. Calculating the percentage of routing messages dropped from an NS 2 trace file....75
Fig 45. The impact of the network growth on the overhead generated per node. 78
Fig 46.The impact of the network growth on the quantity of routing information processed per node. 79
Fig 47. The impact of the transmission range on the overhead generated per node....80
Fig 48. The impact of the transmission range on the quantity of routing information processed per node. 81
Fig 49. The overall quantity of routing messages processed by DSDV and FSR. 82
Fig 50. The impact of the speed of movement on the overhead generated per node. 84
Fig 51. The impact of the speed of movement on the quantity of routing information processed per node. 84
Fig 52. The effect of the speed of movement on the overhead in a continues movement scenario. 85
Fig 53. The Wireless Mesh Network architecture. (Marwaha et al., 2008) 90
Fig 54. The two Hybrid Wireless Mesh Protocol modes. (Islam et al., 2009). 91
Fig 55. Breaking a Cluster. (Shahverdy et al., 2011) 95
Fig 56. Path localization via the ACO method. (Sensarma and Majumder, 2013) 98
Fig 57. Initial solutions....100
Fig 58. New solutions....101
Fig 59. Exploring the TM for alternative paths. 103
Fig 60. The creation of alternative paths. 103
Fig 61. Applying the genetic algorithm....105
Fig 62. Multi criteria path selection method. 106
Fig 63. Path ranking method. 106
Fig 64. Network Simulator 2. 112

List of tables

Table 1. Comparison between the proactive protocols. 13
Table 2. Comparison between the reactive protocols. 18
Table 3. Comparison between the hybrid / hierarchical protocols. 21
Table 4. Summary of the recently developed routing protocols. 28
Table 5. Comparison between the categories of the MANET routing protocols. 29
Table 6. Simulation parameters. 33
Table 7. The recorded average path length and end to end delay during the network growth experiment. 40
Table 8. The recorded average path length and end to end delay during the network density experiment. 43
Table 9. The recorded end to end delay and path length during the transmission connections experiment. 46
Table 10. The recorded average path length and end to end delay during the nodes velocity experiment. 50
Table 11. Average performances comparison. 51
Table 12. Routing table structure. 61
Table 13. ERBOR routing message format. 61
Table 14. The network growth parameters. 77
Table 15. The network density parameters. 80
Table 16. The network members movement speed parameters. 83
Table 17. The topological parameters influence on DSDV, FSR and ERBOR. 86
Table 18. Summary of the protocols designed for WMN. 93

Chapter I: Routing In Dynamic Networks

I.I. Introduction

Because of their multiple advantages and the recent advances in the field of wireless networks, the Mobile Ad hoc Networks (MANETs) progressively gained in popularity. This type of network is distinguished by its distributed nature and auto-healing / auto-organization features. At any moment, the network members are able to move freely which makes the network topology highly flexible. Furthermore, a MANET doesn't contain any form of physical infrastructure which reduces the deployment cost. Instead of dedicating a preset infrastructure for routing operations, all the network components contribute to forward packets and to execute routing functions in a distributed manner.

The dynamic nature of the topology is the core feature of the MANET. This category of network is designed to maintain the connectivity between the users while they are in motion. Consequently, the classic routing strategies can't be applied in such a situation. The MANET routing protocol must react fast to the topological changes and perform routing functions in a distributed fashion. Therefore, numerous routing protocols were specifically designed for dynamic topologies. Based on the way the MANET protocols construct the paths, three types of protocols exist: (a) proactive protocols, (b) reactive protocols, (c) hybrid protocols.

Proactive protocols build and maintain a path to every possible destination in the network. This strategy requires a periodic broadcast of the topological information maintained by each node. Afterwards, the received routing messages are processed to update the routing table. As a result, this strategy optimizes all the available paths and reduces the transmissions delay since all the possible paths are pre-built. Nevertheless, overtime, the constant broadcast / treatment of control messages can consume a sizeable amount of the network resources.

On the other hand, reactive protocols waist less energy on routing operations by creating a path on request only. This strategy is based on a request / wait for reply cycle and doesn't require a periodic advertisement of the global topological information. During the construction of a path, the intermediate nodes forward the transmission request once only until the destination is found. After that, the destination generates a reply and sends it through the discovered path. Although this method reduces the overhead, the path building process forces the source to wait until a path towards the destination is established which increases the transmission delay. Besides, with the absence of routing information exchange, the source is unable to find a better path unless the one in current use is broken.

The hybrid protocols combine the reactive and the proactive methods. Usually, the proactive method is used to collect the routing information about the neighbourhood. Then, the reactive method is launched when a path to a remote destination is needed. As a result, this approach reduces the transmission delay by comparison with the reactive method and generates less overhead by advertising only the neighbourhood table. However, this type of protocol inherits partially the reactive protocols defects (transmission delays and unawareness of the newly formed paths) and doesn't necessarily produce the shortest path since the search for the destination is zone driven.

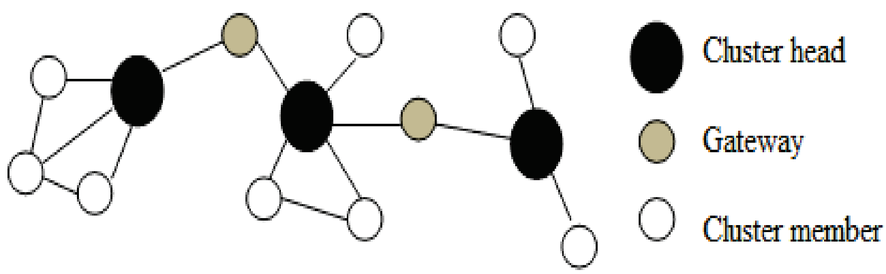

Fig 1. Virtual hierarchical organization for the MANET.

Some protocols organize the network in a hierarchical manner to attribute a role for each node. The idea is to represent the topology in a form of multiple clusters where each cluster, has one representative, in charge of routing functions, called Cluster Head (CH). Secondly, the network members that belong to more than one cluster are used for communication between clusters and are called Gateways (GW). Consequently, the hierarchical organization minimizes the routing operations performed by the cluster members since the routing functions are handled by the CH and GW only. The downside is that the concentration of data flows on the CHs and GWs can lead to network congestion. Fig 1 is an example of how the network is organized by the hierarchical protocols.

Besides their primitive routing strategies, each routing protocol applies additional methods to adapt to a specific requirement. Some protocols are focused on guaranteeing better privacy to the user. Others tend to increase the network life time and / or improve the Quality of Service (QoS). In this chapter, we go through the various methods applied by the routing protocols designed for MANETs.

I.II. Proactive protocols

Typically, the proactive protocols are based on the periodic broadcast of the routing information collected by the network members. Subsequently, the received control messages are processed to refresh the topological view and the detected changes are announced afterwards. This type of protocols auto-optimizes all the possible paths and minimizes the transmission delay. Nevertheless, the constant treatment and advertisement of routing messages, can be expensive in terms of energy and CPU usage. Especially in large networks.

a) Destination Sequenced Distance Vector (DSDV) is a proactive routing protocol based on the Distributed Belman Ford (DBF) algorithm (Perkins and Bhagwat, 1994). In this protocol, a sequence number is used to define the freshness of the path. When a node can choose between multiple paths leading to the same destination, the path with the highest sequence number is favoured. An entry in the routing table of this protocol fundamentally contains: (a) the next hop ID, (b) hop count, (c) the destination ID, (d) the destination sequence number. The main purpose of this protocol is to eliminate routing loops through the use of the sequence number.

The nodes broadcast their routing information periodically and when an important change is detected, the new information is immediately announced. Two types of routing messages are used by DSDV: (a) Full Dump message which contains the entire routing table, (b) Incremental message which contains only the changes since the last advertised Full Dump. When a significant topological alteration is sensed, a Full Dump message is advertised instead of the Incremental message so that the next routing message will contain less information. Additionally, Full Dump messages are advertised after a constant period to refresh the routing view of the neighbours.

When a path to a destination is lost, the hop count is set to infinity. The same operation is made when a link with a neighbour is lost: the hop count is set to infinity in the path attributed to the neighbour and all the paths constructed based on the corresponding neighbour. To announce that a link with a neighbour is broken, a node generates a higher sequence number than the last known sequence number of the neighbour. Subsequently, this information is propagated in the network to delete all the paths leading to the corresponding destination.

b) Optimized Link State Routing (OLSR) is a proactive protocol based on the Link State (LS) algorithm (Clausen and Jacquet, 2003). To run this protocol, every network member chooses a subset of its neighbourhood called Multipoint Relays (MPRs) which is used to forward routing and data messages. Redundant neighbours are excluded from the MPR list to reduce the number of the advertised links. The information describing the neighbourhood is advertised periodically in control messages called HELLO which are used to calculate the MPR ensemble. Then, the MPR nodes are chosen in a way that all the neighbours within two hops distance, are attainable. This method becomes more efficient when the MPR ensemble is reduced.

After the definition of the MPR ensemble, only the selected MPRs are used to forward control messages. The rest of the neighbours process the control messages, but do not forward them. Consequently, this protocol eliminates the asymmetric link problem. In Fig 2 for instance, MH3 can be considered as a redundant neighbour for MH1. Since MH3 is accessible through MH2 and MH4, MH3 is excluded from the MPR list of MH1. On the other hand, MH11 is only accessible through MH7 and therefore this node is selected as a MPR for MH1.

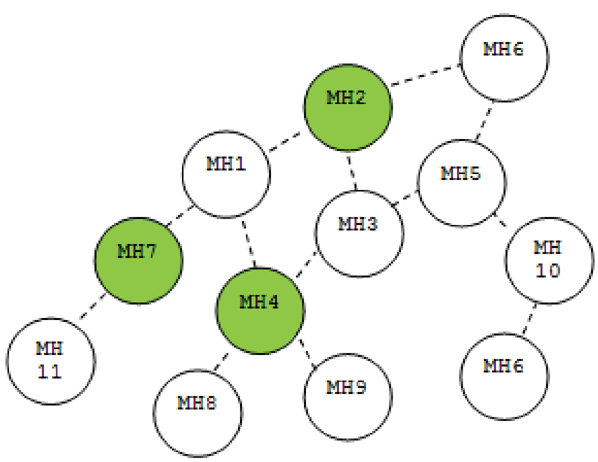

Fig 2. Example of the MPR selection.

c) In Fisheye State Routing (FSR), the Fisheye algorithm is applied to reduce the routing overhead (Gerla et al., 2002). The Fisheye algorithm was initially designed to compress the representation of visual data. When this method is applied on the topological information, the nodes advertise the routing information describing the close neighbourhood more often than the paths leading to the destinations faraway. Consequently, a node obtains a precise routing view describing the close neighbours and the precision decreases gradually for the remote destinations. Although this technique reduces the quantity of overhead, it makes the path built towards the remote destinations, less accurate.

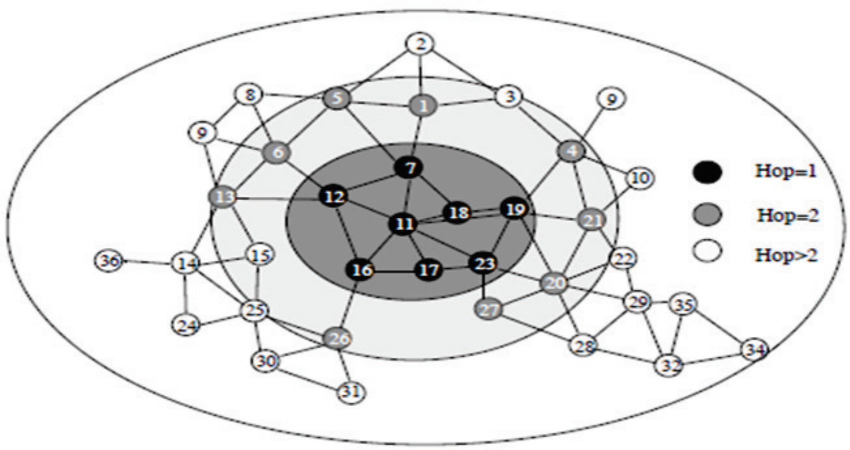

Fig 3. Fisheye state routing. (Gerla et al., 2002).

d) Distance Routing Effect Algorithm for Mobility (DREAM) is a proactive routing protocol that collects the geographical coordinates of the mobile hosts (Basagni et al., 1998). In this protocol, the nodes obtain their locations through a localization system and advertise them in routing messages. Subsequently, the received geographical positions, for each destination, are stored in a Location Table. This protocol adapts the overhead / routing accuracy by two mechanisms: (a) the distance effect, (b) the destination relative speed.

The idea behind the distance effect method is that when two nodes are faraway, they appear to be moving less respectively to each other. Based on this observation, the routing information received from the nodes faraway are less important than the description of the close neighbours. To reduce the overhead quantity, an age parameter is included in the routing messages to determine how far the paths are forwarded. The value of this parameter can be short so that the routing information is announced only in a restricted zone then ignored after that. This type of message is generated frequently to update the nearby nodes. Other routing messages age parameter is higher which means that they are intended to be delivered to a larger part of the network. This type of message is advertised after a constant period to reach the distant network members. Though this strategy is similar to the idea applied in FSR, DREAM collects the geographical distance instead of hop count.

In the second mechanism of DREAM, the frequency with which the routing messages are broadcast depends on the node speed of movement. Accordingly, the nodes moving fast broadcast their routing messages more frequently than those moving slowly. Afterwards, when a path to a destination is required, the geographical description of the destination is used to calculate the intermediate nodes that are close to it and the packets are forwarded through them. Although the methods applied by this protocol can reduce the overhead, the overhead produced by this protocol expands when the nodes move rapidly.

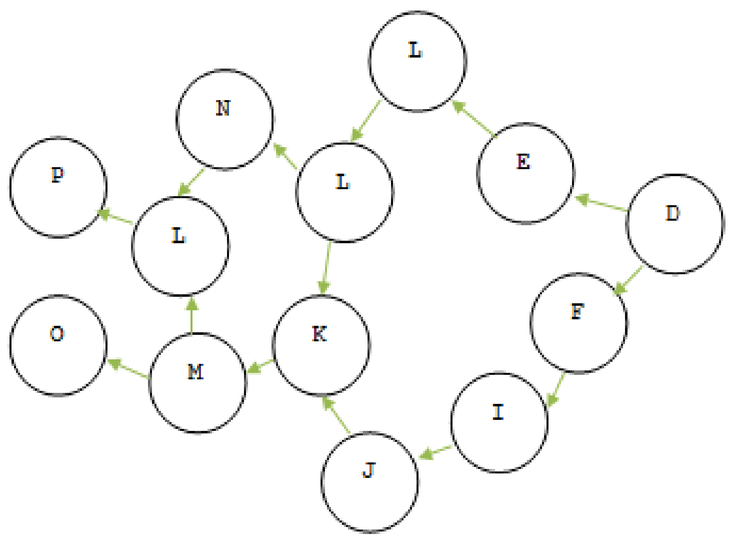

Fig 4. OGMs dissemination.

e) Better Approach To Mobile Ad hoc Networking (BATMAN) is distinguished by its new approach of path selection (Neumann et al., 2008). Instead of calculating the shortest path towards the destination, the node detects which neighbour offers the best path to the destination. In this protocol, the routing information is not exchanged directly like in other protocols. Each node broadcasts routing messages called Originator Messages (OGMs) periodically and the receiving neighbours rebroadcast them afterwards. During this process, the path to each destination is selected based on the number of OGMs received.

Like in other routing protocols, the OGMs are advertised in User Datagram Protocol (UDP) mode. For instance in Fig 3, the OGM generated by D is forwarded by intermediate nodes. During this process, the receivers of the OGM update their routing entries attributed to D. Since the path selection is based on the number of OGMs received from the destination, this protocol chooses the paths that have the lowest packet losses. Though a Time To Live (TTL) parameter is used to limit the distribution of the control messages, the constant rebroadcast of OGMs increases the overhead.

f) In Babel, the path selection process is based on the probability of transmission without packet losses (Juliusz, 2010). To compute this metric, the protocol counts the number of routing messages received from a neighbour during a period of time. Then, to calculate the ETX metric (De Couto et al., 2003), the number of messages received is divided by the excepted number of messages. Afterwards, the neighbours that offer the highest probabilities of transmission without pack loss, are used for path construction. As a result, this protocol promotes the paths that have a low signal interference and a high throughput. On the other hand, like in the BATMAN protocol, the constant broadcast of the routing messages increases the overhead.

Table 1. Comparison between the proactive protocols.

Protocol	Advantages	Inconveniences
DSDV	• Eliminates routing loops through the use of the destination sequence number. • Significant topological changes are immediately announced. • Medium complexity.	• Overhead depends on the topology size. • Paths to the remote destinations are distributed slowly (which causes routing errors)
OLSR	• The use of MPR eliminates the asymmetric links problem.	• Overhead depends on the network size.

		• High complexity.
FSR	• Reduces the overhead by minimizing the broadcast frequency of the routing information describing distant destinations. • Low complexity.	• When the nodes move frequently, the routing information describing the remote destinations becomes less accurate. • Overhead depends on the network density.
DREAM	• Adapts the overhead through the use of the geographical information. • Rapid path recovery.	• Paths produced by this protocol are not necessarily the shortest. • The overhead produced by this protocol depends on the network members' mobility.
BATMAN	• Creates paths with low packet losses / high throughput.	• Generates a considerable volume of overhead (the broadcast interval of the OGMs is smaller than the broadcast interval of other proactive protocols).
Babel	• Spends fewer resources on data retransmission by avoiding signal interferences.	• Generates an important quantity of control messages to calculate the ETX metric.

Table 1 is a comparison between the abovementioned proactive protocols. To summarize, the proactive protocols offer an advantage in collaborative scenarios (the users are in constant communication with each other) in addition to optimizing all the available paths continuously. However, as the size of the network expand, this type of protocol generate a substantial amount of overhead. The most original protocols in this category and the most used in the literature are: (a) DSDV due to its low complexity, (b) OLSR because of its MPR path construction approach, (c) DREAM because of its mobility oriented approach. Which is why we will test these three protocols in the next chapter.

I.III. Reactive protocols

Whereas the proactive protocols main focus is to maintain the paths between all the nodes, reactive protocols create a path on demand only.

a) Designed by Perkins et al. (2003), Ad hoc On demand Distance Vector (AODV) is a reactive protocol that applies a request / wait for reply cycle to construct a path. When a node needs a path to a given destination, the search is initiated by the broadcast of a Route Request (RREQ) message. After receiving a RREQ, the intermediate node creates a reverse path to the source and rebroadcast the request only once. Afterwards, when the destination is reached or a node in possession of a recent path (by comparison with the sequence number included in the RREQ) is attained, a Route Replay (RREP) is forwarded towards the source to establish the transmission path. Subsequently, a path is maintained as long as it is used.

In case of path failure, the source is informed about this event so that the search for the destination can be restarted. Fig 5 is an example of the path maintenance process. When the node L detects that the destination D has moved out of reach, it announces the route failure which is forwarded to the source. Then, S starts the search from the beginning.

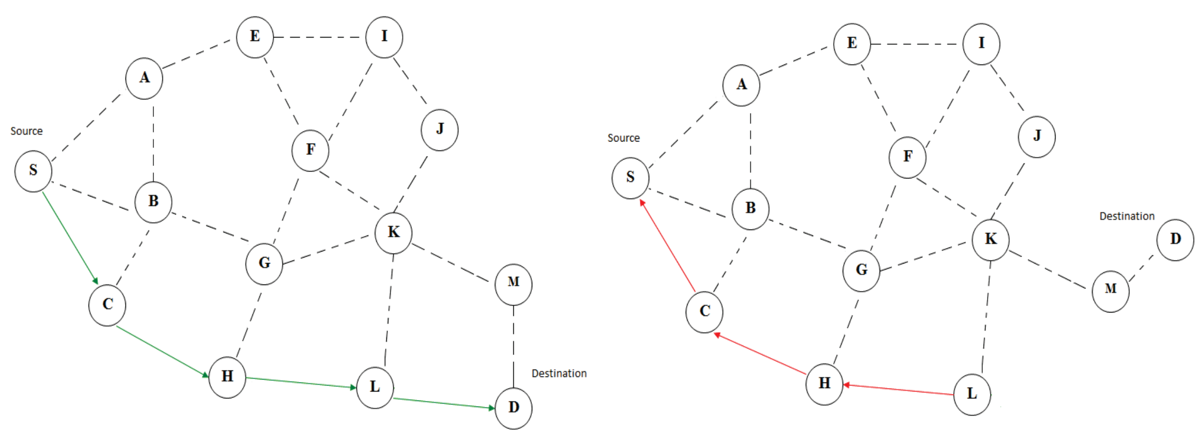

Fig 5. The path maintenance process.

b) Dynamic Source Routing (DSR) applies a similar path construction process to the one applied in AODV (Johnson et al., 2007). During the path discovery, the difference is that the intermediate nodes include their IDs in the RREQ instead of memorizing the reverse path. Moreover, when the path is sent back to the source, the found path is included in all the data packets.

As a result, this protocol requires less storage capacity. On the other hand, including the path in all the RREQ / RREP and data messages introduces a considerable quantity of routing overhead. Fig 6 and 7 illustrate how the search for the destination is carried out by DSR.

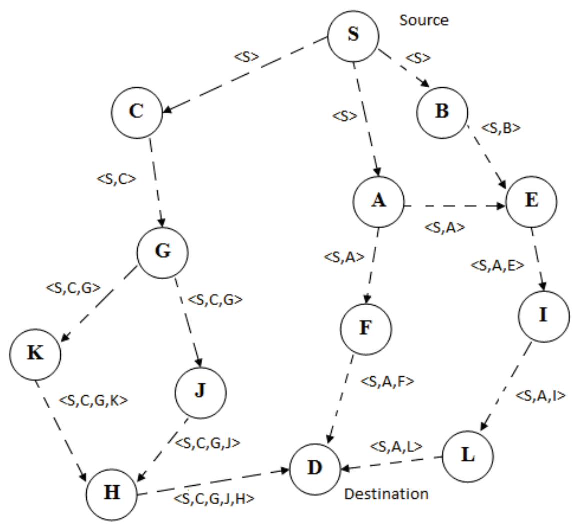

Fig 6. Dissemination of route requests.

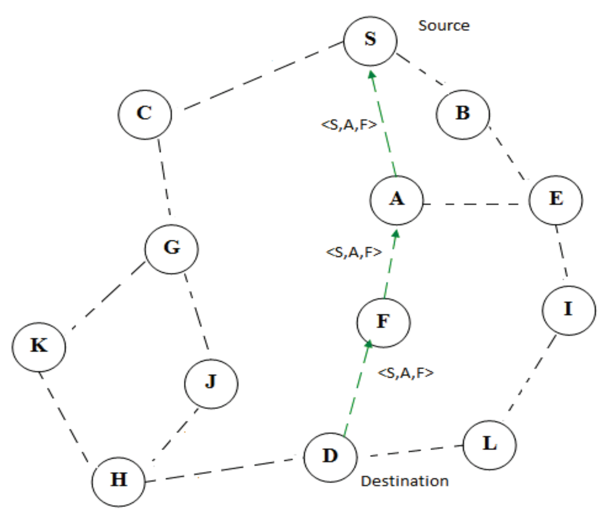

Fig 7. Route reply.

c) Instead of maintaining only one path towards the destination, Ad hoc On demand Multipath Distance Vector (AOMDV) proposed by Marina and Das (2006), and Temporally Ordered Routing Algorithm (TORA) developed by Park and Corson (1997) maintain all the found paths during the search for the destination. After that, in case the primary transmission path is broken, one of the alternative paths is used by the source. Unless all the alternative paths are broken, this approach can significantly shorten the transmission delay and overhead. On the other hand, creating and maintaining multiple paths to the same destination, requires extra processing.

d) Developed by Toh (1997), Associativity Based Routing (ABR) is a reactive protocol that promotes the path that has a high associability degree. To run this protocol, each member of the network attributes an associability degree to its neighbours by counting the number of the HELLO messages received. During the search for the destination, the associability degree of the explored links are included in the request messages. Subsequently, when the destination is reached, the path with the highest associability degree is selected. Also, this protocol incorporates a local path repair mechanism to reduce the transmission delay in case of a path failure. Although the local path repair is effective when the nodes are moving at a medium / low pace, this method increases the transmission delay when it fails.

e) Signal Stability-Based Adaptive Routing (SSA), presented in the work of Dube et al. (1997), is a reactive protocol that selects the paths based on the signal strength between the neighbours. First, for each neighbour, the link signal quality is recorded as “high” or “low”. Then, when the search for the destination is launched, only the path requests received from the “high” signal strength links, are forwarded. As a result, this protocol creates paths formed by “high” quality links. The disadvantage of this method is that, if the first attempt to build a path with “high” quality links fails, the transmission source then restarts the search and the “low” quality links are accepted during the second attempt. Thus, the transmission delay is increased.

f) Ko and Vaidya (2000) proposed another protocol called Location Aided Routing (LAR) that collects the geographical coordinates of the mobile hosts. This reactive protocol explores the existing geographical information about the destination (last known location, time elapsed since the last communication, speed and direction of movement) to near down the location where the destination might be found. First, the source calculates the relative distance that separates it from the destination and includes it in the request message. Afterwards, the intermediate nodes receiving the request message will rebroadcast it only if they are closer to the destination. In order to enlarge the search area, the distance separating the destination from the source, is increased at the beginning of the process. Consequently, the probability of finding the destination, is increased. However, supposing that the next hop leading to the destination is the closest one to it (geographically), can be false. As a result, the path repair mechanism can falsely fail which leads to restarting the search for the destination and including the entire network in the process. Hence, if the path repair fails, the transmission delay is increased.

Table 2. Comparison between the reactive protocols.

Protocol	Advantages	Inconveniences
AODV	• Avoids routing loops by the use of the destination sequence number. • Low complexity.	• When a path in current use is broken, the source is forced to restart the search for the destination from the beginning, which increases the transmission delay.
DSR	• Avoids routing loops by including the entire path in the data messages and route request / reply. • Requires less storage capacity / routing operations.	• Generates a high quantity of overhead especially in large networks.
AOMDV / TORA	• Creates multiple paths to the same destination for a better adaptability to topological changes. • Lowering the overhead caused by path failures.	• Maintaining alternative paths requires extra processing.
ABR	• Creates paths through the nodes that have a low sate of mobility. • Applies a local repair mechanism to reduce the transmission delay.	• When the local repair process fails, the transmission delay is augmented.
SSA	• In order to avoid packet losses, SSA builds paths based on strong links.	• When the process of building a path with “high” quality links fails, the search for the destina-

		tion is restarted to accept "low" quality links. Consequently, the transmission delay is increased.
LAR	• Through the use of the geographical information, LAR minimizes the participants in the path recovery and accelerate the process.	• Path construction based on the geographical locations does not necessarily produce the shortest path.

A comparison between the reactive protocols discussed in this section is presented in Table 2. To recapitulate, the reactive family was initiated to lower the routing overhead. However, this category of protocol cannot sense the availability of better paths when the topology changes. And most importantly, path failures elevate the overhead and the transmission delay. Most of the recent proposed reactive protocols are based on: (a) AODV because of its simplicity, (b) AOMDV because of its multipath approach, (c) DSR due to its adaptability to security features, (d) LAR due to its mobility awareness methods. Thus, in the next chapter, the abovementioned protocols are tested.

I.IV. Hybrid and hierarchical protocols

Hybrid protocols usually create multiple routing zones by applying the proactive approach. Then, the reactive approach is initiated to reach the remote destinations.

a) Proposed by Hass et al. (2002), Zone Routing Protocol (ZRP) applies this idea to create a routing zone around each node as shown in Fig 8. During the reactive path building phase, if the destination is within the routing zone, a path replay is sent back to the source. Hence, the delay introduced by the path discovery process is reduced by comparison with the classic reactive approach. Nevertheless, since the search for the destination is directed through routing zones, the path produced by this protocol is not always the shortest.

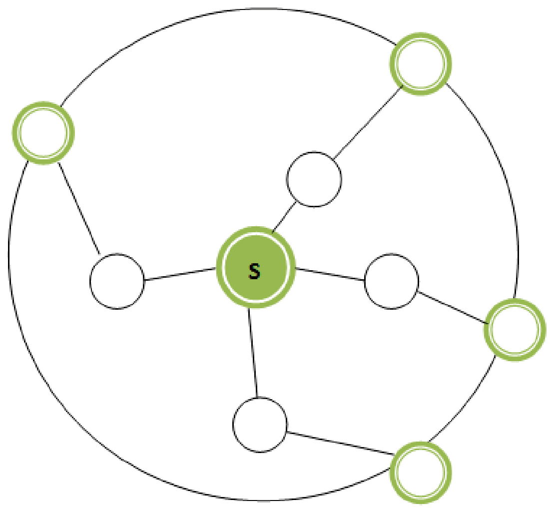

Fig 8. Zone Routing Protocol.

b) Joa and Lu (2002) designed Zone Based Hierarchical Link State (ZHLS) which is also a hybrid protocol that creates routing zones based on the geographical locations of the network components. Depending on the positions of the nodes, the network is arranged in a form of multiple routing zones. Furthermore, the nodes that belong to the same zone create paths towards each other and the links between the routing zones are announced in the entire network. During the reactive search for the destination, the transmission delay is reduced since the route replay is sent once the zone, to which the destination belongs to, is found. On the other hand, the geographical routing zones of this protocol cannot merge. Fig 9 is an example of how ZHLS arranges the nodes in groups based on their positions.

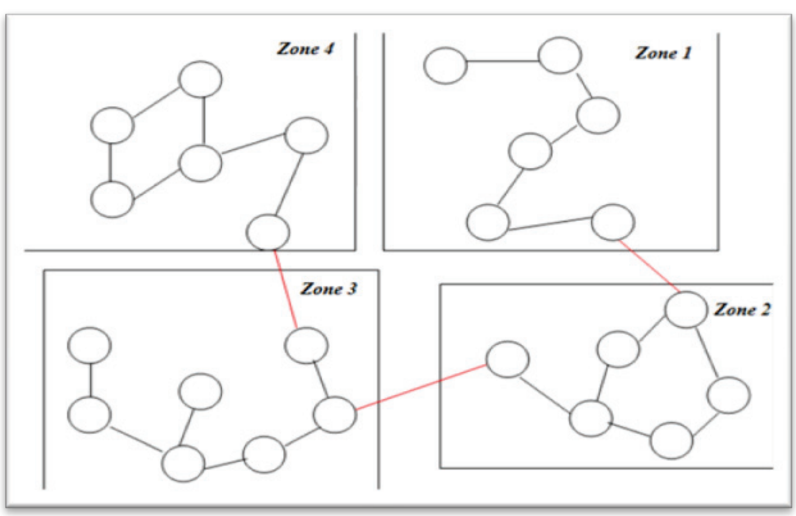

Fig 9. Zone based hierarchical link state.

Protocols that organize the network in a hierarchical manner are very similar to the hybrid protocols. The difference between these two strategies is that the hierarchical organization of the network elects a subset of the network responsible for the routing operations whereas in the hybrid protocols, all the nodes contribute uniformly. This subset of nodes can be elected based on different parameters such as the remaining energy, the nodes mobility state and the density of the topology (Sharmila and Amalanathan, 2014; Hong and Jie, 2014).

c) Pei et al. (1999) designed Hierarchical State Routing (HSR) which is a protocol that arranges the network into multiple hierarchical levels. To apply this protocol, each CH at level N becomes a member in a cluster at level N+1. Afterwards, the nodes are identified by the sequence of CHs IDs relaying them to the root cluster. Consequently, GWs have multiple IDs since they can be reached by at least two CHs.

d) In Cluster Based Routing Protocol (CBRP), the mobile hosts advertise periodically their neighbourhood table (Jiang et al., 1999). On the basis of this information, a role is assigned to each node (CH, GW or CM). Furthermore, the reactive method is applied when the destination does not belong to the same cluster. Accordingly, the route request is forwarded between clusters until the destination is located. Like DSR, this protocol includes the explored CHs addresses during the discovery process. Additionally, in case of path failure, an attempt to repair the broken segment is initiated.

Table 3. Comparison between the hybrid / hierarchical protocols.

Protocol	Advantages	Inconveniences
ZRP	• Accelerates the path building process by the use of the local routing zones.	• The path produced is not always the shortest.
ZHLS	• Accelerates the path building process by the use of geographical routing zones.	• Routing Zones cannot merge. • The path produced is not always the shortest.
HSR	• The paths to every node are pre-built with a reasonable overhead	• The produced paths are not optimal.

	(less than the proactive protocols).	
CBRP	• When the primary path is lost, this protocol applies a repair mechanism to reduce the transmission delay. • The reactive search for the destination is directed through clusters which reduces the transmission delay.	• This protocol includes the CHs and GWs explored by the request messages which increases the overhead. • Path repair can regress the protocol performance in a high speed mobility scenario.

Table 3 summarizes the characteristics of the aforementioned hierarchical and hybrid protocols. Both of the abovementioned strategies where designed to lower the overhead by comparison with the proactive protocols whilst accelerating the path discovery process during the reactive phase. On the other hand, this type of protocol does not always produce the shortest path. Particularly, for the hierarchical category, the concentration of data flows on GWs / CHs can lead to congestion. As the basic hybrid / hierarchical protocols do not offer any mechanism that can influence the QoS, we deemed unessential to test protocols from this category. For example, although tests of ZRP showed perceivable moderation of the routing overhead, the hybrid strategy of this protocol affected negatively the QoS performance: (a) routing loops, (b) high path length, (c) high packet losses ratio.

I.V. Recent advances

At first, most of the protocols tailored for MANET were oriented best effort. With the continuous development of the mobile wireless devices capacities, researchers developed routing protocols able to offer a better QoS and privacy to the mobile users. In this section we discuss the most recent development from this perspective.

I.V.1. QoS Oriented Protocols

Recently, multiple upgraded versions of the initial routing protocols, based on methods such as the Ant Colony Optimization (ACO) (Dorigo and Gambardella, 1997) and the Artificial Bee Colony (ABC) (Marjan and Hamideh, 2015), have been developed to improve the QoS. Given that the purpose of the ACO meta-heuristic is to find the best path between the nest and the food source, this approach conveys perfectly to the QoS routing problem. For example:

a) Multi-Objective AODV (MOAODV) is an extension of AODV, based on the ACO algorithm, that takes into consideration multiple metrics (delay, load and path reliability) besides hop count (Persis and Robert, 2015).

b) Likewise, AODV Ant Like Mobile Agents (AODV-ALMA) proposed by Ayyadurai and Ramasamy (2008), applies the ACO algorithm to establish a path towards the most suitable Internet gateway.

c) In the protocol proposed by Gurpreet et al. (2014)., the end to end delay is estimated based on the travelling time of the artificial ants between the source and destination. In addition to improving the end to end delay, this protocol improves the throughput by avoiding potentially congested routes.

d) Visu and Kannan (2013) designed a hierarchical protocol that applies both the ACO and the ABC. At first, the artificial bees are dispatched in HELLO messages to locate the CHs offering the largest available energy. Afterwards, the ACO method is applied frequently to compute the optimal path towards the destination CH. Besides an efficient Cluster Head selection, this protocol produced a higher throughput by comparison with cluster based AODV.

e) Santhi and Alamelu (2012) proposed a protocol based on AODV that gathers the available QoS metrics during the path discovery session. Afterwards, the destination applies fuzzy inference rules that favours paths with low end to end delay first, then maximizing the bandwidth as the subsequent priority. Then, the obtained fuzzy cost is combined with the path estimated expiration time (durability) to select the most suitable path. The results produced by this protocol showed a perceptible improvement over other fuzzy based protocols, especially in terms of end to end delay and Packet Delivery Ratio (PDR).

f) Moreover, to optimise the routing messages broadcast period, Natsheh et al. (2007) proposed an extension of AODV based on fuzzy logic. Instead of periodically advertising HELLO messages invariably, the broadcast interval is decided based on the node velocity and transmission range. Further than a smart moderation of the overhead, this method improved the PDR.

Due to its self-organised nature, the MANET performance evidently depends on the willingness of the nodes to contribute in the network traffic. From a QoS perspective, managing selfish nodes is an important issue that many researchers attempted to solve (Jebakumar et al., 2015; Janakiraman and Rajendiran, 2015; Sengathir and Manoharan, 2015). For instance:

a) Dimitris et al. (2012) proposed a reputation based mechanism that rewords / panellizes the nodes based on their cooperativeness.

b) Also, a trust estimation approach, based on Mobile Agents, is proposed by Krishnan et al. (2015). To avoid the degradation of QoS due to misbehaving nodes, the designers of this protocol use two agents to detect the irregular behaviour and packet losses. Afterwards, based on the trust estimation, the potentially malicious nodes are excluded from the transmission paths.

c) Distinctively, Record and Trust-Based Detection (RTBD) includes a novel trust estimation mechanism that takes into consideration the behaviour of the nodes during the path discovery session (Senthilkumar et al., 2014). To detect and decline the cooperation with selfish nodes, this algorithm observes the delay with which the RREQ packets are forwarded and uneven packet losses. Thus, detecting the selfish nodes unwilling to participate in the path construction phase.

I.V.2. Energy Aware Protocols

Since the MANET components are usually equipped with restricted energy resources, numerous algorithms were oriented towards the increase of the network lifetime (Thomas, 2009; Abbas and Hamid, 2012; Vivek and Charu, 2015). Fundamentally, the primal strategies to boost the network lifetime is to locate the path with the highest minimum remaining battery power (Toh, 2001) and to promote load balance (Kim et al., 2003). Consequently, avoiding the failure of the low powered nodes positioned in the shortest paths and preserving the network connectivity as long as possible (Domingo et al., 2003).

a) The protocols proposed by Hyungjik and Sunwoong (2015), and Senthil and Eswariah (2013), are extensions of AODV and AOMDV that select paths based on the remaining resource of the intermediate nodes.

b) Besides improving the network lifetime by considering the intermediate nodes power level, Power Control AODV (PCAODV) regulates the transmission power, for data and RREQ packets, instead of continuously using maximum transmission power (Madhanmohan and Selvakumar, 2012). Other than a natural increase of the network survivability, regulating transmission power reduces the probability of transmissions conflicts / interferences. Consequently, the tests of PCAODV showed significant improvement over the standard AODV in terms of PDR and end to end delay.

c) In addition to improving the QoS, the protocols developed by Poonkuzhali et al. (2014) and Haidar et al. (2014) improve the network lifetime by promoting load balance. Same, through a parallel routing approach, Load Balancing AOMDV (LBAOMDV) developed by Alghamdi (2015), and Parallel Routing Protocol (PRP) proposed by Day et al. (2011), distribute the data flow on disjoined paths to increase the throughput and, eventually promote the even distribution of power level. However, parallel routing introduces the complex packet re-ordering issue.

I.V.3. Mobility Aware Protocols

Because of the unpredictable nature of the MANET topology, path failures caused by the nodes movements influence the routing protocols performances significantly (Nassef, 2010). Hence, many routing algorithms were designed to construct stable paths (Ali and Abdallah, 2015).

a) In order to limit the probability of path failures, Mobility Adjustment Routing (MAR) selects a path based on the mobility state of the intermediate nodes (Alagiri et al., 2014). In addition to the election of the GWs and CHs based on the remaining residual energy and mobility state, the maximum mobility state of the intermediate nodes is forwarded during the path building process. Thus, enabling the selection of the path with the highest stability as well as dismissing paths formed by nodes with low battery power.

b) For the same purposes, Nihad and Mustafa (2015) developed Fuzzy AODV that combines the residual energy and the relative mobility to compute a stability value during the path discovery phase.

c) Proposed by Basarkod and Manvi (2014), On-demand Bandwidth and Stability based Unicast Routing (OBSUR) is based on DSR, oriented towards creating stable paths. This reactive protocol estimates continuously the relative mobility, remaining bandwidth and buffer of the neighbours. Afterwards, during path construction, the quality of the explored links are carried by DSR in order to choose the most appropriate path, from a QoS perspective. By promoting path stability, this protocol reduces the overhead caused by route failures as well.

d) Correspondingly, Adaptive Mobility aware AODV (AMAODV) is an upgraded version of AODV based on multiple metrics (relative distance, velocity and remaining buffer) to avoid packet losses caused by path breakdowns (Bisengar et al., 2012). Thus, by considering the relative mobility state of the intermediate nodes and their heterogeneous capacities, this protocol improves considerably the PDR.

e) Ali et al. (2014) proposed a variant of OLSR that takes into consideration the MPR nodes mobility state during the MPR selection process. As a result, improving the resilience of the paths disseminated through the MPRs.

To estimate the durability of a link, several protocols are based on the Link Expiration Time (LET) metric (Su et al., 2000). For a link between two nodes within range of each other, this metric is derived from: (a) the speed and direction of movement of both nodes, (b) distance.

f) Developed by Senthilkumar et al. (2009), Power aware Multiple QoS constraints Routing Protocol with Mobility Prediction (PMQRPMP) is one of the early protocols applying this concept.

g) Similarly, the protocol developed by Prabha and Ramaraj (2015) is an extension of AOMDV, based on the BAT meta-heuristic (Maghrebi et al., 2014), that estimates the availability of the link rather than just the relative mobility. Besides achieving load balance, BAT AOMDV gives a better estimation of the path durability to limit packet losses. Like to the ACO approach, the BAT meta-heuristic is inspired from the natural echolocation behaviour of bats.

I.V.4. Security Oriented Protocols

While the collaborative nature of the MANET is a key characteristic, it introduces the possibility of internal attacks (Yosra et al., 2014; Alnabhan et al., 2014). For example, malicious nodes can advertise false short paths to intercept the communication or, attempt to intrude on a transmission by broadcasting the RREQ with a higher transmission power to insure being a part of the shortest path. Thus, from a security perspective, multiple studies were directed to provide a better privacy for the MANET users (Pravin and Geethanjali, 2014; Gundala and Saravanan, 2014; Marjan, 2014).

a) For instance, in order to limit the probability of the message being captured or reconstituted, the protocol proposed by Madhusudan and Vinod (2012) creates parallel disjoined paths and forwards the encrypted segments of the message through them.

b) Also, assuming that the most logical way to intercept the data, is for the attacker to locate and position itself in the shortest path, Randomized Reverse AODV (RRAODV) lowers the probability of the data being captured by choosing a random path to transport the packets (Santhi and Parvathavarthini, 2014).

c) Moreover, the algorithm developed by Pattanayak and Rath (2014) is based on Mobile Agents. In this approach, a Mobile Agent is deployed in each cluster to register and monitor the transmissions. Then, in case an inconsistency is detected by a Mobile Agent, the CH is informed about the irregular behaviour and the corresponding node is blocked.

d) For the same purpose, Novel Unique Node Based Clustering (NUNBC) applies a clustering approach and assigns a location based key to each node (Vedhavathy and Manikandan, 2015). Subsequently, a node is able to communicate only if, the key that it provides, is consistent.

e) A variant of DSR based on Round Trip Time (RTT), is proposed by Shams et al. (2013) to detect wormhole attacks. During the search for the destination, this protocol records the RREQ receiving time in each intermediate node. Afterwards, once the destination is reached, the intermediate nodes forward the RREP after including the RREQ and RREP receiving time. Subsequently, the source is able to compute the processing time and RTT for each intermediate node. Another distinctive feature for this protocol, is the control by the intermediate nodes as they operate in promiscuous mode in order to monitor the RREP / RREQ forwarded by the adjacent nodes. Therefore, enabling them to detect and report the attempt to falsify the timestamps included in the routing packets. Finally, the source is able to detect the presence of a wormhole in case the RTT is higher than the defined threshold.

Although other algorithms based on RTT measurement were proposed to solve the same problem (Tran et al., 2007), this algorithm supports muliti-rate transmissions while considering the heterogeneous capacities of the nodes and the various wireless conditions, unlike the previous solutions. Consequently, avoiding the wrongful detection of a congestion point as a wormhole.

Table 4. Summary of the recently developed routing protocols.

Protocol	Category	Orientation	Defining Features
MOAODV	Reactive	QoS insurance	• Multi-criteria path selection. • Path discovery based on the ACO meta-heuristic.
ANTALG	Reactive	QoS insurance	• End to end delay estimation.
RTBD	Reactive	QoS insurance	• Nodes cooperativeness analysis during the path construction process.
PHAODV	Reactive	Energy aware	• Interference avoidance. • Taking into consideration the battery status and the heterogeneous capacities for the nodes during the path selection.
LBAOMDV	Reactive / Multi-Path	Energy aware / QoS insurance	• Load balance.
MAR	Hierarchical	Mobility aware / Energy aware	• Path stability and node failure avoidance.
OBSUR	Reactive	Mobility aware / QoS insurance	• Path stability. • QoS constraint grantee.
BAT-AOMDV	Reactive / Multi-Path	Mobility aware / QoS insurance	• Path availability estimation during the discovery process. • Path selection based on multiple metric.
RRAODV	Reactive	Security insurance	• Message interception avoidance.
DSR-RTT	Reactive	Security insurance	• Wormhole detection. • Intermediate nodes control to avoid RTT forgery.

Table 4 summarizes the most unique recent protocols discussed in this section. It’s apparent that the reactive protocols are the most commonly extended nowadays. Especially AODV and AOMDV due to their simplicity. Generally, depending on the orientation / purpose of the protocol, the parameters (speed of movement, signal strength, remaining bandwidth, etc...) are carried in the RREQs / RREPs and the path with highest quality is selected accordingly. On the other hand, this process captures the quality of the path during path discovery only which can randomly change afterwards.

I.VI. Conclusion

Table 5 summarizes the characteristics of the MANET routing protocols categories from a theoretical point of view.

Table 5. Comparison between the categories of the MANET routing protocols.

