Delay Prediction Approach for Cyclic Mobility Models in Ad hoc Networks

Jihen Bokri, Sofiane Ouni and Leila Saidane
CRISTAL laboratory
Ecole Nationale des Sciences de l'Informatique (ENSI)
Tunis, Tunisia
e-mails : jihen.bokri@ensi-uma.tn, Sofiane.Ouni@insat.rnu.tn, leila.saidane@ensi.rnu.tn

Abstract – **Guaranteeing the respect of the deadline on the end-to-end transmission delay for a hard real time traffic is a big challenge, especially in mobile Ad hoc networks. In fact, in such networks, the movement of the nodes can lead to the breakage of the current path between the source and the destination and therefore the establishment of a new path which may provide a transmission delay that exceeds the deadline. In this paper, we propose a delay prediction approach which allows guaranteeing the respect of the deadline regardless of the used path for cyclic mobility models. Indeed, we aim to predict all the possible paths between two nodes and calculate the end-to-end transmission delay in the worst case. Thus, we can guarantee the respect of the deadline if this latter is superior to the calculated delay. By comparing the worst end-to-end delay values calculated by our prediction approach with the delay values obtained by the network simulator NS2 (Network Simulator 2) for the same network, we perceive that the NS2 delays are always lower than the end-to-end transmission delay of the worst case calculated by our approach. This proves that our prediction approach allows guaranteeing the respect of the deadline on transmission delays for hard real time applications.**

Keywords-hard real-time; deadline guarantee; communication delay; prediction; mobility

I. Introduction

The hard real-time applications, which are used in so many domains such as industrial environments, have strict requirements regarding reliability, latency and end-to-end transmission delay. However, respecting the required transmission delay is the most challenging issue since the transmitted data can be easily delayed because of the nodes mobility. In fact, the mobile nodes movements can break the already established path between the source and the destination which requires a new path finding procedure. This can prevent the real-time traffic from satisfying the deadline on the end-to-end transmission delay. For that, many researches are interested in the mobility prediction for Ad hoc networks. However, most of the researches are based on specific equipment, such as GPS (Global Positioning System) [3][6][9][10] or based on non-accurate methods such as probability approaches [4] or approaches which are based on transmission strength [11].

In this paper, we propose a deterministic prediction approach which can guarantee the respect of the deadline on the end-to-end transmission delay for a real-time traffic even in a mobile environment. In fact, our approach predicts all the possible paths between the source and the destination and calculates the end-to-end transmission delay in the worst case for Ad hoc networks having nodes with cyclic mobility models. Thus, we can know in advance if the deadline on the transmission delay of the real traffic will be respected by comparing it with the calculated worst case transmission delay.

This paper is organized as follows. Section 2 presents the mobility prediction approaches in the literature. Section 3 describes our delay prediction approach. In Section 4, we explain the response time analysis of our approach. Sections 5 and 6 are dedicated to the evaluation of our approach and the conclusion, respectively.

II. Mobility Prediction approaches in the literature

So many researches were interested in the mobility prediction for Ad hoc networks. For example, in [1], J. Wang proposes a mobility prediction method for Wireless Ad hoc networks having a reliable service composition. This latter consists in integrating many services which are provided by different service providers in the network. In this research, the only required information for the mobility prediction is the estimated time that a service provider will be present in the current environment. However, some service providers can overestimate this time of availability which leads to the uncertainty in the mobility prediction. Hence, the author aims to characterize this uncertainty using two different models which are the probabilistic-free model and the probabilistic model. For that, J. Wang presents heuristic algorithms which are based on the fact that each service provider has a predicted future location described as a function of time.

While the probabilistic model cannot be used for Hard real time communications since it is not deterministic, the probability-free model is also not suitable for such kind of transfer. In fact, it aims to minimize the risk of uncertainty about the links between nodes but it does not provide a sophisticated solution.

In [3], a mobility prediction solution is proposed, which exploits a mobile user's non-random traveling pattern. In fact, the author presents an enhancement to unicast and multicast routing protocols which predict the network topology changes. To this end, he utilizes GPS location information to predict disconnections by estimating the expiration time of the link between two adjacent nodes. For that, the researcher assumes that the velocities of the nodes are constant and he presents an algorithm which decides whether a link between two given nodes has expired.

The decision is based on the probability **p,** which is the result of the current time of separation (T_{jk}) for the pair of nodes j and k divided by their maximum time of separation (p= T_{jk}/Max(T_{jk})). The time of separation is given by the diameter of the node's coverage area (D) divided by the relative velocity of these two nodes (V_{jk}). Hence, if q(=1-p)<αp, then the link is expired (while α is a constant factor of persistence).

