Partition Detection in Mobile Ad-hoc Networks

Michaël Hauspie, David Simplot, Jean Carle

▶ To cite this version:

HAL Id: hal-00616862

https://hal.archives-ouvertes.fr/hal-00616862

PARTITION DETECTION IN MOBILE AD-HOC NETWORKS*

Michaël Hauspie, David Simplot and Jean Carle

IRCICA / LIFL - Univ. Lille 1
Bât. M3 - Cité Scientifique
59655 Villeneuve d'Ascq Cedex, France
email: {hauspie, simplot, carle}@lifl.fr

ABSTRACT

A classical problem caused by nodes movement in an ad hoc network is partitioning. Predicting those partitions could be a very useful feature that can be provided to applications in a mobile ad-hoc network environment. Indeed, being aware of a future disconnection in the network can help to ensure a better quality of service by adapting the application behavior. Algorithms already exists to do this but they need position information to be provided by a positioning system. This paper propose an original link robustness evaluation method based on the notion of disjoint paths which allow efficient partition detection without using any kind of positioning system.

1 INTRODUCTION

Wireless networks such as Bluetooth [3] or WiFi (Wireless Ethernet IEEE 802.11b) [2] can grant users data access regardless of their location without wired connection.

Nowadays, those networks are mostly used by directly communicating with a base station linked to a wired network and internet. Another application of such technologies are networks based neither on a base station nor any kind of fixed infrastructure. Those networks are useful when no wired link is available such as in disaster recovery or more generally when a fast deployment is necessary.

In those applications, mobile computers, or nodes, will communicate by routing messages through the network by multi-hopping protocols [7, 9, 11]. These networks are called MANET for Mobile Ad-hoc NETworks [1].

A possible problem in those kind of networks is partitioning. If connection breaks are real failure in a wired environment, in ad-hoc networks, we can not consider it like this because it can occur after a normal network behavior (a node has moved, a user turned his device off...). In the case of node movement (Figure 1), it can be useful to predict partitioning and notify applications without using a positioning system which is often expensive and bulky [4].
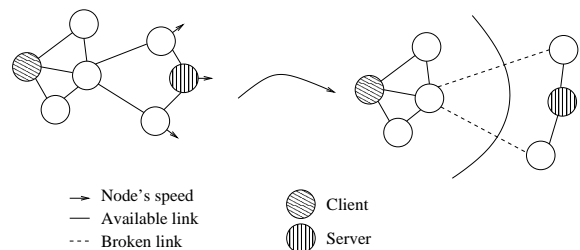

Figure 1: Network partition caused by topology changes.

In this paper, we propose some ideas based on disjoint paths set and we try to show by simulation that these information can be used to have an efficient partition detection.

The paper is organized as follows. First, we will describe the existing work on partition detection. Then, we will propose two metrics and finally we will evaluate them by simulation.

2 RELATED WORKS

In [8], Shah et al. proposed a mechanism for enhancing data access across an ad-hoc network. This mechanism is based on a data replication service which is able to copy some data from one node to several others if a node detects that he will not be able to reach the host of the data because of topology changes. The authors detect the partition by using a global positioning system. Each node collects the position and velocity from its connected group of node and uses those information to predict when two connected groups will be out of range from each other.

Wang et al. used a similar approach in [12] but in a more centralized way. They regroup nodes into "Mobility groups" by using a centralized sequential clustering algorithm running on a server. Each node sends its velocity (computed from the position given by a positioning system) to the server. Knowing the nodes' velocity, the server can predict when the groups will be disconnected and can inform involved nodes.

Although those algorithms are giving good results, their inputs are depending on expensive and bulky hardware (GPS) or/and on a node that can be reached by the others. As we want our method to work on every kind of mobile node equipped with wireless device, we want to focus our work on a totally distributed network that does not use any positioning system. Indeed, it then can be applied on every ad-hoc network regardless of the hardware or software provided by the nodes.

Some work about QoS multi path routing tends to show that a set of disjoint path between two nodes of the network can be useful and computed by a distributed algorithm [6]. We think that disjoint path can also lead to efficient network partition detection.

