Analysis and simulation of Fast-OLSR

Mounir Benzaid[1,2], Pascale Minet[1] and Khaldoun Al Agha[1,2]
[1]INRIA, Domaine de Voluceau - B.P.105, 78153 Le Chesnay Cedex, France
mounir.benzaid, pascale.minet@inria.fr
[2]LRI, Bât 490 Université Paris Sud, 91405 Orsay Cedex, France
alagha@lri.fr

Abstract— With the current increase in ad-hoc mobile networks in public domains (e.g. airports, cities, etc.), and the widespread use of the IEEE 802.11 standard in wireless LAN, there is a growing need to handle and manage fast mobility. Optimized Link State Routing protocol (OLSR) is a proactive protocol that enables routing in Mobile Ad-hoc NETworks (MANETs). Fast-OLSR is a new extension of OLSR designed to meet the need for fast mobility in ad-hoc networks. Combining Mobile-IP with Fast-OLSR ad-hoc routing could offer a complementary solution to the UMTS for fourth generation (4G) mobile networks. In these mobile wireless networks greatly overlapping coverage areas can be produced. In this paper, we evaluate the performances of Fast-OLSR in an overlapping coverage context.

Keywords: mobile ad-hoc networks, wireless networks, fast mobility, ad-hoc routing protocol, OLSR.

I. INTRODUCTION

At present, there are two main types of mobile wireless networks: cellular and ad-hoc mobile networks. When using the cellular concept, a large area is divided into sub-areas called cells. Each cell has its own base station, which provides a radio link for each mobile node in the cell. In a cellular environment, it can be possible to provide fast mobility by means of handoff operations and to guarantee zero packet loss by using a planning scheme with overlapping cells and also soft handoffs in a CDMA based system (IS95, CDMA, etc.). In situations where it is not possible to deploy a cellular infrastructure, an ad-hoc network is used [1], [2].

A Mobile Ad-Hoc NETwork (MANET) [3] is a collection of mobile nodes that communicate using a wireless medium, forming an autonomous network. With the appearance of hot spots in public domains, there is an increasing need to handle and manage fast mobility. Extending the coverage area of ad-hoc networks and taking into account fast mobility in routing protocols could offer a complementary solution to the UMTS in fourth generation (4G) mobile networks. The design of a fast and efficient routing protocol is necessary to the performance of an ad-hoc network in particular in the case of large and dense networks with fast moving nodes.

OLSR [1] is one of the protocols discussed in the MANET working group of IETF (Internet Engineering Task Force Internet). This protocol, inherits the stability of a link-state routing protocol and the availability of routes when needed due to its proactive nature. However, when a node is moving fast, the links with its neighbors are valid only during a short time interval. Hence to minimize packet loss, broken links between the node and its neighbors must be quickly detected. In a recent work [4], we have focused on how to deal with fast moving nodes in the OLSR routing protocol, and we have proposed the Fast-OLSR extension to account for fast nodes while keeping the routing overhead as low as possible. In this paper, we study the effects of the overlapping areas between adjacent nodes and the maximum velocity on packet loss rate. The remainder of this paper is organized as follows. In Section II, we briefly present OLSR and show how this protocol can be extended to take into account fast mobility. The analysis of the Fast-OLSR protocol is presented in section III. In section IV, we evaluate the performance of the Fast-OLSR protocol in terms of loss rate with overlapping areas.

II. FAST-OLSR PROTOCOL

OLSR [1] is an optimization of a pure link state routing protocol based on the concept of *multipoint relays* (*MPRs*). First, using *MPRs* reduces the size of the control messages: rather than declaring all links, a node declares only the set of links with its neighbors that are its "*MPRs*". The use of *MPRs* also minimizes flooding of control traffic. Indeed only *MPRs* of the sender forward control messages. This technique significantly reduces the number of retransmissions of broadcast control messages [5]. OLSR provides two main functionalities: the first is Neighbor Discovery by means of *Hello* messages, and the second is Topology Dissemination by means of *Topology Control* (*TC*) messages.