Although this research presents a mobility prediction solution which allows to identify in advance the links which will expire, this approach is not suitable for Hard real time transmissions since it is based on probability to decide about the link expiration.

Some researches concentrate on the link availability estimation to deal with the mobility issue. In fact, Huang and Bai [4] presented an approach of link availability estimation within a random mobility Ad hoc network. According to the researchers, the link availability is defined as the probability that a link remains available for a period of time t. Hence, they present an analytical expression of link availability based on the estimation of the initial distance between two mobile nodes.

Obviously, this approach cannot be used in the case of Hard real time transmissions since it is based on estimations and probabilities.

Other researches, such as Chegin et al. [5], focus on some specific areas to predict the movements of the mobile nodes and consequently the links duration. In fact, Chegin et al. [5] are interested in predicting the links expiration time in an urban area. For this purpose, they use a map file containing the locations of all the vertical and horizontal streets, as well as the cross points of the roads. Then, they run a prediction algorithm which brings out a prediction table including the links expiration times. This prediction table will be used later by the routing algorithm in order to select the optimal path.

However, the presented solution in [5] is specifically used, as the authors said, with the proactive routing protocols. Also, they assume that the mobile nodes are localized at any time using the GPS which is not always obvious.

In [6][9], the authors present a mobility prediction method which predicts the future distance between two neighboring nodes using learning automaton. In fact, Mousavi et al. [6][9] base their work on the proposed prediction scheme in [7] and they enhance it to have more accurate results.

In [7], a node predicts its future position according to its current position, speed and direction (which are given by a GPS) using the following equations ((1) and (2)):

$$x(t_{0+a}) = x(t_0) \pm s * (t_{0+a} - t_0) * \cos(\theta) \qquad (1)$$

$$y(t_{0+a}) = y(t_0) \pm s * (t_{0+a} - t_0) * \sin(\theta) \qquad (2)$$

where $\mathbf{t_0}$ is the current time, **α** is the time increment in seconds so that **(t_0+α)** is the next sampling time, **(x(t_0+α),y(t_0+α))** is the position of a node at **(t_0+α)**, **s** is the current speed and **θ** is the direction angle of node motion.

In [6], Mousavi et al. use a similar predictor to the one proposed by Mir et al. [7], but with an additional term **(1/alpha)** called the scaling coefficient of the estimator. Hence, the calculation of the future position in works [6] and [9] will be given by the following coordinates ((3) and (4)):

$$x(t_{0+a}) = x(t_0) \pm 1/alpha * (s * (t_{0+a} - t_0) * cos(\theta)) \qquad (3)$$

$$y(t_{0+a}) = y(t_0) \pm 1/alpha * (s * (t_{0+a} - t_0) * sin\,\theta) \qquad (4)$$

The added coefficient (1/alpha) varies according to the mobility models, speeds and sampling rates in order to have a more accurate prediction. The added coefficient is estimated using a learning automaton [8].

Although this work presents a prediction mobility method providing future positions which are close to the reality (as their results showed), it cannot be used for Hard real time transmissions since such transfers require a strict accuracy.

Other mobility prediction methods are interested in cluster based Ad hoc networks. For example, the proposed research in [10] aims to predict the cluster changes in order to use the provided information in the route maintenance. Sathyaraj et al. [10] integrate the proposed scheme into the reactive routing protocol DSR (Dynamic Source Routing) and show that it offers better packet delivery ratio.

However, the solutions provided by these studies are valid only with clustering mechanisms which can be wasteful in time and bandwidth (Control overhead).

In [11], the authors propose a novel routing protocol (MAODV: Multicast Ad-Hoc On-Demand Distance Vector), which is an enhancement for the Ad hoc On-demand Vector routing protocol (AODV: Ad-Hoc On-Demand Distance Vector). In fact, in MAODV, a mobility prediction algorithm is added to AODV in order to control the detected routes and predict the neighbor node's mobility, so that it can deal with the network topology changes. The prediction is based on estimating the neighboring nodes distance each period of time according to the transmission strength. Thus, if the neighboring node is moving away, a new route should be established before the breakage of the current path.

The main idea of the proposed solution by Meng et al. [11] is the estimation of the distance between the current node and its neighbor. Since this estimation is only based on the transmission power, it cannot provide an accurate value of the distance.