3 LINK ROBUSTNESS EVALUATION

In this section, we are aiming at compute the available link robustness between two nodes of a network. We want to evaluate a link wich is not a simple path. We call link between two nodes the set of paths allowing them to communicate. This link robustness can be seen as its capacity to maintain communication between the two nodes. We will say that a link is strong if the physical disconnection risk is weak.

3.1 Preliminaries

Our evaluation methods are based on neighborhood and paths notions that are to be defined. We use $G(V, E)$ as a graph representing the ad hoc network where V represents the set of wireless mobile hosts and E represents the set of edges. Let v and w be two nodes. A path p between v and w is a series of nodes $o_1, o_2, \ldots, o_n$ such as $o_1 = v, o_n = w$ and for all $i \in [\![1, n[\![, o_{i+1} \in N(o_i)$. Let $\tilde{p} = \bigcup_{i=1}^{n} \{o_i\}$, we denote by $|p|$ the number of hops in the path ($|p| = n - 1$). The set of all paths between v and w is denoted by $P(v, w)$.

It is straightforward that each path is not interesting for communications. For instance, extra-long paths or paths with loops are not interesting. Moreover, it is well known that optimal paths in terms of numbers of hops are few, weak and sensible to topological modifications [10]. That is why in this paper, for two given nodes v and w, we will consider a subset of $P(v, w)$ which will be called *sub-optimal loop-free paths*.

Let v and w be two nodes, and $p \in P(v, w)$. p will be called an **optimal** path if and only if $\forall p' \in P(v, w)$ $|p'| \geq |p|$. If p is optimal, $|p|$ is called **distance** between v and w and is denoted by $d(v, w)$. Notice that an optimal path is also a loop-free path[1]. At last, p is called **k-sub-optimal** ($k \geq 0$) if and only if p is loop-free and $|p| < d(v, w) + k$. k represents the number of hops added to an optimal path. We denote by $SOP_k(v, w)$ the set of k-sub-optimal paths between v and w.

Defining SOP_k is motivated by the need of loop-free paths between v and w. Indeed, since optimal paths are few and weak, we need to take account of some other paths near the optimality in number of hops. The set of *Loop-Free-Paths* (denoted by $LFP(v, w)$) can then be determined by

$$LFP(v, w) = \lim_{k \to \infty} SOP_k(v, w) \qquad (1)$$

For a network of size $|V|$, the longest loop-free path is at most of size $|V| - 1$. So, we have $LFP(v, w) = SOP_{|V|-1}(v, w)$. We think that there exists a maximal value of k above which SOP_k does not change significantly in our evaluation. Decreasing k will help us to efficiently compute the metric. Indeed, a fewer k would reduce the network congestion produced by a distributed algorithm computing a set of disjoint paths. Moreover, it is not desirable to store a lot of paths on each node of an ad hoc network.

3.2 Link robustness

Our evaluation method is based on the following idea. The more paths there is between two nodes, the stronger is the path allowing them to communicate. Indeed, if a path breaks, the nodes can go on communicating if and

---

[1] A path $p \in P(v, w)$ is called *loop-free* if and only if $\forall i, j \in [\![1, n]\!]$, $o_i = o_j \Rightarrow i = j$.

only if it stills being a valid path between them. Figure 2 shows two possible cases.
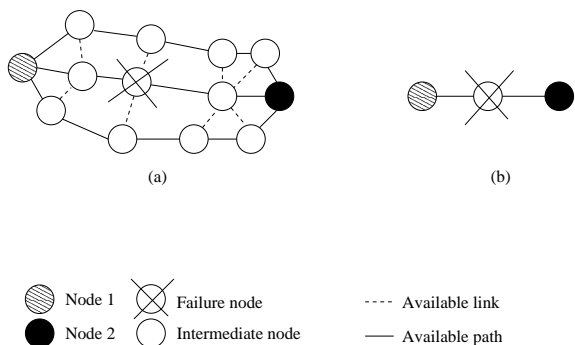

Figure 2: Topology examples.