Category	Advantages	Inconveniences
Proactive	• Complete knowledge of the topology. • Low transmission delay. • Continuous path optimization.	• High overhead / complexity. • The convergence depends on the broadcast frequency of the routing messages.
Reactive	• Low complexity. • Quick to react to the changes affecting the transmission paths.	• Overhead dependable on the nodes movement and the number of transmission demands. • Ignorance of the newly formed paths
Hybrid	• Medium complexity. • Speeding up the path discovery process.	• The overhead depends the density of the topology mainly. • Doesn't guarantee the optimal paths.
Hierarchical	• Medium complexity. • Speeding up the path discovery process.	• Doesn't guarantee the optimal paths. • The concentration of the data traffic and routing operation on the CHs / GWs can exhaust their resources or cause network congestion.

Based on the usage scenario, each routing strategy performs differently. While the reactive protocols are very effective when the transmission requests are occasional, the proactive protocols are more appropriate in collaborative scenarios. Optimizing a path based on a any criteria introduces extra processing and overhead. Hybrid and hierarchical protocols are intended to limit the overhead / routing complexity in large networks but, do not necessarily produce the shortest path. Hence, although many contributions have been proposed lately, the performance of the protocols remain situational.

In the next chapter, we present an extensive evaluation of several protocols. By testing the protocols in various conditions, we are able to extract the adaptability of the protocols. Thus, enabling us to select the most appropriate protocols for the rest of this study.

Chapter II: Comparison of the MANET Routing Protocols

II.I. Introduction

To compare the performances of the routing strategies in various scenarios (Table 6), multiple series of simulations were setup and executed with Network Simulator 2 (NS 2) (1995). The comparison of the protocols is based on the QoS parameters: (a) the routing total overhead, (b) PDR, (c) average path length, (d) average end to end delay, (e) throughput. The choices of the tested protocols were made in order to analyse strategies with different features. The main objective for these tests is to examine how the QoS performance factors are influenced when the experiment parameters are changed. It's also important to mention that in order to mimic a realistic scenario, each Constant Bit Rate (CBR) application is launched at a random time rather than from the beginning of the simulation. Thus, avoiding unfairness when comparing reactive protocols with proactive ones. Because, the proactive method takes some time to establish connectivity. Accordingly, lunching the CBR application from the start will most likely result in massive packet losses for the proactive protocols.

Table 6. Simulation parameters.

Parameters	Network growth	Range increase	Transmission connections	Nodes velocity
Network size	**50-100**	50	100	80
Transmission range	250m	**200m - 300m**	250m	250m
Connections	10	10	**2 - 20**	10
Max. speed	4m/s	4m/s	4m/s	**2m/s-16m/s**
Connection type	CBR			
Packet rate	6			
Max. Packet size	512 byte			
Simulation space	1200m x 1200m			
Simulation time	200 s			
Mobility Model	Random Waypoint			
Pause time	2 s			
MAC layer protocol	IEEE 802.11			

II.II. Results and Discussion

Through this section, it's important to emphasise that the tests were conducted in various simulation conditions / scenarios to verify which protocol is the most suitable for each situation. And most importantly, to conclude on which protocol that is the most versatile in various situations.

II.II.1 The network growth

In the first experiment, the network was gradually expanded. To operate, all MANET routing protocols advertise routing messages in order to establish / preserve connectivity. Reasonably, the growth of the network implicates an increase of the routing information announced by the routing protocols (Fig 10).

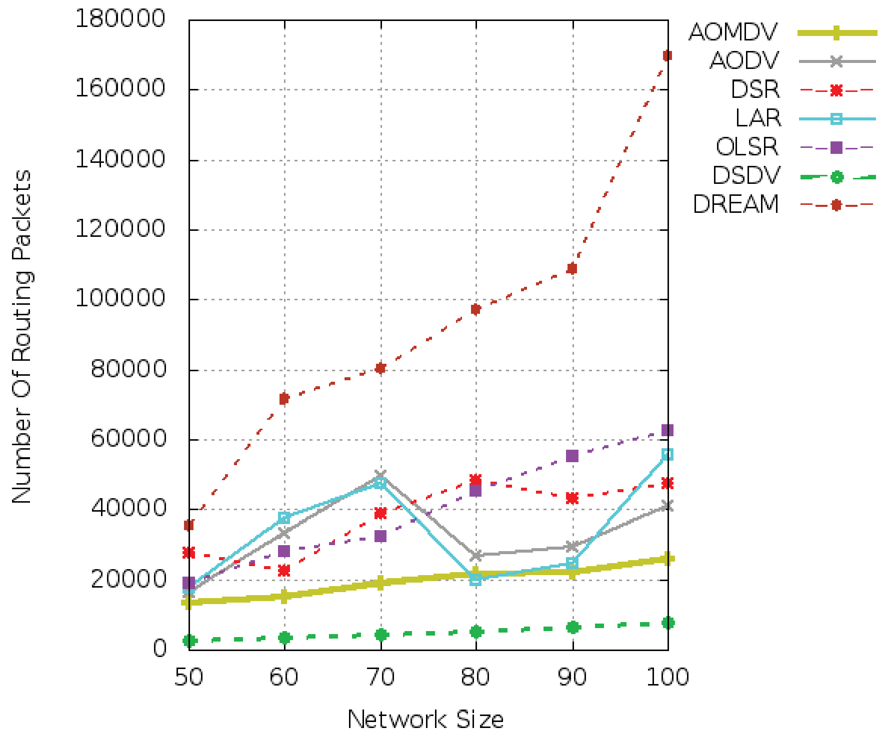

Fig 10. The recorded overhead during the network growth experiment.

Nevertheless, not all the protocols are influenced equally. Because the proactive algorithms construct / preserve a path to every attainable destination, this approach requires a frequent advertisement of the global topological view, in a form of routing messages. Accordingly, as the size of the topology and the routing messages expand, the overhead of the proactive protocols is raised extensively. Particularly, due to the repetitive broadcast of multiple routing messages (Topology Control, MPR Selector and HELLO Messages), OLSR (Francisco, 2004) generates a substantial overhead. Also, DREAM (Jeff and Wilcox, 2002) produces the highest overhead. In addition to its proactive approach, the overhead of this protocol depends on the nodes movement speed and the location of the destination is frequently forwarded to the source. Though this method strengthens the active path, it amplifies the overhead. DSDV on the other hand produces the lowest overhead by announcing the routing messages with the lowest broadcast frequency amongst the tested protocols.

Although operating reactively, DSR generates a considerable overhead because of the insertion of the active path in all the routing / data messages. Similarly to DREAM, LAR announces supplementary geographical coordinates in order to accelerate path recovery. Therefore, the overhead of DSR and LAR is augmented. While AODV and AOMDV build a path when it's necessary only, their recorded overhead is higher than the overhead produced by DSDV. This outcome is caused by the fact that the reactive method flood the entire network with RREQs to establish a path towards the destination. Consequently, to build / maintain only ten paths, the reactive protocols generate a large quantity of routing packets. Conversely, when AODV, LAR and DSR are operational, the routing messages are treated only by the members of the active paths. On the other hand, in order to update the routing table, all the received routing messages are treated by the proactive protocols. Hence, in Fig 11, the quantity of routing messages processed by the proactive protocols is significantly higher by comparison with the reactive protocols. It's noticeable that even if LAR generates as much overhead as AODV, AODV treats a larger quantity of routing packets during this experiment. While LAR generates more routing messages to update the source location table, this protocol can estimate the zone in which the destination is positioned when a path failure occurs. Consequently, whereas AODV searches for the destination in the entire network, LAR avoids overflowing the network and thus, avoids processing a large quantity of RREQs. Likewise, DSR processes a low amount of routing packets because after the establishment of the path, only the nodes forming the transmission paths process the routing messages.

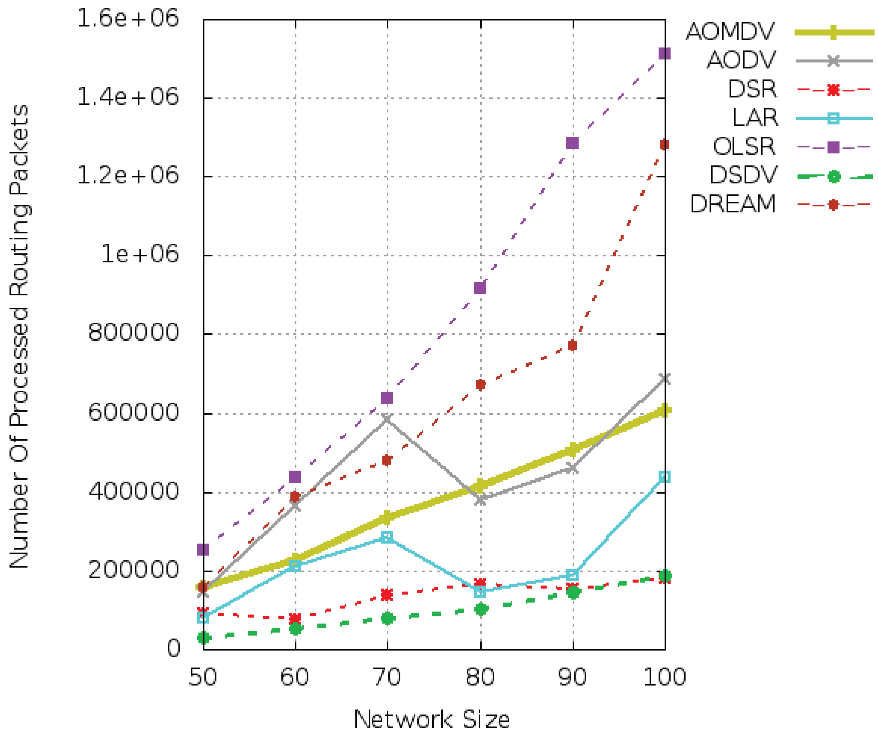

Fig 11. The processed routing information during the network growth experiment.

For similar reasons, though DREAM generates the highest overhead, OLSR treats the largest amount of routing information during this experiment. This outcome is due to the fact that running OLSR requires a periodical processing of all the received control messages. Furthermore, when DREAM is applied, although the destination announces frequently its location and the intermediate nodes retransmit it to the source, only the nodes forming the transmission route process the corresponding routing messages. Thus, limiting the quantity of routing information processed. It's also worthy to point out that in order to create a path to every reachable destination, the routing packets advertised by DSDV and the proactive protocols in general, contain a larger volume of routing information whereas the reactive protocols generate small routing packets as shown in Fig 12. Also, the routing packets advertised by DSR and LAR are larger than the packets announced by AODV and AOMDV due to the inclusion of the entire path (DSR) and the geographical location (LAR).

Unlike OLSR and DREAM, the overhead of AOMDV is small. By maintaining all the discovered disjoined paths leading to the destination, AOMDV is less affected by route failures. Though this method introduces supplementary processing, in case of a path breakdown, AOMDV relies on an alternative path instead of overflowing the network. Hence, during this test, DSDV and AOMDV produce the smallest overhead.

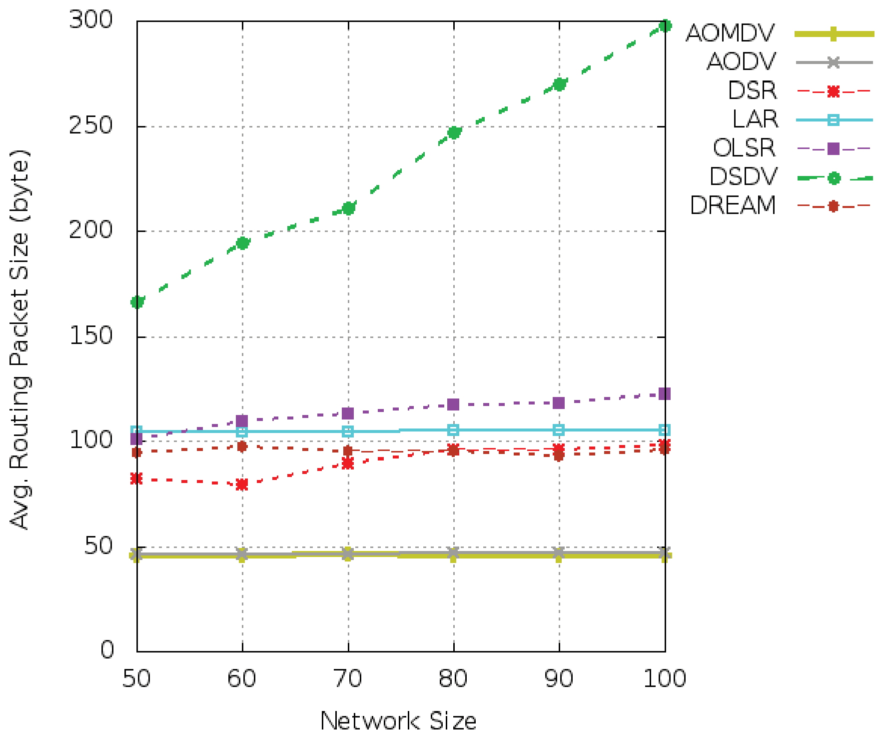

Fig 12. The average routing packets sizes during the network density experiment.

In Fig 13 and 14, the PDR and throughput recorded during this experiment, are illustrated. Due to its high sensitivity to the changes influencing the active routes, AODV reacts faster to a path failure to construct a new path. Thus, attaining the largest PDR as shown in Fig 13. Moreover, OLSR, AOMDV, LAR and DREAM also accomplish considerable PDRs. Peculiarly, the PDR of AOMDV is lower by comparison with AODV. Although AOMDV maintains disjoined backup paths towards the source, the performance of this protocol declines in terms of PDR when the transmission attempt through the alternative path, fails. On the contrary, DSDV and DSR produce lower PDRs by comparison with the remaining protocols. Even if DSDV attempts to speed up the convergence by immediately advertising the vital topological alterations, the updates describing the remote destinations are not forwarded in a timely manner. Same, as DSR relies on the explicit routing directives, this protocol is slow to react to path failures especially those that transpire at a distant point of the path. Accordingly, the PDR accomplished by both protocols, is decreased.

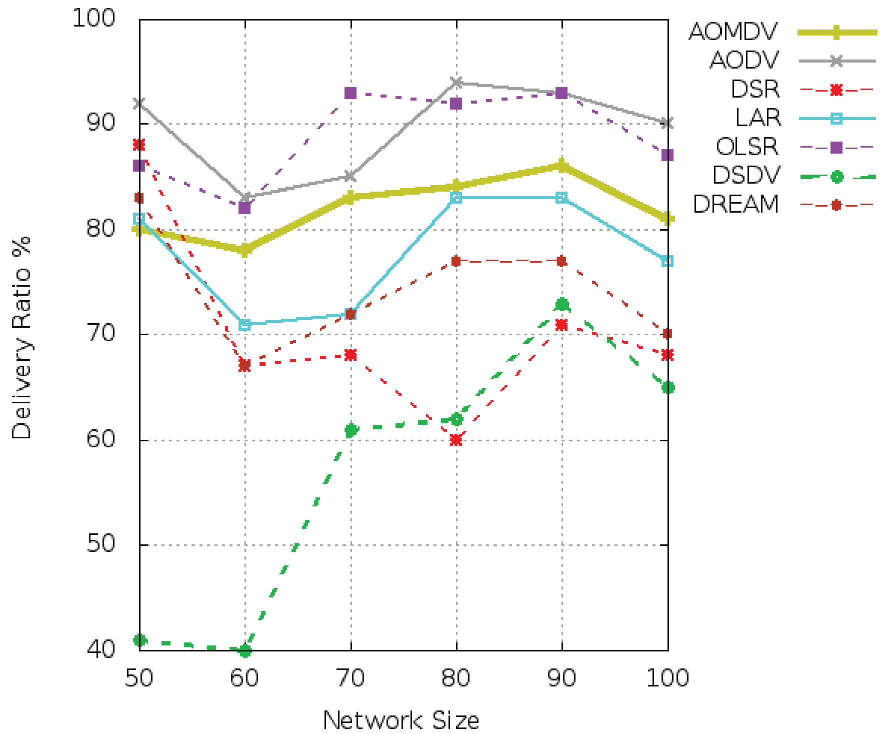

Fig 13. The recorded delivery ratio during the network growth experiment.

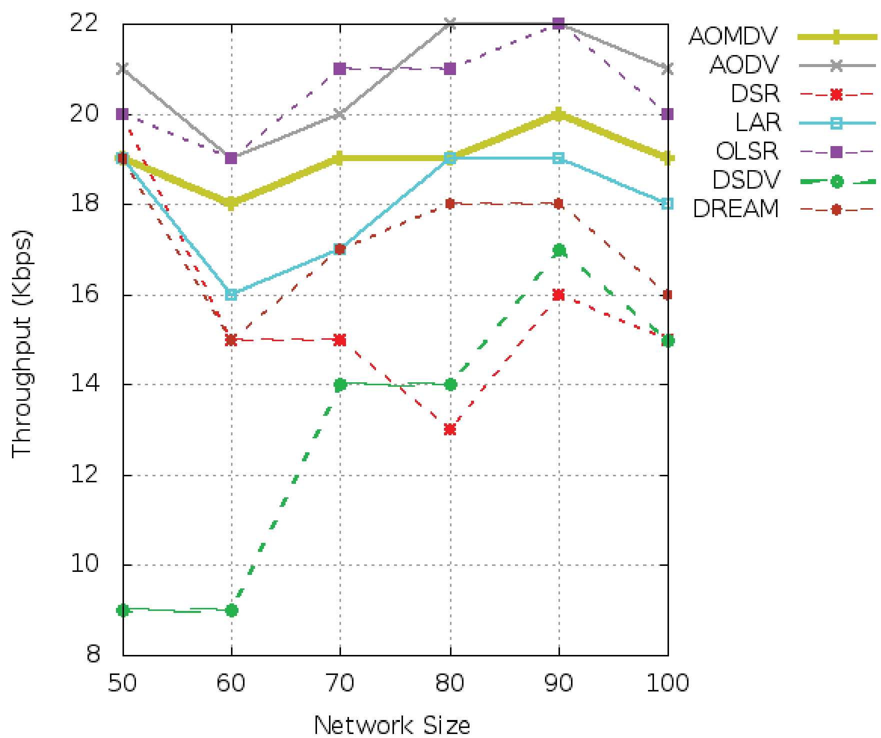

Fig 14. The recorded throughput during the network growth experiment.

From a throughput point of view (Fig 14), the network expansion renders the throughput produced by the protocols, unstable. This outcome is mainly due to the increase of the random topological events that can either make or break a transmission path. Obviously, the throughput value, which is the quantity of data successfully received by the destinations in a unit of time, is strongly related to the PDR. Consistently with the PDR values, DSDV produces the lowest throughput. Likewise, AODV achieves the highest throughput.

In Table 7, we displayed two essential QoS flow parameters obtained from simulation in order to conduct a conclusive comparison: path length (which influence the jitter) and end to end delay.

From a path length perspective (for successful deliveries only), due to its unawareness of the newly formed routes, DSR creates the longest paths during this experiment. Primarily, DSR uses the first obtainable path until it collapses. Then, in case of a path failure, the packets are salvaged by the intermediate nodes and re-routed towards the destination which increases the path length even further. LAR and DREAM also produce longer paths but for unrelated reasons. First, LAR applies a path recovery process based on the existing geographical description. Even if this approach facilitates the search for the destination, it does not produce necessarily the shortest path. Equally, the routing decisions in DREAM are based on the geographical coordinates and not hop count. Thus, both protocols achieve similar performances from an average path length point of view. Otherwise, DSDV, AOMDV and OLSR achieve the lowest average path lengths. Clearly, the performances of DSDV and OLSR are the outcomes of the proactive route updating approach. Then, while the results suggest that AOMDV built shorter paths by comparison with AODV, AODV achieves a much higher PDR than AOMDV. Meaning, that the low average path length of AOMDV during this experiment is partially caused by the failure to deliver packets to the remote nodes.

From an average end to end delay point of view, DSR generates the lowest performance. Although the packet salvaging / recovery method applied by this protocol is intended to improve the PDR, re-routing the packets increases significantly the delay when this operation is successful. Besides, the surplus caused by the inclusion of the active path in the data messages increases as well the end to end delay. It's also perceivable that the end to end delay of DREAM and LAR is slightly lower than the end to end delay of AODV. This performance is mainly due to the path maintenance / recovery based on the geographical information. Nevertheless, the end to end delay of DREAM and LAR is still a little elevated because of the slightly high average path length constructed by these two protocols. In addition to the obvious correlation between the end to end delay and path length, the delay depends also on the load of the nodes participating in the transmission. For instance, although LAR and DREAM usually achieve similar performances in terms of path length due to their related routing strategies, the end to end delay of the two protocols randomly diverge largely because of the arbitrariness of load balance in the constructed paths.

Table 7. The recorded average path length and end to end delay during the network growth experiment.

Protocols		Network Size					
		50	60	70	80	90	100
Delay (ms)	AODV	211,56	176,22	299,93	122,96	151,28	205,47
	AOMDV	54,57	52,25	48,04	40,61	28,74	45,03
	DSR	1815,33	2284,98	3019,94	2896,76	2253,51	1908,73
	LAR	221,93	131,26	239,90	73,05	45,86	101,65
	DSDV	27,14	43,07	62,57	48,36	25,46	39,45
	OLSR	40,41	73,76	35,47	70,38	48,73	89,80
	DREAM	178,98	173,14	144,23	242,22	80,74	137,25
Avg. Path Length	AODV	5,93	6,21	5,90	5,17	4,33	5,03
	AOMDV	5,78	5,73	4,57	4,64	3,84	4,78
	DSR	7,23	6,99	7,03	6,40	5,35	5,59
	LAR	5,75	6,54	6,02	4,91	4,56	5,01
	DSDV	4,21	5,13	4,48	4,04	3,41	4,22
	OLSR	5,11	5,62	4,28	4,42	3,85	4,61
	DREAM	5,89	6,28	5,79	5,33	4,94	5,32

As opposed to the previous results, the average end to end delay of DSDV is the lowest. Although this protocol loses a lot of packets due to path failures, DSDV optimises eventually the paths based on the destination sequence number. As a result, this protocol delivers less packets, mostly to the closer nodes, with a lower delay. Also, AOMDV and OLSR achieve a relatively low average end to end delay. Naturally, the performance of OLSR is reasonable due to the periodical upgrading of all the possible routes. Nevertheless, when a transmission path is broken, the end to end delay is raised. Plus, as the overhead increases due the growth of the network, the additional routing packets queuing / processing time influences negatively the end to end delay of OLSR as well as the rest of the protocols. The end to end delay of AODV is affected by: (a) the path failures, (b) ignorance of the recently available paths. Unlike the rest of the protocols, in case of a path failure, AOMDV can rely on alternative routes to preserve the connectivity with the destination. Besides, because of its low complexity / overhead, the end to end delay of this protocol is less affected by routing packets queuing / processing time. Consequently, from an end to end delay perspective, this protocol achieves a superior performance (while producing a relatively good PDR performance as opposed to DSDV).

II.II.2 The increase of the network density

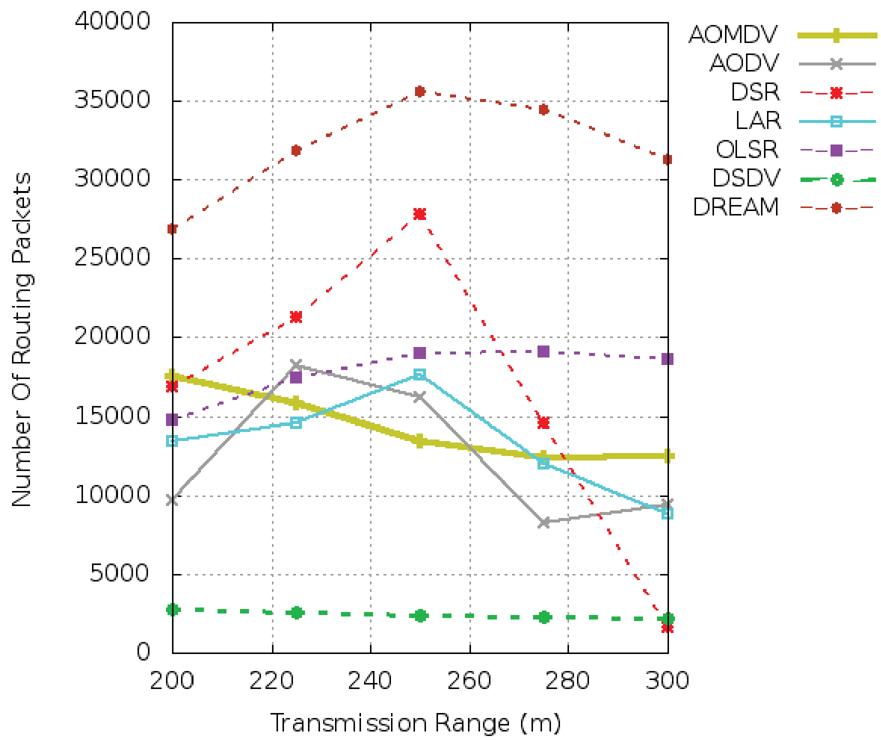

Fig 15. The recorded overhead during the network density experiment.

During this experiment, the transmission range was progressively increased in order to increase the network density. Primarily, the overhead of the protocols is affected in various ways as shown in Fig 15. Generally, the overhead of the proactive protocols is reasonably increased due to the increase of the number of connections between the nodes. On the other hand, the increase of the transmission range has the opposite effect on the reactive protocols in terms of overhead. Obviously, the increase of the transmission range boosts the strength of the routes, thus lowering the occurrences of link / path failures. Therefore, the overhead of the reactive protocols declines.

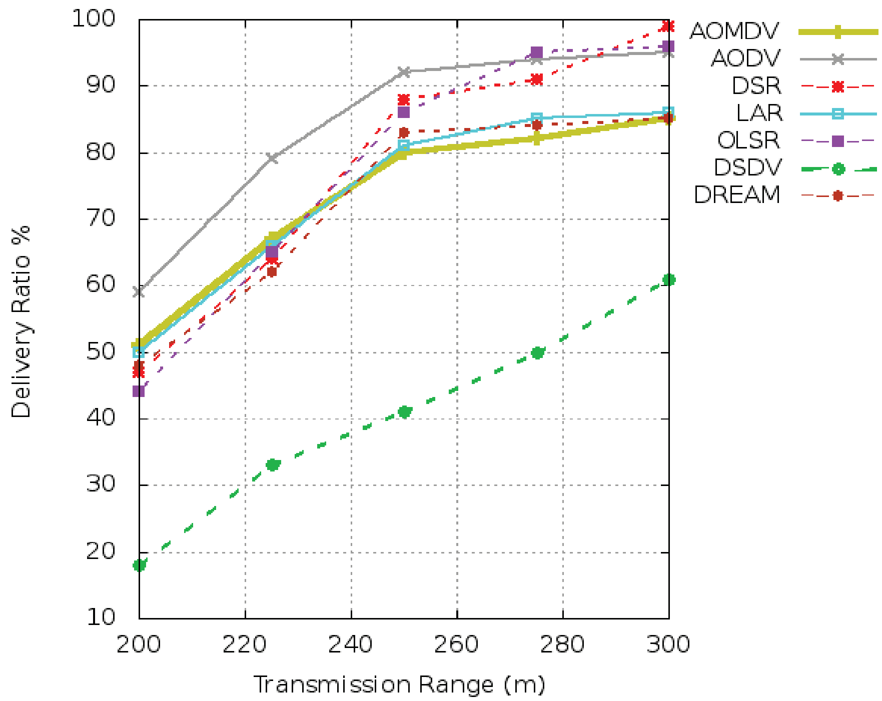

Fig 16. The recorded delivery ratio during the network density experiment.

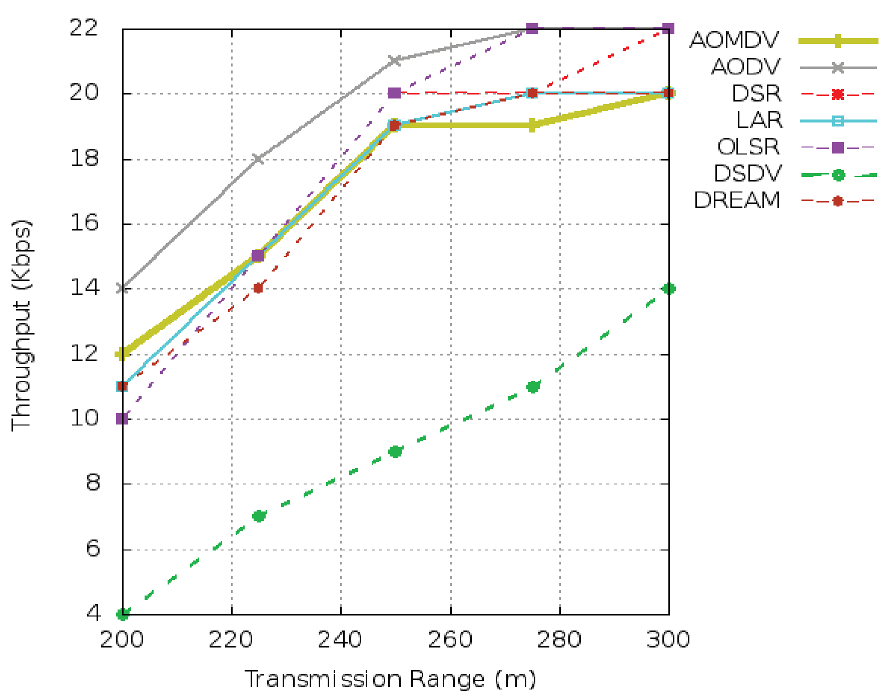

Fig 17. The recorded throughput during the network density experiment.

In Fig 16 and 17, the increase of the PDR and throughput achieved by all the protocols, are the clear outcomes of the increase of the network density. Due to the improvement of the links / paths strengths, packet losses caused by routing errors, are less frequent. Hence, the PDRs of all the protocols are noticeably raised. Likewise, from an average path length and end to end delay perspective (Table 8), all the routing protocols achieve superior performances. Due to the low average length of the paths constructed by the protocols, the packets are transferred faster between the source and destination. Thus, increasing the throughput values as well. All performance metrics considered, the performance of DSR is remarkable. As the length of the paths decreases, the overhead of DSR decreases noticeably. In addition to achieving a noteworthy PDR when the transmission paths are shorter, knowing the integrity of the path provides a significant security option for this protocol.

Table 8. The recorded average path length and end to end delay during the network density experiment.

Protocols		Transmission Range (m)				
		200	225	250	275	300
Delay (ms)	AODV	394,99	292,66	211,56	60,62	86,77
	AOMDV	146,04	101,86	54,57	35,13	36,69
	DSR	2760,79	2826,62	1815,33	1144,05	33,87
	LAR	154,56	175,90	221,93	118,42	74,74
	DSDV	31,80	34,33	27,14	25,21	23,21
	OLSR	38,99	35,44	40,41	30,89	33,51
	DREAM	159,45	73,68	178,98	53,63	37,79
Avg. Path Length	AODV	7,08	6,76	5,93	5,23	4,81
	AOMDV	7,05	6,30	5,78	4,94	4,82
	DSR	7,15	7,43	7,23	5,72	4,51
	LAR	6,52	5,99	5,75	4,95	4,65
	DSDV	4,51	4,67	4,21	4,09	3,99
	OLSR	5,61	5,51	5,11	4,67	4,26
	DREAM	6,14	6,13	5,89	5,03	4,38

II.II.3 Number of transmission connections

From an overhead perspective (Fig 18), the growth of the transmission demands affects the reactive protocols drastically. While the proactive algorithms construct the routes systematically, the reactive ones initiate a search for each connection demand independently. Added to the effect of path failures, the overhead of the reactive protocols is enlarged clearly when the number connection requests is raised. Distinctively, because of the backup routing method, AOMDV is less effected by path failures and consequently produces a lower overhead. In contrast to AOMDV, the overhead generated by DREAM and LAR is noticeably amplified. This result is due to the fact that LAR and DREAM frequently announce the position of the destination, for each connection, to update the location tables. Hence, the elevated overhead produced by these two protocols. On the other hand, the overhead of OLSR and DSDV remained stable. Basically, this result is due to the independence of the proactive path building process form the transmission demands.

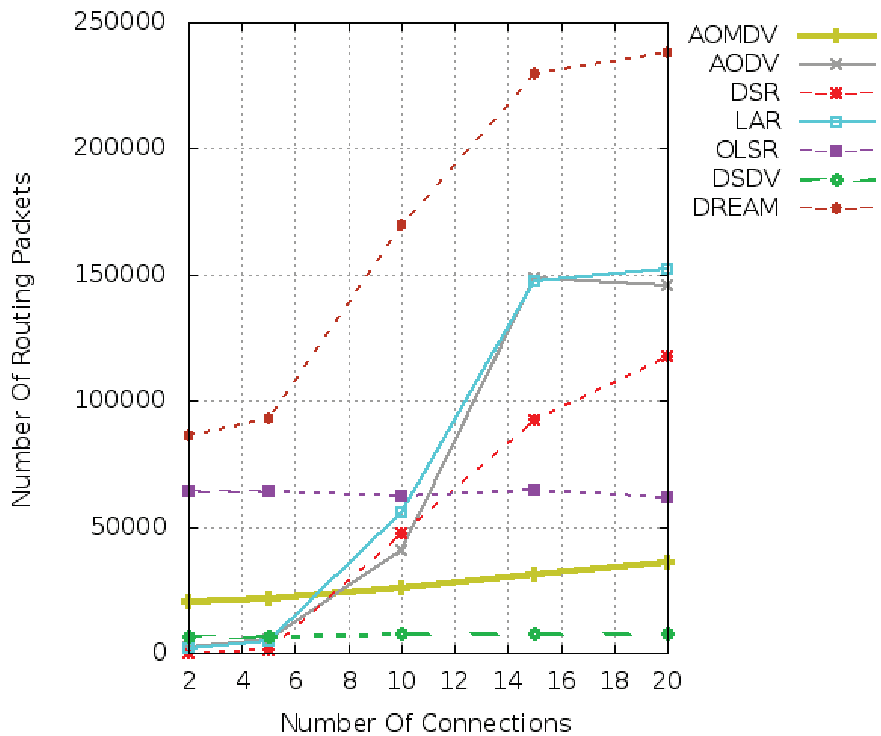

Fig 18. The recorded overhead during the transmission connections experiment.

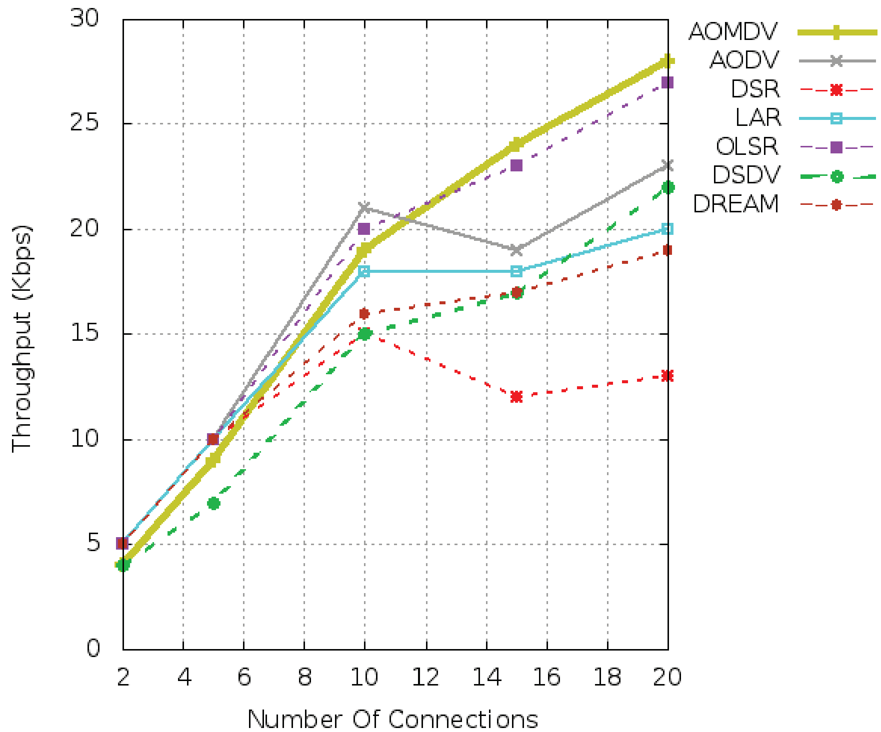

Fig 19. The recorded throughput during the transmission connections experiment.

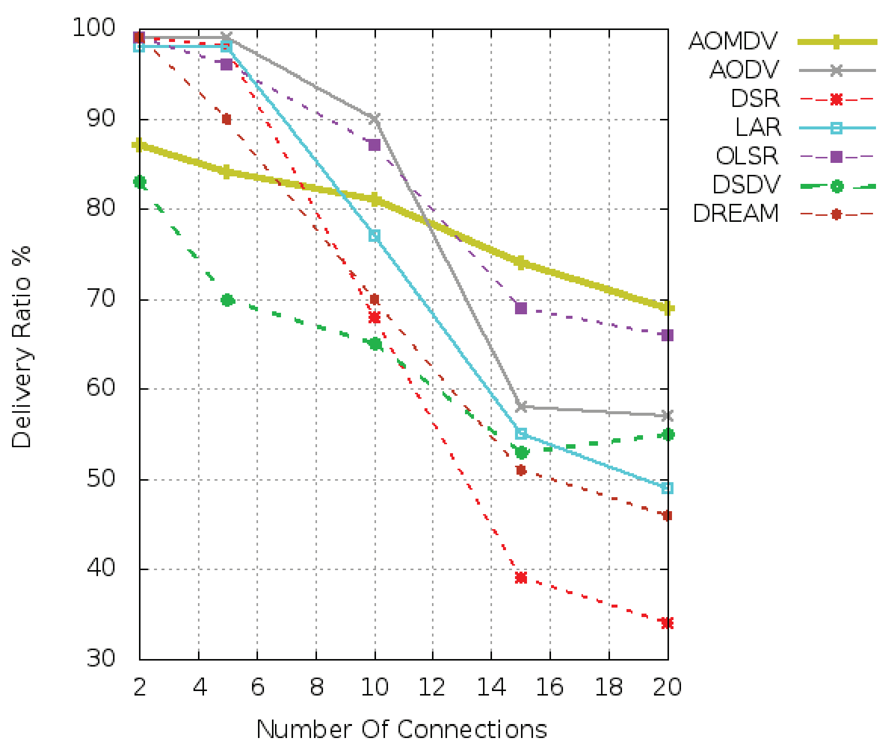

Fig 20. The recorded delivery ratio during the transmission connections experiment.

Mainly, in addition to a reasonable expansion of the throughput (Fig 19), the PDR of all the protocols decreases as shown in Fig 20. While the number of connections is raised, the quantity of packets dropped due to route failures and collisions, is enlarged respectively. Based on the assembled results, AOMDV is fairly the best performing protocol during this experiment. Through maintaining alternative paths, AOMDV is more tolerant to path failures and, unlike the other protocols, can handle multiple transmissions more efficiently. Same, through the proactive path updating approach, the PDR of OLSR is relatively less influenced by path failures during this experiment.

Table 9. The recorded end to end delay and path length during the transmission connections experiment.

Protocols		Number of Connections				
		2	5	10	15	20
Delay (ms)	AODV	55,19	87,43	205,47	501,18	458,42
	AOMDV	10,32	21,25	45,03	110,10	259,94
	DSR	10,69	26,93	1908,73	3278,08	2689,53
	LAR	9.54	17,93	101,65	209,47	261,53
	DSDV	9,94	21,63	39,45	140,63	312,33
	OLSR	11,24	29,83	89,80	543,49	782,22
	DREAM	9,56	105,29	137,25	274,05	224,92
Avg. Path Length	AODV	2,77	3,98	5,03	5,56	5,24
	AOMDV	2,70	3,65	4,78	4,82	4,71
	DSR	2,71	3,46	5,59	5,46	5,48
	LAR	2,70	3,72	5,01	5,22	5,28
	DSDV	2,65	3,22	4,22	3,96	4,18
	OLSR	2,74	3,42	4,61	4,78	4,76
	DREAM	2,74	3,61	5,32	5,20	5,39

Most importantly, from an end to end delay perspective (Table 9), the performances of the protocols deteriorate considerably. While DREAM / LAR build paths based on the geographical information, and AOMDV promotes stable connectivity by creating multiple paths, the rest of the protocols are focused on building the shortest paths. As a result, the end to end delay increase due to the concentration of traffic on the nodes involved in multiple transmissions. On the other hand, DREAM and LAR don't rely on hop count for path selection. Thus, the abovementioned protocols indirectly lighten the stress on the intermediate nodes positioned in the shortest path. Additionally, whereas path failures enlarge the delay for the other protocols, AOMDV maintains connectivity through alternative paths. As a result, lowering the occurrences of path recoveries and the delay introduced by them.

II.II.4 Node velocity

In order to verify the effect of the nodes speed of movement on the performance of the protocols, we made a series of simulations where the nodes velocity is augmented progressively. An understandable consequence of this alteration is a more frequent occurrence of path failures. According to the results in Fig 21, whereas the overhead of OLSR and AOMDV is stable, the overhead of DSR, DREAM, LAR, DSDV and AODV, is increased. This result is consistent with the theoretical descriptions of these protocols.

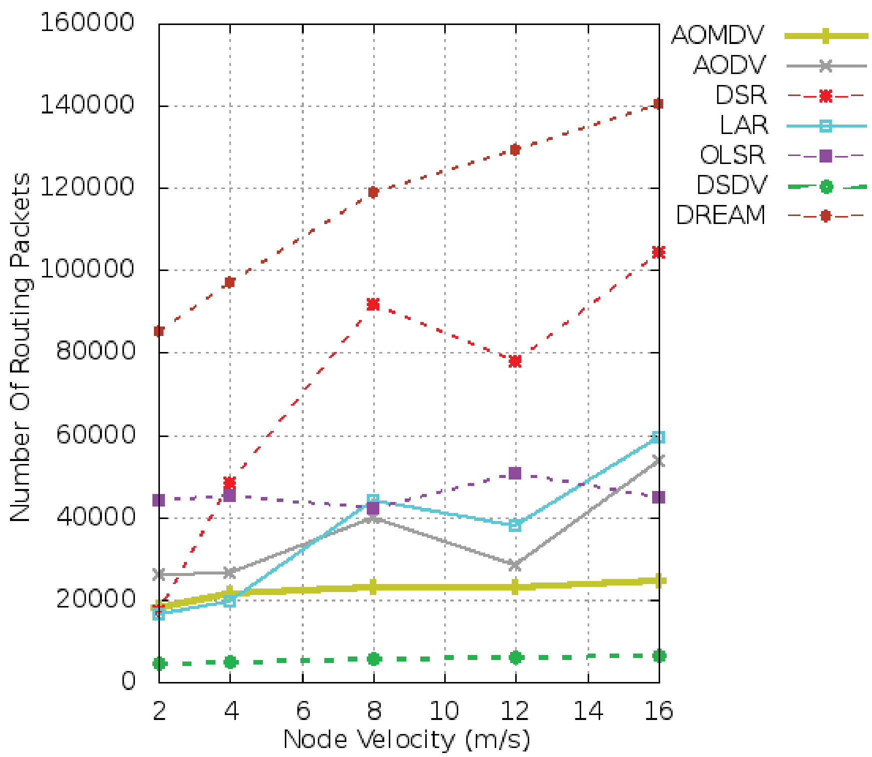

Fig 21. The recorded overhead during the nodes velocity experiment.

Due to the frequent occurrences of path ruptures, DSR and AODV must launch a new path construction process which increases their overhead. Additionally, while applying DREAM and LAR, the nodes advertise their routing messages based on their speed of movement. Hence, the recurrent topological events (link / path failures and node movement) elevate the overhead of the aforementioned protocols. Additionally, the overhead of DSDV is slightly increased since this protocol is based on an event driven method to announce the important changes. On the other hand, the overhead of OLSR is not increased at all since this protocol is based on a periodical broadcast of the routing messages. Likewise, the overhead of AOMDV is less influenced because this protocol attempts to avoid path reconstruction by maintaining the connection to the destination through alternative paths.

From a throughput and PDR point of view, the increase of the nodes speed of movement lowers the performances of most protocols as illustrated in Fig 22 and 23. Although AOMDV reduces the effect of path failures in terms of overhead, the high pace of the topological changes eventually renders the alternative paths unreliable. As a result, postponing path reconstruction while attempting to transmit the data through an alternative path, results into expanding packet losses (-29% PDR). Similarly, OLSR does not distribute the new information about the destination fast enough. Which results into routing errors and subsequently, less packets are delivered to the destination (-15% PDR). On the other hand, when path failures are frequent, AODV immediately starts the search for a new path towards the destination. As a result, although this method increases the overhead, this protocol achieves the highest PDR (-8% PDR). Furthermore, DREAM and LAR circumvent path failures by frequently advertising the location of the destinations that are moving fast. Subsequently, when a path failure is sensed, path recovery is usually less complicated. As a result, both protocols relatively reduce the effect of the nodes mobility on the ongoing transmissions (-10% PDR).