In an OLSR ad-hoc network, when a node is moving fast, its neighborhood changes quickly and the default *Hello* frequency in OLSR is not sufficient to track the node's motion. Therefore, routes to this node become inactive and messages sent to this node may be lost. A higher *Hello* frequency would overcome this problem, but at the cost of an additional control overhead. We have proposed in [4] an extension of OLSR to address the issues for fast mobility. Fast-OLSR has two main objectives: first, the induced control traffic is tuned to node mobility, in such a way that it allows a fast node's motion to be tracked, i.e. the control traffic increases with mobility. Second, the bandwidth consumed by the control traffic must remain reasonable.

The basic idea of Fast-OLSR is to enable a fast moving node, m, to quickly discover a small number of neighbors. Among these neighbors, a small number of *MPRs* are selected to maintain connectivity with other nodes in the network. To achieve this, a fast moving node establishes a small number of *symmetric* links refreshed at a high frequency by means of *Fast-Hellos*. Such links are called Fast links and this high frequency is determined by the refreshing period "*Fast-hello-interval*" (the default value is the *Hello_interval*/10). A node m can select no more than a maximum number of *MPRs*, this number is called "*Max-Default-MPR*". A Fast link that has not been refreshed within "*Fast-neighbor-hold-time*" is considered broken.

Neighbor discovery in Fast-OLSR operates with three main procedures:

- Switching to the *Fast-Moving/Default* mode: when a node detects that it is moving fast (e.g. a high number of changes in its neighborhood), it switches to the *Fast-Moving* mode and starts sending *Fast-Hellos*, and vice versa.
- Establishing Fast links: a node m in Fast-Moving mode sends *Fast-Hello* messages at high frequency. A *Fast-Hello* is smaller then a *Hello*.
- Refreshing Fast links and Detecting new/broken links: Fast links are refreshed by sending periodic *Fast-Hellos*. If a *MPR* of m has not received the *Fast-Hello* of m within *Fast-neighbor-hold-time* (i.e. $3\times$*Fast-hello-interval*), it considers the link with m to be broken and it sends a *TC* message to inform all nodes in the network that it is no longer a *MPR* for m.

Selection of multipoint relays

A fast moving node, m, only partially knows its two-hop neighborhood. Thus, it is necessary to have a simplified heuristic to select the set of *MPRs* of m, denoted *MPR*(m). The set of neighbors in *Default* mode which reply to m with *Fast-Hellos* and have not been selected as *MPRs* of m is called the candidate set and denoted $C(m)$. The selection of *MPRs* is computed as follows:

- Initially *MPR*(m) = $\emptyset$, $C(m) = \emptyset$.
- If m receives a *Fast-Hello* containing the address of m from a node m_d in *Default* mode and $m_d \notin$ *MPR*(m) $\cup\ C(m)$.
  - If the cardinal of *MPR*(m) is lower than *Max-Default-MPR*, m_d is inserted in *MPR*(m).
  - Otherwise, m_d is inserted in $C(m)$.
- If a Fast link with a node in *MPR*(m) is broken, m extracts from $C(m)$ a node m_d, the most recently inserted, and inserts it in *MPR*(m).

This heuristic enables a broken Fast link to be recovered quickly (i.e. after *Fast-neighbor-hold-time*). It also avoids useless changes of m's *MPRs*: when the cardinal of *MPR(m)* is equal to *Max-Default-MPR* and m detects a new candidate node, this candidate cannot be selected as a *MPR*, as long as the *MPRs* of m are still valid.

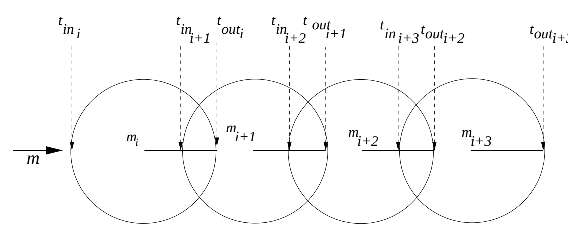
III. Fast-OLSR analysis

In this section we analyse the performances of Fast-OLSR protocol in an ad-hoc network with overlapping areas.