In, Figure 2.a, the nodes have got three paths. If one of them breaks, the nodes can go on communicating. On the other hand, Figure 2.b, it exists only one path connecting node 1 and node 2. If it breaks, the communication will be physically interrupted. In Figure 2 we only consider disjoint paths that seems to be a good robustness criteria as evoked by Haas [6]. Actually, even if it exists hundreds of paths between two nodes, if all of them use the same intermediate node, a simple failure of this node would invalidate all those paths.

For two nodes v and w and a given constant k, the set of parts of $SOP_k(u, v)$ ($2^{SOP_k(u,v)}$) containing only disjoint path containing only disjointed paths is denoted by $DSP_k(v, w)$ (*Disjoint Sub-optimal Paths*) and is defined by :

$$DSP_k(v, w) = \Big\{ S \in 2^{SOP_k(v,w)} \mid$$
$$\forall p, p' \in S \quad \tilde{p} \cap \tilde{p}' \neq \{v, w\} \Rightarrow p = p' \Big\}. \quad (2)$$

Thus, if a node disappear, one and only one paths will break. By this fact, as it stills more than one available disjoint path, one can break without physically breaking the connection. Using this notion of disjoint path, we will propose two link evaluation methods.

We consider k as a fixed system parameter for the definition of our metrics so that the notations remain light.

The first one is only based on the number of disjoint paths available. Indeed, if we have only one paths whereas there were more before, it is strongly possible that this one disappears too, resulting to a physical disconnection. This first metric is denoted LR_1 (for *Link Robustness*) and is defined by :

$$LR_1(u, v) = \max_{A \in DSP_k(u,v)} \{|A|\} . \quad (3)$$

For the second one, we aimed at refining the measure given by LR_1. We can actually notice that the longer is a path, the weaker it is. Thus, all paths do not evenly contribute to the link robustness. Then, We evaluate the probability that at least one path composing the link remains available. The set of disjoint paths that gives the best probability that a path survives gives us our second metric denoted LR_2 :

$$LR_k(v, w) = \max_{A \in DSP_k(u,v)} \left\{ 1 - \prod_{p \in DSP_k(v,w)} P_b(p) \right\}, \quad (4)$$

where $P_b(p)$ is the probability that p breaks. The rough calculation of $P_b(p)$ is given by the probability that each direct connection composing the path breaks. If $p = (o_1, o_2, \ldots, o_n)$, we define :

$$P_b(p) = 1 - \prod_{1 \leq i < n} \mu(o_i, o_{i+1}), \quad (5)$$

where $\mu(x, y)$ represents the probability that the direct connection between the nodes x and y remains available. This probability has to be evaluated too. For now, we will consider that μ is a constant.

4 EXPERIMENTS

The aim of this paper is to propose and evaluate the metrics. So, we compute our robustness using the global knowledge of the network and generating disjoint paths using a depth first search algorithm. We are currently working on a distributed algorithm which can be embedded in a mobile node in order to compute disjoint paths in real conditions.

To evaluate the metrics, we did the following experiments :

1. a random graph is generated in a rectangular area,

2. we randomly choose a server node and one of its neighbors as the client node[2],

3. nodes move according to the random way-point model [5]. Each node choose a random destination and goes for it. When he reaches it, he waits for a random time, choose another destination and so on,

---

[2]We this, we are sure that the nodes are connected at the start of the simulation

4. periodically, we compute the link robustness given by the two methods,

5. when a partition occurs, we go back to step 2 until we reached 300 partitions.

We did our experiments at densities of 4, 6 and 8 nodes by communication area. With higher densities, the network is almost always connected so the physical disconnection is not an important problem. With smaller density, the opposite problem occurs, the network is too unstable to predict anything. We used speeds of 1, 1.5 and 2 meters per second which corresponds to different walking speeds. Nodes communication range are set to 10 meters.