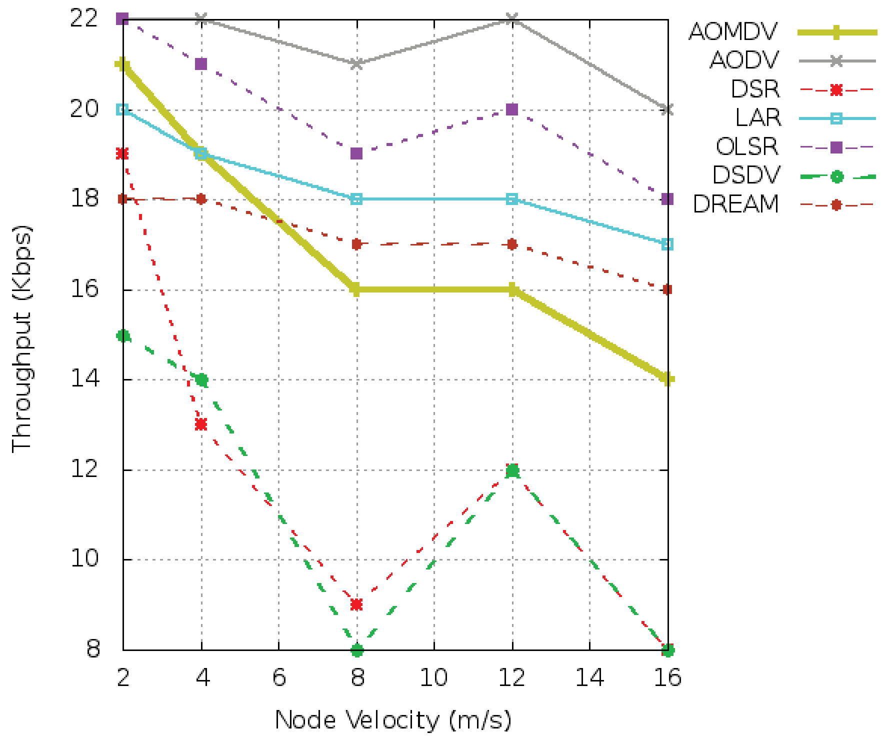

Fig 22. The recorded throughput during the nodes velocity experiment.

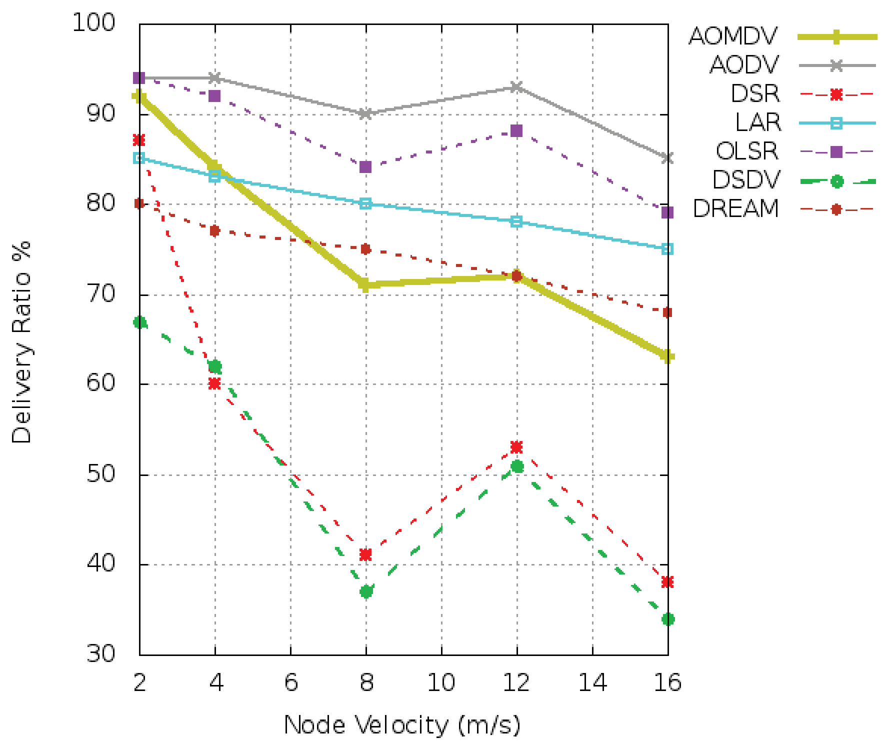

Fig 23. The recorded delivery ratio during the nodes velocity experiment.

Table 10. The recorded average path length and end to end delay during the nodes velocity experiment.

Protocols		Node Velocity (m/s)				
		2	4	8	12	16
Delay (ms)	AODV	157,79	122,96	164,81	176,90	183,87
	AOMDV	40,46	40,61	58,08	32,52	48,08
	DSR	737,18	2896,76	3930,25	3357,41	4715,28
	LAR	96,45	73,05	99,38	139,90	99,08
	DSDV	23,22	48,36	59,12	24,02	61,77
	OLSR	70,15	70,38	60,08	75,32	86,00
	DREAM	117,29	242,22	91,76	108,93	130,45
Avg. Path Length	AODV	5,56	5,17	6,00	4,75	5,60
	AOMDV	5,42	4,64	5,66	4,10	5,18
	DSR	5,72	6,40	7,05	5,49	6,39
	LAR	5,29	4,91	5,89	5,08	5,59
	DSDV	3,68	4,04	4,31	3,43	4,50
	OLSR	5,00	4,42	5,32	4,20	5,17
	DREAM	5,39	5,33	5,90	5,26	5,70

Correspondingly with the previous experiments, the results in Table 10 illustrate that the highest average path length and end to end delay are attained by DSR due to the packets salvaging / recovery attempts. Moreover, whereas AODV restarts path building once a path failure is sensed, searching in the entire network amplifies the delay gradually. The end to end delay of AOMDV is low due to the availability of alternative paths but then again, this protocol suffers a significant PDR drop (Fig 23). Conversely, LAR explores the recent geographical updates about the destination to accelerate path recovery. Which is why this protocol averages a relatively constant end to end delay while the nodes speed of movement is increased. Based on the obtained results, AODV, OLSR and LAR achieve the highest performances during this experiment.

II.III. Conclusion

In this chapter, we conducted three series of experiments in order to: (a) test the protocols in each particular scenario, (b) test the protocols overall adaptability. On the basis of originality, we selected three proactive protocols and four reactive protocols. In Fig 24, the average performances of the protocols are ranked from 0 to 10 respectively with the obtained results (10 being the best performance for the defined factor). It's important to note that the overhead score in Fig 24 takes in consideration the sizes of the routing messages as well.

Based on the recorded results, AOMDV is the most versatile protocol. The only inconvenience noticed while using this protocol, is the decline of its PDR especially during the nodes velocity experiment. This result is essentially due to the ability of AOMDV to postpone path reconstruction. While this approach is rewording from an overhead point of view, further improvement must be done in order to regulate / predict whether AOMDV must launch a path recovery or, rely on an alternative path. Moreover, though OLSR generates a massive overhead, this protocol averages a low path length with an impressive PDR. Thus in the next chapter, we first propose a simple mobility aware extension to improve AOMDV. Subsequently, a proactive protocol is implemented and tested to solve the complexity / overhead issue encountered with OLSR. Table 11 recapitulates the global performances obtained in the previous section.

Table 11. Average performances comparison.

Metric		AODV	AOMDV	DREAM	DSDV	DSR	LAR	OLSR
Total overhead	Avg. packet size (byte)	46,84	45,67	95,35	255,24	100,59	104,84	116.87
	Routing packets	36935	20663	100506	5052	43260	38503	41717
PDR		86,89%	78,37%	73,62%	53,90 %	68,46%	77,36%	84,87%
Avg. path length		5,34	4,95	5,28	4,05	5,92	5,21	4,64
Avg. delay (ms)		206,10	62,38	138,186	53,73	2205,27	127,01	112,21
Throughput (Kbps)		19,76	18,12	16,56	12,26	14,71	17,44	19,42

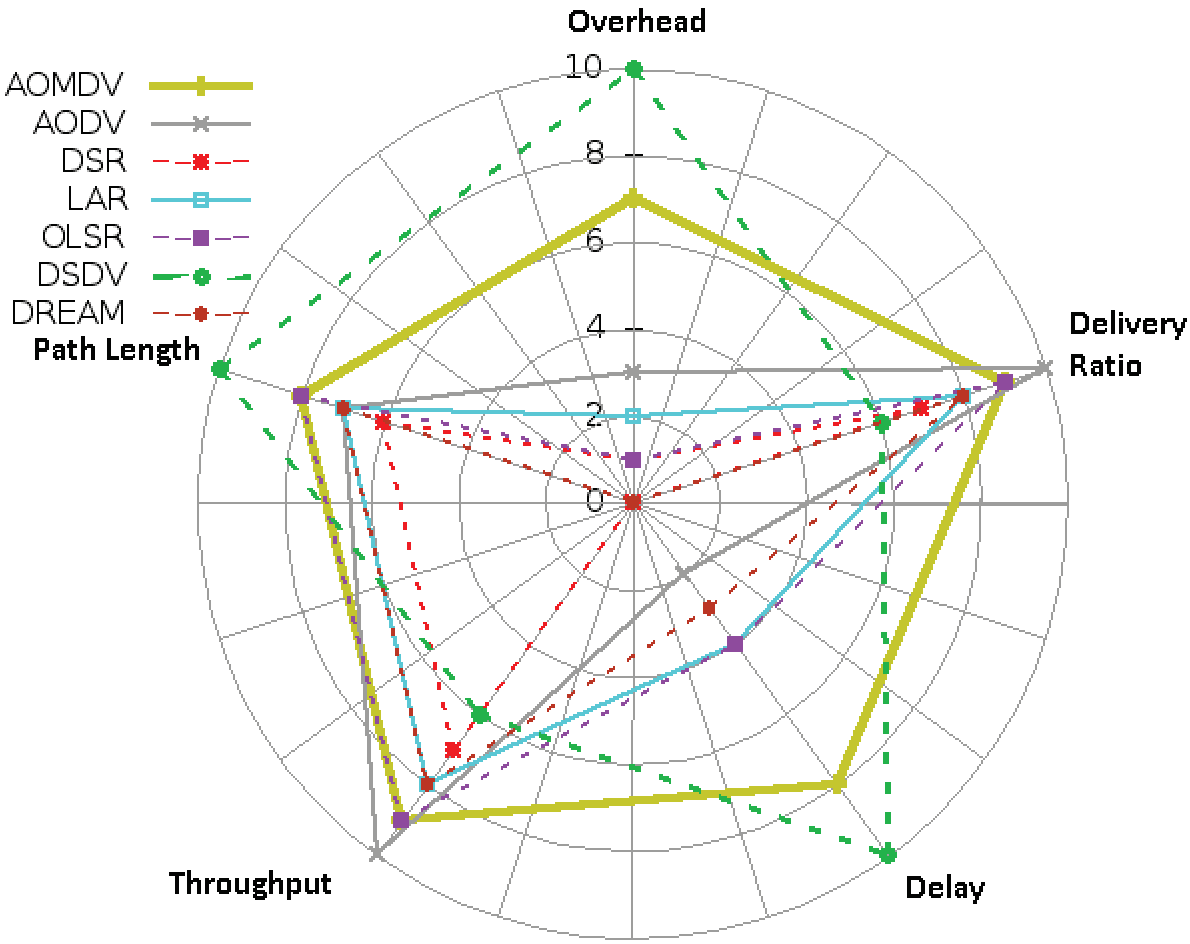

Fig 24. Overall performance ranking based on the simulation results.

Chapter III: Proposition for two Extended Protocols

III.I. Introduction

As it’s been clear in the two previous chapters, the efficiency of each type of routing protocol depends on the situation for which the MANET is used. When the transmission requests are rare, the proactive protocols generate a high quantity of overhead pointlessly. On the other hand, the reactive and hybrid protocols consume fewer resources by building paths on demand. Otherwise, the proactive protocols are more efficient when the user application requires a constant connectivity between the members of the network. Whereas the proactive protocols optimize the available paths automatically which lowers the end to end delay and the path length, the reactive protocols cannot detect a better path. For example in Fig 25, after that the primary path in green is established between node 1 and 11, unless it fails, a reactive protocol cannot detect the presence of the shorter path highlighted in blue.

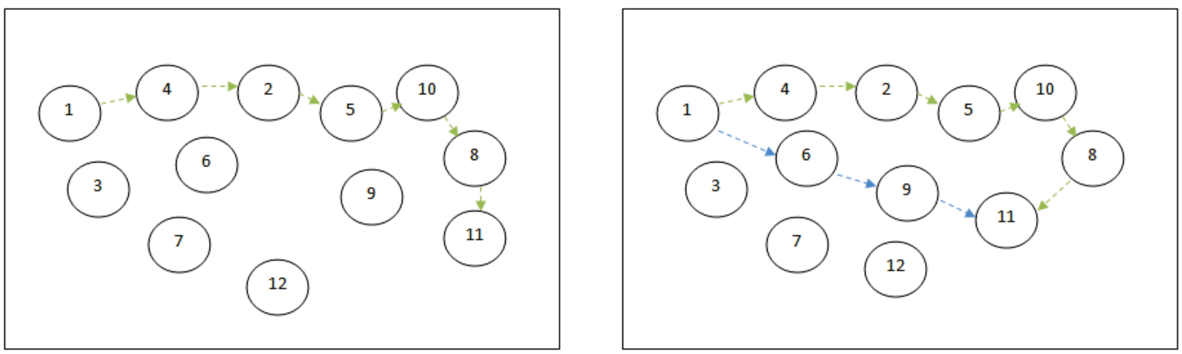

Fig 25. The unawareness of the newly formed paths.

Generally, MANETs are designed to provide connectivity for a group of mobile users. In a collaborative scenario, maintaining communication between the network members is crucial. Although this feature is insured by the proactive protocols, the high quantity of overhead and the periodic treatment of the routing information can consume a considerable amount of the network resources, especially in large / dense topologies. Hence, queuing / processing routing messages hinders the proactive protocols performance in different ways. Also, as it was perceivable in the previous chapter, the nodes mobility influences negatively the reactive protocols. Therefore, we focussed our work on how to reduce the overhead / the volume of routing information processed by the proactive strategy and limit the mobility effect on the reactive protocols.

III.II. Mobility aware extension for the reactive protocols

Though many protocols have been proposed to react better to path failures, most of the proposed algorithms are based on the nodes mobility history. This approach improves somewhat the probability of creating a stable path but in a realistic scenario where the nodes can randomly change their direction / speed of movement, this method doesn't offer much guarantee especially in teams of PDR. Several reactive extensions, labelled as pre-emptive, have been proposed to solve this problem (Boukerche and Zhang, 2004; Ramesh and Subbaiah, 2011). This approach is meant to anticipate a possible path failure to alarm the source / intermediate node beforehand. As a result the source either rebuild a path, attempt a local repair or selects an alternative path.

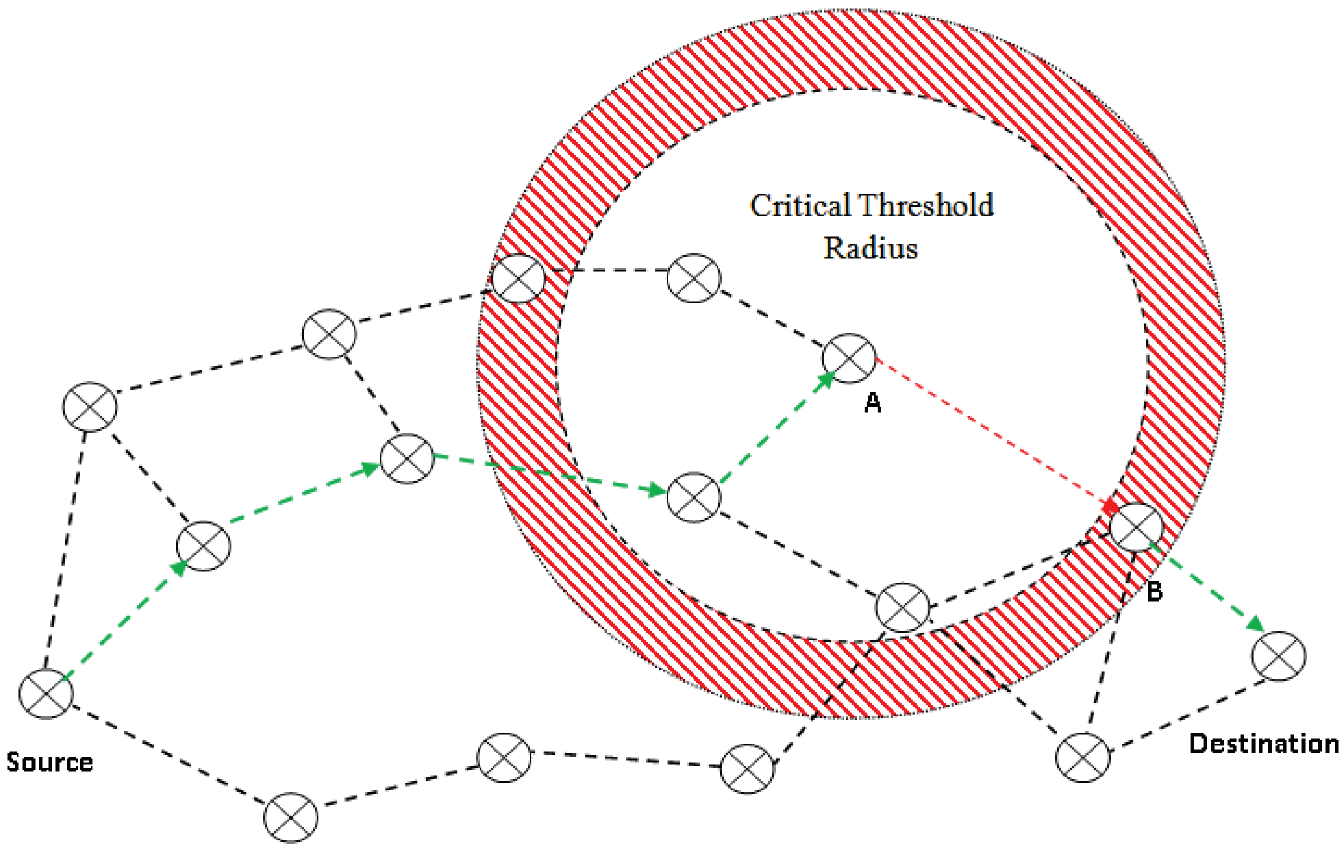

Fig 26. Path failure prediction through signal strength measurement.

```
void Processing_Received_Routing_Message(Routing_Message Received_RM){
 NodeID GeneratorIP=Received_RM.GeneratorIP();
 Neighbour_Info Gen_Neighbour=NeighbourhoodTable.getNeighbourInfo(GeneratorIP);
 boolean Direction;
 Signal_Strength=Received_RM.ReceivingSignalStrength();
 if(Gen_Neighbour==null){
     Gen_Neighbour=new Neighbour_Info(GeneratorIP, Signal_Strength, true);
     NeighbourhoodTable.addNewNeighbour(Gen_Neighbour);
     return ; // If it's a new neighbour, the process is halted
 }
 else {
     Direction= (Signal_Strength-Gen_Neighbour.PreviousSignalStrength())>=0;
     /* Direction = true <=> the two nodes are moving towards each other
      *  Direction = false <=> the two nodes are moving away*/
     NeighbourhoodTable.updateInfo(GeneratorIP, Signal_Strength, Direction);
     }
 if((Critical_Threshold > Signal_Strength) && (!Direction)){
     /* If the link signal strength with the neighbour deteriorate past the
      * Critical_Threshold, a warning message is sent to the corresponding
      * sources*/
     Vector Destination_List=new Vector();
     Destination_List
         .addAll(RoutingTable.isNextHopInActivePath(GeneratorIP));
     Destination_List
         .addAll(RoutingTable.isNextHopInAlternativePath(GeneratorIP));
     /* Destination_List is the list of transmissions that the corresponding
      * neighbour is involved in.*/
     if(Destination_List.isEmpty()) return;
     /* If the neighbour is not involved in any tramissions, the process is
      * halted*/
     NodeID Destination;
     for(int i=0;i<Destination_List.size(); i++){
         Destination=(NodeID)Destination_List.elementAt(i);
         SendWarning(Destination,Signal_Strength);
     }
 }
}
```

Fig 27. Mobility aware algorithm to predict imminent path failures.

Inspired from the previous pre-emptive protocols, we propose an algorithm that can be implemented as an extension for a reactive protocol (AOMDV preferably). First, after that the primal routing approach is applied to search for the destination, regardless of the path quality metric, the Minimum Signal Strength (MSS) of each discovered path is collected and registered at the source. Afterwards, through the periodic HELLO messages broadcast, the intermediate nodes of the active and alternative paths alike, measure the signal strength of the downstream nodes to detect the decline of signal strength (i.e. Fig 26). Eventually, an intermediate node that detects that a downstream node has moved beyond the Critical Threshold (CT) warns the source about the possibility of a link failure. As a result, the source can choose to launch a path recovery or select a substitute path with a solid remaining MSS. Reasonably, this method introduces additional processing / control overhead. However, the slight raise of the overhead comes with a possibly significant profit in terms of PDR. Particularly, in the case of AOMDV, the alternative paths are better managed and the probability of using a deteriorating backup route, is narrowed.

The algorithm presented in Fig 27 shows how the received HELLO messages are processed to apply the pre-emptive approach. Based on the variation of the signal strength, the algorithm detects when two nodes are moving away from each other. Subsequently, when the signal quality deteriorates past the defined CT, the sources are warned about the possible imminent link breakage. For simplicity sake, the algorithm that we propose here deals only with the possibility of link failure. It's possible to extend this method to administer the path discovery process. For instance, while applying AOMDV, instead of forwarding all the RREQs received from different links, packets received from links that are on the verge of failing are dropped accordingly. As a result, limiting the overhead / maintenance cost.

III.III. Lowering the overhead and complexity of the proactive protocols

One of the commune methods applied by all the proactive protocols is the frequent advertisement of the topology state in a form of routing messages. This process provides the routing protocol with essential information to create and update the topological view. However, over the long term, periodically performing this task consumes a considerable portion of the nodes energy. Considering that the MANET is usually composed of embedded devices, the routing information advertised by the proactive protocols must be reduced to preserve the nodes resources.

For instance, OLSR applies the MPR selection method to limit the dissemination of control messages. In this protocol, only the links between a node and the selected MPRs, are disseminated in the network. In FSR, a network member advertises the description of the close neighbours more often than the description of the destinations faraway. Although both strategies are meant to limit the overhead, all the proactive protocols broadcast the full routing table after a period of time. This process influences negatively the nodes resources especially in a large network.

The second commune characteristic between all the proactive protocols is the periodic treatment of the received control messages. Even though OLSR reduces the number of the announced links, the network members must process all the received routing messages. This routine task can drain a piece of the node energy especially when it's located in a dense area. In Fig 28 for example, MH1 is located in a dense area of the network. Hence, treating all the routing information received from the neighbours becomes more complex. Other than the direct impact on the necessary computation time and the energy to process the routing information, this can also have an undesirable influence on the network performance in other ways (end to end delay and jitter).

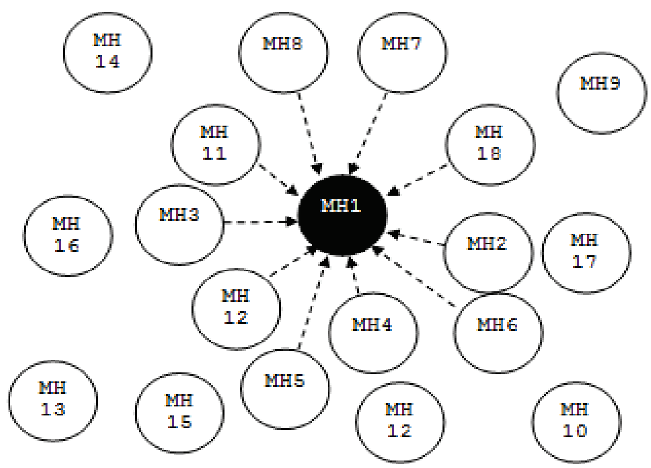

Fig 28. The network density effect on the proactive protocol.

Since the large volume of overhead and the complexity of routing information processing are the main drawbacks of the proactive strategy, we focused our work on how to reduce these negative aspects. First, the large volume of overhead is mainly triggered by the broadcast of the entire routing table. Our idea is to reduce the occurrence of this event by advertising routing updates. Instead of advertising the full routing table after a constant period of time, a node broadcasts the full routing table only when at least one new essential neighbour is detected. Generally, other proactive protocols broadcast the full routing table after a predefined period to insure that all the possible paths are announced in the network and to avoid routing loops. The algorithm we propose in this work creates and maintains a path to every possible destination without regularly broadcasting the full routing table. As a result, this method generates less overhead and smaller routing messages.

Secondly, we include the MPR selection method to reduce the number of routing messages treated in the network. Instead of processing all the received information like in OLSR, only the messages received from the selected MPRs, are processed. To achieve this, the nodes examine the source ID field of the received routing message. After that, the message is fully processed only if it's received from a selected MPR. Otherwise, the message is ignored. Consequently, the routing operations are executed with a lower computational time. Then, the MPR selection method and the manner with which the nodes share their topology information, are combined so that the full routing table is advertised only when a new MPR is detected.

This feature is what sets apart our proposition from other variant of protocols aimed to lower the overhead. For instance, to lower the overhead of OLSR, the variant proposed by Abdelali et al. (2015) incorporates an MPR selection method to limit the number of TC messages advertised necessary for the MPR selection process. However, to the best of our knowledge, the abovementioned protocol like the rest of the proactive protocols must process all the received routing messages to create a complete view of the topology. The resulting protocol is called Effective Routing Based on Overhead Reduction (ERBOR).

To execute ERBOR, the network members store the routing information and the latest topological changes in three tables and the MPR list: (a) Routing Table (RT), (b) Next Update Table (NXUT), c) Delayed Update Table (DUT). The routing table contains a path to every reachable destination. Based on the routing messages received from the selected MPRs, the paths are added, updated or deleted. When a topological change is sensed, the node saves the changes in the NXUT so that they can be advertised. After broadcasting the content of the NXUT, this table is cleared to capture only the new changes.

Table 12 is an example of how ERBOR structures the routing table. The updates contain: (a) the Destination ID, (b) the hop count, (c) the type of the update. In specific situations, the updates are delayed for a short period to avoid routing anomalies. The delayed updates are saved in the DUT first, then added to the NXUT in case they are still valid. Otherwise, if a removal update corresponding to a temporary update is received, the corresponding path and the temporary update in the DUT, are removed. Hence, this table is used essentially to avoid the collision of opposed updates. Afterwards, the content of the DUT is cleared after that it's combined with the NXUT.

Table 12. Routing table structure.

Destination_ID	Next_Hop	Hop_Count
MH3	MH2	3
MH5	MH7	4
MH6	MH2	2
MH8	MH7	3
MH9	MH7	2

III.II.1 Routing messages treatment algorithm

The algorithm presented in Fig 29 represents the method used by ERBOR to process the received update message. The routing messages produced by the nodes contain either the NXUT or the entire routing table. In both situations, the message is a set of routing updates. In case where the message contains the entire routing table, all the updates are path updates. Otherwise, the message contains both the positive and negative updates. Table 13 illustrates the structure of the routing messages generated by ERBOR.

Table 13. ERBOR routing message format.

RM_source_ID		RM_type
RM_sequence_Number		
Destination_ID_1	Hop_count_1	Update_type_1
Destination_ID_2	Hop_count_2	Update_type_2
.	.	.
.	.	.
Destination_ID_n	Hop_count_n	Update_type_n

```java
public void ProcessingRoutingUpdateMessage (ControlMessage UpdateMessage){
int i,n=UpdateMessage.Update_List.size();
int CM_GeneratorID= UpdateMessage.getGeneratorID();
boolean Skip_Update;
RoutingEntry DestinationEntry;
UpdateEntry Update;
if(!Control_Message_Is_Received_From_MPR_Node(UpdateMessage))return ;
for(i=n-1;i>=0;i--){
    Update= (UpdateEntry) UpdateMessage.Update_List.elementAt(i);
    Skip_Update=false;
    if(Update.getDestinationID()==ID_Current_node) Skip_Update=true;
    if(isMPR(Update.getDestinationID())){
        Skip_Update=true;
        if(!Update.getUpdateType())
             InsertInDUT(new UpdateEntry (true,Update.getDestinationID(),1));
        else if(Update.getHopCount()==1)
             PatchAllPaths(Update.getDestinationID(),CM_GeneratorID);
        }
    if(!Skip_Update) {
        DestinationEntry=FindRoutingEntry(Update.getDestinationID());
        if(DestinationEntry!=null){
        if(!Update.getUpdateType()){
            Process_Delete_Entry( Update, DestinationEntry, CM_GeneratorID);
        }else {
        Processing_Update_For_An_Existent_Path (Update, DestinationEntry, CM_GeneratorID);
        }
        }else Registering_A_New_Routing_Entry (Update, DestinationEntry, CM_GeneratorID);
    }
  }
}
```

Fig 29. Update message processing algorithm.

III.II.2 Path update treatment

When a path update is received, the receiver starts by checking if a path to the same destination already exists. If it’s the case, the received update is applied only if the new path is shorter than the existent one by at least two hops or it’s received from the corresponding next hop. Otherwise, the receiver might have a shorter path. In this case, the existent path is first added to the DUT for verification, then advertised if it’s still valid. As a result, the nodes collaborate to build the shortest possible path. For example in Fig 30, MH3 detects and generates a path update describing a new route leading to MH2. When MH1 receives this update, it can’t be applied since MH1 has already a shorter path towards MH2. On the other hand, MH4 can apply this update because it offers a shorter path. This treatment is accomplished by the procedure in Fig 31.

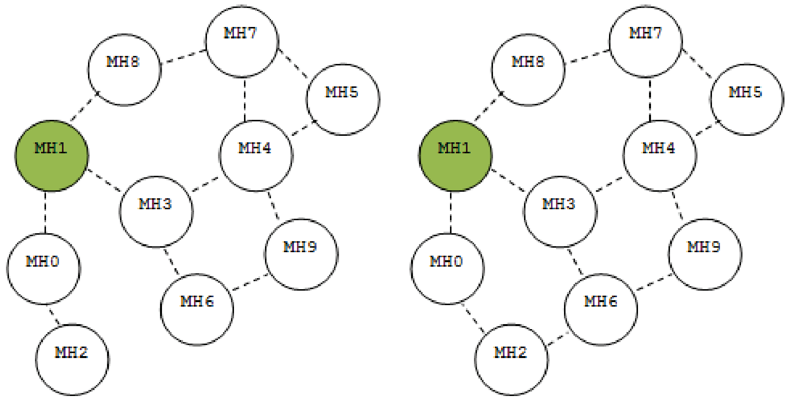

Fig 30. Path update treatment.

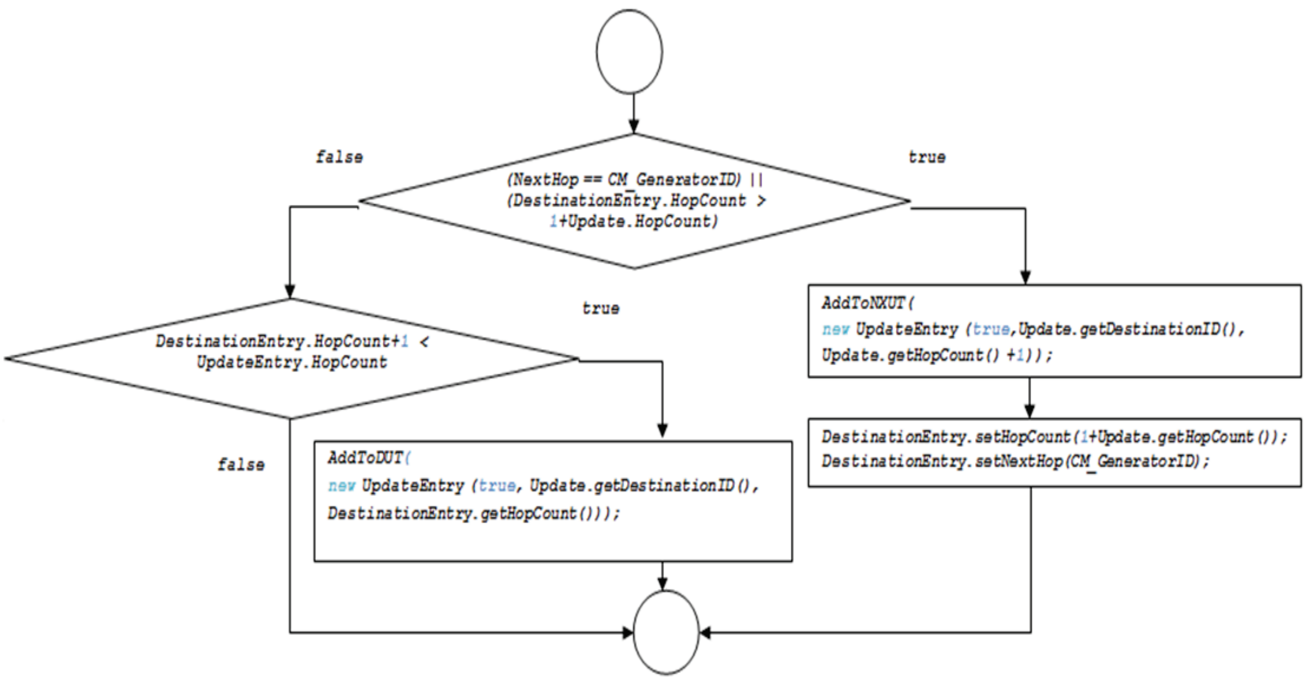

Fig 31. Processing a received update for an existent path.

Another possibility is that the received path describes a new destination. In such a case, the receiver adds a new entry describing this path only if there hasn't been a recent deleted path to the same destination. In order to verify this condition, the receiver searches in the NXUT for a deletion update describing the same destination. This treatment is executed by the procedure in Fig 32.

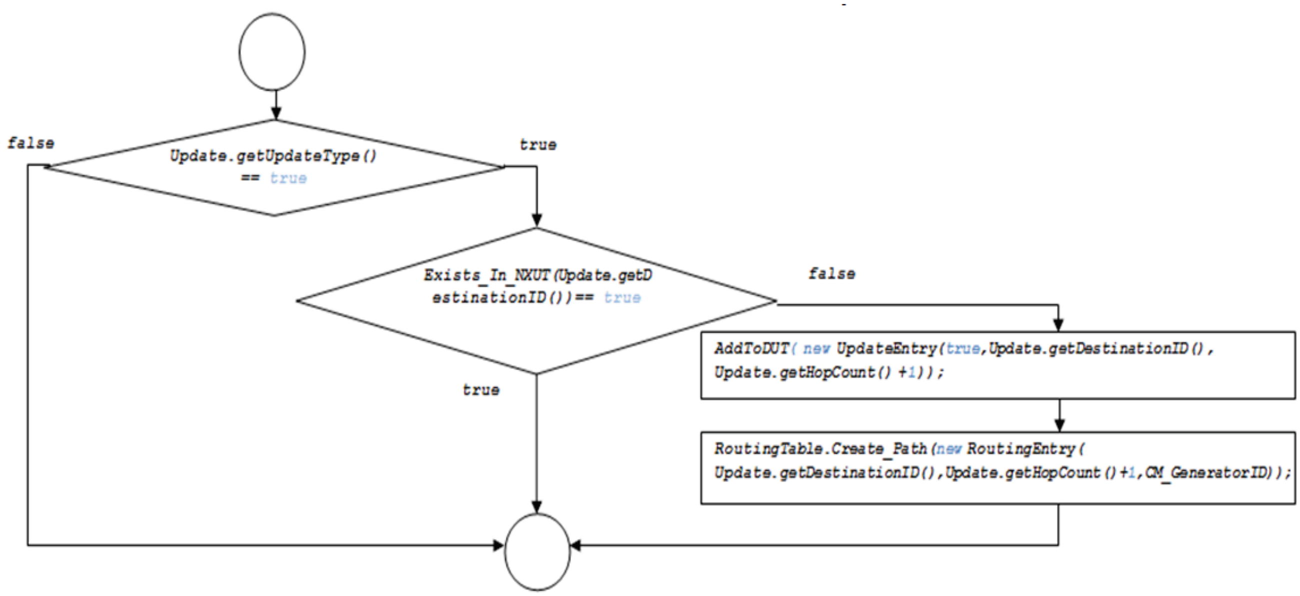

Fig 32. Processing a received update for a new path.

III.II.3 Deletion updates treatment

A deletion update is applied by a node only if it's received from the corresponding next hop. To verify this condition, the receiver compares the next hop towards the destination with the ID of the control message generator. If the two values are equal, the corresponding routing entry is removed and a deletion update is added to the NXUT. Subsequently, the removal update is advertised in the next routing message to inform the adjacent nodes about the topological adjustment.

If the source of the update is not the next hop towards the destination, the receiver of the update might have an alternative path towards the same destination. In this case, this alternative path is added to the DUT so it can be verified. Then, the alternative path is append to the NXUT only if no removal update is received for this path before the next broadcast. Otherwise, the temporary update and the alternative path are removed after generating a removal update to announce that this path is no longer valid. The diagram in Fig 33 explains how this method is implemented.

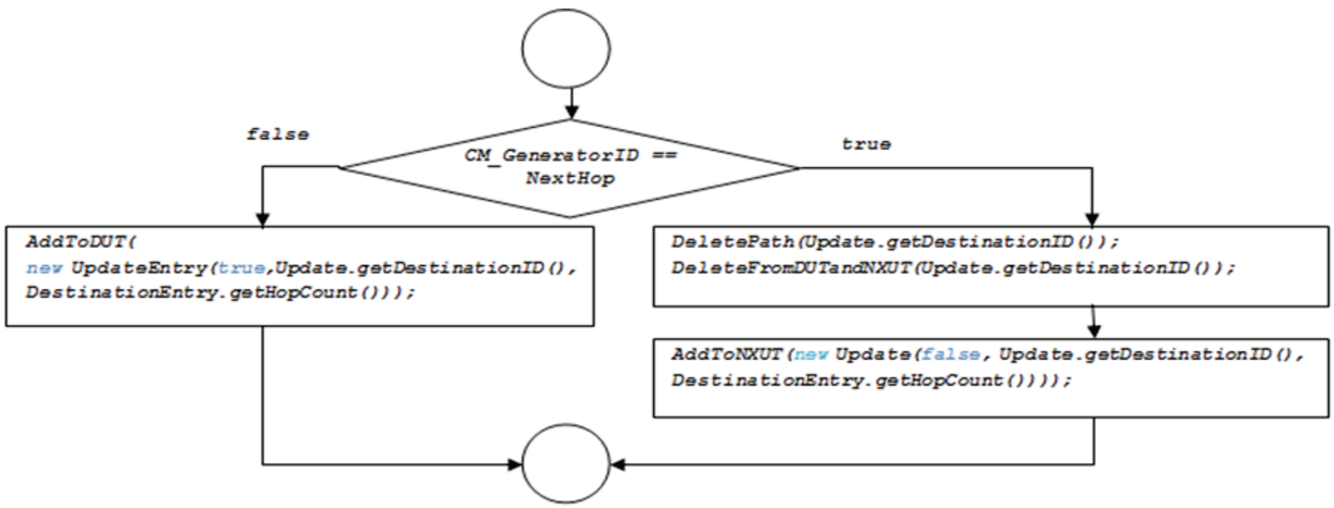

Fig 33. Processing a deletion update.

In Fig 34 for instance, MH5 generates a deletion update after sensing that MH4 has moved out of range. When MH3 receives this update, the entry attributed to MH4 is deleted and a removal update is advertised afterwards. This removal update is applied only by the nodes that use MH3 as the next hop to reach MH4 (MH2 for example). For the rest of the network, this update is ignored since MH3 is not a part of the path to reach MH4. For instance, MH4 is reachable through MH9 for the node MH1. Consequently, MH1 and all the nodes that built a path through it, to reach MH4, ignore the update that's originated from MH3. This method reduces the unnecessary dissemination of routing updates in the unaffected areas of the network and thus reduces the overhead engendered by the movement of the nodes.

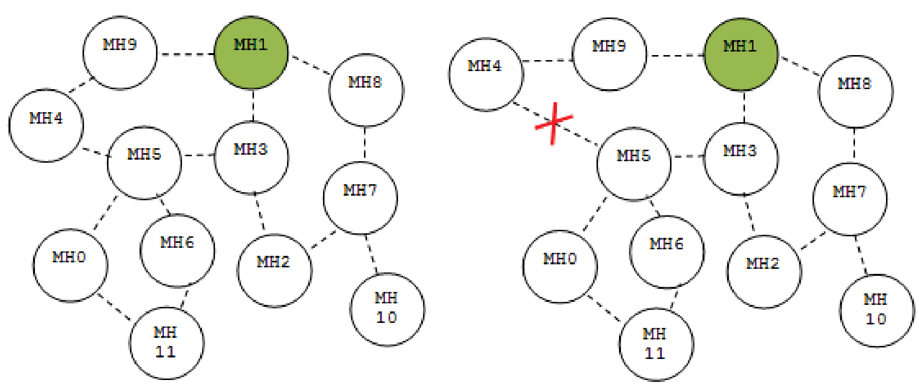

Fig 34. Removal update scenario.

III.II.4 The routing information filter algorithm

A commune aspect shared between the proactive protocols is that all the received routing messages must be treated. Generally, the messages received from the neighbours contain a certain amount of redundant information. For instance in Fig 35, MH1 can obtain the same network view through MH2 and MH3. Processing the same information repetitively is a waste of energy and CPU capacity especially in dense areas of the network. Therefore, we propose a method of MPR selection that eliminates the treatment of the redundant routing information.

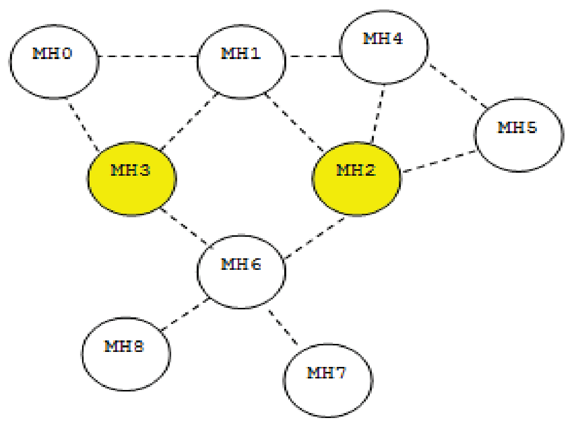

Fig 35. Redundant information treatment.

Instead of treating all the received messages, ERBOR determines if it's necessary to process a routing message by searching for an existent path leading to the generator of the message. Basically, a routing message is fully processed only if it's received from a selected MPR. Otherwise, if the generator of the message isn't selected as a MPR, the receiver verifies if it can be considered as a new MPR. To do so, the protocol searches in the routing table for a path leading to the source of the routing message. If this condition is verified, the source of the message is not an MPR because, all the paths that can be built through it, are redundant. Thus, the received routing message is ignored. The schema in Fig 36 describes how this treatment is executed by ERBOR.

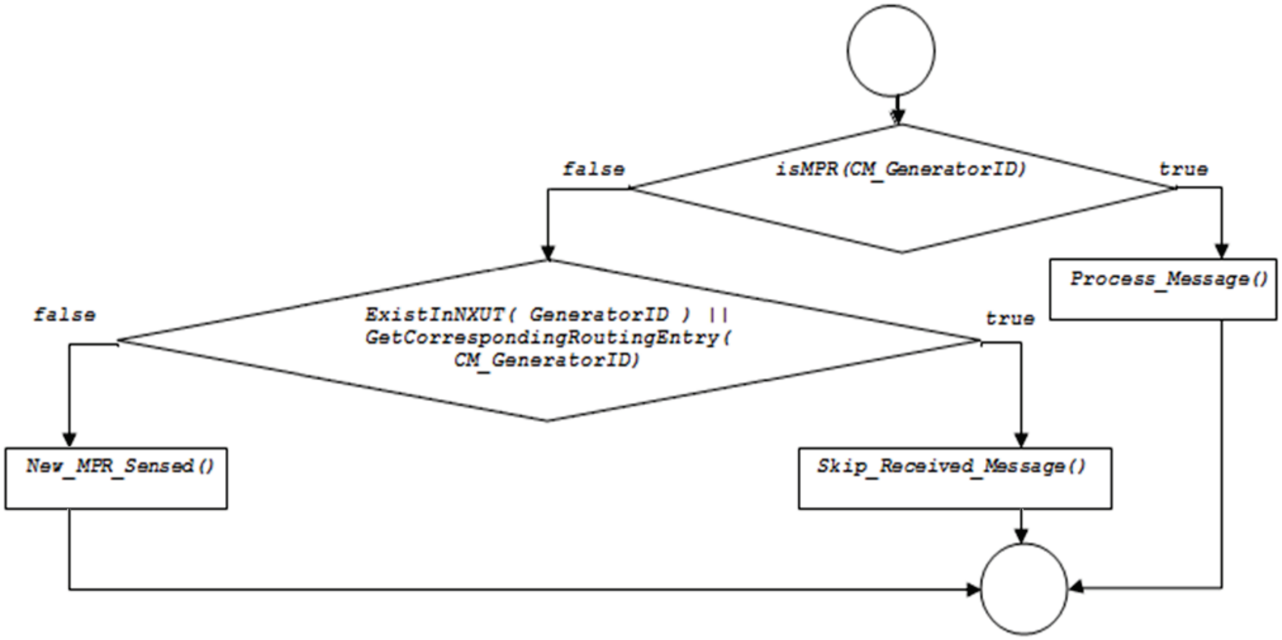

Fig 36. Routing information filter.

On the other hand, if the source of the routing message isn't reachable through any existent MPR, the receiver checks if there's a recent update describing the source of the message. Then, in case were there's no recent update describing the source of the message, the generator of the routing message is considered as a new MPR. Afterwards, the two neighbours that select each other as MPR broadcast their full routing table. Based on the received routing table of a new MPR, a node calculates a complimentary list of paths and adds them to the DUT. Subsequently, the complementary paths are advertised if they are still available. Finally, a link between two nodes is considered as symmetric only if both nodes receive a path update describing this new link.