Most of these researches are based on the GPS which provides information about the localization of the nodes. In fact, Mehdi [3] uses the GPS to predict the topology changes and predict thereafter the disconnections between the adjacent nodes, while in [6][9][10], the GPS is used to predict the future position of the mobile node.

Furthermore, the mobility solution presented by Chegin and Fathy [5] uses the GPS to build a prediction table of the links which will be used by the proactive routing algorithms. However, these methods require an additional equipment to provide the basic information which is the GPS, which is not always obvious.

Other researches are based on probabilistic methods as a research presented by Huang and Bai [4] which proposes an estimation approach of the links availability. This approach is defined as a probability that the link remains available during a period of time t. Also, the heuristic algorithm presented by Wang [1] provides two models: A probabilistic model and a non-probabilistic model which aims only to minimize the risk of uncertainty about the links. Some other mobility solutions are based on inaccurate information. In fact, in [11], the mobility prediction algorithm which was added to the routing protocol AODV is based on the transmission strength. This latter is used to estimate the distance between two neighboring nodes which cannot provide an accurate value.

Other mobility proposals are dedicated to the cluster-based Ad hoc networks like in [10] where the authors are interested in predicting the clusters change in order to maintain the constructed paths between the nodes.

In our mobility approach, we developed a deterministic delay prediction method for mobile Ad hoc networks. This method is based on accurate and certain information and it does not require additional equipment like the GPS.

III. OUR DELAY PREDICTION APPROACH

In order to check if a specific deadline for a hard real-time traffic can be satisfied, we should be sure that each used path between the source and the destination provides an end-to-end transmission delay which is lower than the deadline (We assume that there is always an available path between the source and the destination). For that, our approach allows to predict all the possible paths between two given nodes, so that we can calculate the worst end-to-end delay for each path and check if the deadline on the transmission delay will be satisfied. Furthermore, we should predict the instants in which the topology changes and consequently we should re-verify the available paths and the deadline satisfaction.

In this approach, we assume that the coordinates of the node n_i, in its trajectory at the instant $\boldsymbol{t}$ are described by the time functions $\boldsymbol{x_i(t)}$ and $\boldsymbol{y_i(t)}$. So, we note the position of the node n_i at the instant $\boldsymbol{t}$: $\boldsymbol{M_i(t)}\big(\boldsymbol{x_i(t)}, \boldsymbol{y_i(t)}\big)$. We suppose also that each mobile node n_i has a closed predefined trajectory. Actually, each node has a cyclic movement within a defined route (Like the industrial robots which move in a factory). With this hypothesis, we have two different cases. The first case is the periodic movement which occurs when the velocity is constant. The second case is the non-periodic movement which occurs when the velocity is variable. However, in the first case, in which all the nodes have periodic movements; the prediction of the paths change is easier since we can find a global period in which the same network behavior is repeated. So, we will start with the first case: periodic movements.

A. Paths prediction in the case of periodic movements

The periodic movement is the repetition of the same behavior each time period *T*. If we suppose that each node n_i has a periodic movement with its own time period T_i, we can find a time period *T* in which the same network behavior is repeated. Hence, if we verify the deadline satisfaction in the time period *T*, we will have a verification result for all time. Thus, we can predict the possible paths and calculate their transmission delays only in the instants of topology change belonging to the time period *T*.

In this section, we assume that all the nodes of the Ad hoc network have a periodic movement and that they start their movement at the same time, which is the case of mobile robots in most industrial environments. Based on this assumption, we will define the set of available paths between the source and the destination in each topology change instant belonging to the period *[0, T]*. Thus, we have to determine the value of this period *T*. For that, we use property 1.

Property 1: The same network behavior is repeated each period *T* if and only if *T* is the Smallest Common Multiple (SCM) of the time periods T_i which are the time periods of the mobile nodes n_i according to their trajectories, as shown in (5):

$$T = SCM_i\{T_i\} \tag{5}$$

where $\boldsymbol{T_i}$ is the period in which the movement of the mobile node n_i is repeated according to its trajectory.

The set of possible paths between the source and the destination (noted E_{ch}) is, therefore, defined as the set of possible paths between *0* and *T*, since the same network behavior is repeated each time period *T*.