Figure 3 and 4 show the evolution of the value computed by both methods. The density used was 8 nodes by communication area. Nodes' speed was 2 meters per second which is a fast walk speed leading to sudden disconnections.
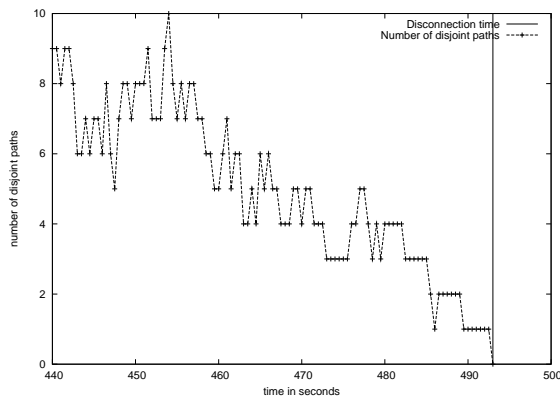
Figure 3: Evolution of the link robustness value for the first metric

We can observe that robustness values computed by both methods falls just before the physical disconnection. What is observed at this density and this speed is almost the same with other speeds and densities.

To detect if a partition occur, we introduce the notion of threshold. If the metric falls under a given value (called threshold) during a given amount of time (about 1 second in our experiments), we raise a "warning" flag. The relevance of this warning flag will depend on the efficiency of the metric (*i.e.* the number of disconnections it predicted) and on the average time between the prediction and the effective disconnection. If this time is too important, the application would always need to change its behavior according to the warning flag. If this time is too short, the application would not have enough time
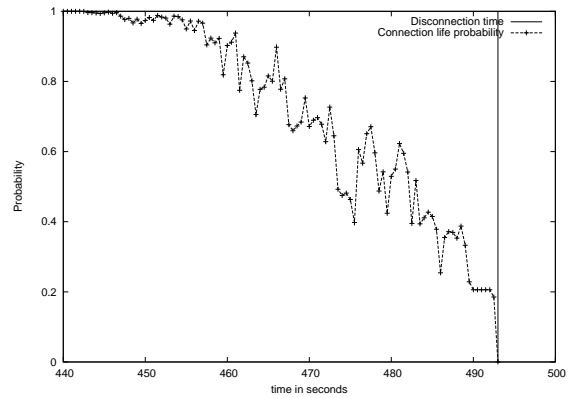
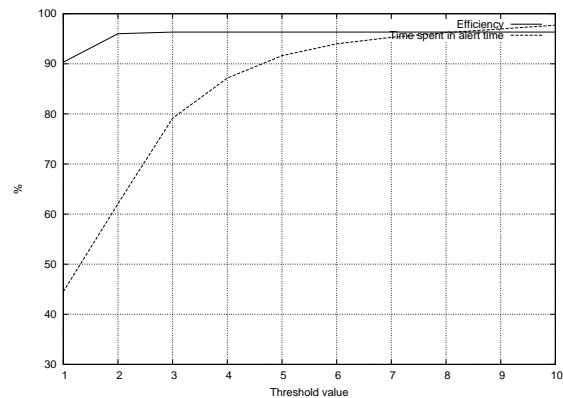Figure 4: Evolution of the link robustness value for the second metric

Figure 5: Evaluation of the first metric at 0.5 m/s with density 6

to changes its behavior and the potential QoS criterion would not be respected.

So, to achieve evaluating the metrics, we computed their efficiency and the time spent in alert for a range of thresholds. Figures 5 to 10 show the results for the two metrics. Figures 5 to 8 show how the speed influences on those metrics whereas Figures 9 and 10 show the density's one.

