# A Cluster-Based Load Balancing Between Satellite Gateways in a MANET

Michael Crosnier*†, Riadh Dhaou†, Fabrice Planchou* and Andre-Luc Beylot†
*Astrium, 31 avenue des cosmonautes, 31402 Toulouse Cedex 04, France
†University of Toulouse, IRIT-ENSEEIHT, 2 rue Charles Camichel,BP7122, 31071 Toulouse Cedex 7, France

***Abstract*—Mobile Ad-hoc Network, associated with satellite connectivity, is a promising solution to provide communication for safety professionals where the standard terrestrial network is not available. Since satellite links are throughput and delay constrained, load distribution is a key mechanism in order to meet safety requirements. As a consequence, this paper presents a load balancing mechanism which distributes traffics among different satellite gateways of a mobile ad-hoc network. The principle is based on the OLSR routing protocol and relies on the correspondence between satellite gateway load and the size of the cluster served by this gateway. The specificity of the proposed mechanism is to tailor the load balancing procedure to the satellite parameters. Besides, the principle is very simple and the complexity lies in the additional mechanisms that limit untoward aftermath of load balancing with inappropriate scenarios.**

***Index Terms*—Load balancing, MANET, OLSR, satellite gateways.**

## I. Introduction

Civil security is experiencing an upturn thanks to the next generation of safety networks. In the current context, the mobility need becomes considerable in order to enhance international and particularly European cooperation .Thanks to its automatic configuration and its dynamic topology, MANET provides the opportunity to deploy a public safety network where no standard communications are available. For instance a MANET may be set up either in isolated areas such as mountainous regions or in a disaster theater where the common infrastructure is destroyed. We have proposed a safety network with more resources than actual terrestrial systems and with a wide coverage through the MONET project (Mechanisms for Optimization of hybrid ad-hoc networks and satellite NETwork [1]). The project is supported by the European commission. In the MONET framework, we have designed a specific architecture, keeping in mind safety requirements (Fig. 1). Portable Wi-Fi nodes, borne by safety professionals or integrated to vehicles, create a MANET located on a disaster theater. The back-office is an anchor of the MONET network. It is static and connected to the MANET thanks to satellite links. The innovation of this architecture is the interaction between a MANET and a satellite system. Satellite links connect the field of intervention to external networks. They do not prevent nodes motion and allow an independent position of the nodes on terrestrial infrastructure. Consequently, the proposed network can be deployed without awareness of the intervention location. Actions abroad become possible within a context of international safety cooperation. A satellite

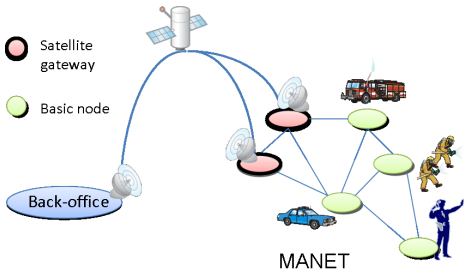


Fig. 1. MONET architecture

receiver can be mobile and be located in large areas thanks to satellite widespread coverage. Despite of this advantage in terms of mobility, the satellite integration raises new issues. It introduces a long latency, adverse for delay-constrained traffics, such as videoconference or voice over IP. In addition, the satellite links with mobility or roaming needs are also throughput limited. The most restrictive link is no longer the MANET but the satellite system. Therefore, the satellite link becomes a bottleneck and is detrimental to resource demanding applications as videoconference. One of the main issues for MANET network lies in the management of its dynamic topology by its routing protocol. We have chosen the proactive OLSR routing protocol [2]. Proactive protocols imply less delay than reactive protocols such as AODV, as depicted in [3], which is beneficial for delay constrained applications. Since OLSR protocols relies on a distance metric, traffic distribution among gateways often becomes uneven because of applications disparity. Indeed, the traffics may be concentrated through one gateway whereas another gateway is left unused. This trend is enhanced by the difference of capacity between satellite links (e.g. a factor of 4 is not rare between two satellite links), hence the necessity of load balancing mechanisms to reduce congestion on satellite entities.