In order to enhance the efficiency of the MPR selection method, when one of the selected MPRs becomes reachable through another MPR, one of the two MPRs is deleted from the MPR list. The state of the redundant MPR is set to "inactive" and all the paths based on it, are patched through the first MPR that announced this new link. After the broadcast of the routing message, all the "inactive" MPRs are removed from the MPR list. In the message processing algorithm (Fig 29), this operation is applied in *PatchAllPaths(Disabled_MPR_ID,Update_Generator_ID)*.

Like in other MANET routing protocols, ERBOR routing messages are advertised in UDP mode. The *RM_sequence_Number* field in Table 13, is used by the nodes to verify that all the messages advertised by the MPRs have been received. When a node detects that one of the routing messages hasn't been received, the entry attributed to the corresponding MPR is removed and a delete update is generated. As a result, the two neighbours either rebroadcast their full routing table or another MPR offers an alternative path to join them.

III.II.5 Protocol testing

In order to make sure that ERBOR works correctly, we implemented a live routing simulator that can execute the protocol in a distributed manner. The role of this software is to generate a virtual network and display the routing information gathered by the nodes while running the movement scenario. Afterwards, the user can verify the progress of the topological information maintained in every node. Unlike other simulation tools, this routing simulator enables the user to directly change the course of the simulation without restarting the experiment. Considering the distributed and interactive nature of the MANET, this feature is a substantial advantage for a simulation of a dynamic network. During the implementation, this software was organized in two layers: (a) the routing protocol layer, (b) the protocol monitor layer. The basic architecture of these two layers is represented by the Unified Modelling Language (UML) Class Diagrams in Fig 37 and 38. This software was written in Java (Jw, 1995) using Netbeans IDE 7.0 (Sun Microsystems, 2000).

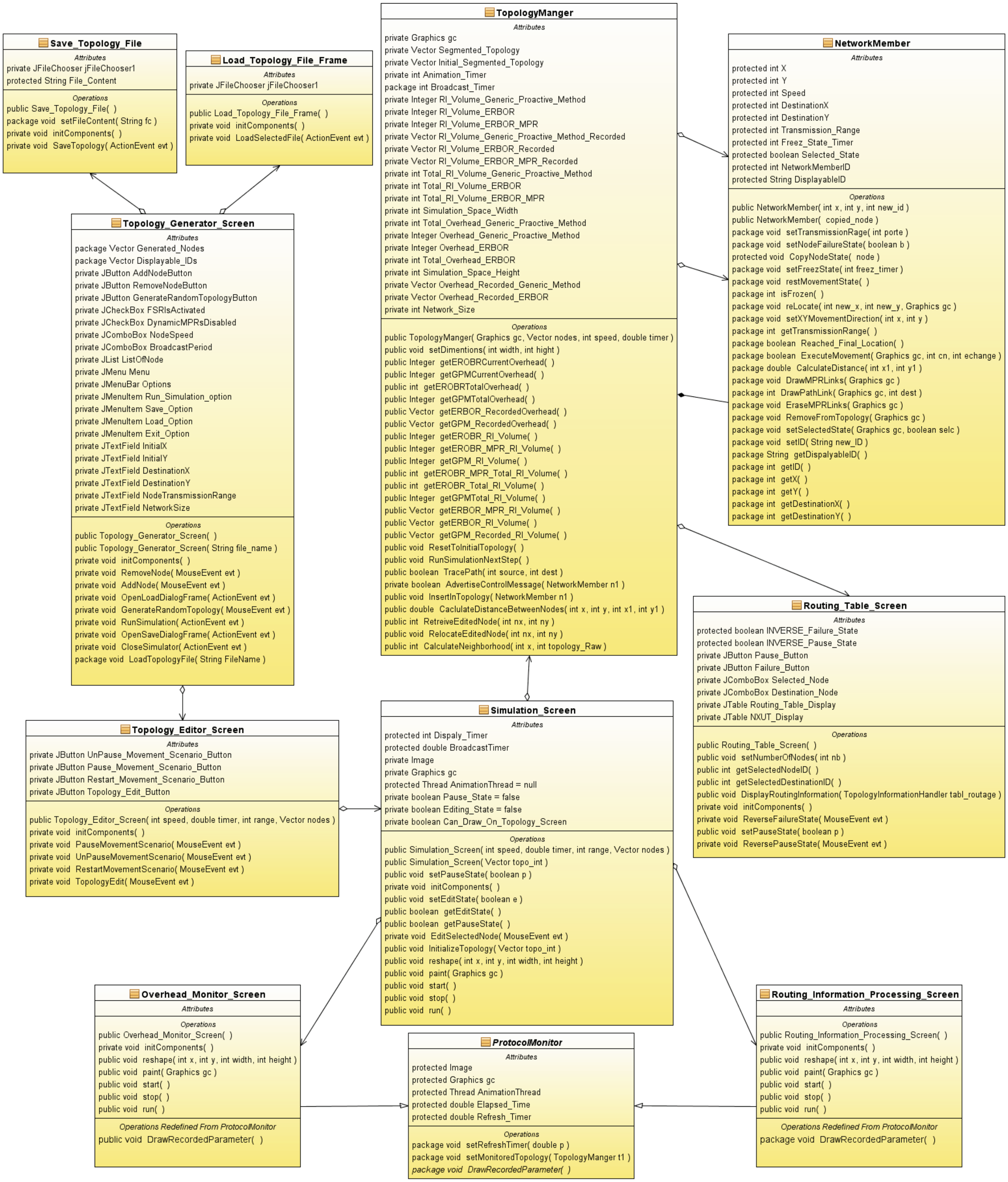

Fig 37. Class diagram for the visual components.

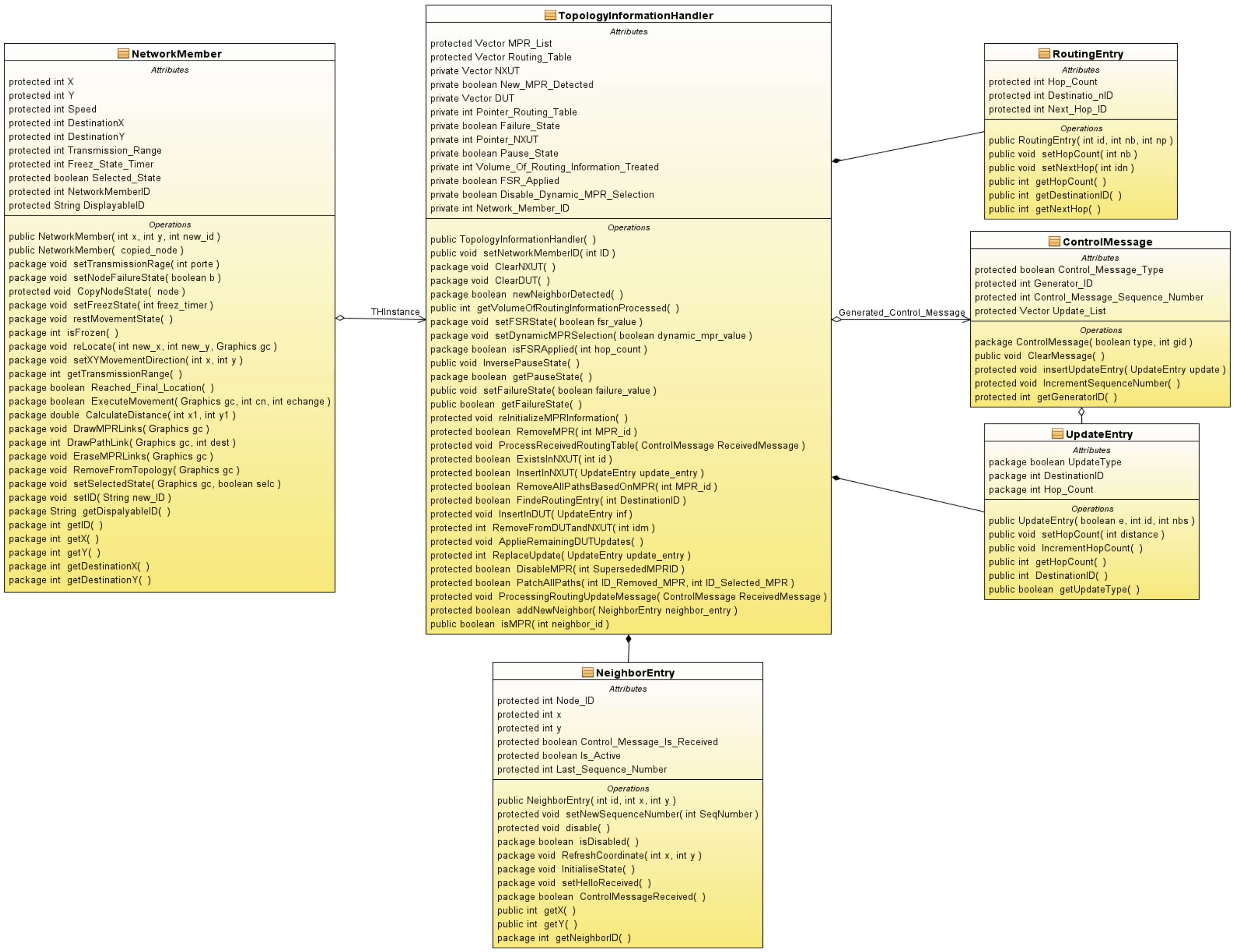

Fig 38. Class diagram for the protocol layer.

To test the protocol properly, the movement of the nodes must be random. Therefore, the first function of this software was to generate random topologies. The user defines the network size only and the simulator generates a network with the specified size. To generate a random movement scenario, the simulator chooses randomly the initial locations and the coordinates to which the nodes are moving toward. This task was important during the test of the protocol in dynamic topologies. In addition, the user can create a specific topology with a predefined movement scenario by choosing the initial and final position for each node. Fig 39 represents the interface of the topology generator.

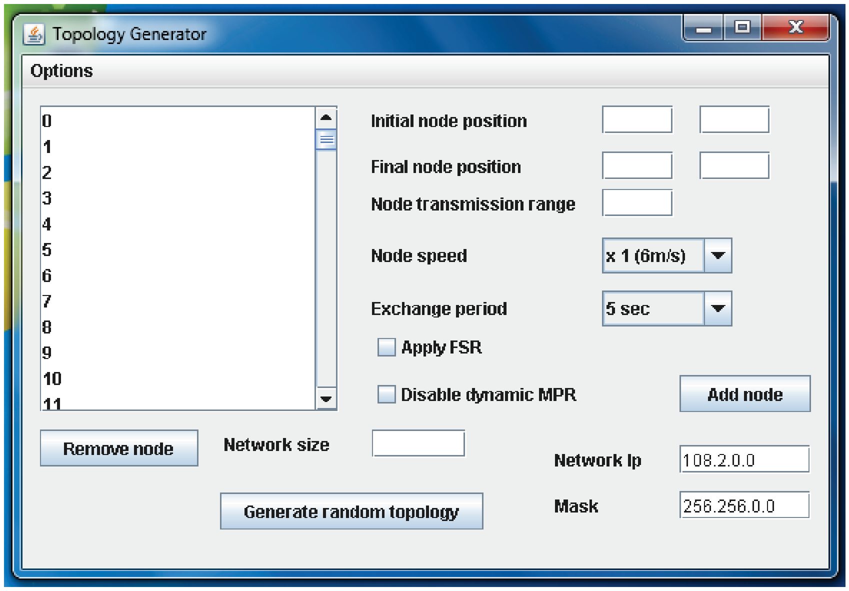

Fig 39. The topology generator.

When the simulation is launched, multiple frames are displayed to monitor the topology and the protocol state. The user is able to: (a) monitor the evolution of the topology, (b) monitor the routing information collected by the selected node, (c) edit the network topology. The selected network member is marked in blue on the topology screen and all the routing information collected by this node is displayed proactively in the routing table screen. Fig 40 is an example of the various interfaces of the simulator.

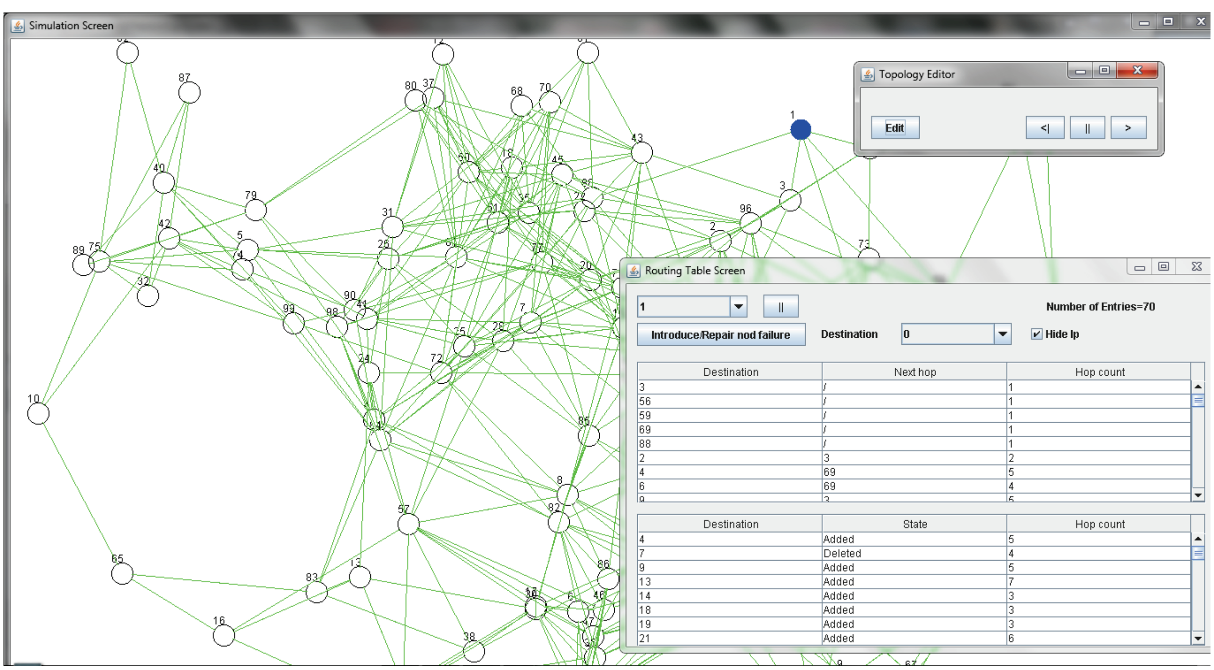

Fig 40. The various visual interfaces accessible for the user.

The link between a node and a selected MPR is displayed by a green connection in the topology screen. To check the accuracy of the topological information created by ERBOR, the user can choose to display a path between a pair of nodes via the routing table interface. It's possible to select a node and the destination to which a route is needed. Then, the path between the selected pair of nodes is highlighted in blue. An example of how a path is visualized between node 0 and 16 is shown in Fig 41.

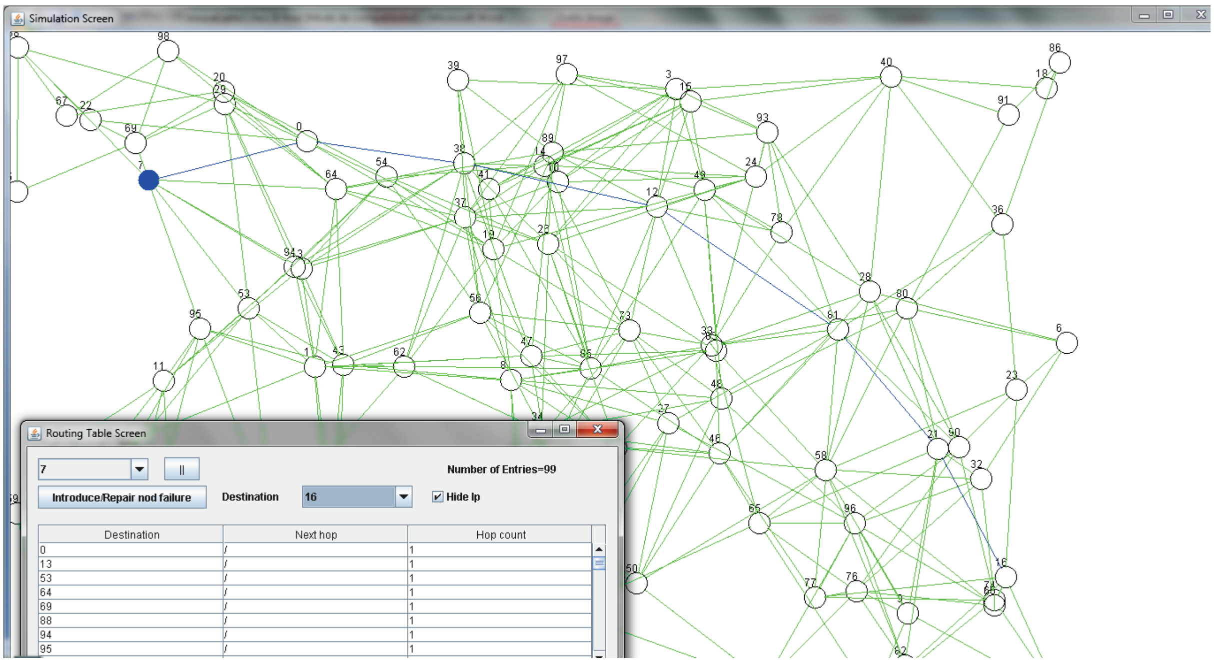

Fig 41. Path visualization.

Considering the unpredictability of the MANET topology, it was essential to examine the routing protocol behaviour in an extensive number of random tests. Therefore, the topologies in which ERBOR was tested, were generated randomly and the network size was augmented gradually. The first tests displayed the creation of routing loops and incomplete dissemination of the routing updates in particular scenarios. This type of routing errors were corrected by the introduction the DUT structure.

Since our primary objective was to create a protocol that consumes fewer resource, the next step was to measure the overhead of ERBOR. The idea is to simply record the sizes of the control messages generated by the nodes and summing them to obtain the overhead volume. In Fig 42, the simulator displays periodically the overhead recorded and the total overhead produced. Two plots in this frame appear: (a) the overhead produced by ERBOR displayed in green, (b) the overhead produced when the nodes broadcast their entire routing table (the generic proactive method) displayed in red.

In this example, the network is composed of 100 nodes moving in 1000m x 1000m perimeter, at a speed of 4m/s and 250m of transmission range. At the beginning, ERBOR generates almost as much overhead as the generic proactive method (full routing table broadcast). This result can be explained by the fact that ERBOR broadcast the entire routing table when the nodes are starting to establish their MPR list. After a short period of time, the overhead of ERBOR starts to decrease progressively because the nodes become familiar with their neighbourhood and the broadcast of the entire routing table is less frequent. Finally, ERBOR generates only small control messages to maintain the links between a node and the selected MPRs. Subsequently, the overhead produced after the protocol convergence, is very low.

Other protocols constantly disseminate the routing messages in the network instead of periodically advertising the routing table. Nevertheless, this method generate a significant amount of overhead as well. The method applied in ERBOR is based on reducing the occurrences of the routing table advertisement. As a result, ERBOR generates less overhead than the classic proactive method. This result can change from a situation to another depending on the parameters of the topology.

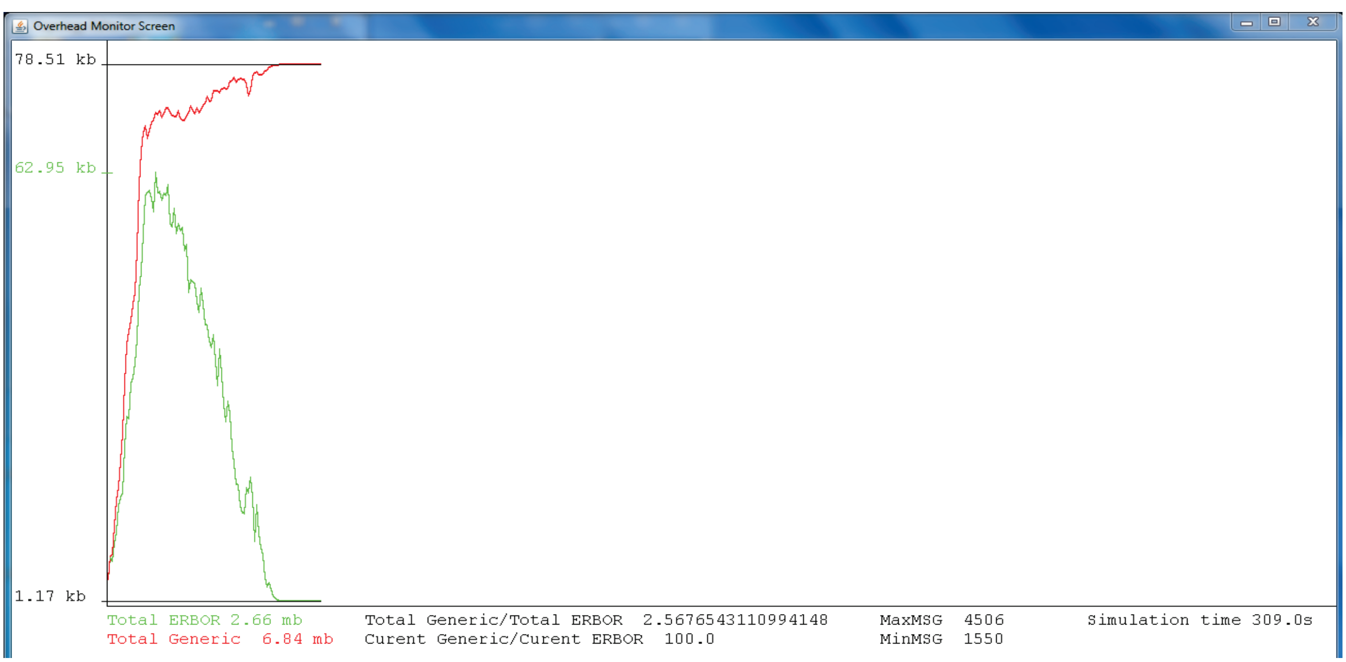

Fig 42. The overhead measurement screen

Moreover, since ERBOR includes a method that eliminates the treatment of the redundant routing information, it was necessary to include a frame that measures the volume of routing information processed in the entire network. This frame displays the volume of routing information received by the nodes and the volume of routing information processed by ERBOR. Three plots are shown in this screen: (a) the volume of routing information treated if all the received routing messages contained the full routing tables of the neighbours, (b) the volume of routing information treated if all the routing messages generated by ERBOR are processed, (c) the volume of routing information treated when only the routing messages received from the selected MPRs are processed (which is the method applied by ERBOR). The result in Fig 43 clearly shows that the control message filter applied in ERBOR reduces significantly the amount of routing information processed.

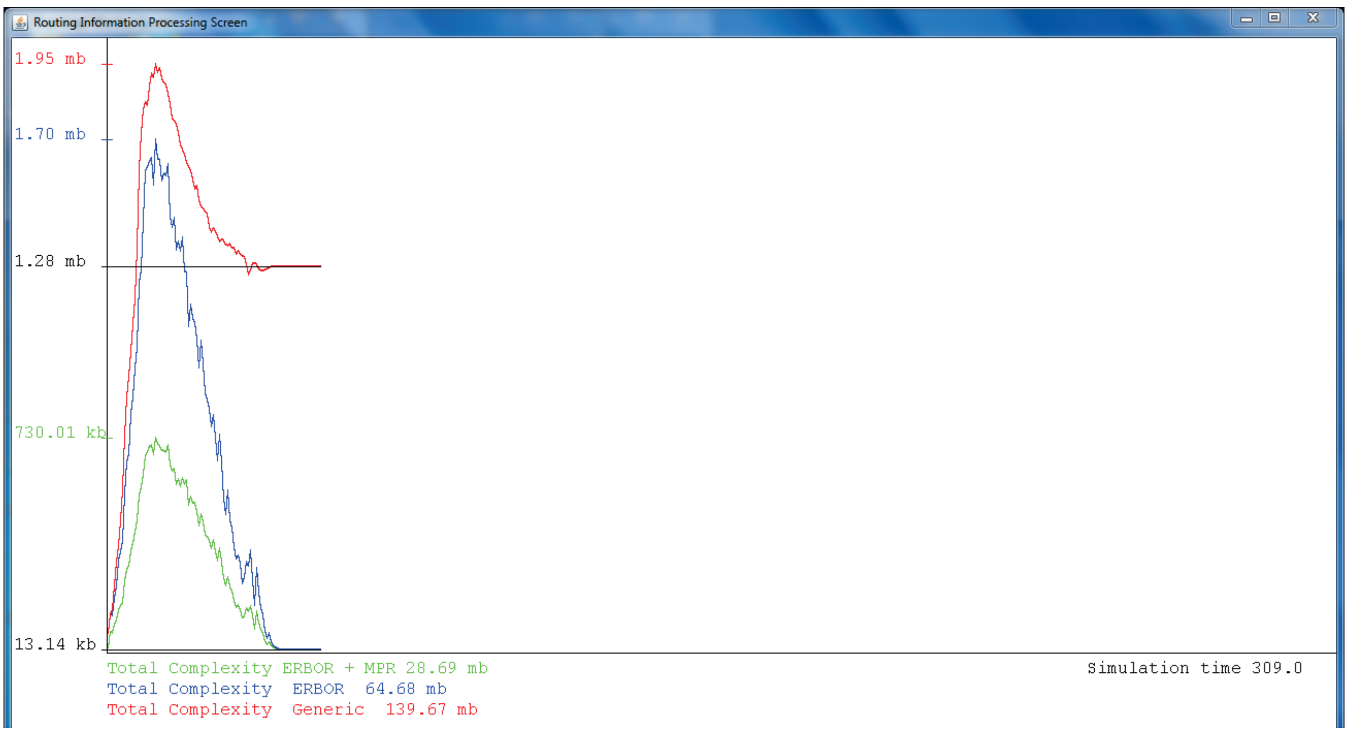

Fig 43. Routing information processed.

```
static void Percentage_OF_MSG_RTR_Lost(String Trace_File_Path){
int Messages_RTR_Received=0,Messages_MAC_Lost=0,i;
String Line,tokens[] = new String[100];
int Sum_Size_Lost=0,Total_MSG_RTR;
java.util.StringTokenizer st;
try{
    FileReader  in=  new FileReader (Trace_File_Path);
    BufferedReader br = new BufferedReader(in);
    while ((Line = br.readLine()) != null) {
        st = new java.util.StringTokenizer(Line, " ");
        i=0;
        while(st.hasMoreElements())tokens[i++]= st.nextToken();
        if (tokens[0].equals("r")  && (tokens[3].equals("RTR")))
            Messages_RTR_Received++;
        if (tokens[0].equals("D")&& tokens[3].equals("MAC")) {
            Sum_Size_Lost+=Integer.parseInt(tokens[7]);
            Messages_MAC_Lost++;
        }
    }
} catch (Exception Exp){
    System.out.println("Error "+Exp.getMessage());
   }
Total_MSG_RTR=Messages_RTR_Received+Messages_MAC_Lost;
System.out.println("Percentage lost MSG = "
          +100.0*Messages_MAC_Lost/Total_MSG_RTR+" %");
System.out.println("Average Size lost MSG="
        +Sum_Size_Lost*1.0/Messages_MAC_Lost);

}
```

Fig 44. Calculating the percentage of routing messages dropped from an NS 2 trace file.

Although our simulator does not implement the MAC and Physical layers, the overhead / routing information processed, which are the only metrics captured by this software, are not significantly affected by the aforementioned layers. In the simulation conditions where the protocol is tested, there is no transmissions between the nodes and yet the proactive protocol overhead continues to expand as explained in previous chapters. Indeed, in a complete simulation, routing messages can be dropped at the receivers MAC layer because of collisions. However, in scenarios with similar settings, the percentage of routing messages dropped due to collisions while applying DSDV is usually lower than 6%. Reasonably, the routing messages losses increase when their sizes are raised, which is the case for DSDV (Fig 12) and most of the proactive routing protocol in general. On the other hand, ERBOR is designed specifically to lower the sizes of routing messages. And most importantly, in ERBOR, not all the routing messages received must be processed which lowers significantly the effect of routing messages loss. Thus, we deemed the effect of the MAC and Physical layers not significant for overhead measurement of ERBOR. To illustrate our hypothesis, the algorithm in Fig 44, calculates the percentage of routing messages dropped and their average size from a NS 2 trace file. When applied properly, this procedure displays that most of the dropped routing messages are large size, which is the case of FSR in dense topologies.

III.IV. Results and discussion

The main disadvantage of all the proactive protocols is their inadaptability to large / high density topologies. Previous studies have illustrated that the increase of the network size, leads directly to the exponential growth of the proactive protocols overhead. For instance, Johann et al. (2004) explain that the large overhead produced by the proactive protocols is caused by the frequent broadcast of the entire routing table. Furthermore, the obtained results by Deepak et al. (2013) and Kavita and Abhishek (2011) indicate that the increase of the network size affects negatively the reactive / hybrid protocols less than the proactive protocols.

Besides, from a network density prospect, the treatment of a large quantity of routing information can exhaust the node resources at high rate. This aspect affects significantly the MANET members since they are usually equipped with a limited computational capacity. To compare ERBOR, in this context, with other protocols from the same category, we conducted several series of NS 2 simulations with DSDV and FSR (Sven, 2005). FSR and DSDV were chosen because they generate the lowest overhead in the proactive routing category. Especially FSR which is a protocol that is specifically developed to lower the overhead in large networks. The simulation parameters were changed in order to measure the overhead when: (a) the network size is expanded, (b) the topology density is raised, (c) the movement speed of the nodes is raised. Generally, ERBOR generates the lowest overhead during the tests and treats less routing information than FSR and DSDV.

It's important to note that the movement scenarios used in this section are simple random movement. Meaning that each node walks to a random position, with a constant speed, then stops. Also, unlike the previous tests the transmission range is set to 150m instead of 250m due to the massive number of nodes (150-250). Finally, the choice to measure the overhead by the cumulative size of the routing messages rather than the number of routing packets is meant to express the complexity of the tested protocols.

III.III.1 The network growth influence

Table 14. The network growth parameters.

Parameters	DSDV	FSR	ERBOR
Broadcast period	10sec	5sec	3sec
Simulation space	1500m x 1500m		
Transmission range	150m		
Speed of movement (constant)	4m/s		
Network size	**150-250**		
Simulation time	400s		

To test the influence of the network growth on the proactive protocols, we ran a series of simulations where the network size was augmented gradually. The obtained results in Fig 45 and 46 show how the overhead and the routing information processed by the tested protocols, are influenced. As expected, FSR generates less overhead and treats less routing information than DSDV. Although the Fisheye algorithm improves considerably the adaptability of the protocol to large topologies, increasing the network size leads eventually to the increase of the network density. As a result, the size of the neighbourhood description, advertised by FSR, is increased as well. The parameters of this experiment are declared in Table 14.

Besides the periodic broadcast of the global topological information, DSDV is based on an event driven method to announce the important topological changes. Although this approach has a slight benefits from a convergence perspective, this methods elevates the overhead. Moreover, the systematic advertisement of the entire routing table increases the overhead as well. Hence, the increase of the volume of routing information processed and the large quantity of overhead when DSDV is applied.

In contrast to the two previous protocols, the growth of the network appears to have a lot less influence on ERBOR. From an overhead point of view, the results can be explained by the fact that ERBOR limits the quantity of overhead by advertising the entire routing table of a node, only when a useful connection is established. Thus, the increase of the network size does not directly augment the sizes of the routing messages. Moreover, when the topology is stable, FSR and DSDV continue to broadcast the neighbourhood / routing table periodically. On the other hand, ERBOR advertises only the detected changes. Accordingly, whereas the overhead / routing information processed of DSDV and FSR continue to expand, ERBOR consumes a negligible quantity of resources when the network members are steady.

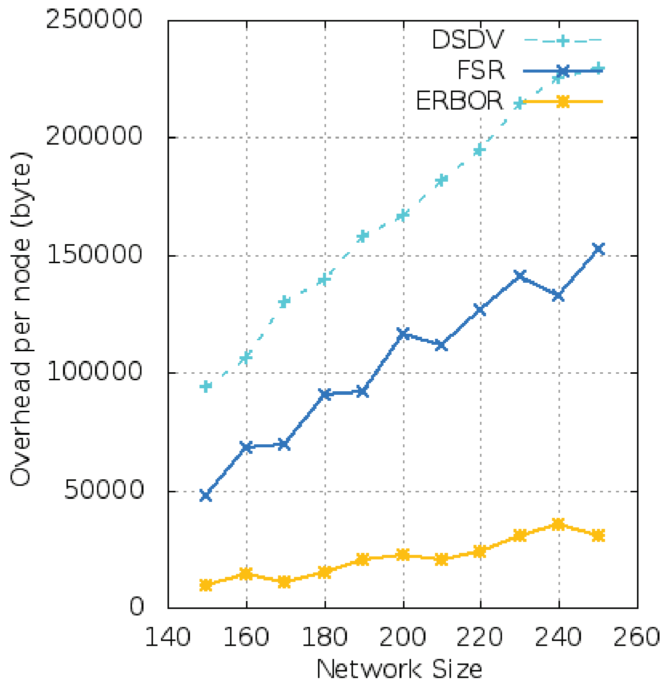

Fig 45. The impact of the network growth on the overhead generated per node.

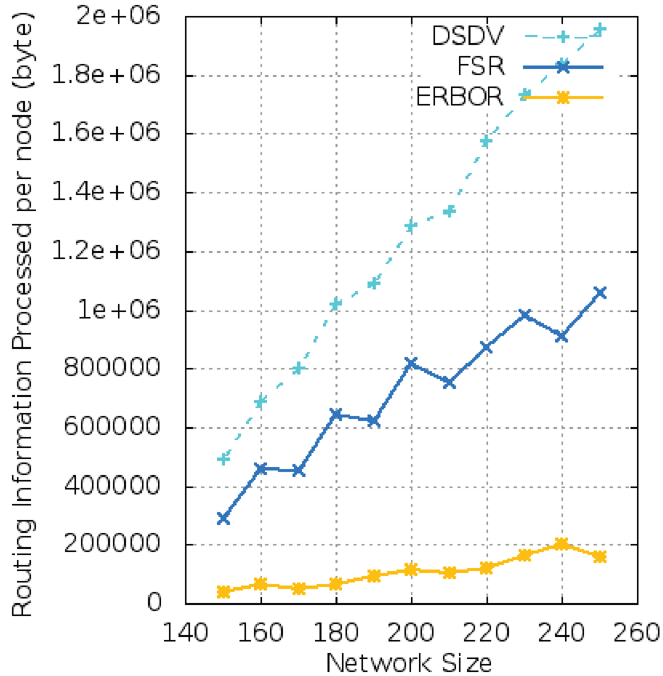

Fig 46.The impact of the network growth on the quantity of routing information processed per node.

As explained earlier, a shared mechanism between the proactive protocols is the periodic treatment of all the received routing messages. Considering the sizes of the routing messages (due to the growth of the network) advertised by DSDV and FSR, both protocols process a significant quantity of routing information in order to maintain the topological view. On the other hand, ERBOR treats only the messages received from the selected MPRs. Combined with the way ERBOR restrains the control message size, the method of MPR selection reduces considerably the influence of the network growth on the volume of routing information processed by the nodes.

III.III.2 The network density influence

Another parameter that can heavily influence the proactive method, is the network density. To test how this factor can impact the proactive protocols, we ran a series of simulations where the network density is increased by augmenting the transmission range of the nodes. Consequently, the number of neighbours sensed by a node, is augmented as well. The results in Fig 47 and 48 illustrate that the increase of the network density leads to the increase of the overhead and the volume of routing information processed, by all three protocols. Table 15 represents the parameters of this simulation.

Table 15. The network density parameters.

Parameters	DSDV	FSR	ERBOR
Broadcast period	10sec	5sec	3sec
Simulation space	1500m x 1500m		
Transmission range	**150-250m**		
Speed of movement	4m/s		
Network size	200		
Simulation time	400s		

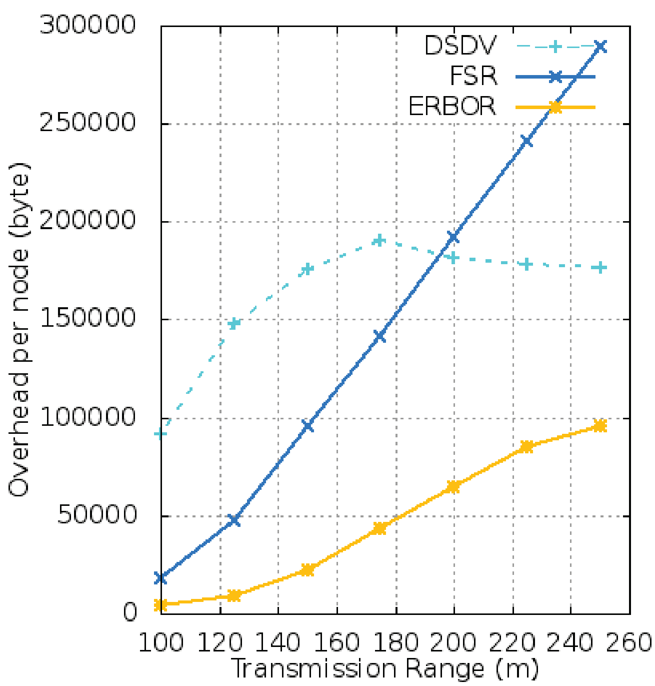

Fig 47. The impact of the transmission range on the overhead generated per node.

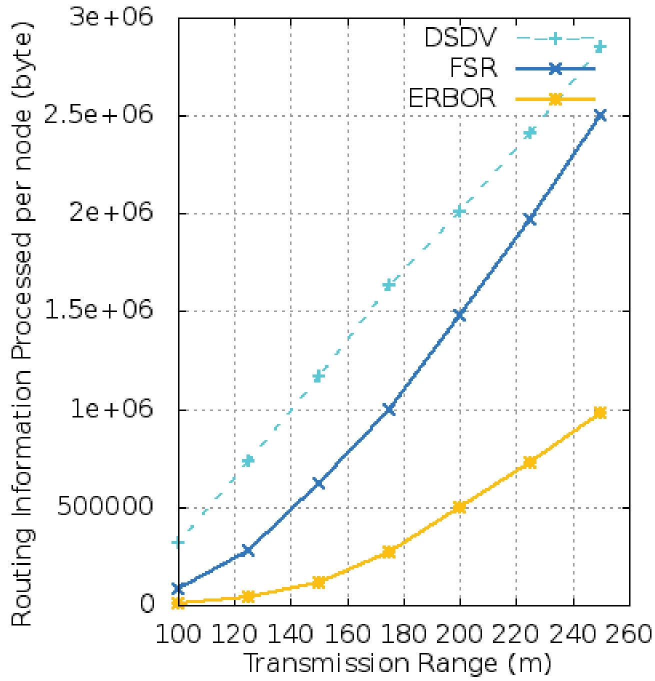

Fig 48. The impact of the transmission range on the quantity of routing information processed per node.

As mentioned in the theoretical description of FSR, the strategy of this protocol is to broadcast the description of the adjacent nodes more often than the description of the remote destinations. Thus, when the number of close nodes increases due to the network density, the overhead of FSR expands past the quantity generated by DSDV. On the other hand, the overhead of DSDV is relatively stable because the dissemination method used in this protocol does not depend on the number of the adjacent nodes. More importantly, since all the received routing messages must be processed, the increase of the network density causes the elevation of the volume of routing information processed by both protocols.

Due to the MPR selection method, ERBOR advertises the entire routing table only when it is necessary. Although this method reduces the volume of overhead, the increase of the network density leads inevitably to the increase of the detected MPRs. Nonetheless, the effect of the network density is significantly reduced by comparison with FSR and DSDV. Moreover, unlike FSR and DSDV, only the essential routing messages are processed. Thus, the influence of the network density on ERBOR, is limited.

Another perceptible outcome is that, although FSR generates more overhead than DSDV, DSDV processes more routing information. While this result can seem odd, it can be simply traced to the event driven nature of DSDV. In FSR, the broadcast of the neighbourhood description is done periodically. As a result, the increase of the routing messages sizes, increases directly the volume of routing information processed. On the other hand, the broadcast of the routing messages is partially event driven when DSDV is applied. To lower the overhead, when an important event is detected, DSDV broadcasts the entire routing table to limit the sizes of the subsequent incremental messages. Moreover, due to the increase of the network density, any sensed change deemed as essential, will trigger the immediate broadcast of the routing message which is received by all the adjacent nodes. Hence, the elevation of the routing information processed by DSDV. Moreover, as the routing messages advertised by FSR expand exponentially in this experiment, a significant percentage is dropped due to collision. In Fig 49, the recorded quantity of routing messages processed by DSDV and FSR, during this series of simulations, are displayed.

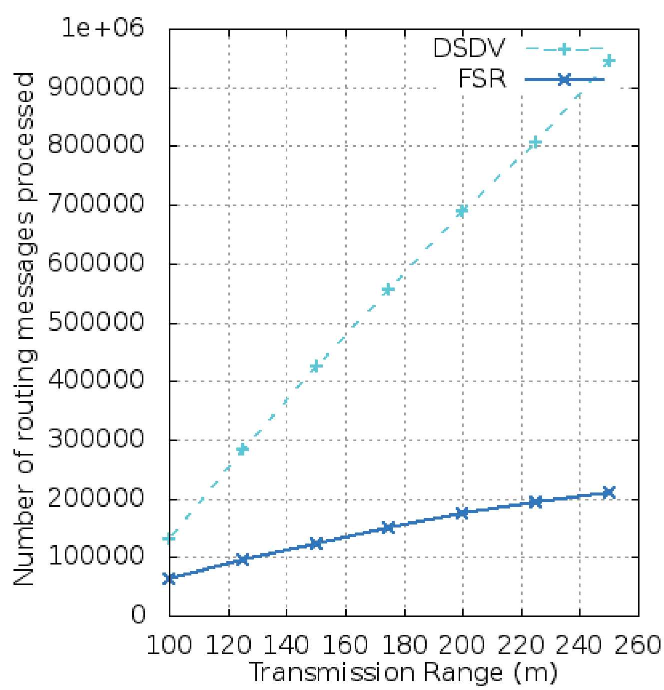

Fig 49. The overall quantity of routing messages processed by DSDV and FSR.

III.III.3 Network members movement

To test the adaptability of the three protocols based on the nodes speed of movement, we made a series of simulations where the nodes speed of movement is augmented gradually. In a simple movement scenario, the obtained results in Fig 50 and 51, show that DSDV and FSR are influenced randomly. On the other hand, the overhead / routing information processed of ERBOR, decreases. The parameters of this experiment are shown in Table 16.

Table 16. The network members movement speed parameters.

Parameters	DSDV	FSR	ERBOR
Broadcast period	10sec	5sec	3sec
Simulation space	1500m x 1500m		
Transmission range	150m		
Speed of movement	**2-6m/s**		
Network size	200		
Simulation time	400s		

First, it is obvious that increasing the nodes speed of movement does not influence FSR directly. Unlike DSDV, the routing messages advertisement method of FSR is independent from the topological events. As it is shown in Fig 50, the highest values of overhead are generated by DSDV. Regardless of the difference between the overhead produced by DSDV and FSR, this protocol generates a higher number of routing messages when the topology changes frequently.

Unlike DSDV, ERBOR periodically announces only the changes based on the updates received from the MPRs. Thus, the effect of the topological changes on the overhead of ERBOR, is reduced. Evidently, when the nodes are moving fast, they reach their final position faster and the topology becomes stable. Since ERBOR announces the routing updates periodically only, the overhead of this protocol is lowered after its convergence. Otherwise, when the nodes are moving slowly, the topological events are more continuous. Hence, it is also noticeable that ERBOR generates more overhead when the nodes are in continuous motion.

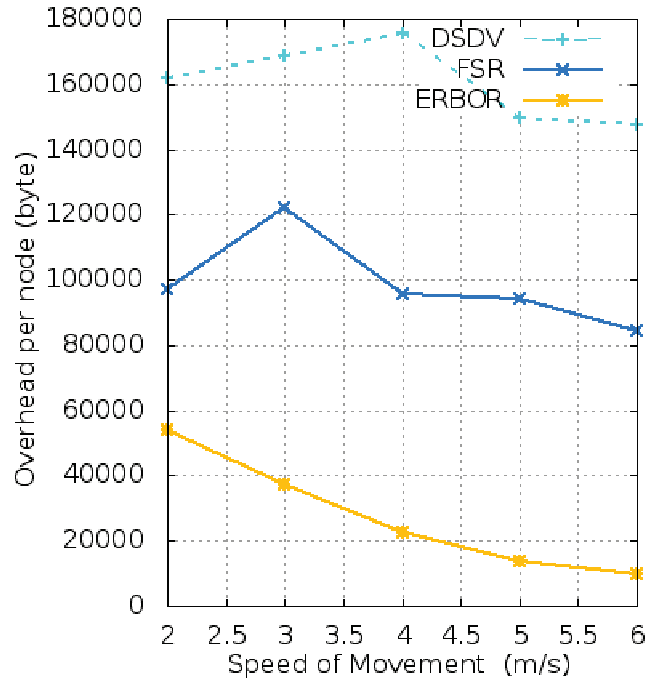

Fig 50. The impact of the speed of movement on the overhead generated per node.