General observations For both metrics, we can say about speed influence that the faster the nodes move, the worse is the prediction. Indeed, if nodes move faster, the topology is less stable so it is harder to have a good prediction. Concerning the influence of the density, as it grows, the stability of the link grows because the number of disjoint paths is potentially higher. This can be observed in Figures 9 and 10.
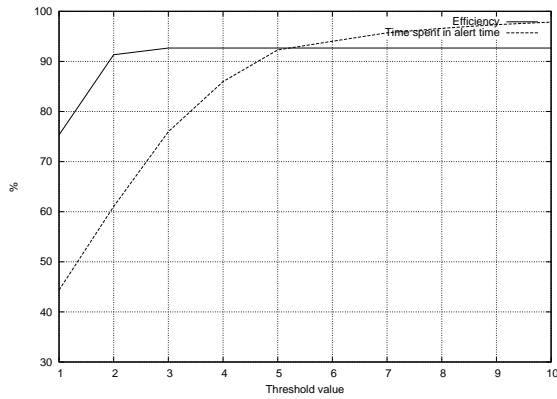
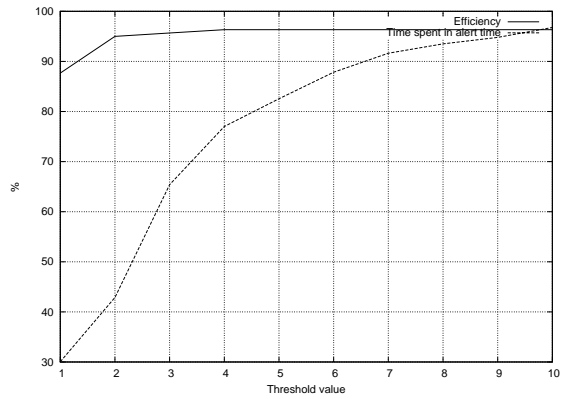Figure 6: Evaluation of the first metric at 2.0 m/s with density 6

Figure 9: Evaluation of the first metric at 2.0 m/s with density 8
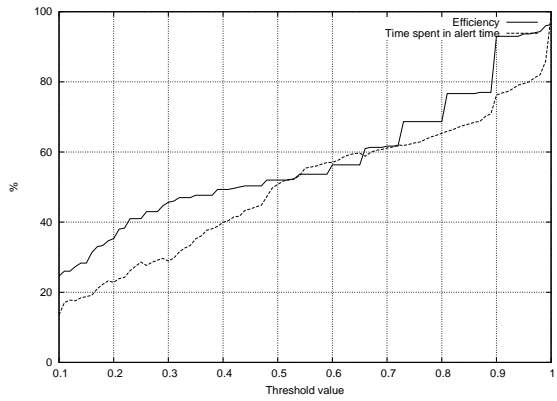

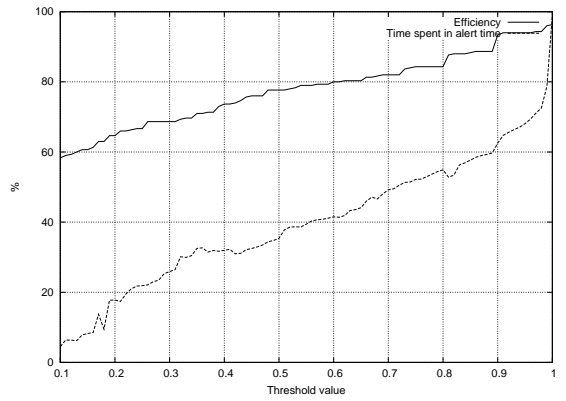Figure 7: Evaluation of the second metric at 0.5 m/s with density 6

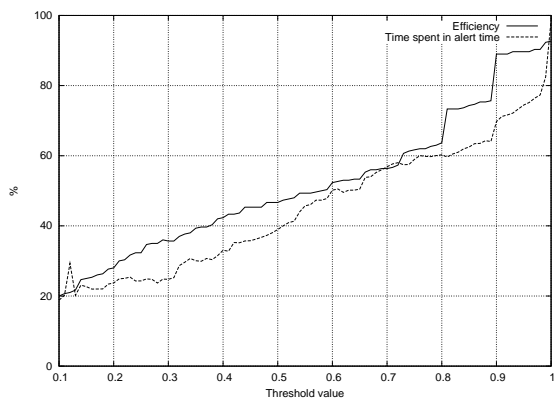Figure 10: Evaluation of the second metric at 2.0 m/s with density 8