A. Models and notations

We assume that neither link layer notification nor information on signal attenuation are available. In this analysis we use the following three models.

- Nodes model: we distinguish three types of nodes:
  - a node m in *Fast-moving* mode.
  - a node C located in a random position that sends a Constant Bit Rate (CBR) traffic to m.
  - m_i, $\forall i \in [1, n]$ fixed nodes in *Default* mode.
- Overlapping model: as depicted in Figure 1, two adjacent nodes m_i and m_{i+1} have overlapping coverage areas. We use the following notations:
  - D_i: the diameter of the coverage area of node m_i.
  - $nbcover_i$: the maximum number of coverage areas which m belongs to when it is located in the coverage area of m_i. In Figure 1, $nbcover_i$ is equal to 2.
  - $\alpha_{i,i+j}$: the diameter fraction of the coverage area of m_i that is common to the coverage area of m_{i+j} on the trajectory of m, for j $\in [1, nbcover_i - 1]$.
- Mobility model: the node m is moving horizontally from m_1 to m_n. The velocity of m, denoted v, is constant. We use the following notations:
  - t_{in_i} (resp. t_{out_i}): the time when m enters (resp. leaves) the coverage area of m_i.
  - t_{TC_i}: the time when C receives a *TC* advertising that m_i is a *MPR* of m.
  - $t_{TC'_i}$: the time when C receives a *TC* advertising that m_i is no longer a *MPR* of m.
  - $t_{effective_i}$: the time when C uses m_i to reach m.
  - $t_{prop_{pkt}}$: propagation time of a packet between m and C.

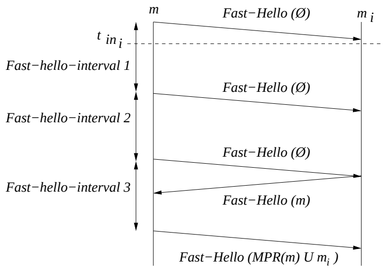


89 Fig. 1. A layout of four cells.

B. Properties

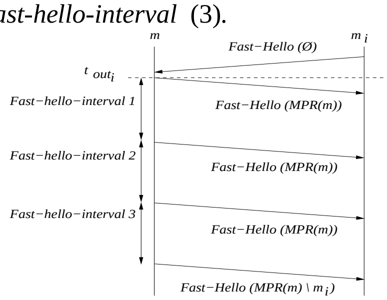
We can show that Fast-OLSR has the following properties:

- **Establishment of a Fast link:** the maximum time needed to establish a Fast link is bounded by 3 $\times$ *Fast-hello-interval* (1). This property is illustrated by the following diagram.

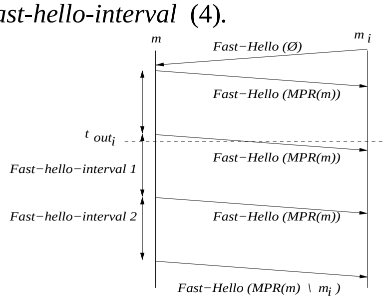


First, m sends an empty *Fast-Hello* (if it has just switched to *Fast-Moving* mode) or a *Fast-Hello* containing the addresses of its *MPRs* (if they still have valid entries in *MPR*(m)). After a maximum of 3 × *Fast-hello-interval* m receives the *Fast-Hello* of a node m_i (in *Default* mode) containing the address of m. Then, m will select and declare m_i as a *MPR* in the next *Fast-Hello* if the cardinal of *MPR*(m) is lower than *Max-Default-MPR* (see section II).

- **Detection of a broken Fast link:**
  the maximum time needed to detect the breakage of a Fast link is: 3 × *Fast-hello-interval* (2).
- **Extraction from *MPR*(m):**
  if a node m leaves the coverage area of m_i ∈ *MPR*(m) at time t_{out_i}, then m extracts m_i from *MPR*(m),
  – at the latest at time:
  $t_{out_i} + 3 \times$ *Fast-hello-interval* (3).