In the literature, there is no algorithm that encompasses all the requirements of the MONET project. The procedure must be adapted to the OLSR routing protocols, draw advantage from the specific MONET architecture and, principally, take into account satellite characteristics. The overall performance is limited because of collisions and congestions within the MANET and in the neighborhood of gateways in common load balancing algorithms. They are focused on the Wi-Fi interface and do not deal with another gateway device as the satellite in our architecture, as in [4] and [5]. Since the satellite

978-1-4673-2676-6/12/$31.00 ©2012 IEEE

links are very limited, these algorithms are not adapted to the MONET architecture, and do not deal with the difference of capacity between satellite gateways. Furthermore, the satellite delay will jeopardize the load balancing efficiency of algorithm proposed in the literature. [8] proposes an algorithm that used a network manager behaving as the back-office, yet it is not suitable to the long satellite latency for an uplink load balancing.[9] creates, as well, a centralized controller, located in one gateway, the information exchanges, necessary to the algorithm are sent through the wired link. The responsiveness of the load balancing becomes unsatisfactory because of the long satellite delay. Secondly, a major part of load balancing papers is either agnostic or dependent on another routing protocol. Many are tailored to reactive protocols such as AODV as analyzed in [6]. Yet, these techniques cannot be transposed into OLSR without substantial additional overhead as they rely on IP source routing. Most of the agnostic mechanisms are only theoretical or add redundant information dissemination such as [7] and do not draw benefits from the OLSR specificities such as broadcast diffusion with multi-point relay (MPR). As a consequence, this paper aims to propose a simple load balancing algorithm. It is based on the satellite load instead of Wi-Fi interface parameters. In addition, it considers not only the uplink load balancing but also the downlink, drawing advantage from the back-office and being adapted to the OLSR protocol. The remaining of this paper is organized as follows. In section II, we present general load balancing algorithm. Section III explains the simulation results and finally section IV draws conclusion.

## II. Load Balancing Algorithm

Load balancing may be divided into three stages: measurement, decision and execution. The first phase consists in measuring all necessary parameters such as load and distance metric and disseminating this information to the suitable entities. During the decision phase, in-charge entities decide to trigger the next step depending on measurement values. This stage ensures that the load balancing execution is advantageous to users traffics. Then the execution stage encompasses routing modification in order to improve the traffic distribution among the different satellite links. Because of satellite delay and links asymmetry in terms of capacity, load balancing mechanisms have to be different between uplink and downlink. Since all downlink packets go through the back-office, this entity manages the downlink execution phase. As a result, the three stages of downlink load balancing are performed by the back-office. The uplink load balancing turns out to be more difficult to design since there is no central entity.

### A. Measurement phase

It stands to reason that the satellite gateway load is the main metric. To evaluate it, various methods may be used such as throughput measure or queue occupancy rate as in [6]. The distance parameter is the main metric used by the MANET routing protocol, thus our algorithm will consider only the load and distance metric. Measurements must be performed in the transmitting entities to avoid an additional satellite delay. The measurement is obtained thanks to the average amount of data received in the transmission queue of the satellite device during a sliding time window. This time window is statically defined. The back-office and the satellite gateways are respectively in charge of downlink and uplink load measures. However, the back-office is not aware of the distance metric. OLSR computes the shortest path depending on the number of hops metric and each gateway maintains distance information to each node that dwells in the MANET. Thus, each gateway informs the back-office of the distance metric sending its routing table that contains the number of hops for all MANET nodes through a new routing message named OLSR-GW.