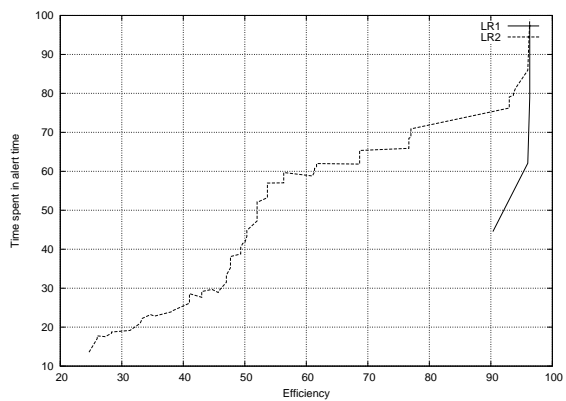Figure 8: Evaluation of the second metric at 2.0 m/s with density 6

Figure 11: Time spent in alert according to the efficiency (varying threshold)

Metrics comparisons Bypassing the general observations, we can say that the first metric, that only takes care about disjoint paths, produces far better results. Actually, the gap between the efficiency and the alert time is far more important in the first metric. If we want 90 % of efficiency, in the first metric we can take a threshold of 1 and we will only spent 30 to 50 % of connection time in alert depending on the density and on the speed of the nodes. To reach the same efficiency with the second metric, we would need to spend more than 80 % of the total connection time in alert which is not acceptable and tends to always raise an alert flag. Figure 11 shows in an even better way that LR_1 has a better efficiency/alert time ratio than LR_2 using a density of 6 nodes by communication area and 0.5 m/s speed. This can be explained by the fact that this metric is not as stable as the first one and so, topology modification could cause great fluctuation of its value.

5 CONCLUSION

In this paper, we propose two metrics for link evaluation in a mobile ad-hoc network environment. Using the evaluation of our first metric, we can predict a physical disconnection between the nodes if this connection is due to a network topology change. Our experiments shows that this kind of prediction is possible without using expensive devices like positioning systems if we have a distributed algorithm giving us a set of disjoint path between two nodes. So, it is now relevant to work on this kind of algorithm. Moreover, a set of disjoint path can be used on QoS multi path routing protocols.

REFERENCES

[1] MANET (Mobile Ad-hoc NETwork) group of IETF (Internet Enginering Task Force).
URL: http://tonnant.itd.nrl.navy.mil/manet/.

[2] IEEE standards boards part 11, wireless LAN medium access control and physical layer specifications, 1999.

[3] Specifications of the bluetooth system v1.0 b, December 1999.

[4] Benlarbi-Delaï, A., D. Simplot, J. Cartigny, and J.-C. Cousin. Using 3D indoor microwave phase sensitive stereoscopic location system to reduce energy consumption in wireless ad-hoc networks. In *Smart Objects Conference (sOc'2003)*. Grenoble, France, 2003.

[5] Johnson, D. B. and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[6] Papadimitratos, P., Z. Haas, and E. Sirer. Path set selection in mobile ad hoc networks. In *MobiHoc 2002*. June 2002.

[7] Royer, E. M. and C.-K. Toh. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications Magazine*, pages 46–55, April 1999.

[8] Shah, S., K. Chen, and K. Nahrstedt. Cross-layer design for data accessibility in mobile ad hoc networks. In *Proc. of 5th World multiconference on systemics, cybernetics and informatics (SCI 2001), Orlando, Florida*. July 2001.

[9] Stojmenovic, I., editor. *Handbook of Wireless Networks and Mobile Computing*. John Wiley & Sons, 2002.

[10] Toh, C.-K. Associativity based routing for ad hoc mobile networks. *Wireless Personal Communications Journal, Special Issue on Mobile Networking and Computing Systems*, 4(2):103–139, March 1997.

[11] Toh, C.-K. *Ad Hoc Mobile Wireless Networks, Protocols and Systems*. Prentice Hall, 2002.

[12] Wang, K. H. and B. Li. Group mobility and partition prediction in wireless ad-hoc networks. In *Proceedings of IEEE International Conference on Communications (ICC 2002)*, volume 2, pages 1017–1021. New York City, New York, April 2002.