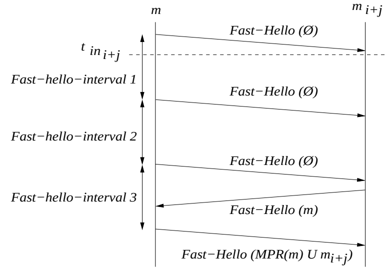


  – at the earliest at time:
  $t_{out_i} + 2 \times$ *Fast-hello-interval* (4).

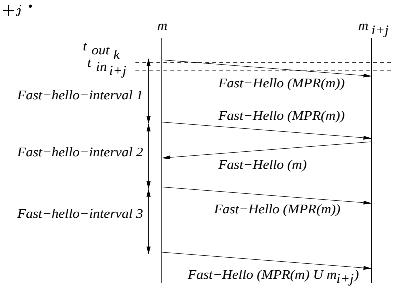


- **Insertion in *MPR*(m):**
  if at time $t_{in_{i+j}}$, m being in the coverage area of node m_i, enters the coverage area of m_{i+j} which is the most recent neighbor to reply with a *Fast-Hello* containing the address of m with $j \in [1, nbcover_{i-1}]$, then m inserts m_{i+j} in *MPR*(m):

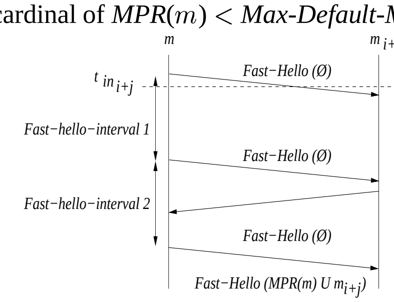
  ∗ at the latest at time:
  - $t_{in_{i+j}} + 3 \times$ *Fast-hello-interval* (5)
    if the cardinal of *MPR*(m)<*Max-Default-MPR*



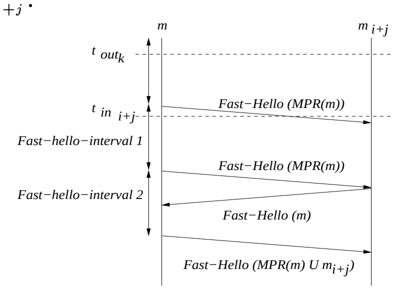
  - *Max*($t_{in_{i+j}}$,t_{out_k})+3× *Fast-hello-interval* (6)
    if the cardinal of *MPR*(m) = *Max-Default-MPR*, and at time $t_{out_k} \geq t_{in_{i+j}}$ m leaves the coverage area of node m_k. m_k is the first *MPR* extracted from *MPR*(m) at a time t $\geq t_{in_{i+j}}$.



  ∗ at the earliest at time:
  - $t_{in_{i+j}} + 2 \times$ *Fast-hello-interval* (7)
    if the cardinal of *MPR*(m) < *Max-Default-MPR*



  - *Max*($t_{in_{i+j}}$, t_{out_k}) + 2 × *Fast-hello-interval* (8)
    if the cardinal of *MPR*(m) = *Max-Default-MPR*, and at time $t_{out_k} \geq t_{in_{i+j}}$ m leaves the coverage area of node m_k. m_k is the first *MPR* extracted from *MPR*(m) at a time t $\geq t_{in_{i+j}}$.



- **Useless choice of a *MPR*:**
  when m selects m_{i+j} as a *MPR*, m must still belong to the coverage area of m_{i+j} (9).