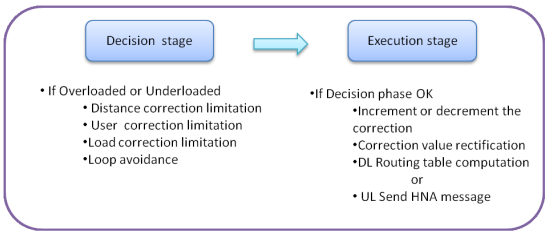


Fig. 2. Load balancing algorithm

### B. Execution phase

Despite the distinction between uplink and downlink load balancing mechanisms, the algorithm bedrocks are similar as both are cluster-based (Fig. 2). We want to design a very simple execution phase, integrated to the OLSR protocol. The most elementary mechanism is based on the adaptation of the cluster size according to the load. It leads to an increase or decrease of the number of nodes served by a gateway if it is underloaded or overloaded, respectively. In order to integrate this behavior in OLSR, we transpose the load measure into a new metric named correction (corr). The principle is simple, at each execution phase, the correction value is incremented if it is overloaded according to the decision algorithm. This correction value is sent thanks to a new field in the Host and Network Association (HNA) message. Each MANET node receives it and computes the shortest path algorithm with the distance metric plus the correction. For instance, in fig. 3, the satellite gateway 1 is overloaded and therefore, the correction value is incremented. Consequently, the number of nodes served by this gateway, and thus the load, are automatically reduced.

### C. Decision Phase

The simplicity of the execution phase has consequences on the decision phase. The originality of this algorithm is to concentrate the complexity within the decision phase since the gateways and the back-office have more energy and computing resources than basic MANET nodes. An entity is considered as overloaded and underloaded when the load measurement exceeds a threshold ($Threshold_{overloaded}$ and

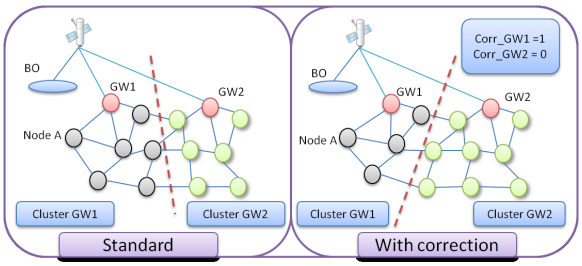


Fig. 3. Correction example

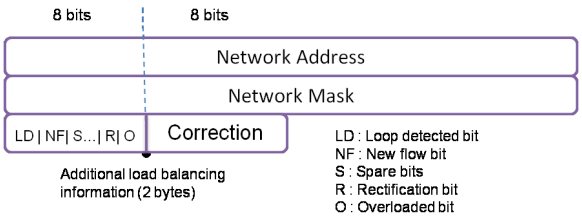


Fig. 4. HNA message

$Threshold_{underloaded}$). These thresholds are static and by default their value is 0.9 for $Threshold_{overloaded}$ and 0.2 for $Threshold_{underloaded}$.

During the cluster-based execution, several nodes change of gateway at each correction modification. The decision phase has to ensure that a correction value is beneficial concerning the overall performance. In order to cap the number of scenarios that lead to an adverse decision afterwards, limitation mechanisms are implemented. In addition, the periodicity of the load balancing algorithm induces a limited responsiveness to sudden load changes. For instance, when an application ends, an overloaded gateway with a correction value of 20 may become underloaded. If the correction should be 5 to reach the nominal operating point (neither underloaded nor overloaded), the algorithm needs 15 seconds assuming the load balancing period is equal to 1 s. Therefore, unnecessary executions reduce the responsiveness of the load balancing algorithm.

*Distance correction limitation* The distance correction limitation is implemented by the uplink and the downlink algorithms. This limitation precludes the fact that the distance between two gateways can be lower than their correction difference; i.e. for each gateway, each difference between corrections of other gateways must be lower than the number of hops between them. Packets from nodes beyond the correction difference are sent to the closest gateway and automatically transmitted through its satellite interface. Therefore, a higher correction value will have no impact on the traffic distribution (e.g. in the fig.3, the node A sends its packets to the gateway 1 even if the difference between the correction values exceeds 4).

*User correction limitation* In order to ward off scenarios when few users consume all the satellite resources, the correction value is changed only if the number of users exceeds a threshold ($threshold_{user}$). Indeed, during this scenario, the correction modification is inefficient because of the small number of active users. There is a high probability that this change has no effect. Besides, if a change occurs, the entire load is transferred to the other gateway which becomes overloaded while the previous gateway is now underloaded.

*Load correction limitation* The third limitation takes into account the scenarios when all gateways are overloaded. Without this mechanism, all the satellite gateways would increment their correction value without any impact on the routing because the correction differences would remain constant. As a result, correction values are frozen when all the gateways are overloaded. For downlink, this limitation is very simple to implement because the back-office concentrates all load balancing stages which is not true for the uplink. Gateways are not aware of load states of the others. Consequently, all gateways set an overload flag in the HNA message (Fig. 4) in order to disseminate their overload state to the other.