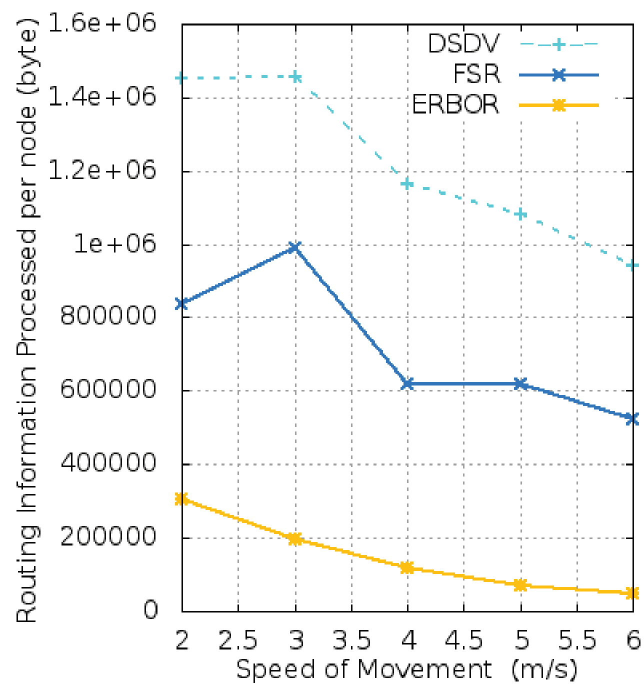

Fig 51. The impact of the speed of movement on the quantity of routing information processed per node.

On the other hand, in a continuous movement scenario, FSR generates the largest volume of overhead as revealed in Fig 52. Though in such a scenario, DSDV understandably generates a much higher number of routing messages, this protocol eliminates routing loops and lowers the sizes of the subsequent incremental messages. Conversely, due to the periodical nature of FSR, the faster the topology is changing, the more frequent routing loops appear. Which increases considerably the sizes of the routing messages advertised by this protocol. Nevertheless, the volume of overhead of both protocols is exponentially increased. Unlike DSDV and FSR, the overhead of ERBOR is decreased. While ERBOR periodical approach is a similarity to FSR, ERBOR advertises mostly updates. Moreover, ERBOR disseminates the topological events faster due to its higher frequency of routing messages broadcast (3s as opposed to 5s / 15s of FSR). As a result, contrarily to FSR, ERBOR deletes expired path entries faster to avoid routing loops and lower the routing messages sizes.

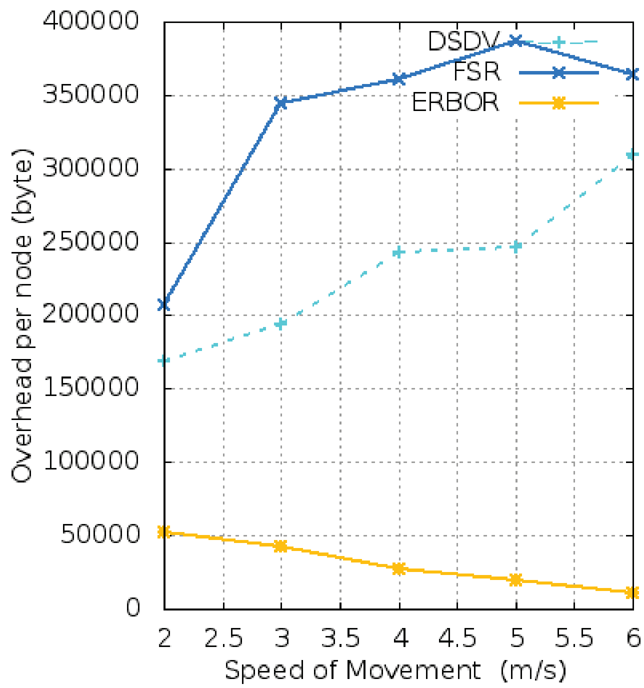

Fig 52. The effect of the speed of movement on the overhead in a continues movement scenario.

III.V. Conclusion

In this chapter, we proposed an implementation for a pre-emptive reactive method intended to limit the nodes mobility effect on the protocols performances. By anticipating path failures, we believe that this sort of mechanism is the best solution to avoid packet losses for AOMDV. Moreover, while other proactive methods focus on how to reduce the number and size of control messages, the proactive algorithm that we propose takes into consideration the routing messages treatment. Instead of processing all the received routing messages, ERBOR avoids treating the redundant information by selecting only the essential neighbours that can offer a complete view of the network. As a result, the nodes spend less resources to execute the routing protocol. Additionally, ERBOR generates less overhead by advertising the entire routing table only when it's necessary. The results obtained in this section showed that ERBOR lowers the overhead / routing information processed significantly by comparison with FSR and DSDV. Table 17 summarizes the results obtained in this section.

Table 17. The topological parameters influence on DSDV, FSR and ERBOR.

Protocol	Parameters	Network size	Network density	Node speed of movement
DSDV	Overhead	High	Medium	High
	Processed information	High	High	High
FSR	Overhead	Medium	High	Low
	Processed information	Medium	High	Low
ERBOR	Overhead	Low	Medium	Low
	Processed information	Low	Medium	Low

As the results in the previous chapter showed, the end to end delay of OLSR particularly expanded during the network growth experiment. When the overhead and quantity of the routing messages processed expands, the complexity of this protocol is amplified exponentially which results into additional transmission delay and energy depletion. Though ERBOR doesn't necessarily produce the shortest path, especially in a large network, providing connectivity with a very low complexity is a major advantage. With its low complexity / overhead, the broadcast period can be tuned in order to accelerate the convergence. In the next chapter, we will discuss a particular type of the MANETs, the Wireless Mesh Network (WMN). In this type of network, the hybrid routing approach is the most suitable due to the existence of central infrastructure. Thus, making our proposed protocols suitable components of a hybrid protocol tailored for WMNs.

Chapter IV: Proposal Of A Load Balancing Mechanism For Wireless Mesh Networks

IV.I. Introduction

Presently, the WMN architecture is emerging as a very profitable networking technology because of its multiple advantages (self-configuration, self-healing and low maintenance cost …etc). Due to the existing similarities between the WMN and MANET, routing protocols made for MANETs can be used in a WMN. The main difference between the WMN and the MANET is the existence of Mesh Routers (MRs). Usually, the MRs have access to an unlimited source of energy and a large bandwidth which makes them well equipped to provide QoS for the Mesh Clients (MCs). Furthermore, the MR that has direct internet access, is identified as a Mesh Gateway (MG). Thus, many modifications have been proposed to adapt the Ad-hoc routing protocols to make full use of the MRs. Theoretically, the most suitable approach for the WMN is the hybrid / hierarchical strategies where the routers exchange their routing tables and the clients launch the reactive search when a path to a remote destination is needed. Consequently, the resources of the clients are preserved.

In this type of network, one of the important problems is the traffic load distribution. The concentration of data flows on the MRs can lead to QoS degradation. Also, attributing the client to the nearest router is not always an ideal solution. Conversely, distributing the clients uniformly on MRs can significantly enhance the network performance. Therefore, many studies have been conducted in order to find a solution for this problem. The general idea is to attribute the clients to the less loaded router. Also, finding alternative paths towards the destination or another less loaded router can considerably enhance the network performance. In this chapter, we discuss the various methods used to improve the QoS offered by WMN. Furthermore, we propose an algorithm for load balancing.

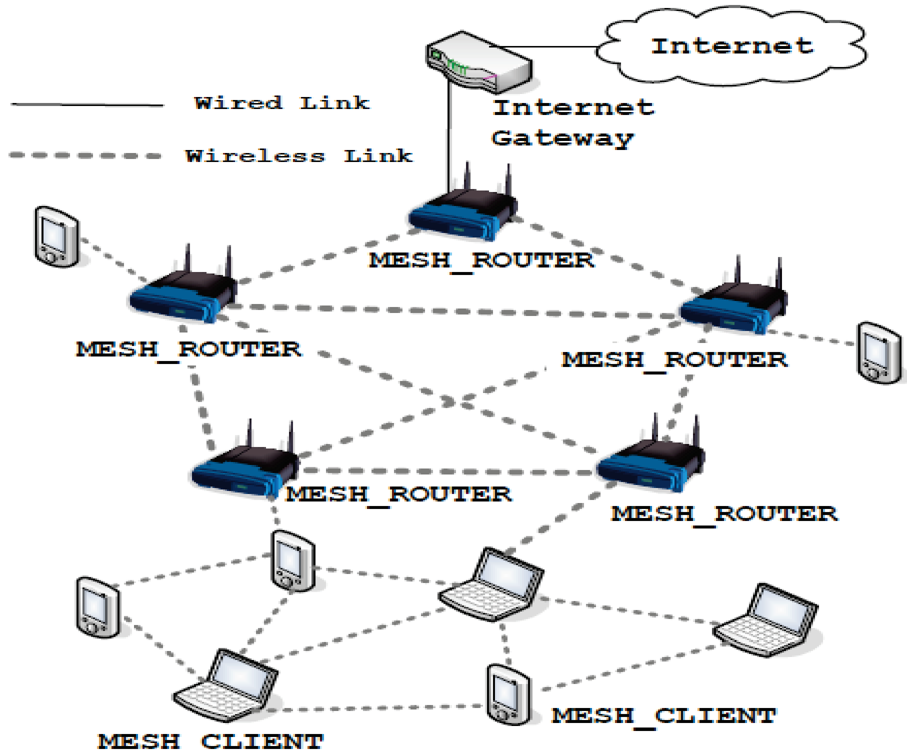

Fig 53. The Wireless Mesh Network architecture. (Marwaha et al., 2008)

IV.II. Routing in WMN: State of the Art

In a dynamic network where the topology can change randomly, maintaining the routing information up to date is a difficult task. This feature is commune between the MANET and the WMN which is why Ad-hoc routing protocols are applicable on WMNs. However, because of the limited resources of the mobile nodes, QoS guarantee mechanisms are difficult to integrate in MANET routing protocols. Hence, most of the routing protocols focus on speeding up the network convergence, lowering the overhead and path building / maintenance optimization.

Recently, many MANET routing protocols have been designed to support the QoS as discussed in the first chapter. However, WMNs contain MRs and MGs that can provide a better QoS to the clients. Subsequently, when a MANET routing protocol operates on a WMN, the MRs resources are not properly explored since the protocol does not treat the MRs any different than the clients. Thus, in order to optimize the usage of the WMN resources, many protocols have been proposed. For example:

a) Hybrid Wireless Mesh Protocol (HWMP) is a hybrid protocol specifically developed for this type of network (Kai et al., 2009). This protocol employs a reactive (Fig 52a) and a proactive (Fig 52b) path construction mode. Furthermore, path quality is measured by the AirTime Cost Metric (ATCM) (IEEE 802.11s Task Group, 2008). As a result, this protocol makes better use of the MRs during the path building process.

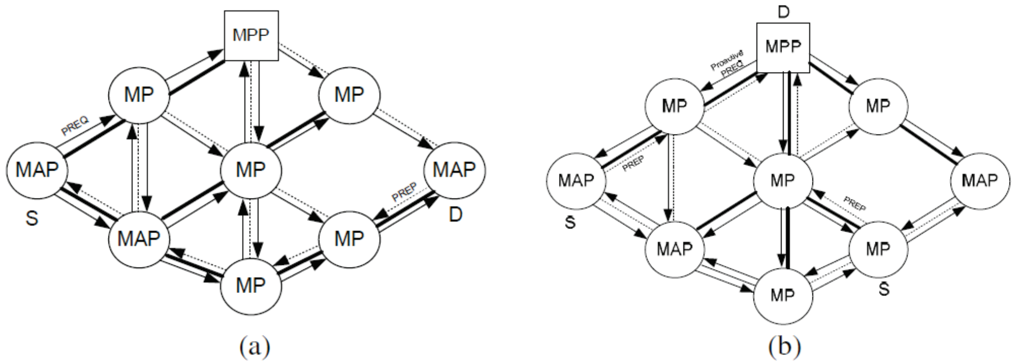

Fig 54. The two Hybrid Wireless Mesh Protocol modes. (Islam et al., 2009).

b) Likewise, Sen (2010) proposed a routing protocol for WMN that applies several methods in order to guarantee QoS for the MCs. This protocol estimates the link reliability by counting the number of control packets received from a neighbour during a period of time. Furthermore, the MPR method is used to reduce the overhead. In addition to the bandwidth estimation method, this protocol uses the available bandwidth on the wired network if there is a batter path through it.

c) Quality Of service Routing in wireless Mesh networks (QUORUM) applies a reactive path building process and a bandwidth reservation method (Kone et al, 2007). While running this protocol, the client nodes estimate the link quality with a neighbour by counting the number of received HELLO messages. This method enables the client to choose good quality links and thus, avoiding data retransmissions. QUORUM also calculates the data transfer delay through the use of DUMMY packets. Before staring the transmission, the source sends DUMMY packets towards the destination in order to verify the end to end delay. While performing this test, all DUMMY packets are set to have similar proprieties of the data packets. Another important mechanism in QUORUM is the management of misbehaving nodes. To avoid the negative impact of the misbehaving nodes on the network performance, QUORUM defines this type of node as the nodes that do not share any routing information. Accordingly, this type of node is excluded from the transmission paths and their data packets are not forwarded. In addition, to improve the QoS, this protocol applies an admission control mechanism during the reactive path discovery process. To do so, an intermediate node accepts a new data flow only if, the bandwidth required by the source, can be provided.

d) Wireless Mesh Routing (WMR) is a protocol similar to QUORUM that combines the proactive and the reactive approach (Xue and Ganz, 2002). This protocol is based on the periodic broadcast of the distance separating the client nodes from the MRs. Like QUORUM, this protocol uses admission control to guarantee the required bandwidth for the new data transfer. The novel approach of WMR consists of the calculation of the shared existing bandwidth between the node and all its direct neighbours in order to get a better estimation of the existing bandwidth.

e) Distributed end to end Allocation of Time Slots for Real time Traffic (DARE) is a protocol that does not depend on a specific routing technique (Carlson et al., 2006). This protocol is applied after that the path is established. In the purpose of avoiding signal interferences, DARE applies a time slot reservation in all the neighbours of a transmitting node. In order to reserve the necessary resources, the source sends a Request To Reserve (RTR) message containing the QoS constraints. All the intermediate nodes verify if the QoS requirements can be guaranteed before retransmitting the request. During this phase, the RTR is forwarded only if the requested resources are available. Otherwise, the RTR message is dropped. Then, when the destination is reached, it sends a Clear To Reserve (CTR) towards the source via the reserved path.

Each time a RTR is accepted, a new “preliminary” entry is created for the corresponding stream. This “preliminary” entry expires after a period of time if no corresponding CTR message is received. On the other hand, in case a CTR message is received, the status of the corresponding “preliminary” entry is changed to “fixed” which means that, the requested resources are reserved. After that, DARE reserves a time slot interval in each intermediate node and all its neighbours. To achieve this, all the nodes within interference range, receive also a RTR message to reserve a specific time slot. In case a time slot is already reserved in a neighbour, a request to update the reserved time slot, is generated in order to communicate the available time slot. Afterwards, during the reserved time slot, the neighbours of the transmitting / intermediate nodes do not transmit their data.

f) Multi-path Multi-gateway Anycast WMN routing protocol Based on Ant colony (MMAMBA) is a novel routing protocol designed for WMN (Song et al., 2010). This protocol incorporates the ACO technique. Basically, the designers of this protocol apply the ant seeking mechanism to search for the best path. To do so, a modified version of AOMDV is incorporated in this protocol to carry out the artificial ants.

g) Furthermore, in Gateway-Centralized Multi-hop Routing protocol (GCMR), the MGs apply a proactive approach to build paths to all the clients (Zhao et al., 2010). Based on: (a) QoS status of the nodes forming the paths, (b) average interference, (c) average distances between the nodes, GCMR estimates the changes in the QoS properties of a path. Thus, making this protocol decide which path is the most appropriate for a defined period of time. Furthermore, in order to lower the overhead, this protocol uses a TTL to limit the dissemination of routing messages. Performance estimation of this protocol showed a noticeable improvement over HWMP in terms of: (a) delivery ratio, (b) average end to end delay, (c) average jitter. Table 18 is a recapitulation of the protocols discussed in this section.

Table 18. Summary of the protocols designed for WMN.

Protocol	Type	Main features
a) HWMP	Hybrid	• Periodic broadcast of the paths leading to the GWs to speedup path construction and to optimise the use of the MRs / GWs.
b) TORP	Hybrid	• Link quality estimation through routing packets analysis. • Overhead reduction. • Bandwidth estimation.
c) QUORUM	Reactive	• Link quality estimation and retransmission avoidance. • End to end delay estimation. • Malicious nodes management
d) WMR	Hybrid	• Bandwidth reservation. • Shared bandwidth estimation.
e) DARE	Independent mechanism	• Time-slot reservation in a two hop radius for interference / transmission conflict avoidance.
f) MMAMBA	Reactive	• QoS estimation by applying the ACO algorithm.

g) GCMR	Hybrid	• Link quality prediction over a period of time.

IV.III. Load balancing: State of the Art

Forwarding the data packets through the closest MR / MG leads inevitably to network congestion. Understandably, a fair distribution of the clients data flows on the MRs and MGs is also important to improve the network performance. Thus, multiple propositions were made to distribute the traffic load evenly. For instance:

a) Load Balancing in Multi Gateway WMN (LBMGW) is a metric that was proposed to select the less loaded accessible MG (Mohammad et al., 2013). The main idea of this metric is to combine hop count, delay and gateway load during the selection of a MG for a new transmission.

b) Likewise, New Metric for HWMP (NMH) is a metric that takes into account the two hop density of the channel and the estimated delay per hop (Sidi et al., 2013).

c) Moreover, the Gateway Load-Balancing (GWLB) protocol was designed to insure load balancing in a distributed fashion (Juan et al., 2008). This protocol uses the Weighted Cumulative Expected Transmission Time Load Balancing (WCETT-LB) metric which is a congestion aware routing metric implemented in the MR (Liang and Denko, 2007).

d) The load balancing method proposed by Shahverdy et al. (2011) is based on a clustering approach. To avoid turning a MR / MG into a congestion point, this method splits the cluster into two new clusters (Fig 55) when the MR workload attains a predefined threshold. Though dividing a cluster can be time consuming, this approach can prevent QoS degradation in some situations. To limit the probability of a new cluster failure, this method takes in consideration several parameters during the election of the new gateway: (a) the node speed of movement, (b) remaining resources (energy, bandwidth and processing capacity), (c) location of the node. Afterwards, the gateways that have no workload are merged in one cluster. Thus, selecting one node responsible of coordinating the MCs traffic instead of several.

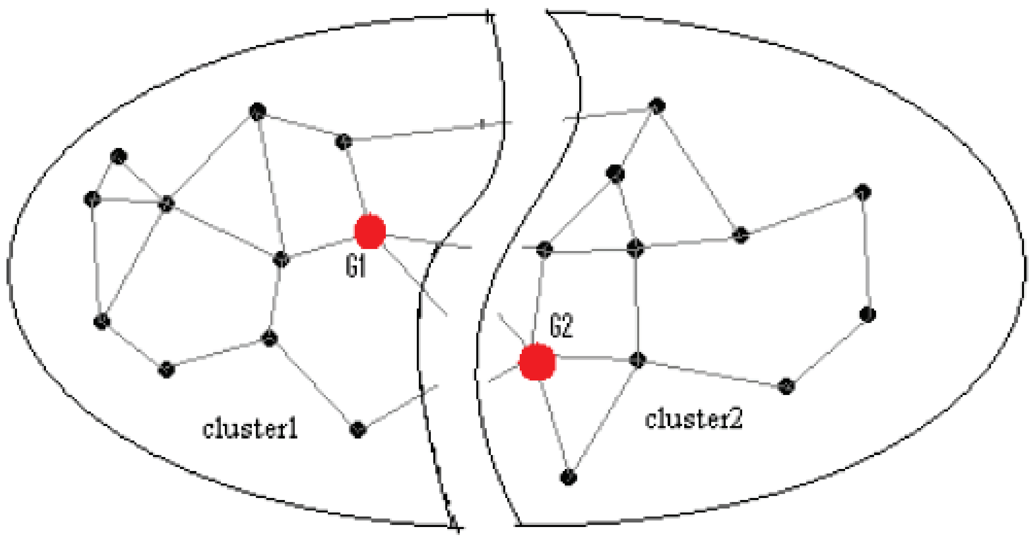

Fig 55. Breaking a Cluster. (Shahverdy et al., 2011)

e) Basically, to improve the network performance, the Multi Access Routers (MARs) protocol uses multiple MRs simultaneously for one data flow (Vinh et al., 2010).

f) Additionally, a load balancing method is used, called Reactive Load Balancing (RLB) proposed by Sensarma and Majumder (2013). To apply this method, the destination generates query packets and the intermediate nodes forward them to the source. Through examining the delay that occurs during the transmission of the query packets, the source can determine which path is less loaded. Unlike other similar methods where both the source and destination must generate query packets, this protocol produces a lower overhead by applying this approach only in one direction (from the source to destination). A summary of the load balance methods discussed in this section is presented in Table 19.

Table 19. Summary of the load balance methods designed for WMN.

Protocol	Type	Main features
a) LBMGW	Metric	• Combines hop count, delay and traffic load during path construction.
b) NMH	Metric	• Combines two hop radius channel density with delay per hop to avoid transmission conflicts.
c) GWLB	Protocol	• Featuring the WCETT metric which is derived from ETX. This metric combines the link reliability, transmission estimated time and the MR load.

d) Cluster based load balance	Protocol	• Splits a cluster to lighten the workload of a overloaded MR.
e) MARs	Protocol	• Parallel routing through multiple paths to promote load balance.
f) RLB	Protocol	• Path load estimation through end to end delay obtained from destination to source.

Whereas the MANET is the essence of the WMN, the existence of infrastructure makes this type of network suitable to guarantee a better QoS for the users. Thus, in this section, we discussed the additional strategies used in the WMN routing protocols to improve the overall QoS. Although the presence of MRs / MGs is an advantage, continuously attributing a new traffic to the closest MR eventually results into network congestion and QoS degradation. Thus, load balancing is a key method that not only promotes equality amongst the users, it also significantly enhance the general performance of the network. In the next section, we will present a load balancing algorithm based on the Genetic Algorithm (GA). The main idea is to create a two dimensional neighbourhood table in the MRs / MGs and search for alternative paths in order to avoid pairing a new connection with a highly loaded router. Furthermore, the GA is applied to extract the most suitable solution when a large number of alternative paths is possible.

IV.IV. A Load Balancing Algorithm For WMN Based On the Genetic Algorithm

Generally, guaranteeing the QoS in a WMN is relatively easier when the connexion requests are below the MR / MG capacities. On the other hand, when the number of connexions are overpowering, optimizing the network performance becomes a very complex problem. Furthermore, similarly to the MANET, the increase of the MCs number also expands the complexity of this problem due to the increase of the overhead and the mobility of the nodes.

As a part of our work, we proposed ERBOR which is a proactive routing protocol that generate a low volume of overhead. More importantly, the proposed protocol decreases the complexity of routing information treatment which is very useful in a dense topology. As a result, ERBOR can be used as a component of a WMN routing strategy. But, instead of maintaining the global topological information, ERBOR is used to maintain only local routing zones around the

MCs. Thus, further reducing the resources spent by the MCs on routing messages treatment. As for the MCs mobility issue, we illustrated how the reactive protocols handle this problem and how the pre-emptive strategy lowers the effect of the mobility on the protocol performance. Based on the results previously obtained, especially in the case of increasing the number of connexions, AOMDV is the most suitable solution for the reactive search procedure in WMN. The more pressing issue that we will discuss in this section is load balancing. The algorithm that we propose is inspired from the GA. Based on the routing information advertised by the MCs, the MR / MG computes possible alternative paths to connect the MC with the less loaded MR. Moreover, the GA is applied in case there's a lot of possible solutions.

IV.V. Overview on evolutionary computing

IV.V.I. Ant Colony Optimization

The ACO method is inspired from the behaviour of the ants while they are searching for a food source. At the beginning, the ant start by random walks from the nest. As the ants travel, they mark their paths with a quantity of pheromone that evaporates overtime. Once a food source is located, the ants start increasing the quantity of pheromone in the path relaying the nest to the food source. Subsequently, the rest of the ants tend to follow the path with the highest pheromone intensity.

For example, in Fig 56, A is the nest and B is the food source. Initially, the ants travel in a straight line from A to B (Fig 56a). Then, when an obstacle is placed between A and B (Fig 56b), the leading ants choose randomly which direction to take (E or F). The ants choosing the path AFB arrive at the food source faster then travel back to A, leaving a higher quantity of pheromones after a few iterations. Eventually, the intensity of the pheromone in the shorter path attracts the rest of the ants (Fig 56c). As this artificial intelligence method is appropriate for solving the optimal path problem, many QoS routing protocols applying this method have been proposed as mentioned in the first chapter. In computer science, this strategy is represented by a matrix of pheromone values that is iteratively updated based on the artificial ants movements and the pheromone evaporation parameter.

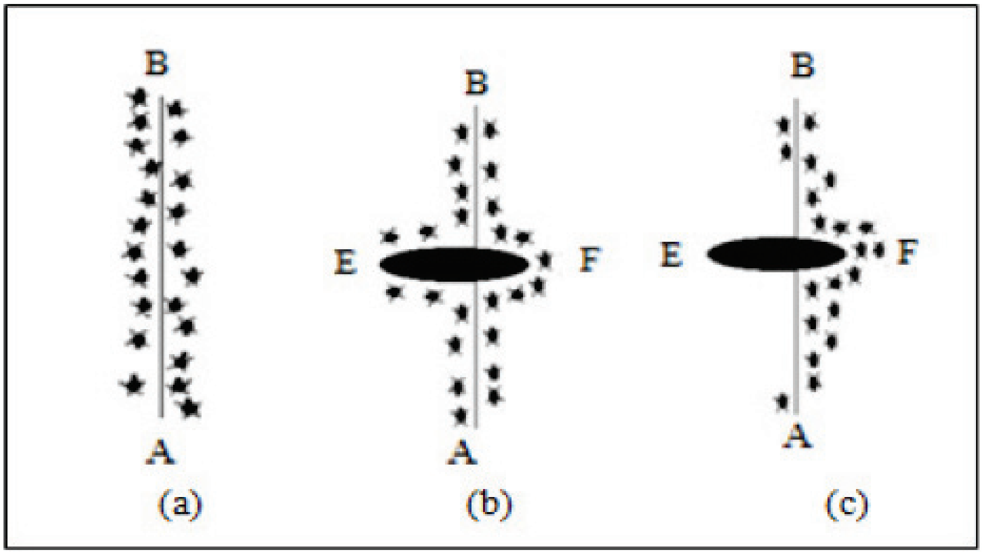

Fig 56. Path localization via the ACO method. (Sensarma and Majumder, 2013)

IV.V.II. Genetic Algorithm

The GA is a method inspired from the field of biology. This method is particularly used for solving optimization problems when a vast set of solutions is possible. To apply this algorithm, the solution must be represented as a chromosome. Afterwards, in order to produce new solutions, potentially better than the initial ones, two chromosomes are crossed. This process is repeated several times until the population stops changing.

For instance, this strategy can be used to solve the shortest path problem. In Fig 57, we consider a space with obstacles where the problem in hand is to find which path is the shortest from point A to B. At first, the algorithm starts by creating an initial population. In this case, the population is a set of paths leading from A to B. To do so, we choose a random point between A and B, called C, and create a path from A to C then from C to B. Thus, by repeating this action N times, we obtain an initial population with a size N.

Initially, the population is a set of random solutions. To produce shorter paths based on the initial population, we proceed with a mutation action from two of the initial solutions. The only condition for this operation to be possible, is that the two solutions must intersect in at least one point. After accomplishing the mutation, four new solutions appear that contain one fragment from each initial solution. Afterwards, the new solutions are added to the population ensemble. To maintain control over the population size, the set of solutions is sorted based on the length of the paths. For the next iteration, only the best N solutions are used and the rest of the population is removed. For example, in Fig 57, we create two random paths between A and B (Fig 57a and Fig 57b). To generate two new solutions, we perform a mutation of the two solutions based on point I. As a result, we obtain two new solutions as shown in Fig 58. Subsequently, the path obtained in Fig 58b is shorter than the initial paths.

Once the population stops changing, the algorithm is stopped. Although this strategy seams complex, the GA makes optimizing a set of solutions a lot simpler than exhaustively searching for the best solution in a large ensemble. In a work related to our field of study, Xhafa et al. (2010) applied the GA to optimize MR positioning. The purpose of this study was to find the near-optimal positioning of the MR in order to establish network connectivity and maximizing users coverage.

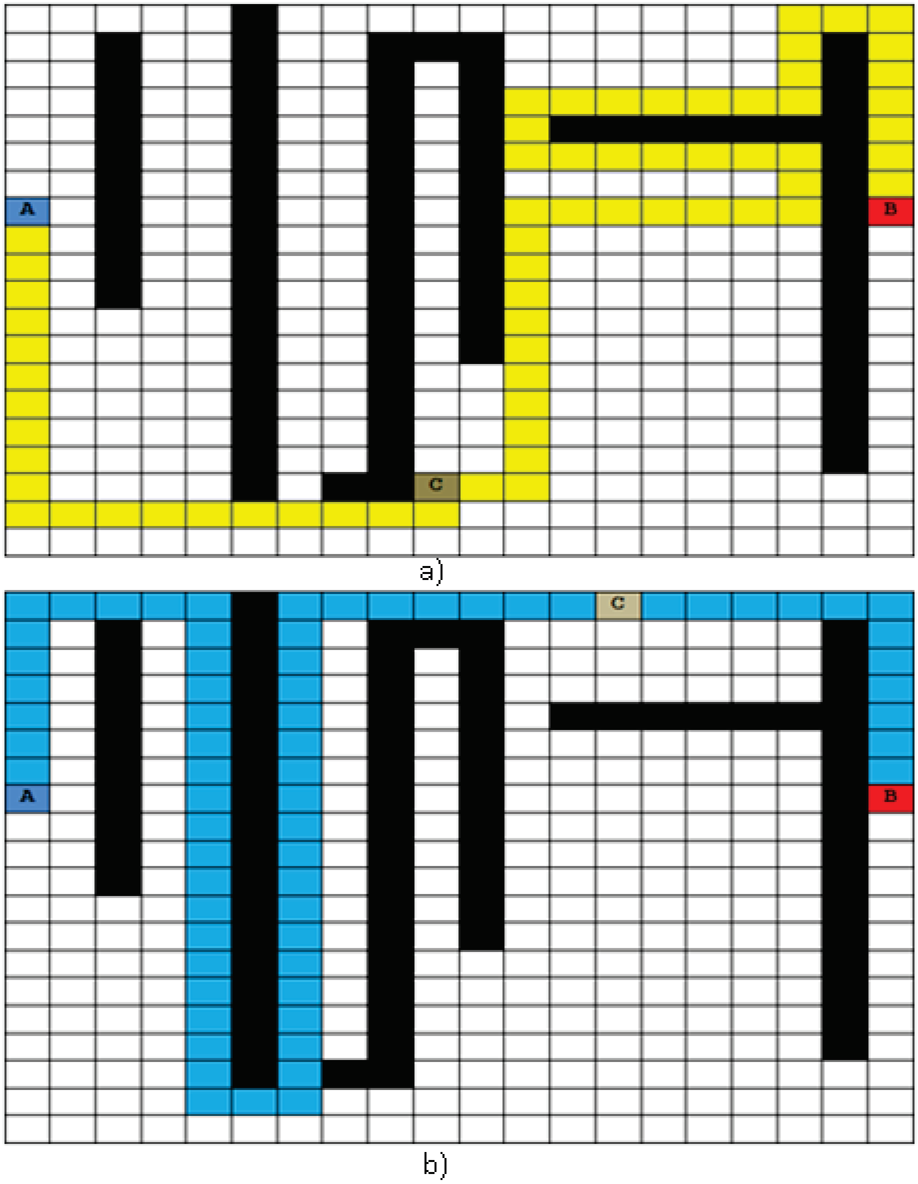

Fig 57. Initial solutions.

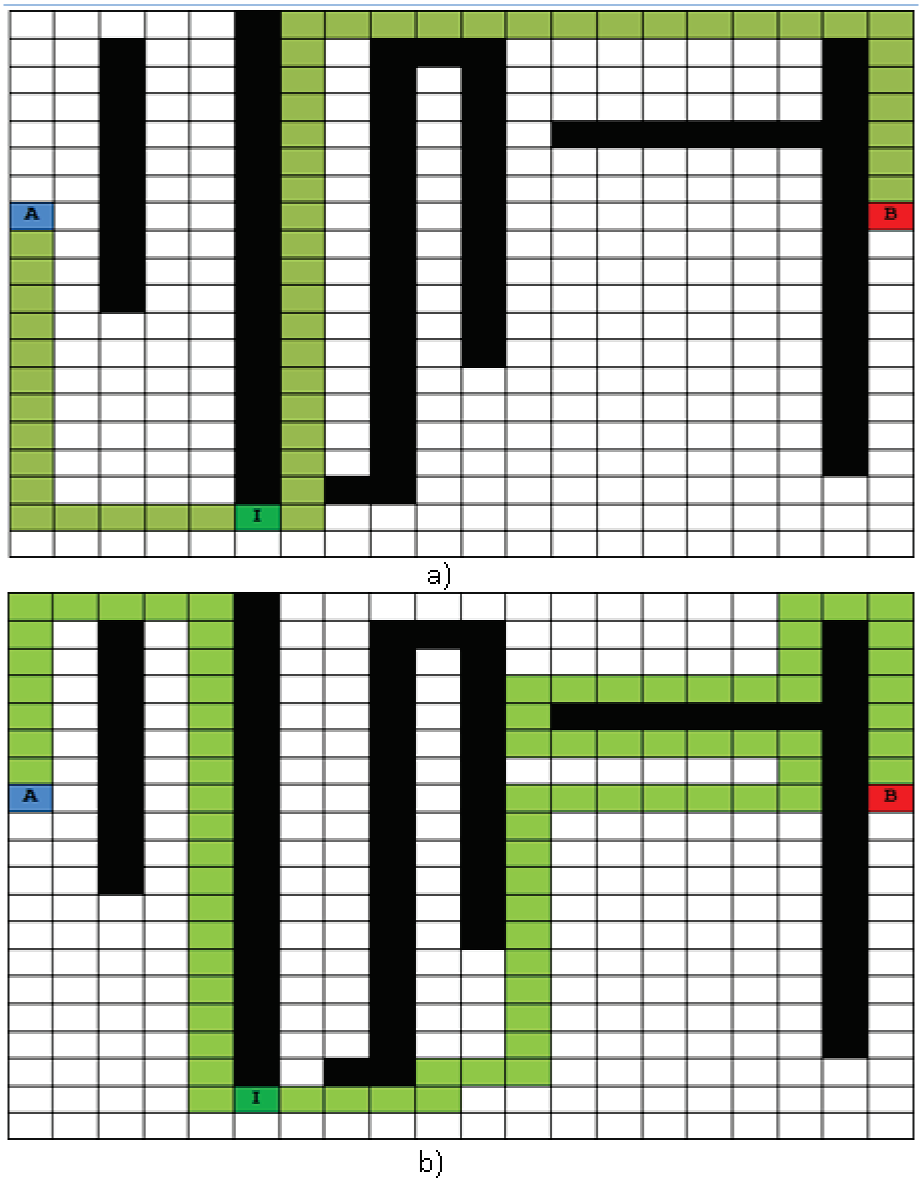

Fig 58. New solutions.

IV.V.III. Chemical Reaction Optimization

The Chemical Reaction Optimization (CRO) is an evolutionary method inspired from the chemical reaction process that was first used to improve the performance of Peer to Peer streaming systems (Albert and Victor, 2010). Similarly to the GA, the CRO is applied on a set of random solutions represented in a form of molecules. To create new solutions, a set of predefined actions are applied iteratively until the stopping criteria is met.

Two sorts of actions are applicable during this process: (a) exploitation actions in which the number of input molecules is the same number of outputs, (b) exploration actions in which the population size is either decreased or increased. To maintain an appropriate size of the population, the application of the two types of actions must be balanced. Similarly to the GA, the new solutions are a product of the molecular changes of the initial solutions. This method has been recently used in the work of Taisir et al. (2015) to select maximally distant code words.

IV.VI. Our Proposed Algorithm

Load balancing is a key mechanism that can improve the WMN performance significantly. Consistently forwarding the data traffic through the closest MRs leads eventually to network congestion and unfairness amongst the MCs. A possible solution for this problem is to compute alternative paths to forward the data flows without involving the loaded routers. Our idea is to construct a Topological Map (TM) in each router based on the neighbourhood tables of the MCs and MRs. To do so, the network members periodically broadcast HELLO messages containing the description of their neighbourhood. The collection of these HELLO messages form a TM that makes the router / gateway able to find alternative paths to the destination or to another less loaded router / gateway. The algorithm in Fig 59, illustrates how the alternative paths are created.

The algorithm that we propose, searches for alternative paths to the destination or to a reachable less loaded router. The alternative paths are formed by client / router nodes from which the router can receive HELLO messages. Thus, before launching a reactive search for the destination, the router uses the TM in order to compute a path towards the destination. This mechanism eliminates the unnecessary forwarding of data by the router when the destination can be reached with a multi-hop path formed by clients. The same procedure is applied to build a path towards other routers / gateways. Accordingly, if a router is loaded and cannot forward a new traffic as in Fig 60, a path towards another router is calculated and sent to the client.

```
private void Explore_Topology_Map(Node Current_Node, IpAddress Destination_ID,Vector Explored_Path,
        Vector Unexplored_TM){
int i,n;
IpAddress Current_Node_ID=Current_Node.getIpAddress();
Vector New_Path=(Vector) Explored_Path.clone();
Vector Saved_Image_Of_TM=new Vector();
Vector Neighbours_List=getNeighbours(Current_Node_ID,Unexplored_TM);
Saved_Image_Of_TM.addAll(Unexplored_TM);
 if(Neighbours_List.size()!=0)New_Path.add(Current_Node);
 else return ;
if(isNeighbour(Destination_ID,Neighbours_List)){
                                        New_Path.add(Current_Node);
                                        Path_ID++;
                                        Path_List.add(new Alternative_Path(New_Path,Path_ID));
                                        return ;
                                        }

n=Neighbours_List.size();
int j;
for(i=0;i<n;i++) {
        for(j=0;j<n;j++)
              if(i!=j) getNeighbours(Current_Node_ID,Saved_Image_Of_TM);
              Path_ID++;
              Explore_Topology_Map( (Node) Neighbours_List.elementAt(i),Destination_ID,New_Path,Saved_Image_Of_TM);
              Saved_Image_Of_TM=new Vector();
              Saved_Image_Of_TM.addAll(Unexplored_TM);
        }
}
```

Fig 59. Exploring the TM for alternative paths.

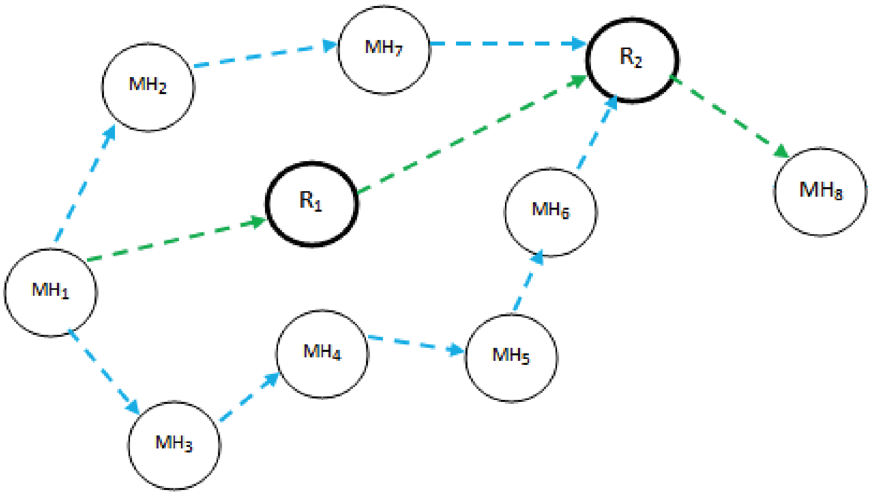

Fig 60. The creation of alternative paths.

The difference between the TM and the routing table is that the routing table contains the paths towards all the possible destinations. On the other hand, the TM is formed by collecting only the partial routing information. Moreover, in order to maintain the routing table up to date, a periodic exchange / processing of the global routing information is necessary whereas while updating the TM, the neighbourhood tables received are only stored. For a better path optimization however, the TM must be periodically refreshed and the algorithm is ran to search for better paths. Furthermore, the size of the TM depends on the radius of the neighbourhood contained in the HELLO messages. Evidently, increasing the neighbourhood radius can expand the number of alternative paths. However, the growth of the TM size also increases the complexity of path calculation.

The execution of the algorithm begins initially with an empty ensemble of paths and the starting point which is the data source. Then, the program explores the neighbours of the data source searching for the destination. The procedure is repeated on every node accessible and after exploring a node, it is deleted from the copy of the TM which the algorithm uses on the subsequent neighbours. Subsequently, every time the destination is found in a neighbourhood table, a new path is added to the list of alternative paths. The result is a list of paths leading to the destination and if no path is found, the client launches a path construction process.

In addition to the neighbourhood radius, the quality of the solutions calculated by the algorithm depends on the metrics advertised by the members of the network. Since the purpose of this algorithm is load balancing, the HELLO message must contain at least the traffic load (remaining buffer) of the corresponding node. Besides, Path length is automatically deduced. For example, the available bandwidth or the mobility state can also be included. Obviously, it is simple to sort the alternative paths based only on one aspect. On the other hand, when the number of alternative solutions expand and the quality of the path depends on multiple aspects, identifying the most suitable path becomes much more complex. Thus, in Fig 62 and Fig 63, we propose a multi criteria sorting algorithm to rank the possible solutions. Furthermore, the algorithm in Fig 61 is based on the GA. In this procedure, the best possible solutions based on each criteria are transformed by mutations in order to search for new possible paths. Applying this artificial intelligence method is useful when there is a wide number of potential solutions.

```java
private void Apply_Genetic_Algorithm(Vector Based_Traffic_Load,Vector Based_Path_Length
        ,Vector Based_Bandwidth,double Genetic_Print){
Vector Mutated_Solutions;
double New_Genetic_Print;
int i,j,n,Initial_Population_Size=Based_Traffic_Load.size();
Alternative_Path Initial_Solution,Solution_Based_TL,Solution_Based_PL,Solution_Based_Bw;
do{
New_Genetic_Print=Genetic_Print; Genetic_Print=New_Genetic_Print;
Mutated_Solutions=new Vector();
for(i=0;i<Initial_Population_Size;i++){
    Initial_Solution=(Alternative_Path)Based_Traffic_Load.elementAt(i);
    for(j=0;j<Initial_Population_Size;j++){
        Solution_Based_PL=(Alternative_Path)Based_Path_Length.elementAt(j);
        Solution_Based_Bw=(Alternative_Path)Based_Bandwidth.elementAt(j);
        if(Initial_Solution.Intersect(Solution_Based_PL).size()!=0  &&
                !ID_is_Equal(Initial_Solution,Solution_Based_PL))
        Mutated_Solutions.addAll(Mutate(Initial_Solution,Solution_Based_PL));
        if(Initial_Solution.Intersect(Solution_Based_Bw).size()!=0 &&
                !ID_is_Equal(Initial_Solution,Solution_Based_Bw))
        Mutated_Solutions.addAll(Mutate(Initial_Solution,Solution_Based_Bw));
        Initial_Solution=(Alternative_Path)Based_Path_Length.elementAt(i);
        if(Initial_Solution.Intersect(Solution_Based_Bw).size()!=0 &&
                !ID_is_Equal(Initial_Solution,Solution_Based_Bw))
        Mutated_Solutions.addAll(Mutate(Initial_Solution,Solution_Based_Bw));
    }
}
n=Mutated_Solutions.size();
Alternative_Path New_Solution;
for(i=0;i<n;i++){
    New_Solution=(Alternative_Path)Mutated_Solutions.elementAt(i);
    for(j=0;j<Initial_Population_Size;j++){
        Solution_Based_TL=(Alternative_Path)Based_Traffic_Load.elementAt(j);
        if(Solution_Based_TL.getMaxTL()>New_Solution.getMaxTL()){
            New_Genetic_Print+=-Solution_Based_TL.getTotalScore()+New_Solution.getTotalScore();
            Based_Traffic_Load.setElementAt(New_Genetic_Print,j);break;
        }
    }
    for(j=0;j<Initial_Population_Size;j++){
        Solution_Based_PL=(Alternative_Path)Based_Path_Length.elementAt(j);
        if(Solution_Based_PL.getLength()>New_Solution.getLength()){
            New_Genetic_Print+=-Solution_Based_PL.getTotalScore()+New_Solution.getTotalScore();
            Based_Path_Length.setElementAt(New_Solution,j);break;
        }
    }
    for(j=0;j<Initial_Population_Size;j++){
        Solution_Based_Bw=(Alternative_Path)Based_Bandwidth.elementAt(j);
        if(Solution_Based_Bw.getAvailableBw()<New_Solution.getAvailableBw()){
            New_Genetic_Print+=-Solution_Based_Bw.getTotalScore()+New_Solution.getTotalScore();
            Based_Bandwidth.setElementAt(New_Solution,j);break;
        }
    }
}
}while(New_Genetic_Print!=Genetic_Print);
}
```

Fig 61. Applying the genetic algorithm.

```
private Alternative_Path Multi_Criteria_Sort(Vector Path_List){
    if(Path_List.size()==0)return null;
Vector Based_Traffic_Load,Based_Path_Length,Based_Bandwidth;
Based_Traffic_Load=new Vector();Based_Path_Length=new Vector();Based_Bandwidth=new Vector();
Alternative_Path Solution, Best_Solution;
double Genetic_Print=0;
int Initial_Population_Size=10;
int i,j,n=Path_List.size();
for(i=0;(i<Initial_Population_Size)&&(i<n);i++){
Best_Solution=(Alternative_Path)Path_List.elementAt(i);
Based_Traffic_Load.add(Best_Solution);Based_Path_Length.add(Best_Solution);Based_Bandwidth.add(Best_Solution);
for(j=1;j<n;j++){
    Solution=(Alternative_Path)Path_List.elementAt(j);
    if(Solution.getMaxTrafficLoad()<Best_Solution.getMaxTrafficLoad())Based_Traffic_Load.setElementAt(Solution,i);
    if(Solution.getLength()<Best_Solution.getLength())Based_Path_Length.setElementAt(Solution,i);
    if(Solution.getAvailableBandwidth()>Best_Solution.getAvailableBandwidth())Based_Bandwidth.setElementAt(Solution,i);
    }
Genetic_Print+=((Alternative_Path)Based_Traffic_Load.elementAt(i)).getTotalScore()+((Alternative_Path)Based_Path_Length.
        elementAt(i)).getTotalScore()+((Alternative_Path)Based_Bandwidth.elementAt(i)).getTotalScore();
}
Apply_Genetic_Algorithm(Based_Traffic_Load,Based_Path_Length,Based_Bandwidth,Genetic_Print);
return getOptimizedSolution(Based_Traffic_Load,Based_Path_Length,Based_Bandwidth);
}
```

Fig 62. Multi criteria path selection method.

```
Alternative_Path getOptimizedSolution(Vector Based_Traffic_Load,Vector Based_Path_Length,Vector Based_Bandwidth){
    double Lowest_Traffic_Load,Largest_Bandwidth;
    int Lowest_Length,i,n=Based_Traffic_Load.size();
    Lowest_Traffic_Load=((Alternative_Path)Based_Traffic_Load.elementAt(0)).getMaxTL();
    Lowest_Length=((Alternative_Path)Based_Path_Length.elementAt(0)).getLength();
    Largest_Bandwidth=((Alternative_Path)Based_Bandwidth.elementAt(0)).getAvailableBw();
    Alternative_Path Highest_Ranked_Solution=(Alternative_Path)Based_Traffic_Load.elementAt(0);
    Highest_Ranked_Solution.getRank(Lowest_Traffic_Load,Largest_Bandwidth, Lowest_Length);
    /*getRank(Lowest_Traffic_Load,Largest_Bandwidth, Lowest_Length)= ((100*Lowest_Traffic_Load/getMaxTrafficLoad())
     * +(100*Lowest_Length/getLength())+(100*getAvailableBandwidth()/Largest_Bandwidth))/3;*/
for(i=0;i<n;i++){
   if(((Alternative_Path)Based_Traffic_Load.elementAt(i)).getRank(Lowest_Traffic_Load,Largest_Bandwidth, Lowest_Length)
           >Highest_Ranked_Solution.rankValue())Highest_Ranked_Solution=(Alternative_Path)Based_Traffic_Load.elementAt(i);
   if(((Alternative_Path)Based_Path_Length.elementAt(i)).getRank(Lowest_Traffic_Load,Largest_Bandwidth, Lowest_Length)
           >Highest_Ranked_Solution.rankValue())Highest_Ranked_Solution=(Alternative_Path)Based_Path_Length.elementAt(i);
   if(((Alternative_Path)Based_Bandwidth.elementAt(i)).getRank(Lowest_Traffic_Load,Largest_Bandwidth, Lowest_Length)
           >Highest_Ranked_Solution.rankValue())Highest_Ranked_Solution=(Alternative_Path)Based_Bandwidth.elementAt(i);
}
    return Highest_Ranked_Solution;
}
```

Fig 63. Path ranking method.