To collect the set of available paths, we start with defining the set of neighbors of a given node n_i at the instant t (noted $NE_i(t)$). $NE_i(t)$ is the set of nodes n_j such as n_j is the neighbor of the node n_i at the instant t. Hence:

$$\boldsymbol{NE_i}(t) = \{\boldsymbol{n_j}/\boldsymbol{n_j}\ is\ the\ neighbor\ of\ \ \boldsymbol{n_i}\ at\ the\ instant\ \boldsymbol{t}\} \tag{6}$$

When one of the sets of neighbors changes, the available paths between the source and the destination may also change. In fact, a link breakage can occur on one of the paths because of the movement of the node which maintains this link. For that reason, we consider that the instants of neighboring change are the instants of paths change (E_{inst}). So, we should define at these instants, the new possible paths between the source and the destination.

 ISBN: 978-1-61208-418-3

We define E_{inst} as the set of instants of paths change between *0* and *T*. Therefore, the set of all the possible paths between the source and the destination (E_{ch}) is defined as the union of the sets of possible paths at the instants belonging to the set E_{inst}. Hence:

$$E_{ch} = \bigcup_{t \in E_{inst}} E_{ch}(t) \quad (7)$$

where $E_{ch}(t)$ is the set of possible paths at the instant of paths change *t*. Thus, $E_{ch}(t)$ is defined as the union of all the possible paths at the instant *t*.

$$E_{ch}(t) = \bigcup_k P_k(t) \quad (8)$$

where $P_k(t)$ is the k^{th} path linking the source and the destination at the instant *t*. Therefore, $P_k(t)$ is represented as the set of nodes which make a link between the source and the destination at the instant *t*:

$$P_k(t) = \{S, n_j, n_k, \ldots, n_n, D\} \quad (9)$$

To illustrate this principle, we take the example of the network represented in Figure 1. This network consists of 12 nodes numbered from 0 to 11. At the instant *t*, these nodes are arranged as described in Fig. 1. If we suppose that node 0 is the source and node 11 is the destination, the set of possible paths between the source and the destination at the instant *t* are described in Figure 1. Indeed, the possible paths are the following:

$P_0(t) = \{0, 1, 4, 8, 11\}$
$P_1(t) = \{0, 1, 3, 5, 9, 11\}$
$P_2(t) = \{0, 2, 3, 5, 9, 11\}$
$P_3(t) = \{0, 2, 6, 7, 10, 11\}$

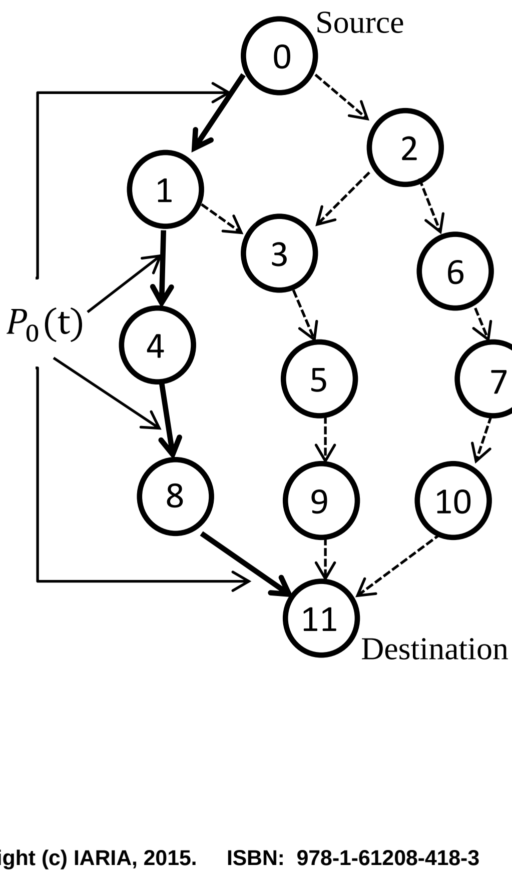


Figure 1. Set of possible paths between the source and the destination

So, the set of all the available paths at the instant t ($E_{ch}(t)$) is equal to:

$$E_{ch}(t) = \bigcup_k P_k(t) = P_0(t) \cup P_1(t) \cup P_2(t) \cup P_3(t)$$

Thus, we propose an algorithm which allows us to find out the set of possible paths between the source and the destination at any instant t belonging to the set of instants of topology change E_{inst}.