*Loop avoidance* This mechanism detects correction loop caused by the load balancing algorithm and prevents its adverse effects. Two gateways increment in turn their correction at each execution phase. As a consequence, the correction change do not improve the load balance and it enhances the jitter and desequencing drawbacks. A loop is detected when the difference between the corrections values of two gateways increase and decrease during a small duration and no new traffics are sent in the network. The detection mechanism controls the variation of the correction differences as well as traffics creation or disappearance. When a loop is detected, the information is broadcast in the MANET and all the gateways freeze their correction value in order to cease the loop phenomenon. To stabilize the load balancing at the right correction value, the gateway with less satellite capacity decrements its correction value. If a flow is created while the correction values are frozen, the load balancing returns to an active state. The new flow information is transmitted to other gateways setting to one the NF bit in the HNA message (fig. 4).

*Correction Rectification* This mechanism is not included in the decision but in the execution stage. The correction rectification is an algorithm implemented to cap the correction value to the minimum. The rectification avoids problems when a new gateway is activated with a correction value equals to 0. For downlink, each correction value is decreased by the minimum of all the corrections. For uplink, the rectification algorithm needs to be synchronized to ward off inconsistency and because of its complexity, the rectification is not similarly implemented. As the correction value is limited by the size of the field in the HNA message, the rectification is only run when a correction value reaches its maximum or when a new gateway is activated, setting to one the rectification flag in the HNA message (Fig. 4).

The fig. 2 summarizes the bedrock of our algorithm. After that the back-office computes the routing table with the new correction value for downlink whereas the satellite gateways send a HNA message with the correction value and the flags

necessary to the limitation and rectification algorithms. The HNA message draws advantages from MPR mechanism in order to save resource for broadcasting. HNA messages are received by basic nodes that recalculate their routing table with the new correction values.

## III. Simulation results

In order to evaluate the efficiency of the proposed load balancing algorithm, network simulation has been performed using network simulator 3 (NS3 [10]). The simulation is composed of 49 MANET nodes, two of them have an additional satellite device to link the MANET to the back-office, similarly as the architecture depicted in fig. 1. The initial positions of nodes form a grid where each node is 40 m apart from its neighbors and two MANET nodes are selected as satellite gateways (nodes 1 and 49, which are located on their respective opposite sides within the grid). The YANS Wi-Fi model with a theoretical data rate of 24 Mbps is used in ad-hoc mode in order to simulate the MANET. The channel implementation relies on a log-distance propagation loss model with an exponent value of 2.25 and the NIST error model. A satellite link are simulated thanks to two point-to-point devices (uplink and downlink) with a delay of 300ms. The first satellite link has an uplink capacity of 512 kbps and downlink capacity of 1024 kbps. The second one has an uplink capacity of 256 kbps and downlink capacity of 512 kbps. The basic OLSR protocol in NS3 has been modified to support the proposed load balancing mechanism e.g. modification of the HNA message, the limitation mechanism and the increase or the decrease of the correction value. In order to run realistic scenarios, traffics are based on safety applications. Therefore, three main applications has been selected: voice over IP, videoconference and extensive file download. Voice over IP is the essential application for safety professionals and videoconference whereas extensive file download are necessary to the next generation of safety networks. During a period $T_{application}$, sending nodes are randomly selected among the different MANET nodes. Then these nodes randomly select the beginning of the application in the $T_{application}$ interval. At each $T_{application}$ period, this procedure is repeated. These random scenarios provide the possibility to test the load balancing, and not only in favorable conditions. Some parameters are further detailed in table I.

Four scenarios patterns (A, B, C and D in table II) have been tested to highlight the load balancing efficiency in various conditions. They differ according to their mobility model (a static or randomwalk mobility model) and the presence of internal voice traffics. The randomwalk model embodies the players'motions at a pedestrian speed (up to 1.4 m/s). To get statistical figures, 73 scenarios have been performed for each pattern in which only the random variables are different. Fig. 5 and fig. 6 present cumulative distributions of differences concerning packet loss rate, between two simulations of the same scenario with and without load balancing. They concern voice and videoconference traffics, accordingly. Table II displays general statistics for all applications.