If property 9 is not met, and m_{i+j} is the last hop used to reach m, then any traffic sent to m via m_{i+j} is lost. We must then have: $\forall i \in [1, n]$ such that m_i is a *MPR* of m and $\forall j \in [1, nbcover_i - 1]$, such that m_{i+j} is chosen as a *MPR* to replace m_i:

$$t_{out_i} + (3 \times \textit{Fast-hello-interval}) < t_{out_{i+j}} \quad (10)$$

Condition 10 can be expressed in terms of distance: $\forall i \in [1, n]$ and $\forall j \in [1, nbcover_i - 1]$:

$$\alpha_{i,i+j} \times D_i + (3 \times \textit{Fast-hello-interval} \times v) < D_{i+j} \quad (11)$$

C. Evaluation of time to change routes at node C:

Let us consider a node C reaching m via its MPR m_i, we now compute the time $t_{effective_{i+j}}$ when C uses m_{i+j} to reach m. The time $t_{effective_{i+j}}$ depends on the length of the new route via m_{i+j}, because Fast-OLSR maintains only one route (the shortest one) to reach a destination. This time $t_{effective_{i+j}}$ is computed in two cases:

- if the new route via m_{i+j} is shorter than the old one via m_i, then $t_{effective_{i+j}} = t_{TC_{i+j}}$ (12)
- otherwise, $t_{effective_{i+j}}$ = *Max*($t_{TC_{i+j}}$,$t_{TC'_i}$) (13), with $t_{TC'_i}$ being the time when C knows that m can no longer be reached via m_i.

The maximal value of $t_{TC_{i+j}}$ is computed using (5) and (6):

- $t_{in_{i+j}}$ + (3 × *Fast-hello-interval*)+ $t_{prop_{TC}}$ if cardinal of *MPR*(m) < *Max-Default-MPR*
- *Max*($t_{in_{i+j}}$,t_{out_k}) + (3 × *Fast-hello-interval*) + $t_{prop_{TC}}$ if cardinal of *MPR*(m) = *Max-Default-MPR*, and m_k is the first *MPR* extracted from *MPR*(m) at a time t $\geq t_{in_{i+j}}$;

The minimal value of $t_{TC_{i+j}}$ is computed using (7) and (8):

- $t_{in_{i+j}}$ + (2 × *Fast-hello-interval*)+ $t_{prop_{TC}}$ if cardinal of *MPR*(m) < *Max-Default-MPR*
- *Max*($t_{in_{i+j}}$,t_{out_k}) + (2 × *Fast-hello-interval*) + $t_{prop_{TC}}$ if cardinal of *MPR*(m) = *Max-Default-MPR*, and m_k is the first *MPR* extracted from *MPR*(m) at a time t $\geq t_{in_{i+j}}$;

Also, the maximal value of $t_{TC'_i}$ is: t_{out_i} + (3 × *Fast-hello-interval*) + $t_{prop_{TC}}$, using (3).
The minimal value of $t_{TC'_i}$ is : t_{out_i} + (2 × *Fast-hello-interval*) + $t_{prop_{TC}}$, using (4). So the value of $t_{effective_{i+j}}$ is given by the following formulas:

* if (condition 12 is verified) then
  - if (cardinal of *MPR*(m)<*Max-Default-MPR*) then $t_{in_{i+j}}$ + (2 × *Fast-hello-interval*) + $t_{prop_{TC}}$ $\leq$ $t_{effective_{i+j}}$ $\leq$ $t_{in_{i+j}}$ + (3 × *Fast-hello-interval*) + $t_{prop_{TC}}$ (14).
  - if (cardinal of *MPR*(m)=*Max-Default-MPR* and m_k is the first *MPR* extracted from *MPR*(m) at a time t $\geq t_{in_{i+j}}$) then t_{out_k} + (2 × *Fast-hello-interval*) + $t_{prop_{TC}}$ $\leq$ $t_{effective_{i+j}}$ $\leq t_{out_k}$ + (3 × *Fast-hello-interval*) + $t_{prop_{TC}}$ (15).
* if (condition 13 is verified) then
  - if (cardinal of *MPR*(m) < *Max-Default-MPR*) then *Max*($t_{in_{i+j}}$,t_{out_i}) + (2 × *Fast-hello-interval*) + $t_{prop_{TC}}$ $\leq t_{effective_{i+j}}$ $\leq$ *Max*($t_{in_{i+j}}$,t_{out_i}) + (3 × *Fast-hello-interval*) + $t_{prop_{TC}}$ (16).
  - if (cardinal of *MPR*(m) = *Max-Default-MPR* and m_k is the first *MPR* extracted from *MPR*(m) at a time t $\geq t_{in_{i+j}}$) then *Max*(t_{out_k},t_{out_i}) + (2 × *Fast-hello-interval*) + $t_{prop_{TC}}$ $\leq t_{effective_{i+j}}$ $\leq$ *Max*(t_{out_k},t_{out_i}) + (3 × *Fast-hello-interval*) + $t_{prop_{TC}}$ (17).