1) Presentation of the proposed algorithm

The goal of the proposed algorithm, which we call *ConsPaths*, is to collect and save all the possible paths between any two nodes from the network. For that, the algorithm starts from the source and browses the links between neighbors until reaching the destination. Once arriving to the destination, each traversed path is added to the set of possible paths. To do so, we associate four parameters to the algorithm *ConsPaths*. The first parameter is the global variable $\boldsymbol{G_{SD}}$ which will contain all the available paths. The parameters $\boldsymbol{n_i}$ and ***D*** will represent respectively the source and the destination. The fourth parameter ***P(t)*** is a temporary variable containing the current path which we are discovering. The variable ***P(t)*** is reset each time we start discovering a new path (see algorithm).

The principle of the algorithm *ConsPaths* is to record all the possible paths between the source and the destination in the global variable $\boldsymbol{G_{SD}}$. For that, we use the temporary variable ***P(t),*** which allows to keep each discovered path so that we can add it to the global variable $\boldsymbol{G_{SD}}$. So, ***P(t)*** contains temporarily the path which we are discovering. Indeed, it is initiated with the node source. Then, each node n_i having a direct link (neighbor) with the source node is added to a separate initiated variable ***P(t)*** (We will have an independent temporary variable ***P(t)*** for each neighbor of the source node). After that, each node n_j having a link with the node n_i is added to a separate *P(t)* and so forth until reaching the destination or reaching a node which doesn't have neighbors. In the first case, when we reach the destination, we add the recorded path *P(t)* to the set $\boldsymbol{G_{SD}}$. In the second case, when we reach a node which hasn't neighbors, we discard the current content of *P(t)*, since it does not have a path leading to the destination.

Variables of the algorithm ConsPaths :

$\boldsymbol{G_{SD}}$**:** Global variable which contains the set of all the possible paths.
S : Source node.
D : Destination node.
$\boldsymbol{n_i}$***:*** A node **i** belonging to the network.
P(t) : Temporary variable containing the path which we are discovering.
CP(t) : Intermediate variable containing the current content of ***P(t)***.

```
Algorithm ConsPaths (var G_SD, n_i, D, P(t))
BEGIN
If (n_i = S) Then
{new (P(t))
 P(t) ← {S}}
   Else If NE_i(t) = ∅  Then return (∅)
        Else If (n_i = D) Then {
        G_SD ← P(t) ∪ G_SD
         return ()}
      Else
         CP(t) ← P(t)
         For each node n_j ∈ NE_i(t)
              and n_j ∉ CP(t) do {
           new (P(t))
           P(t) ← CP(t) + {n_j}
           ConsPaths (G_SD, n_j, D, P(t)) }
         EndFor
      EndIf
  EndIf
EndIf
END
```

Figure 2. Algorithm of paths construction in the case of periodic movements

This algorithm is executed at each instant of topology change belonging to the time interval *[0, T]*. So, we should determine these instants.

2) The instants of topology change:

To identify the instants of topology change in the time period between 0 and T, we will use property 2 which allows checking if the neighborhood (and consequently the topology) has been changed.

<u>*Property 2*</u>: We identify t_1 as the instant of neighborhood change, if the neighborhood of at least one node n_i from the network changes compared with the last instant of neighborhood change t_2. Thus, if t_2 is an instant of neighborhood change, then the next instant of neighborhood change will be t_1 if and only if:

$$For\ t_1 > t_2, \exists i / NE_i(t_1) \neq NE_i(t_2) \quad (10)$$

where $NE_i(t)$ is the set of the neighbors of the node n_i at the instant *t.*

Based on property 2, we can check if a given instant is an instant of topology change. So, it remains to determine the instants we should verify if there are instants of topology (neighborhood) change.

3) The instants of verification of neighborhood change:

The neighborhood of a node may change in three cases. The first case: If the node moves and its position changes. The second case: If one of its neighbors moves and comes out from its range. The third case: if a new neighbor comes into its range. All those cases depend on nodes movement. Thus, the instants of verification may be identified according to the nodes movements which are characterized by the speed and the running distance.

Since we are interested in the worst case, we will consider the minimum running distance which can change the neighborhood of a given node regarding the predefined trajectories ($\mathrm{Distance_{min}}$) and the maximum speed on the network ($\mathrm{Speed_{max}}$). Therefore, the minimum time after which the network topology may change ($\mathrm{t_{min}}$) is the quotient of the division of the minimum running distance by the maximum speed ($\mathrm{Speed_{max}}$) (as in (11)).

$$t_{min} = \frac{Distance_{min}}{Speed_{max}} \quad (11)$$

Therefore, if $\mathrm{t_{i-1}}$ is the precedent instant of verification, the next instant of verification $\mathrm{t_i}$ will be (as in (12)):

$$\mathrm{t_i} = \mathrm{t_{i-1}} + \mathrm{t_{min}} \quad (12)$$

B. Paths prediction in the case of non-periodic movements

In the case of non-periodic movements, we cannot find a time period in which the same network behavior is repeated. Thus, we cannot predict all the instants of topology change. In this case, we can only define the set of possible paths between the source and the destination regardless of the time. For that, we use the property of the possible neighbors (property 3).