TABLE I
Physical and MAC parameters

| OLSR and load balancing parameters | | | |
|---|---|---|---|
| HNA Periodicity | 1 s | DL load balancing periodicity | 1 s |
| $Threshold_{overload}$ | 0.9 | $Threshold_{Underload}$ | 0.2 |
| $Threshold_{user}$ | 1 | $min_{cluster}$ | 0 |

| Application parameters | | |
|---|---|---|
| | Voice | Visioconference |
| Number of sending nodes | 16 | 2 |
| Application duration | 60 s | 180 s |
| $T_{application}$ | 600 s | 600 s |
| Data rate | 64 kbps | 256 kbps |
| Packet size | 160 B | 160 B |
| Bidirectional | yes | yes |

TABLE II
General statistics

| Pattern | A | B | C | D |
|---|---|---|---|---|
| Internal traffic | no | yes | no | yes |
| Random Walk | no | no | yes | yes |
| Application | statistics ( mean / max) | | | |
| Voice Loss (%) | 1.4 / 10 | 0.6 / 3.9 | 1 / 7.4 | 0.3 / 2.8 |
| Visio Loss (%) | 4.7 / 41 | 3.9 / 34 | 3 / 28 | 2.1 / 26 |
| Intern voice Loss(%) | - | 0 / 0.1 | - | 0 / 4.5 |