IV.VII. Conclusion

The algorithm that we propose in this work, uses the partial routing information collected by the routers / gateways, to distribute the traffic load fairly and avoid network congestion. The algorithm can also be used with different path quality measurement like the signal strength, remaining energy and link reliability (ETX for instance). Moreover, with the collection of multiple alternative paths, it is possible to rank the paths based on multiple metrics simultaneously. A simple way to choose a balanced solution, is to rank the alternative paths based on each criteria separately. Afterwards, the GA can be applied to mutate the best solutions in each category. Although, the complexity and the overhead of the algorithm seems significant, the routing strategy that we proposed in the previous chapter limits the repercussions of this problem. Finally, besides being inspired from the GA and collecting a two dimension topological map, the novelty of the proposed algorithm is its ability to choose a balance solution for multiple metrics.

General Conclusion

As we enter the wireless broadband era, finding a way to improve the offered QoS for applications has become quickly one of the main challenges. In this thesis, we propose a vast study to show the major impact of the routing protocols developed for MANETs and WMNs from a QoS perspective. While countless contributions have been tailored for this purpose, the complex nature of the dynamic routing problem makes the performance of the algorithms situational. Which is why each protocol is developed to solve a specific problem. Our target was mainly to lower the proactive routing complexity.

As explained, the random mobility issue is the most important problem that all this routing protocol family must be accustomed to. From a PDR perspective, though the reactive protocols usually handle this issue better, the mechanisms applied by this category of protocol (multi-path, packet salvaging, local path repair, etc...) have undesirable side effects (overhead / end to end delay increase or PDR decrease). Based on our analysis, the path failure prediction algorithm is the most appropriate solution to improve a reactive protocol performance from this point of view. Unlike other methods proposed to deal with the mobility problem, this strategy relies less on the mobility history of the nodes.

The most complex issue that we attempted to solve in this work is the proactive routing problem. As it is stated in the literature and showed through the simulation results, the proactive routing complexity / overhead is strongly correlated with the network size. Unlike other proactive protocols, our proposed algorithm doesn't process all the received routing messages and broadcast the entire routing table only when an important link is established. The low complexity of ERBOR makes it potentially suitable for low capacity devices composing a large network.

As for the WMN routing problem, insuring a proper QoS is effected by the same MANET routing issues. The two previous proposed routing algorithms can easily be fit to compound a hybrid / hierarchical WMN routing protocol: (a) ERBOR to lower the complexity / overhead of maintaining a local routing zone, (b) the reactive mobility aware extension to lower the packet losses. More importantly, the presence of the MGs

/ MRs offers physical support to enhance the QoS. When there are not too many transfer requests, guaranteeing the best QoS to a MC is merely a matter of attributing it to the closest MR / MG. Otherwise, when the number of data flows is overpowering, load balancing is the key method to fairly serving all the demands. Thus, we proposed an algorithm to deviate the traffic from the loaded MR / MG. The novelty of our approach is that it brings a two dimensional awareness of the neighbourhood for the MG / MR.

Perspectives

As illustrated in the simulation results, although AOMDV produces better results when the number of data streams are increased, this protocol is the most negatively affected by the increase of the nodes movement speed. Which is why we strongly believe that the mobility extension can significantly enhance this protocol performance. On the other hand, the load balance algorithm raises potential complexity issues.

The work done during this thesis and the obtained results open the door to a range of perspectives. Amongst them, we can mention:

a) Mix the link breakage prediction mechanism with the mechanisms of ERBOR in a single routing protocol for WMN. Then, applying the load balancing mechanism to test the effectiveness of all the proposed methods together.

b) The use of the GA for load balancing may be replaced by other bio-inspired algorithms to determine which would be most efficient.

c) Make tests for each type of online applications (VoIP, VoD, Games ...) with their different needs in terms of QoS, to determine the efficiency of the existing routing protocols.

Abbreviations

ABC	Artificial Bee Colony
ABR	Associativity Based Routing
ACO	Ant Colony Optimization
AMAODV	Adaptive Mobility aware AODV
AODV	Ad hoc On demand Distance Vector
AODV-ALMA	AODV Ant Like Mobile Agents
AOMDV	Ad hoc On demand Multipath Distance Vector
ATCM	AirTime Cost Metric
BATMAN	Better Approach To Mobile Ad hoc Networking
CBR	Constant Bit Rate
CBRP	Cluster Based Routing Protocol
CH	Cluster Head
CRO	Chemical Reaction Optimization
CT	Critical Threshold
CTR	Clear To Reserve
DARE	Distributed end to end Allocation of Time Slots for Real time Traffic
DBF	Distributed Belman Ford
DREAM	Distance Routing Effect Algorithm for Mobility
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
DUT	Delayed Update Table
ERBOR	Effective Routing Based on Overhead Reduction
FSR	Fisheye State Routing
GA	Genetic Algorithm
GCMR	Gateway-Centralized Multi-hop Routing
GW	Gateway
GWLB	Gateway Load-Balancing
HSR	Hierarchical State Routing
HWMP	Hybrid Wireless Mesh Protocol
LAR	Location Aided Routing
LBAOMDV	Load Balancing AOMDV
LBMGW	Load Balancing in Multi Gateway WMN
LET	Link Expiration Time
LS	Link State
MANET	Mobile Ad hoc Network
MAR	Mobility Adjustment Routing
MARs	Multi Access Routers
MC	Mesh Client
MG	Mesh Gateway
MMAMBA	Multi-path Multi-gateway Anycast WMN routing protocol Based on Ant colony

MOAODV	Multi-Objective AODV
MPR	Multipoint Relay
MR	Mesh Router
MSS	Minimum Signal Strength
NMH	New Metric for HWMP
NS2	Network Simulator 2
NUNBC	Novel Unique Node Based Clustering
NXUT	Next Update Table
OBSUR	On-demand Bandwidth and Stability based Unicast Routing
OGM	Originator Message
OLSR	Optimized Link State Routing
PCAODV	Power Control AODV
PDR	Packet Delivery Ratio
PMQRPMP	Power aware Multiple QoS constraints Routing Protocol with Mobility Prediction
PRP	Parallel Routing Protocol
QoS	Quality of Service
QUORUM	Quality Of service Routing in wireless Mesh networks
RLB	Reactive Load Balancing
RRAODV	Randomized Reverse AODV
RREP	Route Replay
RREQ	Route Request
RT	Routing Table
RTBD	Record and Trust-Based Detection
RTR	Request To Reserve
RTT	Round Trip Time
SSA	Signal Stability-Based Adaptive Routing
TM	Topological Map
TORA	Temporally Ordered Routing Algorithm
TTL	Time To Live
UDP	User Datagram Protocol
UML	Unified Modelling Language
WCETT-LB	Weighted Cumulative Expected Transmission Time Load Balancing
WMN	Wireless Mesh Network
WMR	Wireless Mesh Routing
ZHLS	Zone Based Hierarchical Link State
ZRP	Zone Routing Protocol

Appendix

Network Simulator 2

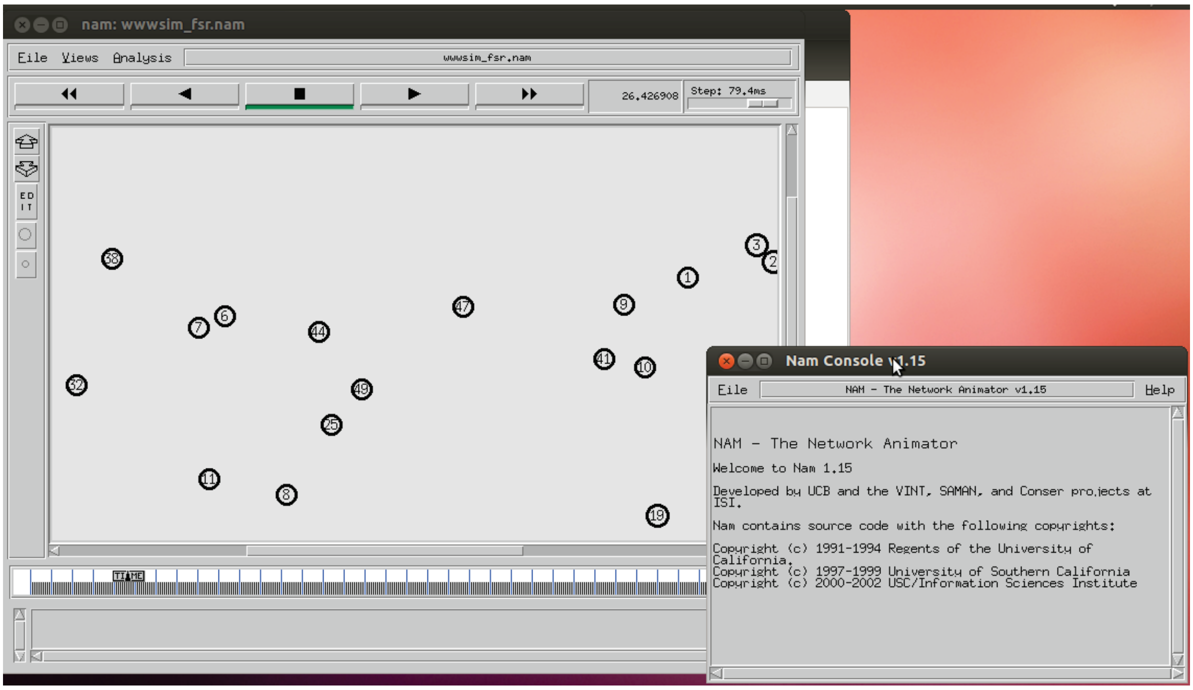

Fig 64. Network Simulator 2.

Network Simulator 2 (NS) is an open software used by researchers to create a virtual network and replicate the behaviour of it components. This simulation tool was written in C programming language in 1995. In order to create a virtual network, the users must define the conditions of the simulation and the components of the network. The following script written in TCL is an example of how to simulate a MANET composed of 100 nodes.

MANET Simulation Script

set val(chan) Channel/WirelessChannel ;# channel type

set val(prop) Propagation/TwoRayGround ;# radio-propagation model

set val(netif) Phy/WirelessPhy ;# network interface type

set val(mac) Mac/802_11 ;# MAC type

set val(ll) LL ;# link layer type

set val(ant) Antenna/OmniAntenna ;# antenna model

set val(ifqlen) 50 ;# max packet in ifq

```
set val(rp) DSDV ;# routing protocol
set val(seed) 0.0 ;#
if { $val(rp) == "DSR" } {
   set val(ifq) CMUPriQueue
} else {
   set val(ifq) Queue/DropTail/PriQueue
}
set val(x) 1500 ;# X dimension of the topography
set val(y) 1500 ;# Y dimension of the topography
set val(stop) 200.0 ;# simulation time
set val(path) /home/acharya/ns-allinone-2.35/ns-2.35
set val(nn)          100          ;# how many nodes are simulated
set val(cp)          "cbr-50-test"
set val(sc)          "p1maxsp1scen-100-test_1000x1000"

Phy/WirelessPhy set CPThresh_ 10.0 ;# Capture threshold (db)
Phy/WirelessPhy set CSThresh_  2.82e-9 ;#Carrier sense threshold(W)
Phy/WirelessPhy set RXThresh_  2.82e-9 ;# Receive power threshold(W)
Phy/WirelessPhy set Rb_ 2*1e6 ;# Bandwidth
Phy/WirelessPhy set Pt_ 0.2818 ;# Transmission power (W)
Phy/WirelessPhy set freq_ 914e6 ;# frequency
Phy/WirelessPhy set L_ 1.0 ;# system loss factor

set ns_ [new Simulator]
set tracefd [open sim_dsrnet.tr w]
```

```
set god_ [create-god $val(nn)]
$ns_ trace-all $tracefd
$ns_ use-newtrace
# set the new channel interface.
#Open the nam file
set namtrace [open confout.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
#Set up topography object to keep track of movement of nodes
set topo [new Topography]
#Provide topography object with coordinates
$topo load_flatgrid $val(x) $val(y)
proc finish {} {
   global ns_   tracefd namtrace
   $ns_ flush-trace
   close $tracefd
   close $namtrace
   exec ./nam confout.nam &
   exit 0
}
create-god $val(nn)
$ns_ node-config -adhocRouting $val(rp) \
   -llType $val(ll) \
   -macType $val(mac) \
   -ifqType $val(ifq) \
   -ifqLen $val(ifqlen) \
```

```
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -channelType $val(chan)\
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -movementTrace ON
for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0 ;# disable random motion
}
puts "Loading connection pattern..."
source $val(cp)
puts "Loading scenario file..."
source $val(sc)
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ initial_node_pos $node_($i) 50
}
for {set i 0} {$i < $val(nn) } {incr i} {
    $ns_ at $val(stop).0 "$node_($i) reset";
}
$ns_ at $val(stop).0001 "finish"
$ns_ at $val(stop).0002 "puts \"NS EXITING...\" ; $ns_ halt"
```

```
puts $tracefd "Confidant Wrote this!"

puts $tracefd "M 0.0 nn $val(nn) x $val(x) y $val(y) rp $val(rp)"

puts $tracefd "M 0.0 sc $val(sc) cp $val(cp) seed $val(seed)"

puts $tracefd "M 0.0 prop $val(prop) ant $val(ant)"

puts "Starting Simulation..."

$ns_ run
```

References

Abbas, N. and Hamid, S.A. (2012) 'Analysis of link lifetime in wireless mobile networks', Ad Hoc Networks, Vol. 10 No. 7, pp.1221-1237.

Abdelali, B. et al, (2015) 'Multi-point relay selection strategies to reduce topology control traffic for OLSR protocol in MANETs', *Journal of Network and Computer Applications*, Vol. 53, pp.91-102.

Afzal, M. and Hossam, H. (2011) 'Efficient content distribution for peer-to-peer overlays on mobile ad hoc networks', Journal of Advanced Research, Vol. 2 No. 3, pp. 265-279.

Alagiri, I., Madhuviswanatham, V. and Krishna, P.V. (2014) 'Efficient Data Transfer by Mobility Adjustment Algorithm for Clustered Mobile Ad-Hoc Networks', Cybernetics and Information Technologies, Vol. 14 No. 2, pp.50-64.

Albert, Y.S.L. and Victor, O.K.L. (2010) 'Chemical-Reaction-Inspired Metaheuristic for Optimization', IEEE Transactions on Evolutionary Computation, Vol. 14 No. 3, pp.381-399.

Alghamdi, S.A. (2015) 'Load balancing ad hoc on-demand multipath distance vector (LBAOMDV) routing protocol', EURASIP Journal on Wireless Communications and Networking, Vol. 2015 Article No. 242, pp.1-11.

Ali, M. and Abdallah, B. (2015) 'A survey of routing protocols based on link-stability in mobile ad hoc networks', Journal of Network and Computer Applications, Vol. 47, pp.1-10.

Ali, M., Fouzi, S. and Abdallah, B. (2014) 'A link-state QoS routing protocol based on link stability for Mobile Ad hoc Networks', Journal of Network and Computer Applications, Vol. 39, pp.117-125.

Alnabhan, M. et al, (2014) 'Emerging Trends in Intrusion Detection in Mobile ad hoc Networks (MANETs)', Information Technology Journal, Vol. 13 No. 13, pp.2065-2078.

Alshanyour, M. and Baroudi, U. (2008), 'Bypass AODV: Improving Performance of Ad Hoc On-Demand Distance Vector (AODV) Routing protocol in wireless ad hoc networks' In AMBI-SYS 2008: Proceedings of the 1st International Conference on Ambient Media and Systems, Quebec, Canada, No. 17,pp.1-8.

Ayyadurai, V. and Ramasamy, R. (2008) 'Internet Connectivity for Mobile Ad Hoc Networks Using Hybrid Adaptive Mobile Agent Protocol', The International Arab Journal of Information Technology, Vol. 5 No. 1, pp.25-33.

Basagni, S. et al (1998), 'A Distance Routing Effect algorithm for mobility' In MobiCom 1998 Proceedings of the 4th annual ACM/IEEE International Conference On Mobile Computing And Networking, Dallas, USA, pp.76-84.

Basarkod, P.I. and Manvi, S.S. (2014) 'On-demand Bandwidth and Stability Based Unicast Routing in Mobile Adhoc Networks', International Journal of Electronics and Telecommunications, Vol. 60 No. 1, pp.27-39.

Bisengar, A., Rziza, M. and Ouadou, M. (2012) 'Queue Length and Mobility aware Routing Protocol for Mobile Ad hoc Network. Networks', International Journal of Communication Networks and Information Security, Vol. 4 No. 3, pp.207-211.

Boukerche A. and Zhang, L.(2004) 'A performance evaluation of a pre-emptive on-demand distance vector routing protocol for mobile ad hoc networks', Wireless Communications and Mobile Computing, Vol 4 No. 1, pp.99-108.

Carlson, E. et al (2006) 'A Distributed End-to-End Reservation Protocol for IEEE 802.11-Based Wireless Mesh Networks', Selected Areas in Communications IEEE Journal. Vol. 24 No. 11, pp.2018-2027.

Che-Liang, L., Chih-Yu, W. and Hung-Yu, W. (2010) 'Cross-layer Mobile Chord P2P protocol design for VANET', International Journal of Ad Hoc and Ubiquitous Computing, Vol. 6 No. 3, pp.150-163.

Choudhary, M., Sharma, P. and Sanghi, D. (2004), 'Secure multicast model for ad-hoc military networks' In ICON 2004: Proceedings of the 12th IEEE International Conference on Networks, Vol. 2, Singapore, pp.683-688.

Clausen, T. and Jacquet, P. (2003) 'Optimized Link State Routing Protocol', IETF Internet Draft, https://www.ietf.org/rfc/rfc3626.txt, (Accessed 2 November 2012)

Day, K. et al (2011) 'Parallel Routing In Mobile Ad Hoc Networks', International Journal of Computer Networks and Communications, Vol. 3 No. 5, pp.77-94.

De Couto, D.S.J., et al (2003) 'A high-throughput path metric for multi-hop wireless routing', In AICMCN 2003: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, San Diego, CA, USA, pp: 134-146.

Deepak, K.P., Rakesh, K. and Daniel, A.K. (2013) 'Performance analysis and behavioral study of proactive and reactive routing protocols in MANET', Int. J. Adv. Res. Comput. Commun. Eng., 2: 1789-1796. http://www.ijarcce.com/upload/2013/april/32-Deepak%20Patel-Performance%20Analysis%20%26%20Behavioural.pdf.

Demin, L., Qiuran, L. and Jiacun, W. (2012) 'Traffic information collecting algorithms for road selection decision support in vehicle ad hoc networks', International Journal of Simulation and Process Modelling, Vol. 7 No. 1/2, pp.50-56.

Dimitris, E.C., Konstantinos, D.G. and Athanasios, D.P. (2012) 'ICARUS: hybrId inCentive mechAnism for coopeRation stimUlation in ad hoc networkS', Ad Hoc Networks, Vol. 10 No. 6, pp.976-989.

Domingo, M.C., Remondo, D. and Leon, O. (2003) 'A simpler routing scheme for improving ad hoc network survivability', In GLOBECOM 2003: Proceedings of the IEEE Global Telecommunications Conference, San Francisco, USA, Vol. 2, pp.718-723.

Dorigo, M. and Gambardella, L.M. (1997) 'Ant colony system: a cooperative learning approach to the traveling salesman problem[J]', IEEE Transaction on Evolutionary Computing, Vol. 1 No. 1, pp.53-66.

Maghrebi, M.F. et al (2014) 'Determining the Amount and Location of Leakage in Water Supply Networks Using a Neural Network Improved by the Bat Optimization Algorithm', Journal of Civil Engineering and Urbanism, Vol. 4 No. 3, pp.322-327.

Francisco, J.R. (2004) OLSR patch for NS 2.35, http://garr.dl.sourceforge.net/project/um-olsr/um-olsr/1.0/um-olsr-1.0.tgz (Accessed 29 October 2013)

Gerla, M. et al (2002) 'Fisheye State Routing protocol for ad hoc networks', IETF Internet Draft. http://tools.ietf.org/html/draft-ietf-manet-fsr-01 (Accessed 15 August 2012)

Gundala, S. and Saravanan, R. (2014) 'Secure Routing in Manet Using Synchronization and Data Authencity', Cybernetics and Information Technologies, Vol. 14 No. 2, pp.24-30.

Gurpreet, S., Neeraj, K. and Anil, K.V. (2014) 'ANTALG: An Innovative ACO based Routing Algorithm for MANETs', Journal of Network and Computer Applications, Vol. 45, pp.151-167.

Haidar, S., Marcel, K. and Bassam, M. (2014) 'PHAODV: Power aware heterogeneous routing protocol for MANETs', Journal of Network and Computer Applications, Vol. 46, pp.60-71.

Hass, Z.J., Pearlman, R. and Samar, P. (2002) 'Zone Routing Protocol for Ad hoc networks', IETF Internet Draft. http://tools.ietf.org/html/draft-ietf-manet-zone-zrp-04 (Accessed 20 August 2012)

Hoda, T. et al (2012) 'An energy-aware distributed clustering protocol in wireless sensor networks using fuzzy logic', Ad Hoc Networks, Vol. 10 No. 7, pp.1469-1481.

Hong, Z. and Jie, L. (2014) 'Dynamical Topology Analysis of VANET Based on Complex Networks Theory', Cybernetics and Information Technologies, Vol. 14 No. 5, pp.172-186.

Hussain, M.A. (2015) 'Deployment of Mobile Ad-hoc Network Ticket Based QoS Routing Protocol for Healthcare', Indian Journal of Science and Technology, Vol. 8 No. 15, pp.648-68.

Hyungjik, K. and Sunwoong, C. (2015) 'A New Energy-Aware Routing Protocol for Improving Path Stability in Ad-hoc Networks' Contemporary Engineering Sciences, Vol. 8 No. 19, pp.859-864.

IEEE 802.11s Task Group (2008) 'Draft Amendment to Standard for Information Technology Telecommunications and Information Exchange Between Systems- LAN/MAN Specific Requirements- Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment: ESS Mesh Networking', IEEE P802.11s/D2.02.

Islam, M.S. Hamid, M.A. and Choong, S.H. (2009) 'SHWMP: A Secure Hybrid Wireless Mesh Protocol for IEEE 802.11s Wireless Mesh Networks', Transaction on Computer Science, pp.95-114.

Janakiraman, S. and Rajendiran, M. (2015) 'A futuristic trust coefficient-based semi-Markov prediction model for mitigating selfish nodes in MANETs', EURASIP Journal on Wireless Communications and Networking, Vol. 2015 Article No. 158, pp.1-13.

Javaid, N. et al (2015) 'Towards optimising routing overhead in wireless multi-hop networks', International Journal of Ad Hoc and Ubiquitous Computing, Vol. 19 No. 1, pp.4-18.

Jebakumar, M.S.P.J.K. et al (2015) 'A unified approach for detecting and eliminating selfish nodes in MANETs using TBUT', EURASIP Journal on Wireless Communications and Networking , Vol. 2015 Article No. 143, pp.1-11.

Jeff, B. and Wilcox, L. (2002) DREAM and LAR patch for NS 2.33. https://doc-0g-3o-docs.googleusercontent.com/docs/securesc/ha0ro937gcuc7l7deffksulhg5h7mbp1/9jr6me6ajnd4mgnscmbpjiscohr3qgbn/1436248800000/04198340273633395013/*/0B7S255p3kFXNM2JwbktMSGxoUkE?e=download (Accessed 3 April 2014)

Jeni, P.R., Julie, A.V. and Bose, A.M. (2014) 'An Enhanced Route Failure Recovery Model For Mobile Ad Hoc Networks', Journal of Computer Science, Vol. 10 No. 8, pp.1561-1568.

Joa, N.M. and Lu, I.T. (2002) 'A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks', IEEE Journal on Selected Areas in Communications, Vol. 17 No. 8, pp.1415-1425.

Johnson, D., Hu, Y. and Maltz, D. (2007) 'The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks for IPv4', IETF Internet Draft, https://www.ietf.org/mail-archive/web/ietf-announce/current/msg03407.html (Accessed 13 September 2012)

Johann, L., José, M.B. and Jorge G.V. (2004) 'Analysing the overhead in mobile ad hoc network with hierarchical routing structure', Technical University of Catalonia, http://research.ac.upc.edu/XARXES/CompNet/papers/HETNETs_paper_23.pdf.

José, J.A. et al (2014) 'Vehicle to Vehicle GeoNetworking using Wireless Sensor Networks', Ad Hoc Networks, Vol. 27, pp.133-146.

Juan, J.G., Ruiz, P.M. and Skarmeta, A.F.G. (2008) 'A Distributed Algorithm for Gateway Load-Balancing in Wireless Mesh Networks', In Proceeding of Wireless Days 08, Dubai, pp. 1-5.

JW (1995) Sun Microsystems, http://www.java.com/ (Accessed 10 January 2011)

Jyoti, P.S., Paramartha, D. and Amlan, C. (2014) 'Weighted delay prediction in mobile ad hoc network using fuzzy time series', Egyptian Informatics Journal, Vol. 15 No. 2, pp.105-114.

Kai, Y., Jian-feng, M. and Miao, Z. (2009) 'Hybrid Routing Protocol for Wireless Mesh Network'. In CIS 2009: Proceeding of the International Conference on Computational Intelligence and Security, Beijing, China, Vol. 1,pp.547-551.

Kanchana, K. et al (2007) 'DUMBONET: a multimedia communication system for collaborative emergency response operations in disaster-affected areas', International Journal of Emergency Management, Vol. 4 No. 4, pp.670-681.

Kavita, P. and Abhishek, S. (2011) 'A Comprehensive Performance Analysis of Proactive, Reactive and Hybrid MANETs Routing Protocols', International Journal Computer Science, 8: 432-441. http://arxiv.org/ftp/arxiv/papers/1112/1112.5703.pdf.

Khalid, S. et al (2015) 'IDHOCNET-A Novel Protocol Stack and Architecture for Ad hoc Networks', International Journal of Communication Networks and Information Security, Vol. 7 No. 1, pp.20-33.

Kim, B.C. et al (2003) 'An ad hoc routing with minimum contention time and load balancing', In GLOBECOM 2003: Proceedings of the IEEE Global Telecommunications Conference, San Francisco, USA, Vol. 1, pp.81 -85.

Ko, Y.B. and Vaidya, N.H. (2000) 'Location Aided Routing in mobile Ad hoc networks', Wireless Networks Journal, Vol. 6 No. 4, pp.307-321.

Koichi, A. et al (2010) 'An ad-hoc unicursal protocol for stable and long-lived communication systems in disaster situations', International Journal of Knowledge and Web Intelligence, Vol. 1 No. 3/4, pp.154-168.

Kone, V. et al (2007) ‘QUORUM- Quality Of Service in Wireless Mesh Networks’, Mobile Networks and Applications, Vol. 12 No. 6, pp.358-369.

Krishnan, M.B.M., Balachander, T. and Rajasekar, P. (2015) ‘Agent Based Trust Estimation for Mobile Ad Hoc Network’, Indian Journal of Science and Technology, Vol. 8 No. S9, pp.223-227.

Liang, M. and Denko. M.K. (2007) ‘A Routing Metric for Load-Balancing in Wireless Mesh Networks’, In AINAW 2007: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops, Niagara Falls, pp.409-414.

Li-Der, C., Ching-Chiang, H. and Jui-Ming, C. (2010) ‘An Early Warning Scheme for Broadcasting Critical Messages using VANET’, International Journal of Ad Hoc and Ubiquitous Computing, Vol. 6 No. 1, pp.1-9.

MadhanMohan, R. and Selvakumar, K. (2012) ‘Power controlled routing in wireless ad hoc networks using cross layer approach’, Egyptian Informatics Journal, Vol. 13 No. 2, pp.95-101.

Madhusudan, G. and Vinod, D.S. (2012) ‘Novel Technique of Multipath Routing Protocol in Ad hoc Network’, International Journal of Computer Networks and Communications, Vol. 4 No. 3, pp.109-119.

Mariann, H. et al (2014) ‘Selected Issues of QoS Provision in Heterogenous Military Networks’, International Journal of Electronics and Telecommunications, Vol. 60 No. 1, pp.7-13.

Marina, M.K. and Das, R.S. (2006) ‘Ad hoc On-demand Multipath Distance Vector routing’, Wireless Communications And Mobile Computing, Vol. 6 No. 7, pp.969-988.

Marjan, K.R. (2014) ‘Identifying IDS Agent Nodes Based on 3-Layered Key Management Framework for MANET’, Walailak Journal of Science and Technology, Vol. 11 No. 2, pp.79-85.

Marjan, K.R. and Hamideh, F. (2015) ‘A Survey on Algorithms Based on Bee Swarms for Ad Hoc Networks’, Walailak Journal of Science and Technology, Vol. 12 No. 1, pp.21-26.

Marwaha, S., Indulska, J. and Portmann, M. (2008) ‘Challenges and Recent Advances in QoS Provisioning in Wireless Mesh Networks’, In Proceedings 8th IEEE International Conference on Computer and Information Technology, Sydney, NSW Australia, pp.618-623.

Mohammad, K.J., Jamali, M.A.J. and Akbarpour, S. (2013) ‘A New Unicast Routing Algorithm for Load Balancing in Multi-Gateway Wireless Mesh Networks’, International Journal of Computer Networks and Communications Security; Vol. 1 No. 6, pp.252-256.

Nassef, L. (2010) ‘On the effects of fading and mobility in on-demand routing protocols’, Egyptian Informatics Journal, Vol. 11 No. 2, pp.67-74.

Natsheh, E. et al (2007) ‘Adaptive Optimizing of Hello Messages in Wireless Ad-Hoc Networks’, The International Arab Journal of Information Technology, Vol. 4 No. 3, pp.191-200.

Network Simulator 2 Website. (1995) http://www.isi.edu/nsnam/ns (Accessed 10 September 2011)

Neumann, A., Aichel, C. and Lindner, M. (2008) ‘Better Approach To Mobile Ad-hoc Networking’, IETF Internet Draft, http://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing-00 (Accessed 13 October 2012)

Nihad, I.A. and Mustafa, I. (2015) ‘Fuzzy approach to improving route stability of the AODV routing protocol’, EURASIP Journal on Wireless Communications and Networking, Vol. 2015 Article No. 235, pp.1-11.

Nora, A.A. et al (2012) ‘Elongation of WSN lifetime using a centralised clustering technique’, International Journal of Systems, Control and Communications, Vol. 4 No. 4, pp.250-261.

Park, V.D. and Corson, M.S. (1997) ‘A highly adaptive distributed routing algorithm for mobile wireless networks’, In INFOCOM 1997: Proceedings of the 6th Annual Joint Conference of the IEEE Computer and Communications Societies, Kobe, Japan. 1405-1413.

Patricia, M. Allan, G. and Carlos, S. (2015) ‘Mobile ad hoc network communication for disaster recovery’, International Journal of Space-Based and Situated Computing, Vol. 5 No. 3, pp.178-186.

Pattanayak, B.K. and Rath, M. (2014) ‘A Mobile Agent Based Intrusion Detection System Architecture For Mobile Ad Hoc Networks’, Journal of Computer Science, Vol. 10 No. 6, pp.970-975.

Pei, G., Gerla, M and Xiaoyan, H (1999) ‘A wireless hierarchical routing protocol with group mobility’, In WCNC1999: IEEE Wireless Communications and Networking Conference, New Orleans, L.A., USA. 1538-1542.

Perkins, C.E. and Bhagwat, P. (1994) ‘Highly dynamic destination-sequenced distance-vector routing for mobile computers’, In SIGCOMM 1994: Proceedings of ACM Conference on Communications Architectures, Protocols and Applications, London, England, pp.234-244.

Perkins, C.E., Belding, R.E. and Das. R.S. (2003) ‘Ad hoc On-Demand Distance Vector Routing’, IETF Internet Draft, https://www.ietf.org/rfc/rfc3561.txt (Accessed 15 November 2011)

Persis, D.J. and Robert, T.P. (2015) ‘Ant Based Multi-objective Routing Optimization in Mobile Ad-Hoc Network’, Indian Journal of Science and Technology, Vol. 8 No. 9, pp.875-888.

Poonkuzhali, R., Yakub, S.M. and Annaji, S. (2014) ‘A Premeetive Link State Spanning Tree Source Routing Protocol For Mobile Ad Hoc Networks’, Journal of Computer Science, Vol. 10 No. 1, pp.85-90.

Prabakaran, T. and Karthikeyan, N.K. (2015) ‘Adaptive clustering and routing using fuzzy decision model in WSN’, International Journal of Mobile Network Design and Innovation. Vol. 6 No. 1, pp.51-62.

Prabha, R. and Ramaraj, N. (2015) ‘An improved multipath MANET routing using link estimation and swarm intelligence’, EURASIP Journal on Wireless Communications and Networking, Vol. 2015 Article No. 173, pp.1-9.

Pravin, R.A. and Geethanjali, S. (2014) ‘A trust-based AODV routing protocol for improved QoS in mobile ad-hoc networks’, International Journal of Trust Management in Computing and Communications, Vol. 2 No. 1, pp.7-21.

Radwan, S.A. and Mohammed, G. (2011) ‘A Driven Backup Routing Table To Find Alternative Disjoint Path in Ad Hoc Wireless’, International Journal of Computer Networks and Communications, Vol. 3 No. 4, pp.46-55.

Ramesh, V. and Subbaiah, P. (2011) ‘Preemptive AOMDV routing for mobile Ad-hoc networks’, International Conference on Sustainable Energy and Intelligent Systems (SEISCON 2011), Chennai, pp: 622-625.

Santhi, G. and Alamelu, N. (2012) ‘Fuzzy-cost based multi-constrained QoS routing with mobility prediction in MANETs’, Egyptian Informatics Journal, Vol. 13 No. 1, pp.19-25.

Santhi, K. and Parvathavarthini, B. (2014) ‘Performance Analysis Of Randomized Reverse Ad Hoc On Demand Distance Vector Routing Protocol In MANET’, Journal of Computer Science, Vol. 10 No. 9, pp.1850-1858.

Sen, J. (2010) ‘A Throughput Optimizing Routing Protocol for Wirless Mesh Networks’, In Proceedings of the 12th IEEE International Conference on High Performance Computing and Communications, Melbourne, VIC Australia, pp.665-670.

Sengathir, J. and Manoharan, R. (2015) ‘Exponential Reliability Coefficient based Reputation Mechanism for isolating selfish nodes in MANETs’, Egyptian Informatics Journal, Vol. 16 No. 2, pp.231-241.

Sensarma, D. and Majumder, K. (2013) ‘A Novel Hierarchical Ant Based QsS Aware Intelligent Routing Scheme For MANETS’, Journal of Computer Networks and Communications, Vol. 5 No. 6, pp.215-229.

Senthil, K., Kumar, S. and Sugapriya, C. (2014) ‘Group Management In Mobile AdHoc Networks’, American Journal of Applied Sciences, Vol. 11 No. 7, pp.1059-1064.

Senthil, M.T., Eswariah, K. (2013) ‘Link Stability With Energy Aware Ad Hoc On Demand Multi-Path Routing Protocol In Mobile Ad Hoc Networks’, American Journal of Applied Sciences, Vol. 10 No. 8, pp.844-848.

Senthilkumar, M., Somasundaram, S. and Amuthakkannan R. (2009) ‘Power aware multiple QoS con-straints routing protocol with mobility prediction for MANET’. International Journal of In-formation Systems and Change Management, Vol. 4 No. 2, pp. 156-170.

Senthilkumar, S., William, J. and Karthikeyan, S. (2014) ‘A distributed framework for detecting selfish nodes in MANET using Record-and Trust-Based Detection (RTBD) technique’, EURASIP Journal on Wireless Communications and Networking, Vol. 2014 Article No. 205, pp.1-10.

Shahverdy, M., Behnami, M. and Fathy, M. (2011) ‘A New Paradigm for Load Balancing in WMNs’, International Journal of Computer Networks, Vol. 3 No. 4, pp.239-246.

Shams, Q. et al (2013) ‘Securing DSR against wormhole attacks in multirate ad hoc networks’, Journal of Network and Computer Applications, Vol. 36 No. 2, pp.582-592.

Sharmila, V.C. and Amalanathan, G. (2014) ‘Construction Of Strategic Connected Dominating Set For Mobile Ad Hoc Networks’, Journal of Computer Science, Vol. 10 No. 2, pp.285-295.

Sidi, O.C., Hassan, M.M. and Geuroui, A. (2013) ‘New Metric For HWMP Protocol’, International Journal of Computer Networks & Communications, Vol. 5 No. 2, pp.49-58.

Sofiane, Z., Soufiene, D. and John, M. (2014) ‘A fast, reliable and lightweight distributed dissemination protocol for safety messages in Urban Vehicular Networks’, Ad Hoc Networks, 2014; 27: 26-43.

Song, L., Jie, C. and Yang, X. (2010) ‘Multi-path Anycast Routing Based on Ant Colony Optimization in Multi-gateway WMN’, In Proceedings of the 5th International Conference on Computer Science & Education, Hefei, pp.1694-1698.

Su, W., Lee, S.J. and Gerla, M. (2000) ‘Mobility Prediction in Wireless Networks’. In Proc. of 21st MILCOM, pp. 22-25, doi: 10.1109/MILCOM.2000.905001

Sun Microsystems (2000) Netbeans Website, http://fr.netbeans.org/ (Accessed 15 January 2011)

Susmit, B. (2010) ‘A distributed algorithm for energy-aware clustering in WSN’, International Journal of Sensor Networks, Vol. 7 No. 1/2, pp.37-43.

Sven, J. (2005) ‘FSR Patch for NS 2, Sourceforge.NET, http://fsr-routing.sourceforge.net/ (Accessed 09 February 2013)

Taisir, E, Waleed N. and Aws, K. (2015) ‘Maximally Distant Codes Allocation Using Chemical Reaction Optimization and Ant Colony Optimization Algorithms’, Journal of Computer Sciences, Vol. 11 No. 8, pp. 892-901.

Thomas, K. (2009) ‘Energy-efficient routing in mobile ad hoc networks: a cautionary tale’, International Journal of Autonomous and Adaptive Communications Systems, Vol. 2 No. 1, pp.70 - 86.

Timo, S. et al (2015) ‘Vehicular Networking Road Weather Information System Tailored for Arctic Winter Conditions’, International Journal of Communication Networks and Information Security, Vol. 7 No. 1, pp.60-68.

Toh, C.K. (2001) ‘Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks’, IEEE Communications Magazine, Vol. 39 No. 6, pp.138-147.

Tran, V.P. et al (2007) ‘Transmission Time-Based Mechanism to Detect Wormhole Attacks’, In APSCC 2007: The 2nd IEEE Asia-Pacific Service Computing Conference, Tsukuba Science City, Japan, pp.172-178.

Vedhavathy, T.R. and Manikandan, M.S.K. (2015) ‘A Novel Unique Node Based Clustering and Location-Key Pair Based Security for Wireless Networks’, Journal of Computer Science. Vol. 11 No. 3, pp.534-539.

Vinh, D.H., Maode M. and Hiroshi, H. (2010) ‘Multi-paths Routing with Load Balancing for Internet Access in Wireless Mesh Networks’, International Journal of Wireless & Mobile Networks, Vol. 2 No. 1, pp.55-75.

Visu, P. and Kannan, E. (2013) ‘Combined Swarm Intelligence Routing Protocol for MANET’, Information Technology Journal, Vol. 12 No. 4, pp.791-797.

Vivek, A. and Charu, G. (2015) ‘Energy aware routing protocols for mobile ad hoc networks - a survey’, International Journal of Information and Communication Technology, Vol.7 No.6, pp.662-675.

Xhafa, F., Sanchez, C. and Barolli, L. (2010) ‘Genetic Algorithms for Efficient Placement of Router Nodes in Wireless Mesh Networks’, In AINA’10: Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications, April 20-23, pp. 465 - 472.

Xue, Q. and Ganz, A. (2002) ‘QoS routing for Mesh-based Wireless LANs’, International Journal of Wireless Information Networks, Vol. 9 No. 3, pp.179-190.

Yosra, B.S. et al (2014) ‘A survey of collaborative services and security-related issues in modern wireless Ad-Hoc communications’, Journal of Network and Computer Applications, Vol. 45, pp.215-227.

Zhao, L., Al-Dubai, A.Y. and Romdhani, I. (2010) ‘Novel QoS-Aware Gateway Centralized Multi-hop Routing for Wireless Mesh Networks’, In CIT 2010: 10th IEEE International Conference on Computer and Information Technology,pp.336-342.

American Journal of Applied Sciences

Original Research Paper

A Novel Proactive Routing Method for Mobile Ad Hoc Networks

Bambrik Ilyas and Didi Fedoua

Department of Computer Science, Laboratory of Research in Informatics of Tlemcen (LRIT), Abou Bekr Belkaid University, Tlemcen, Algeria

Article history
Received: 09-08-2014
Revised: 10-05-2015
Accepted: 12-07-2015

Corresponding Author:
Bambrik Ilyas
Department of Computer Science, Laboratory of Research in Informatics of Tlemecen (LRIT), Abou Bekr Belhaid University, Tlemcen, Algeria
Email: ilyas9111@yahoo.fr

Abstract: In recent years, the Mobile Ad hoc Networks (MANETs) became more popular due to their multiple advantages such as the low setup cost, auto-configuration and auto-healing features. This kind of network is distinguished by its dynamic and distributed nature since it does not contain any form of centralized infrastructure. These characteristics have led to the emergence of new routing protocols specifically designed for dynamic topologies. Some protocols construct and maintain paths before they are needed (proactive protocols). However, this method generates a massive volume of overhead. Other protocols reduce the overhead by constructing paths only when it is necessary (reactive protocols). On the other hand, this method is not well adapted when the topological changes are frequent and can increase the transmission delay. Though the features presented by this type of network seem appealing, maintaining the network connectivity in a distributed fashion is a complicated task and could drain a significant piece of the network resources. Moreover, considering that the network members are usually embedded devices equipped with limited resources, it is vital to minimize the resources spent by the routing protocol. In this study we compare the various routing strategies designed for MANET and propose a proactive routing algorithm that consumes an acceptable quantity of the network members resources. Furthermore, we present how our protocol works and multiple tests to show the protocol reactions to the different topological parameters.