<u>*Property 3*</u>: We define $M_i(t)$ as the position of the node n_i at the instant *t*, NP_i as the set of possible neighbors for the node n_i and *R* the transmission range of the nodes. A node n_j can belong to the set of possible neighbors of the node n_i (NP_i) if and only if there are two instants t_1 and t_2 such as the distance between the position of the node n_i at the instant t_1 ($M_i(t_1)$) and the position of the node n_j at the instant t_2 ($M_j(t_2)$) is inferior to the range of the nodes (*R*) (as in (13)):

$$n_j \epsilon\ NP_i \text{ if and only if } \exists\ t_1, t_2 / \left| M_i(t_1) - M_j(t_2) \right| < R \quad (13)$$

Based on property 3, we can find out the possible paths between a source and a destination. Thus, the algorithm which allows getting the possible paths in the case of non-periodic movements ($ConsPathsNP$) has the same principle as that of the periodic movements except that it does not depend on the time. It allows only getting the set of possible paths according to the possible neighbors from the source to the destination.

<u>*Variables of the algorithm ConsPathsNP :*</u>

$\boldsymbol{G_{SD}}$**:** Global variable which contains the set of all the possible paths.

***S*:** Source node.

***D*:** Destination node.

$\boldsymbol{n_i}$**:** A node **i** belonging to the network.

P: Temporary variable containing the path which we are discovering.

CP: Intermediate variable containing the current content of ***P***.

```
Algorithm ConsPathsNP (var G_SD, n_i, D, P)
BEGIN
If (n_i = S) Then
{new (P)
 P← {S}}
   Else If NP_i = ∅  Then return (∅)
        Else If (n_i=D) Then {
        G_SD←P ∪G_SD
        return ()}
      Else
          CP← P
          For each node n_j ∈ NP_i
               and n_j ∉ CP do {
            new (P)
            P ← CP+{n_j}
            ConsPathsNP (G_SD, n_j, D, P)}
          EndFor
     EndIf
 EndIf
EndIf
END
```

Figure 3. Algorithm of paths construction in the case of non-periodic movements

After finding out the possible paths between the source and the destination, we should analyze the response times of these paths so that we can get the response time (end-to-end transmission delay) in the worst case.

IV. RESPONSE TIME ANALYSIS

If we assume that the longest path has the largest response time, the transmission delay of the longest possible path between two nodes will be the maximum delay and consequently the transmission delay of the worst case [12].

In the case of periodic movements, the set of possible paths is the union of the sets of possible paths at the instants of topology change. So, if we have the maximum transmission delay of each instant of topology change, the worst case delay between the source and the destination will be the maximum of these delays. Thus, $\mathrm{TR}_{\mathrm{E_{ch}}}$, which is the response time between two nodes of the network in the worst case, is given by the following formula:

$$TR_{E_{ch}} = \max_{t\epsilon[0,T]} TR_{E_{ch}(t)} \qquad (14)$$

where $TR_{E_{ch}(t)}$ is the response time in the worst case at the instant of topology change *t*. Hence, its value is the maximum of the response times of the discovered paths at the instant *t*. $TR_{E_{ch}(t)}$ is given by the following formula:

$$TR_{E_{ch}(t)} = \max_{k} TR_{P_k(t)} \qquad (15)$$

where $TR_{P_k(t)}$ is the response time of the path $P_k(t)$ discovered at the instant of topology change *t*.

The response time $TR_{P_k(t)}$ for the path $P_k(t)$ is the sum of the elementary transfer times between two each neighboring nodes which belong to the path, as shown in the following formula:

$$TR_{P_k(t)} = \sum_{(N_i,N_{i+1})\epsilon P_k(t)} ETT\,(N_i, N_{i+1}) \qquad (16)$$

where $ETT\,(N_i, N_{i+1})$ is the elementary transfer time between the neighboring nodes N_i and N_{i+1}. It depends on the number of neighbors (According to Opt/TDMA [13]).