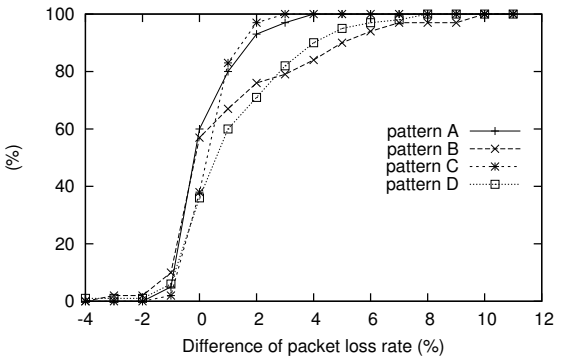


Fig. 5. Cumulative loss packet rate difference distribution for voice

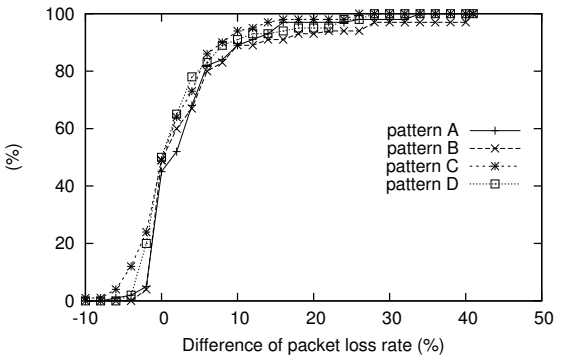


Fig. 6. cumulative loss packet rate difference distribution for videoconference

Basically, the load balancing mechanism provides benefits. It decreases the packet loss rate, since the average packet loss rate difference is positive for the four scenario patterns as for voice and videoconference traffics. It is quite satisfactory for videoconference as the average is equal to 4,5 %. Yet, the gain remains limited for voice applications (1,4 % for the pattern A) since the packet loss rate is already an average of the number of traffics during a simulation. It has more impact on voice application because there are up to 32 voice traffics (two periods of 16 voice traffic) whereas the number of videoconference traffics reaches only 4, as depicted in table II. In addition, many scenarios do not suffer from packet loss. Indeed, the random nature of application parameters (e.g. sending nodes) leads to a high occurrence of well balanced traffics. However, this does not explain the negative results. For instance, the minimum of the packet loss rate degradation, reached for the pattern A, is about 3 %. Only few scenarios lead to a deterioration thanks to the limitation mechanisms (see fig.6 and fig.5). Firstly, the videoconference traffic transfer has an untoward impact on voice applications (e.g. The maximum of voice packet loss degradation occurs when the videoconference applications reach the maximum gain). To solve this problem, we need a QoS Wi-Fi and satellite devices in order to prioritize voice packets as necessary in the MONET network to meet the requirements of civil security. Secondly, the deterioration is due to gateway flapping. It is avoided thanks to the loop detection. However, the frozen state is kept unchanged during a period until an arrival or an end of a flow. These events have to be reduced to only applications with significant data rate such as videoconference and voice to hinder too frequent detection of a new flow. The most weighty result is the substantial gain obtained for several scenarios. The maximum gain for voice applications in terms of packet loss rate is 10 % and for videoconference, it reaches 41 % (table II). When the scenario can take advantage from the load balancing algorithm, the results are quite satisfactory, 10% of scenarios exceed an improvement of 10 % of packet loss rate (fig. 5 and fig.6). As expected, the mobility model has an impact on the performance of the load balancing. The average difference is still positive, yet the occurrence of substantial gain decrease because with a random walk mobility model, the duration when gateways are overloaded are not as long, due to the nodes motion. In the same way, the additional voice traffics, sent between two nodes that dwell in the MANET, slightly decrease the overall performance. This result is a consequence of the additional collisions due to these new packets. However, these traffics do not involve major congestion as the packet loss rates are almost the same with static nodes (the difference does not exceed 0.1 % (see table II). The TCP applications performance are limited as the New Reno implementation in NS3 does not include the windows scaling mechanism. Therefore, the TCP throughput is very limited because of the satellite link delay.

## IV. Conclusion

To provide safety communications in a medium scale disaster, we have designed an architecture aimed at mobility and roaming aspects. In this way, we have interconnected a MANET and a satellite system. The uneven load distribution among activated gateways has to be solved to ensure a high quality of service to the new data consuming applications. We proposed an algorithm divided into a very simple execution phase and a more complex decision phase which is only implemented in gateways and a downlink centralized entity. Many realistic scenarios have been performed to show up its benefits and particularly for high data rate traffics. In order to enhance the efficiency of the load balancing algorithm, we have planned to implement QoS mechanism. This algorithm will be implemented in Wi-Fi nodes and carry out a real full-scale trial.

## References


[1] MONET project, http://Monet.tekever.com/home

[2] Clausen T., Jacquet P.: RFC 3626: Optimized link state routing protocol (OLSR). IETF, October 2003.

[3] Clausen T., Jacquet P., Viennot L.: Comparative Study of Routing Protocols for Mobile Ad-hoc Networks. In 1st IFIP MedHocNet Conference, 2002.

[4] Vinh Pham, Erlend Larsen, Paal E. Engelstad, and Oivind Kure: Performance analysis of gateway load balancing in ad hoc networks with random topologies. In Proceedings of the 7th ACM international symposium on Mobility management and wireless access (MobiWAC '09). ACM, New York, NY, USA, 66-74. 2009.

[5] Sriram Lakshmanan, Raghupathy Sivakumar, Karthikeyan Sundaresan: Multi-gateway association in wireless mesh networks. In Ad Hoc Networks, Volume 7, Issue 3, Pages 622-637, ISSN 1570-8705, May 2009.

[6] Kumar R., Misra M., Sarje A. K.: A Proactive Load-Aware Gateway Discovery in Ad Hoc Networks for Internet Connectivity. International Journal of Computer Networks & Communications (IJCNC), vol. 2, no.5, pp. 120-139, September 2010.

[7] Hoffmann, F.; Medina, D.: Optimum Internet Gateway Selection in Ad Hoc Networks. In Communications, 2009. ICC '09. IEEE International Conference, pp.1-5, 14-18 June 2009.

[8] Ancillotti Emilio, Bruno Raffaele, Conti Marco,: Load-balanced routing and gateway selection in wireless mesh networks: Design, implementation and experimentation. In World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium, pp.1-7, 14-17 June 2010.

[9] Juan J. Galvez, Pedro M. Ruiz, Antonio F.G. Skarmeta: Responsive online gateway load-balancing for wireless mesh networks. In Ad Hoc Networks, Volume 10, Issue 1, Pages 46-61, ISSN 1570-8705, January 2012.

[10] Network Simulator 3, http://www.nsnam.org/