Keywords: Routing Protocol, MANET, Proactive, Reactive, Overhead, Network Resources

Introduction

With the recent technological advances in the field of telecommunication, a new type of wireless network called Mobile Ad hoc Network (MANET) has emerged. Unlike the traditional wireless networks, MANETs are formed by mobile hosts (smart phones, laptops, …etc) solely. Instead of relying on a preset infrastructure, all the network members collaborate in a distributed fashion in order to form multi-hop paths. Furthermore, the network members are able to move freely which makes the network topology highly flexible. Due to the interesting features of the MANET, this network technology has become a core feature in other networks such as the Wireless Mesh Network (WMN) and the Vehicular Ad hoc Network (VANET). The ability of operating without a centralized infrastructure makes this type of network very suitable for situations where deploying an infrastructure is difficult or unwanted and more importantly, this technology can be viewed as an ideal platform to support the pervasive systems.

Since the topology can change randomly, the traditional routing solutions cannot function in this type of network. Therefore, a new type of routing protocol appeared that can operate in a dynamic topology. Based on the manner with which the paths between the network members are created, three categories of MANET routing protocols exist. First, the proactive protocols create and maintain the paths to all the potential destinations in the network. Accordingly, when a network member requires a path to a given destination, the data transfer starts without delay. To apply this strategy, the network members periodically broadcast their topological information and process the incoming routing messages. As a consequence, this type of protocol can consume a portion of the mobile hosts resources (bandwidth, energy and computation time).

Secondly, the reactive protocols create the paths on demand instead of building a path to every possible destination. In this strategy, a path between a pair of nodes is created and maintained as long as it is used. When a path toward a given destination is needed, a request/wait for reply process is launched. Then, the path requests are rebroadcast by the intermediate nodes until the destination is found. Afterwards, a reply is sent to the source through the found route. Thus, unlike the proactive protocols, this category of protocol lowers the overhead when the transmission requests are uncommon.

Although this routing approach preserves the resources of the network, the reactive path building process can introduce a transmission delay. Inevitably, this problem expands when the topology changes frequently. Furthermore, this strategy is ineffective when the number of transmission demands is elevated and can consume as much resources as the proactive method in such a situation. Logically, when the user application requires a constant connectivity between the network members, it is more appropriate to apply the proactive approach than to create a path to each destination separately. Moreover, whereas the proactive approach automatically optimizes the available paths, the reactive protocol first informs the source about the path rupture so that a new search for the destination can be launched.

The mixture between the proactive and the reactive method created a new type of protocol called hybrid. First, the proactive method is applied to collect the routing information about the close neighborhood (3 or 4 hops away). After that, in case a path to a remote destination is needed, the reactive approach is launched. Consequently, this kind of protocol generates less overhead than the proactive protocols and reduces the transmission delay by comparison with the reactive protocols. As a substitute to the reactive path building process, the request messages are forwarded through routing zones until the destination is found instead of overflowing the entire network. Nonetheless, this method does not produce necessarily the shortest paths because, the search for the destination, is zone driven.

Some protocols organize the network in a hierarchical manner to attribute a role for each network member. The idea is to represent the network in a form of multiple clusters where each cluster has one representative responsible of routing functions, labeled as a Cluster Head. Mobile hosts that belong to more than one cluster are used for communication between clusters and are considered as Gateways. As a result, the hierarchical organization minimizes the routing operations performed by the cluster members since the routing functions are insured only by the Cluster Heads and Gateways. The downside is that the concentration of data flows on the Cluster Heads and Gateways can lead to network congestion and eventually, node failure.

Unquestionably, the routing protocol is the kernel function behind all the advantages of the MANET. Considering the differences in the strategies applied by the routing protocols, the performances of the protocol depend on the context for which the MANET is used. Hence, multiple extensions were proposed in order to increase the adaptability of the routing protocols in different situations. Some researchers focused their work on how to balance the traffic load in order to increase the network life time (Day *et al.*, 2011). Other studies were oriented toward Quality of Service (QoS) awareness (Poonkuzhali *et al.*, 2014) and security insurance (Gopalakrishnan and Ganeshkumar, 2015; Chandramohan and Kamalakkannan, 2014).

For instance, in a scenario where the transmission requests are rare, the reactive and hybrid protocols can outperform the proactive protocols. On the other hand, in a situation where the user application requires a constant connectivity between the users, the proactive protocols are more suitable. In addition, the proactive method offers many advantages such as its better awareness of the topological events and low transmission delay. However, the complexity of the proactive protocols make their performances depend on the topological parameters. Besides, the MANET components are usually equipped with a restricted CPU power, limited bandwidth and energy. For that reason, the constant advertisement/processing of the routing information can accelerate the exhaustion of the mobile hosts energy; especially in a large networks.

In order to lower the overhead of the proactive strategy, many methods have been proposed. In this study we present a proactive algorithm based on a new routing information exchange/processing method that aims to reduce the complexity of the routing function. Although the proposed algorithm is proactive, the resulting overhead is fairly reasonable by comparison with other protocols from the same category. Furthermore, we propose several series of simulations to test the proposed solution resiliency based on different topological parameters (nodes speed of movement, network size and density).

This paper is organized as the following: In section I we discuss the different routing algorithms proposed for MANET. In section II, we provide the core algorithm for our protocol. Then, in section III we explain how the protocol is tested. In section IV, we discuss the results obtained from the protocol tests. Finally, section V concludes this paper.

Literature Study

The Proactive Protocols

Due to the distributed and unpredictable nature of the MANET, numerous routing protocols have been

DOI: 10.3844/ajassp.2015.382.402

designed to execute the routing function in dynamic topologies. Destination Sequenced Distance Vector (DSDV) developed by Perkins and Bhagwat (1994) is a proactive protocol that chooses the paths based on how fresh they are. To determine the freshness of the paths, each path is tagged with the corresponding destination sequence number so that the node can select the most recent path. In addition, important topological changes are immediately announced by the network members. Consequently, this protocol efficiently eliminates routing loops and accelerates the convergence. However, when the topology changes frequently, the volume of routing messages produced by this protocol, expands.

Optimized Link State Routing (OLSR), designed by Clausen and Jacquet (2003) is a proactive protocol based on the Link State algorithm. This protocol optimizes the dissemination of the control messages by selecting for each network member, a subset of its neighborhood called Multipoint Relays (MPR). Simply, this subset contains the minimum set of neighbors that can enable a network member to reach all the adjacent nodes two hops away, or less. Afterwards, all the neighbors process the control messages but only the selected MPRs forward them. Consequently, the links that are disseminated in the network, are limited.

The strength of this protocol is its adaptability to dense topologies. Instead of announcing all the links with the neighbors, only the necessary links are broadcast. Although the MPR selection process requires a periodic treatment of the neighbors set, this mechanism eliminates the asymmetric link problem. Also, from a security point of view, relaying only on the selected MPRs eliminates the misbehaving host problem since, the network member that does not share any routing information, is excluded from the MPR list.

The idea applied by Gerla *et al.* (2002) in Fisheye State Routing (FSR), is to use the Fisheye algorithm which was initially designed for the compression of visual data. When applied on routing information, the network members advertise the description of their near neighborhood more frequently than the description of the remote destinations. Thus, the outcome of this method is the reduction of the control message size and a slight inaccuracy while building paths to the remote destinations. However, since the routing decisions are made hop by hop, the routing imprecision can be insignificant because the information about the destination is more accurate for the network members relatively closer to it. Although this method slows down the convergence of the protocol, this strategy is well adapted to large topologies. Furthermore, Andreas *et al.* (2009) proposed an integration of this method in OLSR to lower the overhead. Figure 1 is an example of how FSR organizes the routing information.

Neumann *et al.* (2008) proposed a protocol called Better Approach to Mobile Ad hoc Networking (BATMAN) which incorporates a new method for path building. This protocol generates control messages called Originator Messages (OGMs) and upon receipt, the distance to each destination is incremented and the message is rebroadcast. Subsequently, the paths are selected based on the number of OGMs received. The advantage of this protocol is that it produces paths with high throughput and low packet loss. On the other hand, the OGMs are advertised more frequently than the control messages of other proactive protocols and disseminating them in the network increases the overhead.

Distance Routing Effect Algorithm for Mobility (DREAM), created by Basagni *et al.* (1998), is a proactive protocol similar to FSR. This protocol collects the geographical descriptions of the network members through a localization system and uses them to construct paths. Two mechanisms compose this protocol: (a) the distance effect, (b) the node relative speed. The idea of the first mechanism is that the more a node is distant, the less it appears to be moving. Therefore, this protocol advertises the routing information representing the close nodes more often than the description of the destinations faraway. The difference between this method and FSR is that DREAM measures the geographical distance between the nodes instead of hop count.

In the second mechanism, the nodes moving at a high speed advertise their routing information more often than those moving slowly. Furthermore, this protocol considers that the next hop toward the destination is the closest one to it according to the collected geographical information. However, this perception might not be always true. Hence, making this protocol inaccurate in some scenarios. Besides, the correlation between the broadcast frequency of the control messages and the network member speed of movement, increases the overhead when the network members are moving fast.

Typically, the routing protocols for MANET tend to build the shortest path. Juliusz (2010) created a new proactive protocol, called Babel, based on the ETX metric proposed by De Couto *et al.* (2003). Basically, this protocol calculates the probability of data transmission over a link without packet loss. To achieve this, the number of control messages received from a neighbor and the expected number of control messages over a period of time, are compared. Afterwards, the neighbors proposing the highest probabilities of transmission without packet loss, are chosen for path construction purposes. As a result, this protocol promotes the paths that have a low signal interference. Additionally, the sequence number idea of DSDV, is applied in this protocol. Nonetheless, the problem of this protocol is that the computation of the ETX metric requires a recurrent broadcast of the control messages. Consequently, the routing overhead of this protocol is elevated.

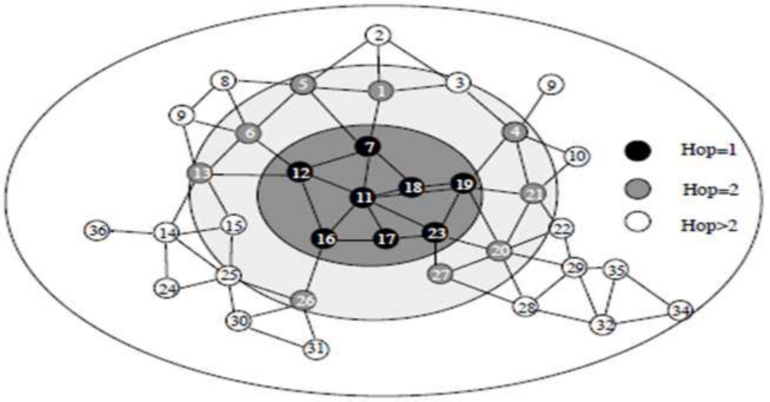

Fig. 1. Fisheye state routing (Gerla *et al*., 2002)

The Reactive Protocols

Whereas the proactive protocols main focus is to maintain the paths between all the network members, reactive protocols create a path on demand only. Designed by Perkins *et al*. (2003), Ad hoc On demand Distance Vector (AODV) is a reactive protocol that applies a request/wait for reply cycle to construct a path. When a network member needs a path to a given destination, the search is initiated by the broadcast of a Route Request (RREQ) message. After receiving a RREQ, the intermediate node creates a reverse path to the source and rebroadcast the request only once. Afterwards, when the destination is reached or a node in possession of a recent path (by comparison with the sequence number included in the RREQ) is attained, a Route Replay (RREP) is forwarded toward the source to establish the transmission path.

The main disadvantage of AODV is its inadaptability to the frequent topological changes. When the network members movement affects the transmission path, first, the source is informed about the path rupture. Then, the search for the destination is restarted from the beginning. To limit the transmission delay problem, Alshanyour and Baroudi (2008) proposed an upgraded version of this protocol by incorporating a local route repair method. This mechanism can reduce the transmission delay due to path rupture especially when the network members are moving slowly or at a medium speed. Furthermore, in the work of Surjeet *et al.* (2014) and Kanniche *et al.* (2011), two bandwidth estimation strategies based on AODV, are proposed in order to provide the QoS support for the data streams.

Designed by Johnson *et al.* (2007), Dynamic Source Routing (DSR) applies a similar path construction process to the one applied in AODV. The difference is that the intermediate network members include their IDs in the RREQ instead of memorizing the reverse path. Moreover, when the path is sent back to the source, the found path is included in all the data packets. As a result, this protocol requires less storage capacity and a rupture in the transmission path is immediately detected which eliminates routing loops. However, including the path in all the RREQ/RREP and data messages generates a considerable quantity of routing overhead.

The reactive protocols developed by Marina and Das (2006) and Park and Corson (1997) create multiple paths toward the same destination. This strategy can improve the protocol adaptability to frequent topological events. Instead of maintaining only one path toward the destination, Ad hoc on demand Multipath Distance Vector (AOMDV) and Temporally Ordered Routing Algorithm (TORA) maintain all the found paths during the search for the destination. After that, in case the primary transmission path is broken, one of the alternative paths is used by the source. Unless all the alternative paths are broken, this approach can significantly shorten the transmission delay. On the other hand, creating and maintaining multiple paths to the same destination, increases the overhead. In order to introduce the QoS awareness in this type of protocol, Sensarma and Majumder (2013) proposed an extension of TORA based on the Ant colony method (Dorigo and Gambardella, 1997).

The Associativity Based Routing (ABR) protocol, developed by Toh (1997), is a reactive protocol that promotes the path that has a high associativity degree. To run this protocol, each member of the network attributes an associativity degree to its neighbors by counting the number of the HELLO messages received. During the search for the destination, the associativity degree of the explored links are included in the request messages. Subsequently, when the destination is reached, the path with the highest associativity degree is selected. Also, this protocol incorporates a local path repair mechanism to reduce the transmission delay in case of a path failure. Although the local path repair is effective when the network members are moving at a medium/low pace, this method increases the transmission delay when it fails.

Signal Stability-Based Adaptive Routing (SSA), presented in the work of Dube *et al.* (1997), is a reactive protocol that selects the paths based on the signal strength between the neighbors. First, for each neighbor, the link signal quality is recorded as “high” or “low”. Then, when the search for the destination is launched, only the path requests received from the “high” signal strength links, are forwarded. As a result, this protocol creates paths formed by “high” quality links. The disadvantage of this method is that, if the first attempt to build a path with “high” quality links fails, the transmission source then restarts the search and the “low” quality links are accepted during the second attempt. Thus, the transmission delay is increased.

Ko and Vaidya (2000) proposed another protocol called Location Aided Routing (LAR) that collects the geographical coordinates of the mobile hosts. This reactive protocol explores the existing geographical information about the destination (last known location, time elapsed since the last communication, speed and direction of movement) to near down the location where the destination might be found. First, the source calculates the relative distance that separates it from the destination and includes it in the request message. Subsequently, the intermediate nodes receiving the request message will rebroadcast it only if they are closer to the destination. In order to enlarge the search area, the distance separating the destination from the source, is increased at the beginning of the process. Consequently, the probability of finding the destination, is increased. However, supposing that the next hop leading to the destination is the closest one to it (geographically), can be false. As a result, the path repair mechanism can falsely fail which leads to restarting the search for the destination and including all the network members in the process. Hence, if the path repair fails, the transmission delay is increased.

The Hybrid and Hierarchical Protocols

Hybrid protocols usually create multiple routing zones by applying the proactive approach. Then, the reactive approach is initiated to reach the remote destinations. Zone Routing Protocol (ZRP), proposed by Hass *et al*. (2002), applies this idea to create a routing zone around each network member. During the reactive path building phase, if the destination is within the routing zone, a path replay is sent back to the source. Hence, the transmission delay is reduced by comparison with the typical reactive approach. Nevertheless, since the search for the destination is directed through routing zones, the path produced by this protocol is not always the shortest.

Joa and Lu (2002) proposed Zone Based Hierarchical Link State (ZHLS) which is also a hybrid protocol that creates routing zones based on the geographical locations of the network members. Depending on the positions of the network members, the network is arranged in a form of multiple routing zones. Furthermore, the nodes that belong to the same zone create paths between each other and the links between the routing zones are announced in the entire network. During the reactive search for the destination, the transmission delay is reduced since the route replay is sent once the zone, to which the destination belongs to, is found. On the other hand, the geographical routing zones of this protocol cannot merge. Figure 2 is an example of how ZHLS arranges the nodes in groups based on their positions.

Protocols that organize the network in a hierarchical manner are very similar to the hybrid protocols. The difference between these two strategies is that the hierarchical organization of the network elects a subset of the network responsible for the routing operations whereas in the hybrid protocols, all the network members contribute uniformly.

Pei *et al.* (1999) designed Hierarchical State Routing (HSR) which is a protocol that arranges the network into multiple hierarchical levels. To apply this protocol, each Cluster Head at level N becomes a member in a cluster at level N+1. Afterwards, the network members are identified by the sequence of Cluster Heads IDs relaying them to the root cluster. Consequently, Gateways have multiple IDs since they can be reached by at least two Cluster Heads.

In Cluster Based Routing Protocol (CBRP), designed by Jiang *et al.* (1999), the mobile hosts advertise periodically their neighborhood table. On the basis of this information, a role is attributed to each network member (Cluster Head, Gateway or Cluster Member). Additionally, the reactive method is applied when the destination does not belong to the same cluster. Accordingly, the route request is forwarded between clusters until the destination is located. Moreover, Vidal and José (2010) proposed an extension of this protocol to provide QoS for the new data flow. Cluster Based QoS Routing Protocol (CBQRP) includes the minimum bandwidth required in the route requests. Essentially, the idea in this protocol is to forward the route request by only the Cluster Heads and Gateways that have enough remaining bandwidth to support the new data flow.

Theoretical Analysis

In addition to the advantages/inconveniences that the protocols inherit from their path building strategy, each protocol incorporates other methods that can improve its performance. For instance, the reactive protocols can reduce the transmission delay by adding a local path repair method. Also, the performances of the routing protocols depend on several aspects such as the network members mobility, network size/density, the user application requirements,…etc.

DOI: 10.3844/ajassp.2015.382.402

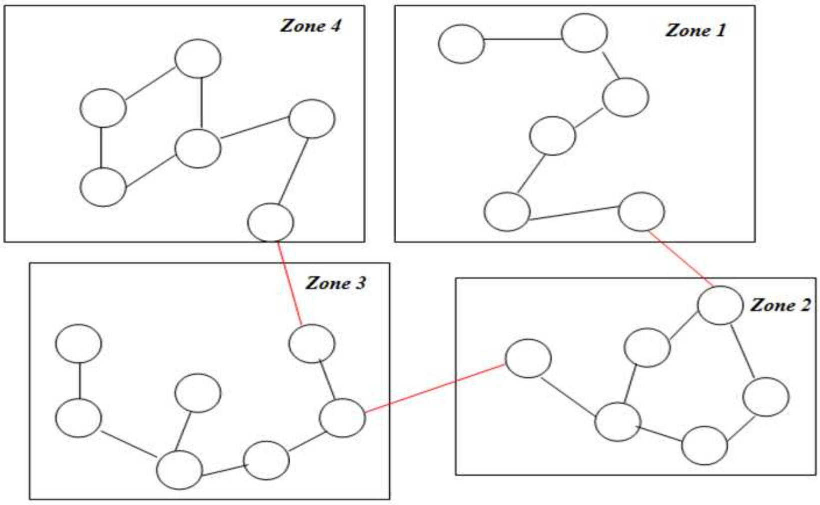

Fig. 2. Zone based hierarchical link state

Table 1. Different protocols characteristics

Parameters Protocol	Resource consumption	Topological awareness	Algorithm complexity	Performance dependency	Advantages
DSDV	Medium	Medium	Medium	Network size/density and the mobility of the nodes.	Fast convergence and routing loop avoidance.
FSR	Medium	Low	Medium	Network density and the mobility of the nodes.	Adaptability to large topologies.
DREAM	High	High	Medium		
OLSR	High	High	High	Network size.	Adaptability to dense topologies.
Babel	High	High	Medium	Network size and density.	High throughput and retransmission avoidance.
BATMAN	High	Medium	Medium		
AODV	Very low	Low	Low	Network size, the number of transmission requests and the mobility of the nodes.	Low overhead and routing loop avoidance.
DSR	High	Low	Very Low		Routing loop avoidance.
AOMDV	Medium	Medium	Low	Network size and the number of transmission requests.	Adaptability to path rupture.
TORA	Medium	Medium	Medium		
LAR	Low	Medium	Low	The number of transmission requests.	Transmission delay reduction.
SSR	Low	Low	Low	Network size, the number of transmission requests and the mobility of the nodes.	Retransmission avoidance.
ABR	Low	Medium	Low	Network size and the number of transmission requests.	Path rupture avoidance.
ZRP	Medium	Low	Low	The number of transmission requests and the network density.	Path discovery acceleration.
ZHLS	Medium	Low	Low		
HSR	Medium	Low	Medium		
CBRP	Medium	Medium	Medium		

For example, the proactive protocols are more suitable when the user application needs to maintain the communication between all the network members. In a collaborative system where each user must be aware of the other teammates status, establishing the paths automatically between the components of the network, is crucial. Whereas the proactive protocols automatically create the paths and update them, frequent topological changes force the reactive protocols to restart the path building process from the beginning. On the other hand, the reactive protocols are usually fairly simple to execute and require very little resources. Table 1 summarizes the characteristics of the MANET routing protocols from a theoretical point of view.

Proposed Algorithm

To achieve proactive routing, the entire network must advertise periodically the collected routing information in a form of control messages. After that, the received control messages are processed by the neighbors to update their topological view. Usually, the proactive protocols broadcast the entire routing table after a short period of time to avoid routing anomalies and refresh the routing view of the neighbors. Another shared mechanism between the proactive protocols is that all the control messages received from the neighbors must be processed.

In a small/medium size network, the proactive protocols can perform very well without a noticeable effect on the network resources. On the other hand, in a large network, the sizes of the control messages grow as well which directly increases the consumption of the network members resources. Our idea is to reduce the overhead of the proactive protocol by limiting the broadcast of the entire routing table. Instead of frequently advertising the full routing table, the network members broadcast the routing updates. Evidently, in order to announce all the available paths, the broadcast of the global topological view is a must in any proactive protocol. However, the periodical broadcast of the routing table is wasteful in terms of energy. Especially in the network areas where there is no significant changes to be announced.

To enable the protocol to converge adequately while generating an acceptable quantity of overhead, the algorithm that we propose broadcasts the full routing table only if at least one new essential neighbor is sensed. This event driven mechanism significantly reduces the effect of the network size on the quantity of overhead produced by the routing protocol. The protocol that we present in this study is called Effective Routing Based on Overhead Reduction (ERBOR). Initially, we started designing this protocol in our preliminary research on the MANET and the existent Ad hoc routing protocols (Bambrik and Didi, 2013).

Another topological parameter that can influence the proactive protocol is the network density. Constantly processing all the control messages received from the neighbors can consume a large portion of energy and computation time. In Fig. 3 for example, MH1 is located in a dense area of the network and must process all the control messages received from its neighbors. The MPR method included in OLSR improves the protocol adaptability to dense topologies by reducing the number of links announced in network. Still, the network members treat all the control messages received from both the neighbors and MPRs.

Understandably, the amount of energy spent by a network member on routing operations strongly depends on the volume of routing information processed by it. For that reason, we included a modified MPR selection mechanism that enables ERBOR to disregard control messages that contain redundant routing information. Instead of processing all the received control messages like OLSR, ERBOR verifies first the control message generator field to determine if the message is useful. Simply, when the generator of a control message is reachable through an existing path, the control message is ignored. Otherwise, the control message is fully processed.

Control Messages Processing

ERBOR is based on three tables and the MPR list. The first table is the Routing Table (RT) which contains for each path: (a) the destination ID, (b) next hop, (c) hop count. The second table is the Next Update Table (NXUT) in which, the updates that have not been advertised yet, are stored. Each update contains: (a) the destination ID, (b) hop count, (c) type of the update (a deletion or path entry). The third table is called Delayed Update Table (DUT) and it contains all the updates that need to be checked before they can be advertised. In case an entry from the DUT is removed before the next control message broadcast, a new removal update is included in the NXUT to announce that the corresponding route is no longer valid. Otherwise, if an entry in the DUT still exists after the subsequent broadcast (the information is still valid), a positive update is added to the NXUT.

The network members periodically broadcast their NXUT and clear them afterwards. Then, between two broadcasts, the control messages received from the neighbors are examined. The control message content can be either the NXUT (recent topological events only) or the full routing table. This approach is slightly similar to the one applied in DSDV with several changes. DSDV broadcasts the full routing table when a significant change is sensed so that the size of next incremental message is reduced. Also, DSDV periodically broadcasts the full routing table of all the network members after a short period. The idea we propose in ERBOR is to broadcast the full routing table of a network member only when it is necessary. Table 2 represents the format of the control messages generated by ERBOR. Also, the *CM_Sequece_Number* is used to detect packet loss.

All the control messages contain a list of updates that can be either a deletion update (negative update) or a path update (positive update). The NXUT contains both types of update. On the other hand, when the control message contains the full routing table, all the entries are positive updates. The algorithm in Fig. 4 shows how the control messages are handled by ERBOR.

Removal Update

A path contained in the routing table can be deleted only if a negative update is received from the corresponding next hop or the link with the next hop is broken. Also, if a path is deleted, all the corresponding temporary path updates in the DUT and NXUT, are deleted as well. Subsequently, for each deleted path, a removal update is added to the NXUT in order to announce the detected change. Another possible situation is that a mobile host receives a removal update for a destination that does not exist in its routing table. Logically, the deletion update is ignored in this scenario. The schema in Fig. 5 explains how this procedure is implemented.

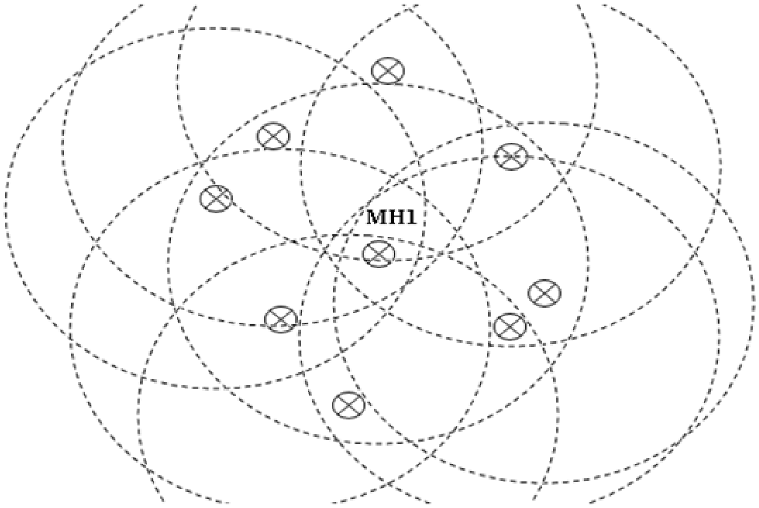

Fig. 3. The effect of the network density

```
public void ProcessingRoutingUpdateMessage  (ControlMessage UpdateMessage){
int i,n=UpdateMessage.Update_List.size();
int CM_GeneratorID= UpdateMessage.getGeneratorID();
boolean Skip_Update;
RoutingEntry DestinationEntry;
UpdateEntry Update;
int GeneratorID= UpdateMessage.getGeneratorID();
if(!C_MSG_Is_Received_From_MPR(UpdateMessage)) return  ;
for(i=n-1;i>=0;i--){
    Update= (UpdateEntry) UpdateMessage.Update_List.elementAt(i);
    Skip_Update=false;
    if(Update.getDestinationID()==ID_Current_node)  Skip_Update=true;
    if(isMPR(Update.getDestinationID())){
        Skip_Update=true;
        if(!Update.getUpdateType())InsertInDUT(new UpdateEntry (true,Update.getDestinationID(),1));
        else if(Update.getHopCount()==1)PatchAllPaths(Update.getDestinationID(),CM_GeneratorID);
        }
      if(!Skip_Update){
      DestinationEntry=GetCorrespondingRoutingEntry(Update.getDestinationID());
      if(DestinationEntry!=null){
        if(!Update.getUpdateType()){
        Process_Delete_Update(Update,DestinationEntry,CM_GeneratorID);
       }else {
            Processing_Update_For_An_Existent_Path(Update,DestinationEntry,CM_GeneratorID);
            }
      }else Creating_A_New_Routing_Etry(Update, DestinationEntry, CM_GeneratorID);
    }
  }
}
```

Fig. 4. Control message processing algorithm

Table 2. Control message format

CM_Generator_ID	CM_Sequence_Number	CM_Type
Destination_ID 1	Hop_count	Update_Type
Destination_ID 2	Hop_count	Update_Type
.	.	.
.	.	.
Destination_ID n	Hop_count	Update_Type

On the other hand, if the next hop is not the originator of the removal update, the path in the routing table can be an alternative path unaffected by the received update. To verify that the available path is still valid, the path in the routing table is added to the DUT. After the next broadcast, this alternative path is included in the NXUT, if no removal update is received for it. Afterwards, the path is advertised to notify the originator of the removal update that an alternative path is available.

To explain how a deletion update is handled, we consider the scenario in Fig. 6. When the link between MH5 and MH3 is lost, MH5 generates a negative update to inform the neighbors about the topological change. Let's suppose that for MH1, the next hop toward MH3, is MH2. After the reception of the update generated by MH5, MH1 does not remove the path attributed to MH3 because, MH5 is not the next hop toward it. Instead, MH1 adds the path describing MH3 to its DUT so it can be verified first and advertised afterwards. This way, MH5 will be informed about the existence of an alternative path toward MH3.

Path Update

A path update, for an existing path, is applied by a network member only if it is received from the corresponding next hop or it has a better metric than the one available in the routing table. Additionally, when a path update describing a new destination is received, this new path is accepted only if, there is no path to the same destination, recently deleted. To verify this condition, the receiver searches for a negative update describing the corresponding destination in the NXUT.

In case a network member receives a path update with a higher hop count by comparison with the one in its routing table, the receiver creates an update describing the available path and adds it to the DUT. The path in the routing table is verified first then added to the NXUT in case it is still valid. The schemas in Fig. 7 and 8 represent the processing methods for the positive updates. In Fig. 9, MH5 detects MH6 as a new neighbor and generates a path update describing this new link. When MH1 processes this information, it first verifies if there is already a path leading to MH6. In this example MH6 is a direct neighbor for MH1 which means that MH1 has a shorter path than the one advertised by MH5.

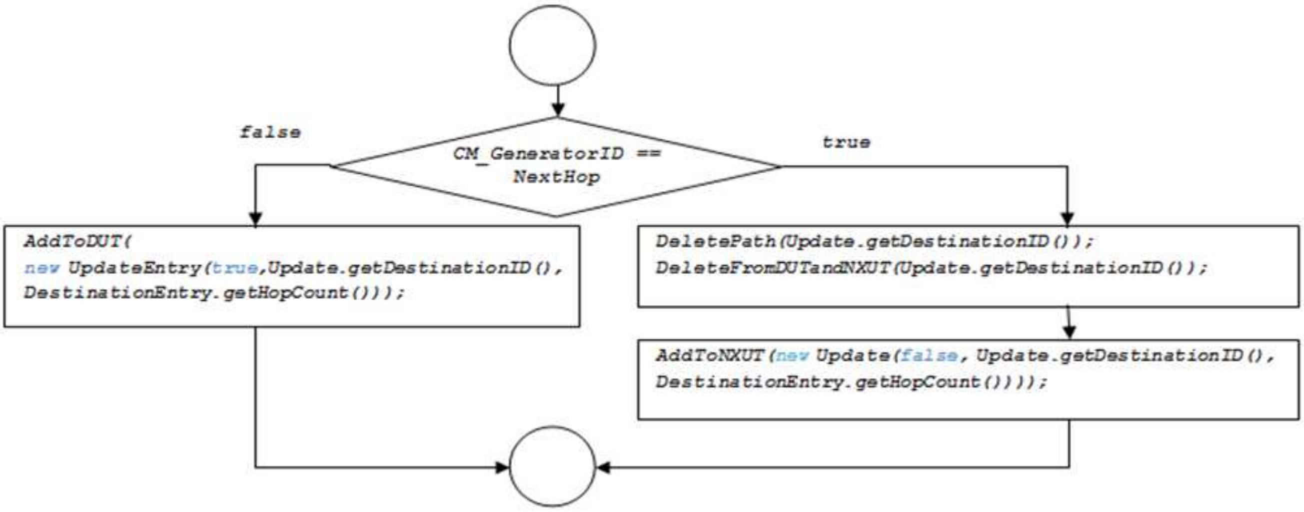

Fig. 5. Deletion update treatment algorithm

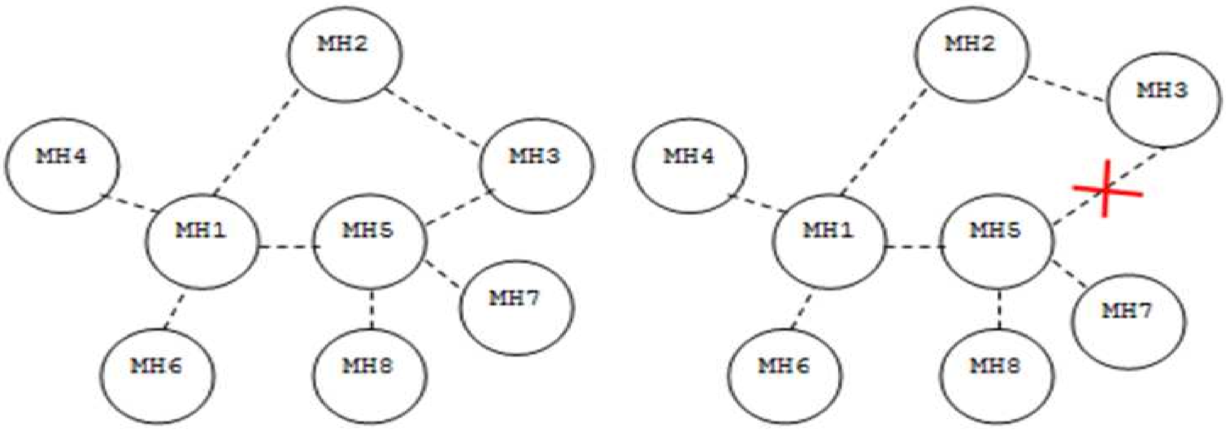

Fig. 6. Deletion update scenario

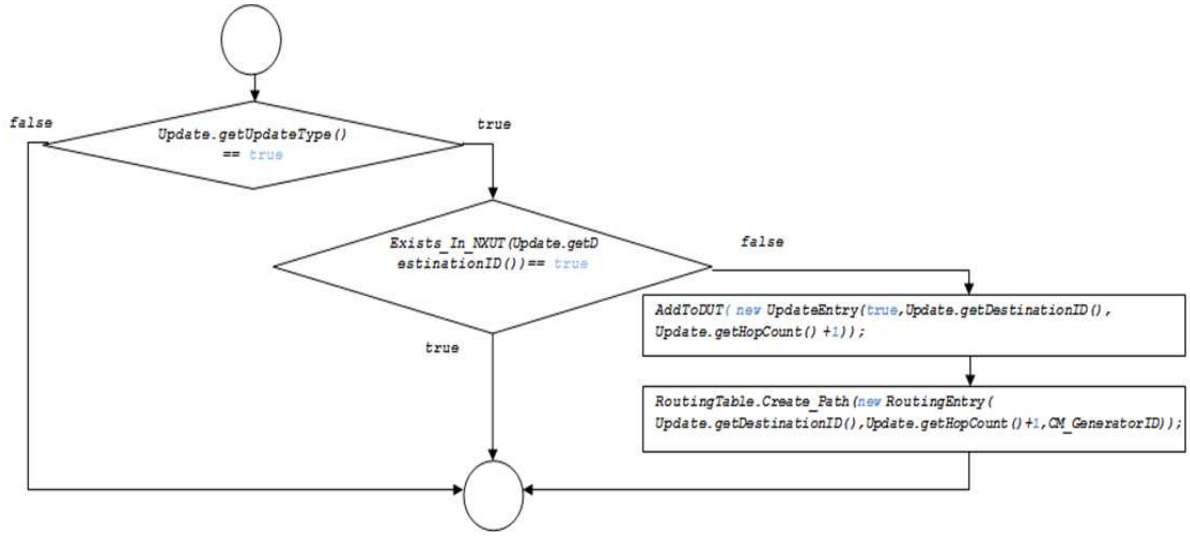

Fig. 7. Creating a new routing entry

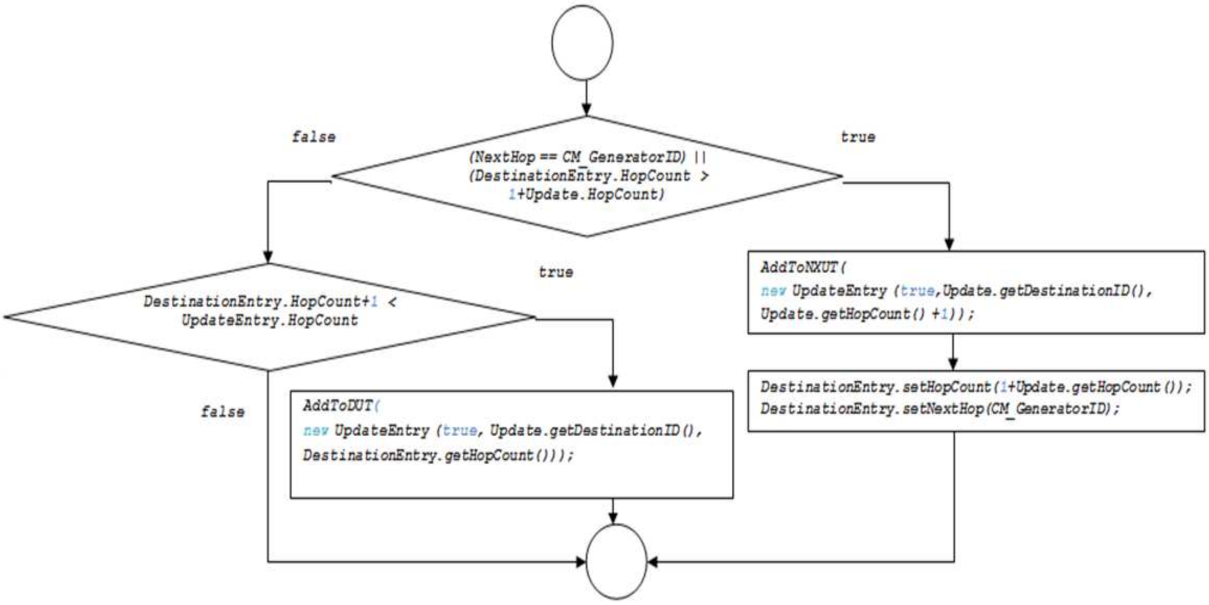

Fig. 8. The algorithm for processing a positive update for an existent path

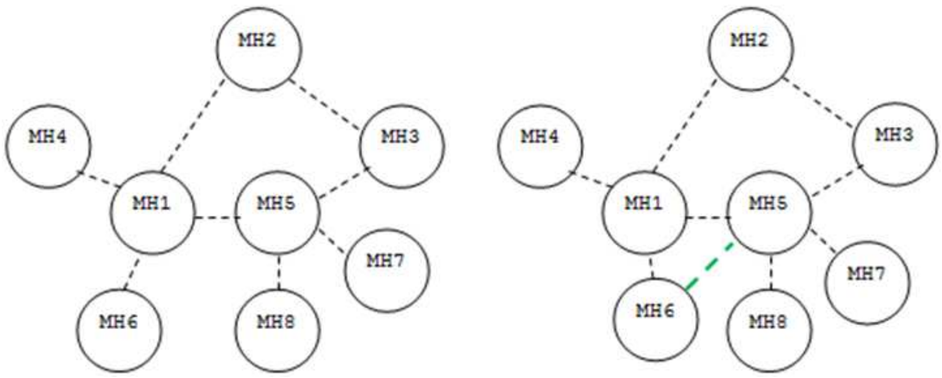

Fig. 9. Path update example

MPR Selection

In a situation where a node has many neighbors, the routing information advertised by some neighbors can be redundant. For example in Fig. 10, MH1 can obtain almost the same routing information from MH8 and MH5. Therefore, it is more efficient to select a subset of neighbors for each network member, that enables the routing protocol to create a complete view of the network. In OLSR, the MPR selection method defines a subset of the neighborhood that enables a mobile host to reach all the neighbors within two hops distance. Nonetheless, the MPRs are used to forward routing information only while the network members process all the received control messages. Our idea is to select a subset of neighbors that enable the protocol to build a path to every potential destination. Afterwards, only the control messages received from the selected neighbors, are treated.

In ERBOR, the receiver of a control message first verifies if the source can be reached through an active MPR. If that is the case, the source of the update is considered as a neighbor only and, the control message received from it, is ignored. Otherwise, if the source cannot be reached by any existing MPR, this neighbor becomes an MPR itself. This way, only the control messages received from the selected MPRs are processed. Accordingly, this method reduces the energy and computation time spent on routing operations. Moreover, only the detection of a new MPR can trigger the broadcast of the full routing table. Figure 11 shows how the network member detects a potential new MPR and, how the control messages received from the redundant neighbors, are ignored.

When two nodes detect each other as MPRs, they broadcast their entire routing table. The receiver of the routing table compares it with its routing table to create a list of paths that are unavailable for the new MPR. Then, this list is added to the DUT so that it can be verified first then advertised. Afterwards, when a neighbor is selected as an MPR, the full routing tables received from it, are processed only to verify if a path has been removed.

This MPR strategy can function better if the MPR list of each node is minimized. To achieve this, the network members remove any MPR reachable through another MPR. When a new positive update for a current MPR node is received, this new path is added to the RT and the corresponding MPR state is set to “inactive”. Then, all the paths based on the “inactive” MPR are patched through the MPR node that offered this new path and the “inactive” MPRs are removed from the MPR list. This method is similar to the one used in hierarchical protocols for Cluster Heads election. For example, in Fig. 12, the movement of MH7 makes it accessible through MH6. Let’s suppose that in this example, MH6 is the first to announce this new link. When MH1 detects this change, it patches all the paths based on MH7 through MH6 and removes MH7 from the MPR list. Consequently, MH1 will be processing less routing packets.