In the case of non-periodic movements, the response time between a source and a destination in the worst case is the maximum of the response times of all the possible paths (as in (17)) :

$$TR_{E_{ch}} = \max_{\mathrm{k}} \mathrm{TR}_{\mathrm{P_k}} \qquad (17)$$

The response time TR_{P_k} of the path P_k is the sum of the elementary transfer times between two neighboring nodes which belong to the path as in (16). However, it doesn't depend on time.

Thus, we have determined the response time in the worst case between two given nodes from the Ad hoc network. In the following section, we will evaluate the results of this analysis by comparing them to the simulation results.

V. EVALUATION

To evaluate our delay prediction, we have performed simulations on NS2 simulator [14] for an Ad hoc network having mobile nodes with closed trajectories. Then, we compared the end-to-end delays obtained from simulations with the worst case delays obtained from our analysis approach (see previous section).

A. Simulation scenarios

The simulated Ad hoc network is composed of twenty nodes randomly positioned in a 1000×1000 area and moving with cyclic movements. In this network, we use Time Division Multiple Access (TDMA) [15] as access method and RT-DSR [2] as routing protocol.

To perform the simulations, we created three traffics which are transmitting in the network and we are interested to the worst case end-to-end delay of the first real-time traffic.

B. Variation of the transmission delay over time

First, we observed the variation of the end-to-end transmission delay when the mobile nodes of the simulation are moving with a radius which is equal to 100 meters. Then, we calculated the worst case end-to-end delay with our analytic approach and we compared the obtained value with the simulation delays. The results are represented in Figure 4.

 ISBN: 978-1-61208-418-3

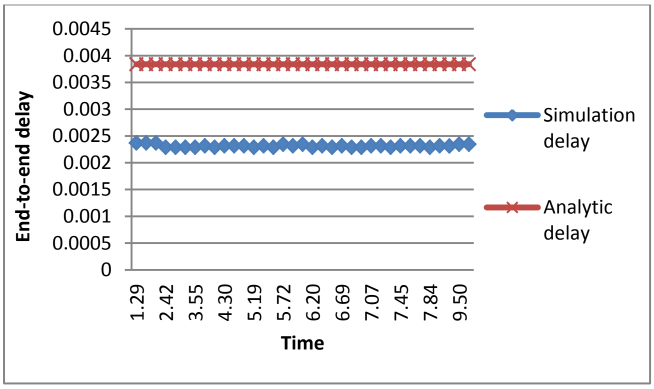


Figure 4. Variation of the transmission delays over time

We note, according to Figure 4, that all the simulation delays are inferior to the delays calculated by our analytic approach. This is due to the fact that our analytic approach gives us the end-to-end delay in the worst case which is a specific case and it may not occur in the simulations. So, we conclude that our analytic approach allows getting an upper bound of end-to-end transmission delays. Thus, if a delay deadline is superior or equal to this boundary, we can be sure that it will be respected.

C. Variation of the worst case transmission delays according to the moving radius

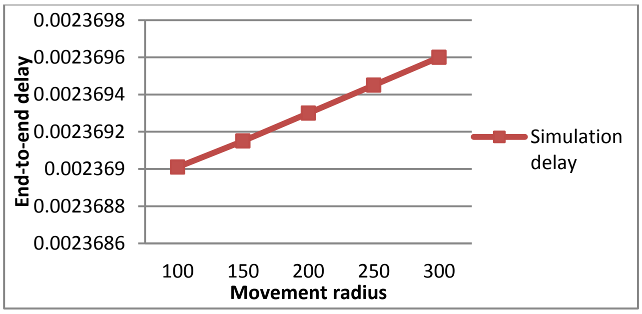
Then, we were interested in the variation of the end-to-end delays according to the moving radius of the mobile nodes. For that, we present in Figure 5 the maximum transmission delays obtained with the simulator NS2 (a) and the comparison between those simulation delays and the transmission delays calculated with our analytic approach (b).

Figure 5 shows that the worst case end-to-end delay increases when the moving radius of the mobile nodes increases in the simulation results as well as in the analytic results. However, the slope of the results calculated by our analytic approach is higher.

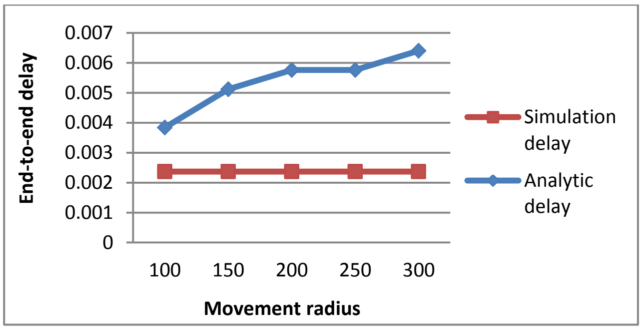
These results show also that the worst case analytic delays are always superior to the simulation delays with a small difference which proves the effectiveness of our approach in estimating the transmission delays.