IV. Simulation

We first describe our simulation runs and then compare the simulation results with the analysis results.

A. Simulation measurements and parameters

The simulation model is the one described in section III, with $n = 54$ (see figure 1) and *Max-Default-MPR*=2. $\forall i \in [1, n]$, $nbcover_i$ =2. The refreshing period of *Fast-hello-interval* is chosen equal to the transmission period of the *SACCH* or *PTCCH* channels used in the cellular networks (i.e. 120 ms) [6].

Let us assume that a CBR stream of packets is being transmitted from node C to mobile m. With our model, C is always two hops from m but the next hop to reach m changes when m moves. We are interested in evaluating the number of lost packets due to handoffs, assuming that neither buffering nor retransmission are available.

- **Packet Loss:** This number is computed as follows. We first compute the number of lost packets during a handoff between nodes m_i and m_{i+j}. Packets are lost during the time interval $[t_1, t_2$) where t_1 is the time when m loses connectivity with m_i and t_2 is the time when m can be reached through m_{i+j}.
- **Parameters:** We have measured the packet loss in various simulations. In all simulations, we assume a constant coverage area D_i =D, $\forall i \in [1, n]$. Each simulation is characterized by two parameters:
  - the velocity of m in kilometers per hour.
  - the overlapping ratio $\alpha_{i,i+1}$: we have $\forall i \in [1, n]$, $\alpha_{i,i+1}$ = α.

B. Simulation results versus analysis results

We have tested the overlapping fraction α to see how it affects the loss rate and compared the simulation results with analysis results. Figure 2 highlights the performance of Fast-OLSR in terms of loss rate, with regard to the value of α when m moves from m_1 to m_n and there is a traffic from C to m. The new route to reach m from C via a *MPR* m_{i+j} is equal, in the number of hops (2 hops), to the old one via m_i. Mobility ranges from 20 km/h to 150 km/h. The two plots in Figure 2 depict the packet loss rate versus mobility with α = 3% and 25%. In each plot the central curve *loss_sim*, i.e. the loss rate simulation curve, is always between the minimum boundary curve *loss_min* and the maximum boundary curve *loss_max* that we found in the theoretical analysis given in section III.

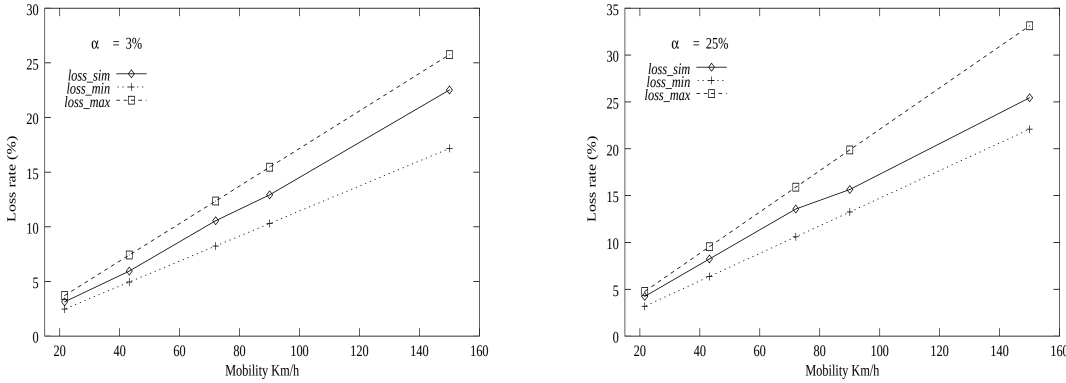


Fig. 2. Loss rate versus mobility.