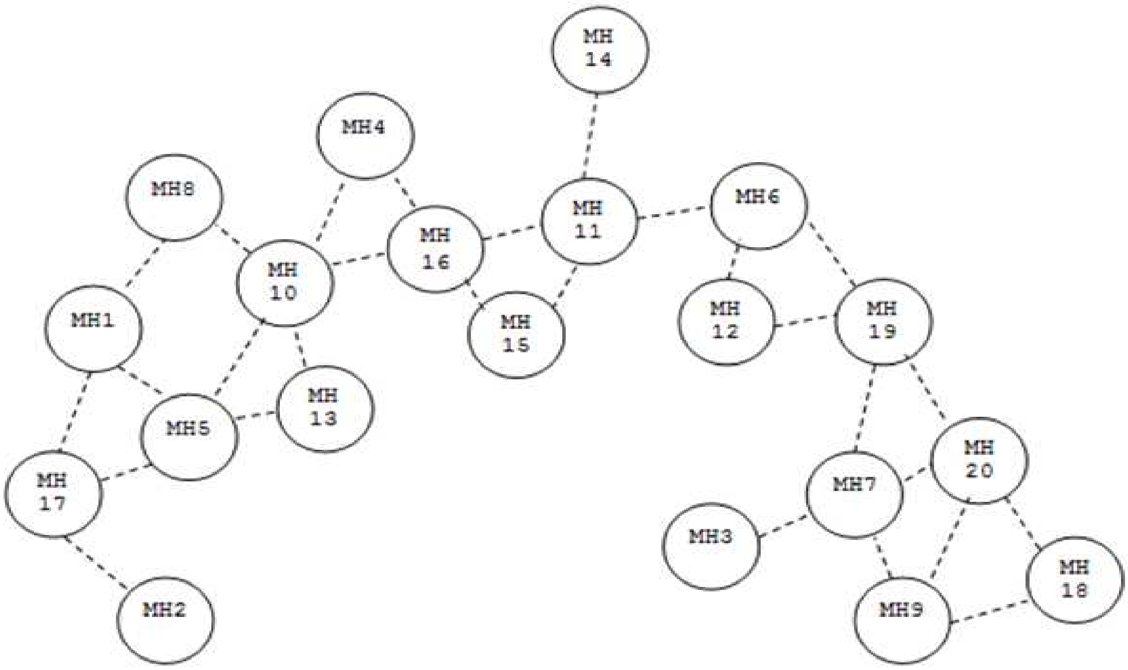

Fig. 10. Redundant routing information treatment

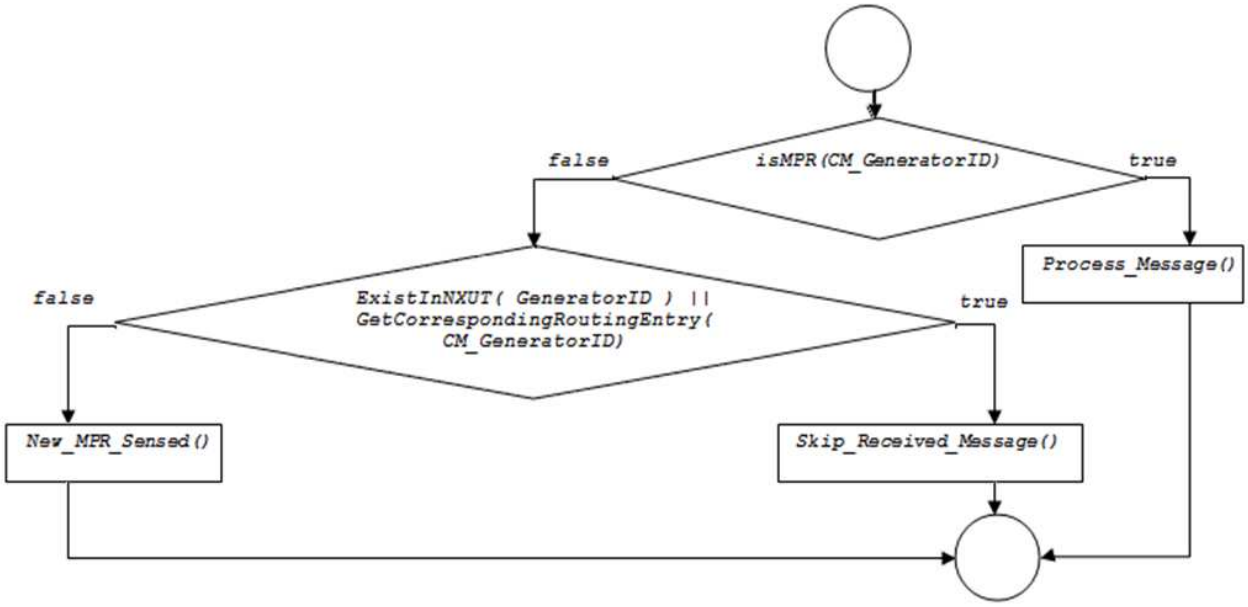

Fig. 11. Control messages sorting algorithm

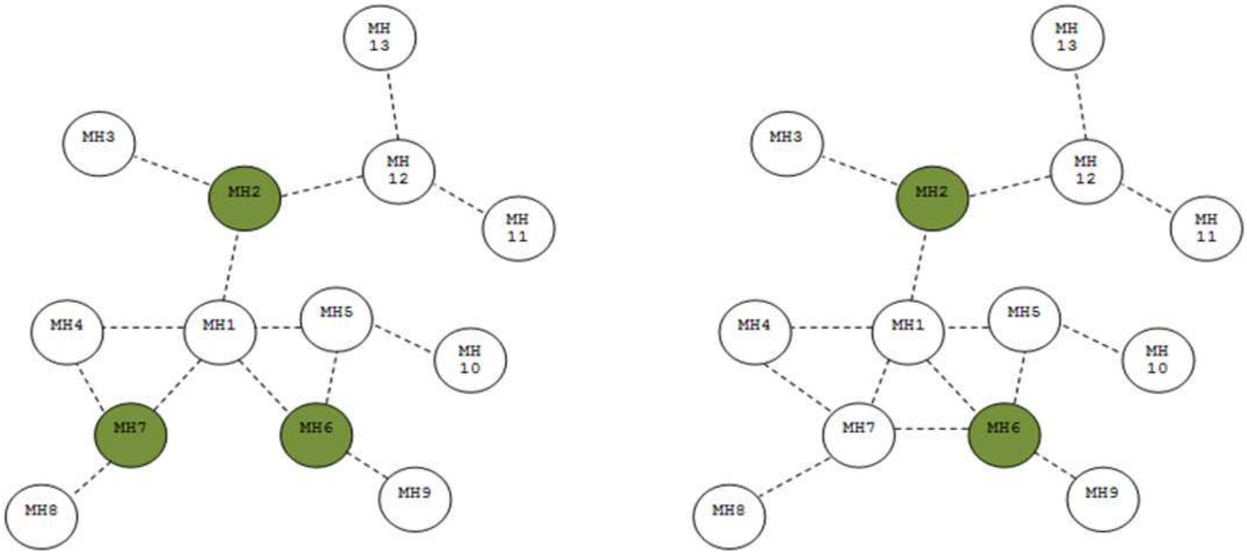

Fig. 12. The MPR set Optimization

Simulation

In order to test ERBOR, we implemented a simulator that can run ERBOR in a dynamic topology. The role of this software is to create a virtual network formed by mobile nodes and execute ERBOR in a distributed manner. First, our purpose was to test the protocol in different topologies. To ensure that the routing protocol works properly in any topology, we created a random movement scenario generator that defines the initial positions of the nodes and the positions to which they are moving toward. The tests were repeated extensively to ensure that the routing information is advertised correctly and no routing loops can appear.

The routing simulator we implemented, was written in Java (JW, 1995) programming language. Initially, the role of the simulator was to execute the routing functions of ERBOR and display the produced routing tables interactively. Thus, the simulator was meant to operate specifically on the Routing Layer. Afterwards, we added a measurement mechanism that calculates the overhead generated by ERBOR to compare it with the overhead generated if the components of the network advertised their routing tables. Since most of the proactive protocols advertise the entire routing table after a defined period, the typical proactive method of routing information exchange can be compared with the method that ERBOR applies. To do so, the overhead generated in the network is measured by summing the sizes of the control messages advertised by the network members. Figure 13 displays the different interfaces of the routing simulator.

At the beginning, the tests showed the formation of routing loops when a subset of the network goes out of coverage and renders the network divided into multiple zones. This problem was solved by including the DUT structure instead of the sequence number method. The user is also able to monitor the progression of the routing table in any network member. Both the routing table (RT + MPR list) and the NXUT are displayed proactively. It is also possible to choose a path between the selected node and a destination to display it on the topology interface, as it is shown in Fig. 14. Additionally, the user is able to suspend the node movement or turn off a node in order to test the protocol reaction to node failure. The user can also suspend the movement scenario, restart it or edit the positions of the network members.

In the overhead measurement screen, the simulator indicates the elapsed simulation time, the total overhead generated by ERBOR and the overhead produced by the classical proactive method (the advertisement of the entire routing table). In Fig. 15, the red plot represents the overhead generated by the full routing table broadcast, whereas the overhead generated by ERBOR is displayed in green. At first, the two methods produce almost the same volume of overhead. This result can be explained by the fact that the network members executing ERBOR advertise their full routing tables when they are starting to establish their MPR list. After the definition of the MPR list, the nodes broadcast more often the routing updates which decrease the overhead. Eventually, when the topology is stable, ERBOR generates only small control messages to maintain the relationship between a node and its selected MPRs. The network in this example was composed of 300 nodes with a transmission range of 250 m for each.

As previously explained, proactive protocols require that all the network members process the entire routing information received from the neighbors. This method costs expensively in terms of energy when the volume of routing information processed is increased. To test the method included in ERBOR, we implemented a frame that displays the volume of routing information processed (as it is displayed in Fig. 16) across the network when: (a) the entire routing table is received and treated (represented by the plot in red), (b) ERBOR treats all the control messages received from the neighbors (represented by the plot in blue), (c) ERBOR treats the control messages received from the MPRs only (represented by the plot in green). The example in Fig. 16 shows that the method of control messages treatment included in ERBOR reduces significantly the volume of routing information treated by the network members. By doing so, this method reduces also the network members CPU usage and the energy spent on routing information processing.

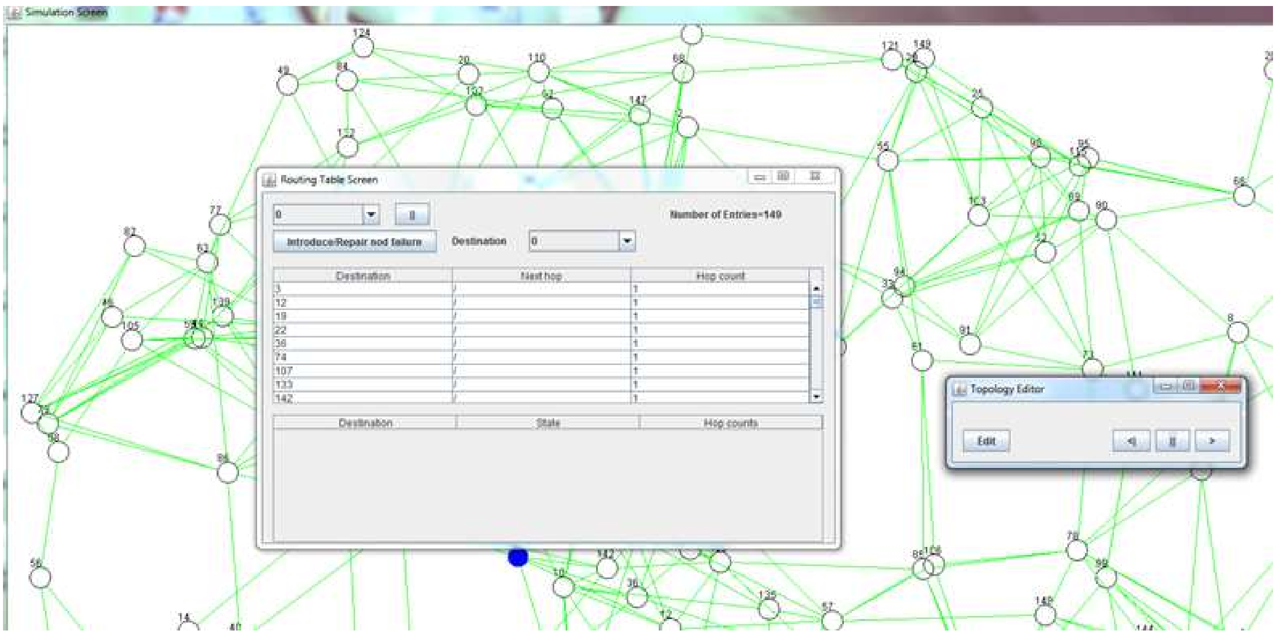

Fig. 13. The routing simulator

DOI: 10.3844/ajassp.2015.382.402

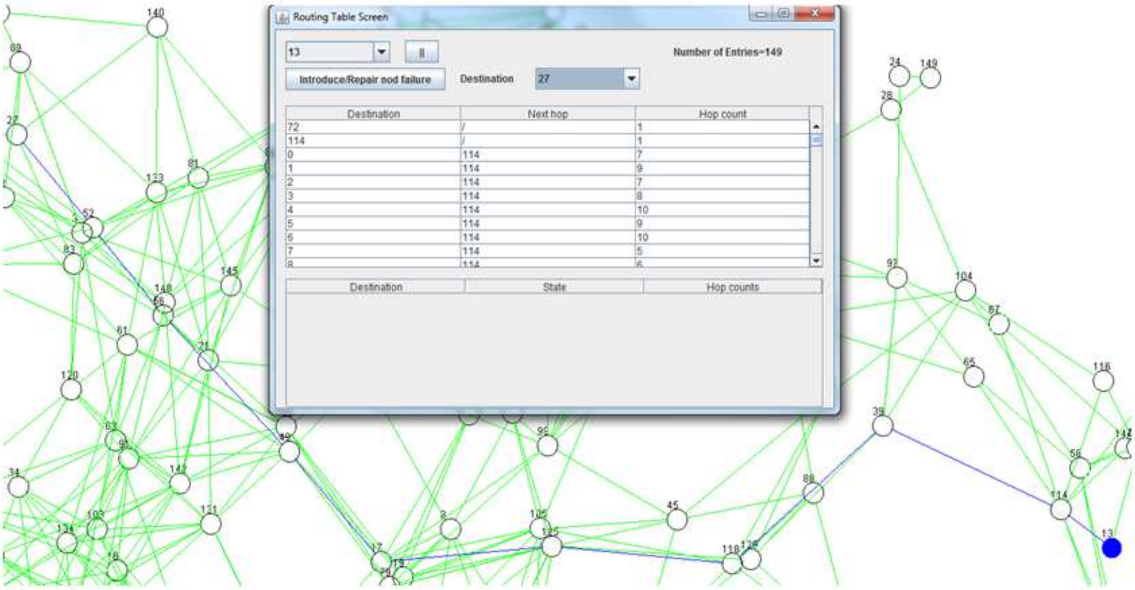

Fig. 14. Path display on the topology screen

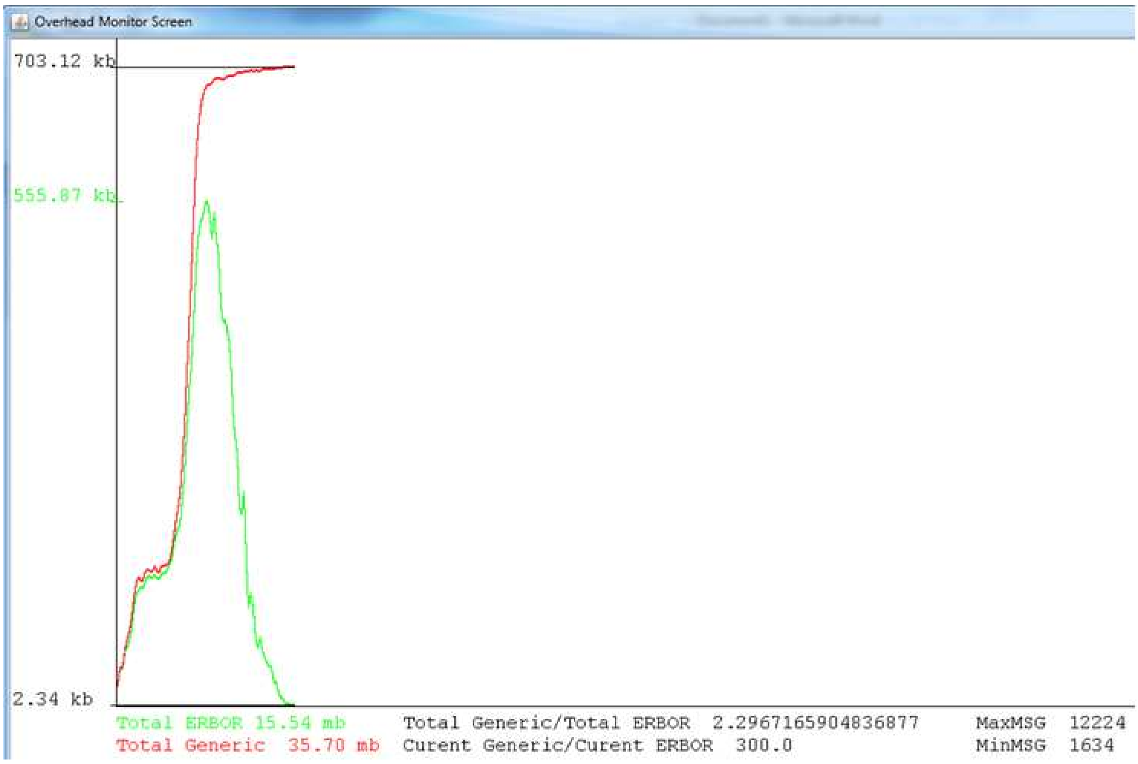

Fig. 15. The overhead measurement screen

In order to facilitate the usage of the simulator, we added an interface that enables the user to define the network topology and a simple movement scenario as it is shown in Fig. 17. The user can also define other parameters like the broadcast period of the control messages and the network members speed of movement. To generate a random topology, the user has to specify the network size and click on "Generate random topology". Moreover, the user can observe the evolution of the topology as it is shown in Fig. 18. At first, this frame was used to make sure that all the nodes detect their neighbors. After implementing the MPR selection method, the MPR connection between two nodes was displayed by a green link.

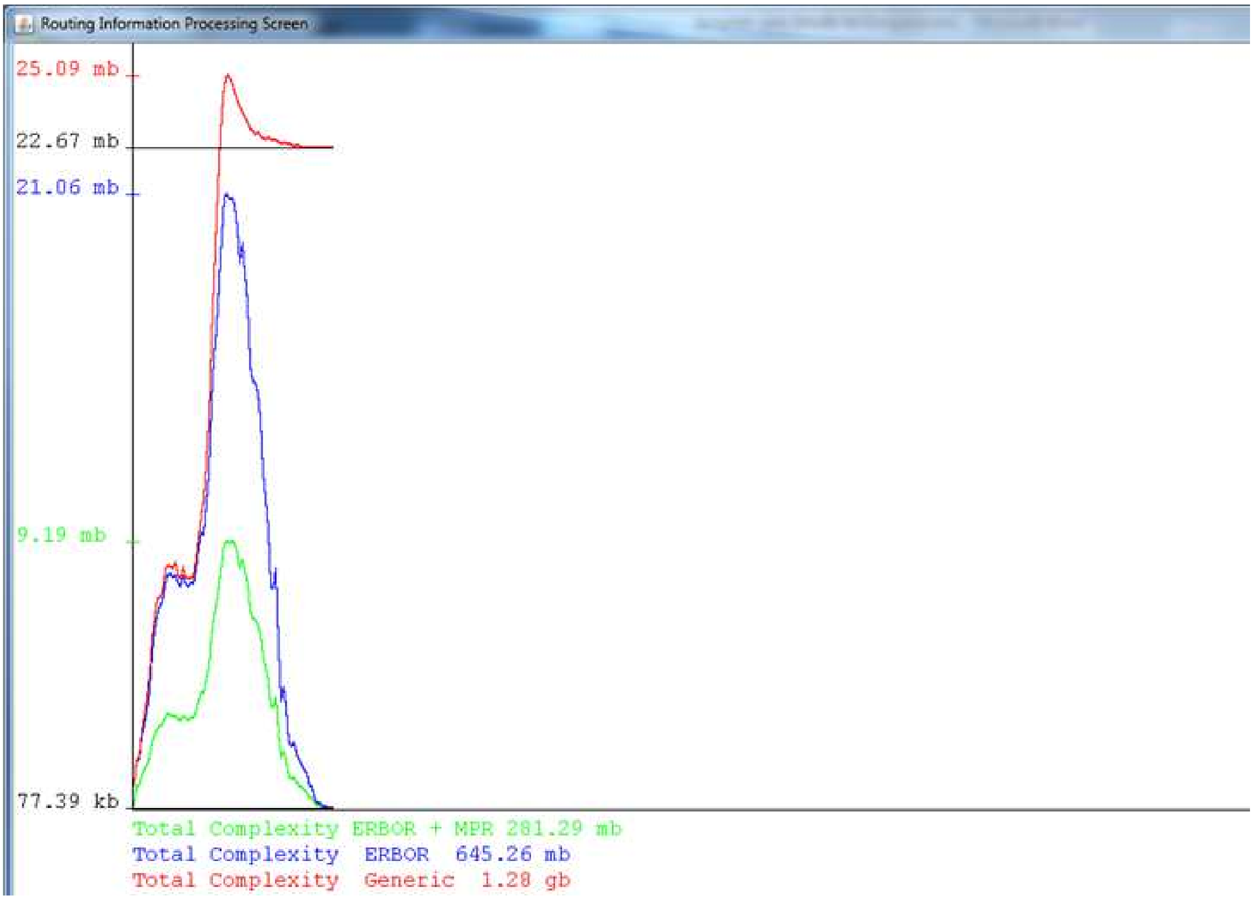

Fig. 16. The volume of routing information processed in the network

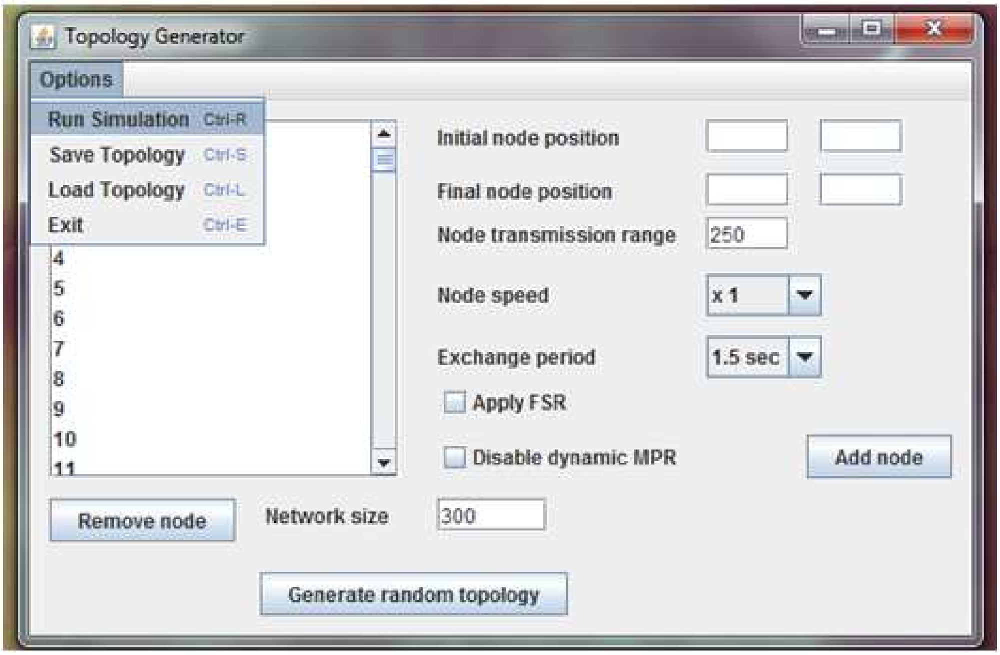

Fig. 17. The topology generator interface

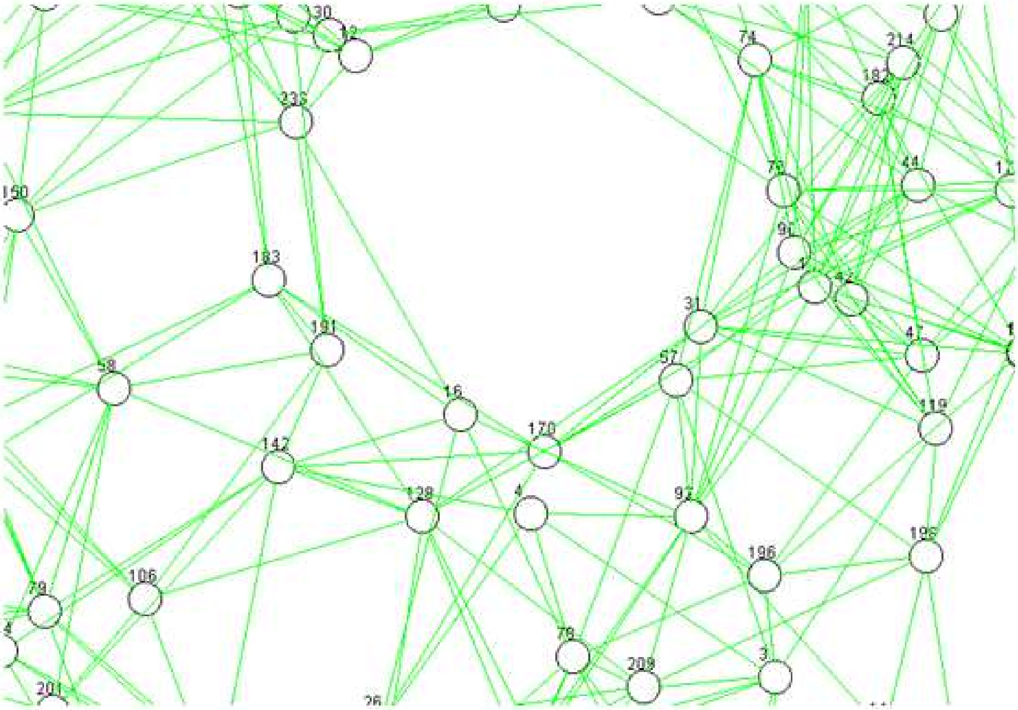

Fig. 18. Topology display

Although this simulation tool is focused only on the routing function, the ability of running the simulation dynamically is a significant advantage. First, the primary role of the simulator is to locate any malfunction of the algorithm which is an easy task when the routing information is displayed interactively. Subsequently, any routing loops/errors can be located. For instance, it is possible to simulate all the network layers with Network Simulator 2 (NS2, 1995), but it is difficult to locate an unusual behavior of the protocol through interpreting a large trace file. Therefore, our focus was to test the protocol in an interactive manner and to make sure that it can operate correctly.

Results

The main disadvantage of all the proactive protocols is their inadaptability to large/high density topologies. Previous studies have illustrated that the increase of the network size, leads directly to the exponential growth of the proactive protocols overhead. For instance, Johann *et al.* (2004) explain that the large overhead produced by the proactive protocols is caused by the frequent broadcast of the entire routing table. Consequently, the resulting overhead is correlated with the value $Network\_Size^2$. Furthermore, the obtained results by Deepak *et al.* (2013) and Kavita and Abhishek (2011) indicate that the increase of the network size affects negatively the reactive/hybrid protocols less than the proactive protocols.

Besides, from a network density prospect, the treatment of a large quantity of routing information can exhaust the network resources at high rate. This aspect affects significantly the MANET members since they are usually equipped with a limited computational capacity. To compare our solution, in this context, with other protocols from the same category, we conducted several series of simulations with DSDV, FSR (Sven, 2005) and ERBOR. The simulation parameters were changed in order to measure the overhead when: (a) the network size is expanded, (b) the topology density is raised, (c) the movement speed of the nodes is raised. The results recorded during the simulations are presented in Fig. 19-24. Generally, ERBOR generates the lowest overhead during the tests and treats less routing information than FSR and DSDV.

Discussion

The Network Growth Influence

To test the influence of the network growth on the proactive protocols, we ran a series of simulations where the network size is augmented gradually. The obtained results in Fig. 19 and 20 show how the overhead and the routing information processed by the tested protocols, are influenced. As expected, FSR generates less overhead and treats less routing information than DSDV.

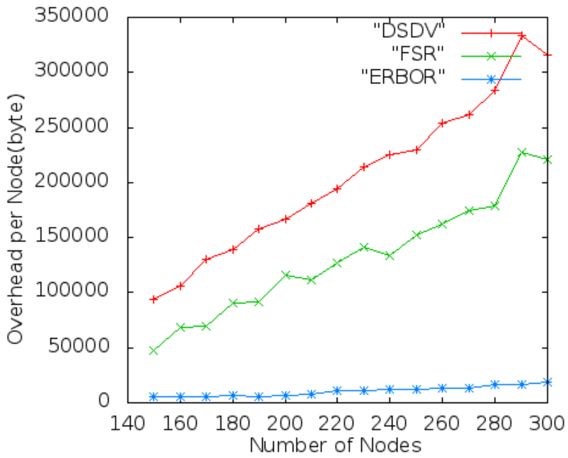

Fig. 19. The impact of the network growth on the overhead generated per node

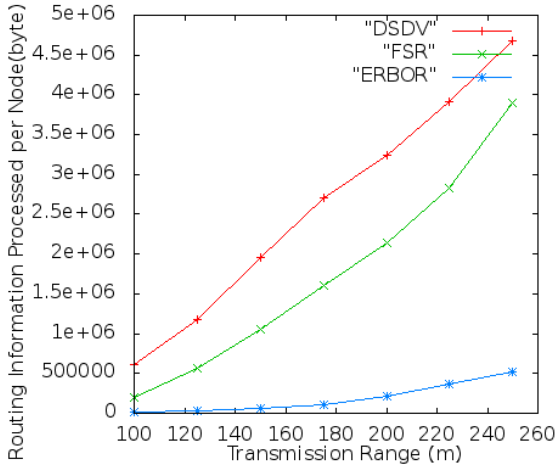

Fig. 22. The impact of the transmission range on the quantity of routing information processed per node

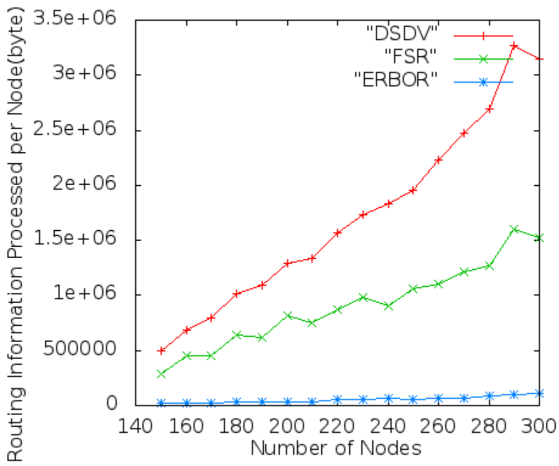

Fig. 20. The impact of the network growth on the quantity of routing information processed per node

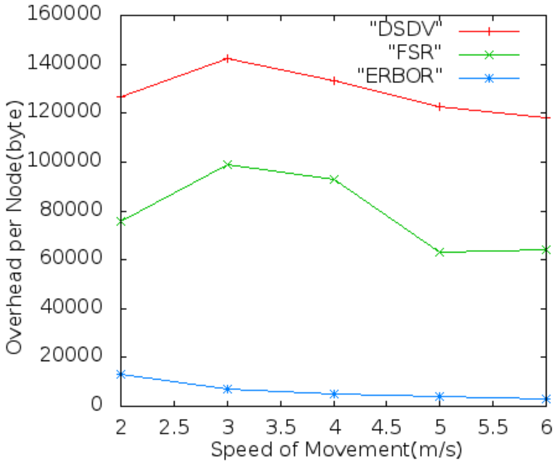

Fig. 23. The impact of the speed of movement on the overhead generated per node

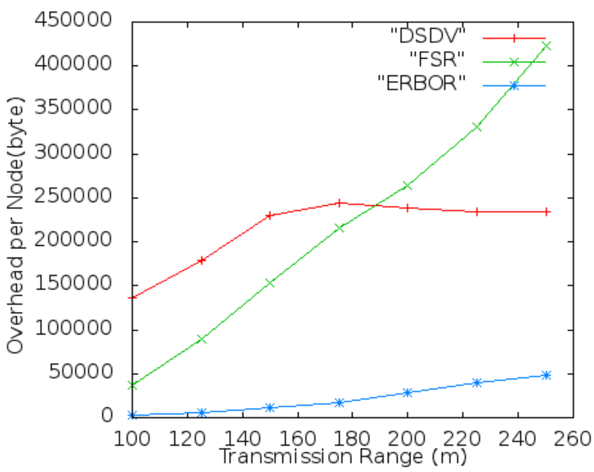

Fig. 21. The impact of the transmission range on the overhead generated per node

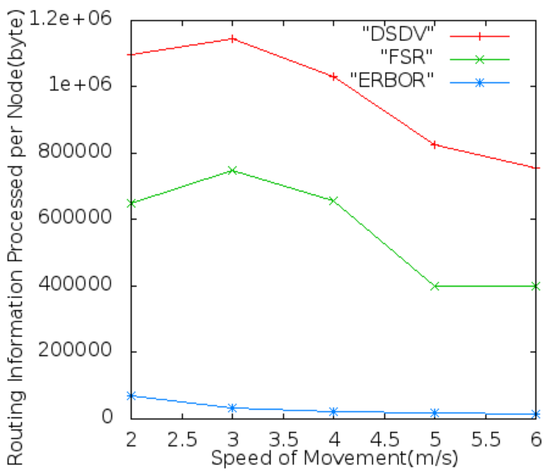

Fig. 24. The impact of the speed of movement on the quantity of routing information processed per node

Although the Fisheye algorithm improves considerably the adaptability of the protocol to large topologies, increasing the network size leads eventually to the increase of the network density. As a result, the size of the neighborhood description, advertised by FSR, is increased as well. The parameters of this experiment are declared in Table 3.

Besides the periodic broadcast of the global topological information, DSDV is based on an event driven method to announce the important topological changes. Although this approach has its benefits from a convergence perspective, this methods elevates the overhead when the network members are still in motion. Moreover, the systematic advertisement of the entire routing table increases the overhead as well. Hence, the massive volume of routing information processed and the large quantity of overhead when DSDV is applied.

In contrast to the two previous protocols, the growth of the network appears to have very little influence on ERBOR. From an overhead point of view, the results can be explained by the fact that ERBOR limits the quantity of overhead by advertising the entire routing table of a node, only when a useful connection is established. Thus, the increase of the network size does not directly augment the sizes of the control messages. Moreover, when the topology is stable, FSR and DSDV continue to broadcast the routing messages periodically. On the other hand, ERBOR advertises only the detected changes. Accordingly, whereas the overhead/routing information processed of DSDV and FSR continue to expand, ERBOR consumes a negligible quantity of resources when the network members are steady.

As explained earlier, a shared mechanism between the proactive protocols is the periodic treatment of all the received routing messages. Considering the sizes of the control messages (due to the growth of the network) advertised by DSDV and FSR, both protocols process a massive quantity of routing information in order to maintain the topological view. On the other hand, ERBOR treats only the messages received from the selected MPRs. Combined with the way ERBOR restrains the control message size, the method of MPR selection reduces considerably the influence of the network growth on the volume of routing information processed by the network members.

The Network Density Influence

Another parameter that can heavily influence the proactive method, is the network density. To test how this factor can impact the proactive protocols, we ran a series of simulations where the network density is increased by augmenting the transmission range of the network members. Consequently, the number of neighbors sensed by a node, is augmented as well. The results in Fig. 21 and 22 illustrate that the increase of the network density leads to the increase of the overhead and the volume of routing information processed, by all three protocols. Table 4 represents the parameters of this simulation.

As mentioned in the theoretical description of FSR, the strategy of this protocol is to broadcast the description of the adjacent nodes more often than the description of the remote destinations. Thus, when the number of close nodes increases due to the network density, the overhead of FSR expands past the quantity generated by DSDV. On the other hand, the overhead of DSDV is relatively stable because the dissemination method used in this protocol does not depend on the number of the adjacent nodes. More importantly, since all the received routing messages must be processed, the increase of the network density causes the elevation of the volume of routing information processed by both protocols.

Due to the MPR selection method, ERBOR advertises the entire routing table only when it is necessary. Although this method reduces the volume of overhead, the increase of the network density leads inevitably to the increase of the detected MPRs. Nonetheless, the effect of the network density is significantly reduced by comparison with FSR and DSDV. Moreover, unlike FSR and DSDV, only the essential routing messages are processed. Thus, the influence of the network density on ERBOR, is limited.

Another perceptible outcome is that, although FSR generates more overhead than DSDV, DSDV processes more routing information. While this result can seem odd, it can be simply traced to the event driven nature of DSDV. In FSR, the broadcast of the neighborhood description is done periodically. Thus, the increase of the routing messages sizes, increases directly the volume of routing information processed. On the other hand, the broadcast of the routing packets is event driven when DSDV is applied. To lower the overhead, when an important event is detected, DSDV broadcasts the entire routing table to limit the sizes of the subsequent incremental messages. Moreover, due to the increase of the network density, any sensed change deemed as essential, will trigger the immediate broadcast of the routing message which is received by all the adjacent nodes. Hence, the elevation of the routing information processed by DSDV. In Fig. 25, the recorded quantity of routing messages processed by DSDV and FSR, during this series of simulations, are displayed.

Network Members Movement

To test the adaptability of the three protocols based on the nodes speed of movement, we made a series of simulations where the nodes speed of movement is augmented gradually. The obtained results in Fig. 23 and 24, show that DSDV and FSR are influenced randomly. On the other hand, the overhead/routing information processed of ERBOR, decreases. The parameters of this experiment are shown in Table 5.

DOI: 10.3844/ajassp.2015.382.402

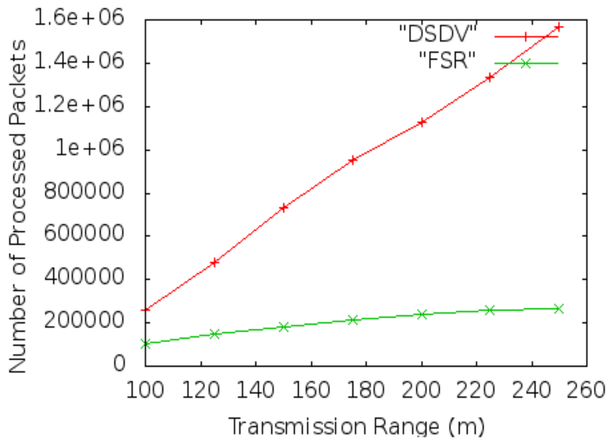

Fig. 25. The overall quantity of routing packets processed by DSDV and FSR

Table 3. The network growth parameters

Parameters	DSDV	FSR	ERBOR
Broadcast period	15 sec	5 sec	5 sec
Simulation space		1500×1500 m	
Transmission range		150 m	
Speed of movement		4 m s^{-1}	
Network size		150-300	
Simulation time		400 s	

Table 4. The network density parameters

Parameters	DSDV	FSR	ERBOR
Broadcast period	15 sec	5 sec	5 sec
Simulation space		1500×1500 m	
Transmission range		150-250 m	
Speed of movement		4 m s^{-1}	
Network size		250	
Simulation time		400 s	

Table 5. The network members movement speed parameters

Parameters	DSDV	FSR	ERBOR
Broadcast period	15 sec	5 sec	5 sec
Simulation space		1500×1500 m	
Transmission range		150 m	
Speed of movement		2-6 m s^{-1}	
Network size		250	
Simulation time		400 s	

Table 6. The topological parameters influence on DSDV, FSR and ERBOR

Parameter	Protocol	Network size	Network density	Node speed of movement
DSDV	Overhead	High	Medium	High
	Processed information	High	High	High
FSR	Overhead	Medium	High	Low
	Processed information	Medium	High	Low
ERBOR	Overhead	Low	Medium	Low
	Processed information	Low	Medium	Low

First, it is obvious that increasing the network members speed of movement does not influence FSR directly. Unlike DSDV, the routing messages advertisement method of FSR is independent from the topological events. As it is shown in Fig. 24, the highest values of overhead are generated by DSDV. Regardless of the difference between the overhead produced by DSDV and FSR, this protocol generates a higher quantity of routing messages when the topology changes frequently.

Unlike DSDV, ERBOR periodically announces only the changes based on the updates received from the MPRs. Thus, the effect of the topological changes on the overhead of ERBOR, is reduced. Obviously, when the network members are moving fast, they reach their final position faster and the topology becomes stable. Since ERBOR announces the routing updates periodically only, the overhead of this protocol is lowered after its convergence. Otherwise, when the network members are moving slowly, the topological events are more continuous. Hence, it is also noticeable that ERBOR generates more overhead when the network members are moving slowly. Table 6 summarizes the results obtained in this section.

Conclusion

After implementing ERBOR and putting it to the test, we obtained positive results that can be used to improve it in different aspects. For instance, a local path repair technique like the one used in ABR can limit the routing update dissemination. Instead of immediately announcing that a link with an MPR is lost, the node announces this routing update in its neighborhood to search for an alternative path. If the path repair process fails, a path removal update is announced. This method can be effective when the network members are moving at slow/medium speed. Otherwise, the route repair attempt will only slowdown the convergence. A simpler idea is to include the FSR method in order to accelerate the convergence without elevating the overhead. In future studies we intend to include other methods to enhance the protocol adaptability to the user application requirements.

Acknowledgement

The authors would like to thank the reviewer for his effort and valuable comments.

Funding Information

The authors of this paper have no financial funding to report.

Author's Contributions

Both authors have contributed equally during this work.

Bambrik Ilyas: Was in charge of: (a) implementing the protocol and the development of the simulator, (b) interpreting the results.

Didi Fedoua: Contributed in: (a) the study of the related works, (b) protocol testing.

Ethics

The corresponding author states that this paper is original. Both authors have read the paper and approved of its content.

References

Alshanyour, M. and U. Baroudi, 2008. Bypass AODV: Improving Performance of Ad Hoc On-Demand Distance Vector (AODV) Routing protocol in wireless ad hoc networks. Proceedings of the 1st International Conference on Ambient Media and Systems, Feb. 11-14, Quebec, Canada. DOI: 10.4108/ICST.AMBISYS2008.2895

Andreas, T., L. Thomas, G. Hannes, P. Bernd and K. Aaron *et al*., 2009. OLSR development.

Bambrik, I. and F. Didi, 2013. Effective routing based on overhead reduction. Int. J. Applic. Innovat. Eng. Manage., 2: 102-110.

Basagni, S., I. Chlamtac, V.R. Syrotiuk and B.A. Woodward, 1998. A distance routing effect algorithm for mobility (DREAM). Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking, Oct. 25-30, Dallas, TX, USA, pp: 76-84. DOI: 10.1145/288235.288254

Chandramohan, K. and P. Kamalakkannan, 2014. Traffic Controlled-Dedicated Short Range Communication: A Secure Communication Using Traffic Controlled Dedicated Short Range Communication Model In Vehicular Ad Hoc Networks For Safety Related Services. J. Comput. Sci., 10: 1315-1323. DOI: 10.3844/jcssp.2014.1315.1323.

Clausen, T. and P. Jacquet, 2003. Optimized link state routing protocol. IETF Internet Draft.

Day, K., A. Touzene, B. Arafeh and N. Alzeidi, 2011. Parallel rrouting in mobile ad hoc networks. Int. J. Comput. Netw. Commun., 3: 77-94. DOI: 10.5121/ijcnc.2011.3506

Deepak, K.P., K. Rakesh and A.K. Daniel, 2013. Performance analysis and behavioral study of proactive and reactive routing protocols in MANET. Int. J. Adv. Res. Comput. Commun. Eng., 2: 1789-1796.

Dorigo, M. and L.M. Gambardella, 1997. Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Trans. Evolutionary Computat., 1: 53-66. DOI: 10.1109/49.779923

De Couto, D.S.J., A. Daniel, J. Bicket and M. Robert, 2003. A high-throughput path metric for multi-hop wireless routing. Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, Sept. 14-19, San Diego, CA, USA, pp: 134-146. DOI: 10.1145/938985.939000

Dube, R., C.D. Rais, K.Y. Wang and S.K. Tripathi, 1997. Signal stability-based adaptive routing for ad hoc mobile networks. IEEE Personal Commun. Magazine, 4: 36-45. DOI: 10.1109/98.575990

Gerla, M., G. Pei, H. Xiaoyan and T.W. Chen, 2002. Fisheye state routing protocol for ad hoc networks. IETF Internet Draft.

Gopalakrishnan, S. and P. Ganeshkumar, 2015. Secure and Efficient Transmission in Mobile Ad hoc Network to Identify the Fake ID's Using Fake ID Detection Protocol. J. Comput. Sci., 11: 391-399. DOI: 10.3844/jcssp.2015.391.399

Hass, Z.J., R. Pearlman and P. Samar, 2002. The Zone Routing Protocol (ZRP) for ad hoc networks. IETF Internet Draft.

JW, 1995. Sun Microsystems.

Jiang, M., J. Ji and Y.C. Tay, 1999. Cluster based routing protocol. National University of Singapore.

Joa, N.M. and I.T. Lu, 2002. A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks. IEEE J. Selected Areas Commun., 17: 1415-1425. DOI: 10.1109/49.779923

Johann, L., M.B. José and G.V. Jorge, 2004. Analysing the overhead in mobile ad hoc network with hierarchical routing structure. Technical University of Catalonia.

Johnson, D., Y. Hu and D. Maltz, 2007. The Dynamic Source Routing Protocol (DSR) for mobile ad hoc networks for IPv4. RFC.

Juliusz, C., 2010. The babel routing protocol. IETF Internet Draft.

Kanniche, H., F. Louati, M. Frikha and F. Kamoun, 2011. A QoS routing protocol based on available bandwidth estimation for wireless ad hoc networks. Int. J. Comput. Netw. Commun., 3: 219-239. DOI: 10.5121/ijcnc.2011.3114

Kavita, P. and S. Abhishek, 2011. A Comprehensive Performance Analysis of Proactive, Reactive and Hybrid MANETs Routing Protocols. Int. J. Comput. Sci., 8: 432-441.

Ko, Y.B. and N.H. Vaidya, 2000. Location aided routing in mobile Ad hoc networks. Wireless Networks J., 6: 307-321. DOI: 10.1023/A:1019106118419

Marina, M.K. and R.S. Das, 2006. Ad hoc on-demand multipath distance vector routing. Wireless Commun. Mob. Comput., 6: 969-988. DOI: 10.1002/wcm.432

NS2, 1995. Network simulator 2 website.

Neumann, A., C. Aichel and M. Lindner, 2008. Better approach to mobile ad-hoc Networking. IETF Internet Draft.

Park, V.D. and M.S. Corson, 1997. A highly adaptive distributed routing algorithm for mobile wireless networks. Proceedings of the 6th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'97), Kobe, Japan. pp: 1405-1413. DOI: 10.1109/INFCOM.1997.631180

Pei, G., M. Gerla and H. Xiaoyan, 1999. A wireless hierarchical routing protocol with group mobility. IEEE Wireless Communications and Networking Conference (WCNC1999), New Orleans, L.A., USA. 1538-1542. DOI: 10.1109/WCNC.1999.796996

Perkins, C.E. and P. Bhagwat, 1994. Highly dynamic destination-sequenced distance-vector routing for mobile computers. ACM SIGCOMM' 94 Conference on Communications Architectures, Protocols and Applications, New York, USA. 234-244. DOI: 10.1145/190314.190336.

Perkins, C.E., R.E. Belding and S. Das, 2003. Ad hoc On-Demand Distance Vector (AODV) Routing. DOI: 10.17487/RFC3561

Poonkuzhali, R., S.M. Yakub and S. Annaji, 2014. A premeetive link state spanning tree source routing protocol for mobile ad hoc networks. J. Comput. Sci., 10: 85-90. DOI: 10.3844/jcssp.2014.85.90

Sensarma, D. and K. Majumder, 2013. An efficient ant based QoS aware intelligent temporally ordered routing algorithm for MANETs. Int. J. Comput. Netw. Commun., 5: 189-203. DOI: 10.5121/ijcnc.2013.5415

Surjeet., A. Parkash and R. Tripathi, 2014. Bandwidth constrained priority-based routing algorithm for mobile ad hoc networks. Int. J. Commun. Netw. Syst. Sci., 7: 141-150. DOI: 10.4236/ijcns.2014.75016

Sven, J., 2005. FSR Patch for NS2, Sourceforge.NET.

Toh, C.K., 1997. Associativity-based routing for ad hoc mobile networks. Wireless Personal Commun., 4: 103-139. DOI: 10.1023/A:1008812928561

Vidal, J.G. and B.M. José, 2010. Cluster based QoS routing protocol for MANET. Int. J. Comput. Theory Eng., 2: 1793-8201.