VI. CONCLUSION

In this paper, we proposed a delay prediction approach for Ad hoc networks. In fact, we presented an algorithm which allows getting the worst case end-to-end transmission delay. By comparing the delay values calculated by our approach with the simulation results obtained from NS2 simulator for the same network, we conclude that the simulation delays are always inferior to the calculated worst case delays with a small difference. This proves the effectiveness of our approach which allows deciding if we can satisfy a required deadline for real time traffic.



-a- Variation of the maximum transmission delay in the simulations according to the moving radius of the nodes



-b- Comparison between the simulation transmission delays and the analytic transmission delays according to the moving radius of the nodes

Figure 5. Variation of the worst case transmission delay according to the moving radius of the nodes

REFERENCES

[1] J. Wang, "Exploiting Mobility Prediction for Dependable Service Composition in Wireless Mobile Ad Hoc Networks", IEEE Transactions on services computing, vol. 4, no. 1, January-March 2011.

[2] S. Ouni, J. Bokri, and F. Kamoun, "DSR based Routing Algorithm with Delay Guarantee for Ad Hoc Networks", Journal of Networks, Academy Publisher, vol. 4, iss. 5, July 2009, pp. 359-369.

[3] H. Mehdi, "Mobility Prediction With LLT Algorithm In Wireless Networks", International Conference on Information, Networking and Automation (ICINA), 2010.

[4] J. Huang and Y. Bai, "A Novel Approach of Link Availability Estimation for Mobile Ad Hoc Networks", Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE, May 11-14, 2008.

[5] M. Chegin and M. Fathy, "Optimized Routing Based on Mobility Prediction in Wireless Mobile Adhoc Networks for Urban Area", Fifth International Conference on Information Technology: New Generations , Las Vegas, Nevada, April 2008.

[6] S. M. Mousavi, H. R. Rabiee, M. Moshref, and A. Dabirmoghaddam, "Model Based Adaptive Mobility Prediction in Mobile Ad-Hoc Networks", Wireless Communications, Networking and Mobile Computing, 2007 (WiCom 2007), September 2007, pp. 1713 – 1716.

[7] Z. H. Mir, D. M. Shrestha, G. Cho, and Y. Ko, "Mobility Aware Distributed Topology Control for Mobile Multi-hop Wireless Networks", ICOINS 2006, LNCS 3961, 2006, pp. 257-266.

[8] A.S. Poznyak and K. Najim, "Learning Automata and Stochastic optimization", New York: Springer, 1997.

[9] S. M. Ousavi, H. R. Rabiee, M. Moshref, and A. Dabirmoghaddam , "Mobility Aware Distributed Topology Control in Mobile Ad-Hoc Networks with Model Based Adaptive Mobility Prediction", Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007, October 2007, pp. 86-86.

[10] B. H. Sathyaraj and R. C. Doss, “Route Maintenance using Mobility Prediction for Mobile Ad hoc Networks”, Mobile Adhoc and Sensor Systems Conference, November 2005.

[11] L. M. Meng, J. X. Zang, W. H. Fu, and Z. J. Xu, “A Novel Ad hoc Routing Protocol Research Based on Mobility Prediction Algorithm”, Wireless Communications, Networking and Mobile Computing, 2005, 23-26 September 2005.

[12] J. Bokri, S. Ouni, and F. Kamoun, “The Worst Path Estimation for Real Time Communications over Ad hoc Networks”, The International Conference on Wireless Networks (ICWN'12), Las Vegas, Nevada, 16-19 July 2012.

[13] S. Ouni, J. Bokri, and F. Kamoun, “Opt-TDMA/DCR: Optimized TDMA Deterministic Collision Resolution Approach for Hard Real-Time Mobile Ad hoc Networks”, WSEAS Transactions on Communications 11/2013, 2013, pp. 12(11):570-583.

[14] Z. Wu, “Network simulator 2 for wireless : My experience”, Technical report, Rutgers University, 2007.

[15] I. Jawhar, and J. Wu, “Qos support in Tdma-based mobile ad hoc networks”, Journal of Computer Science and Technology, pp. 20(6):797–810, 2005.