A loss interval incurred at node m, denoted t_{loss_i}, is computed as follows:

$$t_{loss_i} = t_{effective_{next(i)}} - t_{out_i} \text{ (18).}$$

such that $t_{effective_{next(i)}} - t_{out_i}$ expresses the latency before C uses effectively $m_{next(i)}$ to reach m after m left the coverage area of its current *MPR* m_i. The loss rate obtained by simulation, *loss_sim*, is obtained by summing t_{loss_i} divided by the simulation duration.

The boundaries are obtained using formula 16 with $t_{prop_{TC}}$ being negligible and *Max*($t_{in_{i+j}}$,t_{out_i}) = t_{out_i}. So t_{loss_i} between two adjacent *MPRs* becomes: $2\times$*Fast-hello-interval* $\leq t_{loss_i} \leq 3\times$*Fast-hello-interval* (19).

We then get $loss\_min = \frac{2\times n\times \textit{Fast-hello-interval}}{t_{out_n} - t_{in_1}}$, $loss\_max = \frac{3\times n\times \textit{Fast-hello-interval}}{t_{out_n} - t_{in_1}}$. This can also be written $loss\_min = \frac{2\times n\times v\times \textit{Fast-hello-interval}}{D\times(n-(n-1)\times\alpha)}$, $loss\_max = \frac{3\times n\times v\times \textit{Fast-hello-interval}}{D\times(n-(n-1)\times\alpha)}$

As Figure 2 shows, the loss rate becomes greater as the speed increases. Indeed, the new route being equal in number of hops to the old one, C is ensured to use the new route only after the breakage detection of the old one (i.e. $3\times$ *Fast-hello-interval* + $\frac{\alpha\times D}{v}$).

V. Conclusion

In this paper we have analysed and studied the effects of overlapping in Fast-OLSR ad-hoc routing. The performances of Fast-OLSR are evaluated by simulation. The results lead us to conclude that in order to guarantee zero packet loss in an overlapping context with Fast-OLSR it is possible to use soft-handoffs with data buffering. In our case, data must be buffered during $3\times$ *Fast-hello-interval* $+ \frac{\alpha\times D}{v}$.

In further work, we will study the impact of the value of *Max-Default-MPR* and selection of *MPR* nodes on the performance of Fast-OLSR.

References

[1] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayuam, L. Viennot, “Optimized Link State Routing Protocol,” draft-ietf-manet-olsr-07.txt, IETF, http://www.ietf.org, November 2002.

[2] J. Broch, D. Maltz, D. Johnson, Y. Hu, J. Jetcheva, “A performance comparison of multi-hop wireless ad hoc network routing protocols,” in *ACM Mobicom’98*, (Dallas, USA), October 1998.

[3] Mobile Ad-hoc Networks (MANET), *manet-charter*. http://www.ietf.org/html.charters/manet-charter.html, 2002.

[4] M. Benzaid, P. Minet, K. Al Agha, “Integrating fast mobility in the OLSR routing protocol,” *Fourth IEEE Conference in Mobile and Wireless Communications Networks*, Stockholm, Sweden, September 2002.

[5] A. Qayyum, A. Laouiti, L. Viennot, “Multipoint relaying technique for flooding broadcast messages in mobile wireless networks,” in *HICSS: Hawaii International Conference on System Sciences*, (Hawaii, USA), January 2002.

[6] 3GPP TS 03.64, *Radio Access Network, General Packet Radio Service (GPRS), overall description of the GPRS radio interface Stage 2,v8.10.0*, February 2002.