Replication decision algorithm based on link evaluation for services in MANET*

Michaël Hauspie† David Simplot Jean Carle

May 2002

Abstract

When using some services across a wireless network, a node should expect to use it in good conditions. The first criterion that comes in mind in wireless environment is network partitioning (*i.e.* when the client can no more reach the host of the service it is using). This criterion, or QoS attribute, can be handled by predicting partitioning and using service replication which consist of electing a new host for the service and duplicating it on this new host. But wireless networks QoS implies more criterion that could also be handled by service replication. The main problem is then to detect when the replication must occur. In this paper, we propose a metric for link quality evaluation and a fast and reliable protocol to practically compute it.

Keywords
ad-hoc networks, QoS, data replication, link evaluation, broadcast

1 Introduction

Wireless networks such as Bluetooth [2] or WiFi (Wireless Ethernet IEEE 802.11b) [3] can grant users data access regardless of their location. 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 battle field communications or disaster recovery. Here, mobile computers, or nodes, will communicate by routing messages through the network by multi-hopping protocols [6, 8, 11]. These networks are called MANET for Mobile Ad-hoc NETworks [1].

As nodes can communicate together, they can share data or applications (services). The nodes that are using a service (so-called clients) must be provided with enough bandwidth or little delay regarding to the type of the service they are connected to. In [7], the authors proposed a method aiming to enhance the data access rate by duplicating the data across the network when necessary. This concept can be adapt to high density networks (*e.g.* connectivity greater than 90%) to increase the link quality between servers and the clients. One of the main problem with service replication methods is to detect when the replication must occur. In this paper, we present an efficient method to do a fast and reliable evaluation of the link quality between server and clients.

The paper is organized as follow, Section 2 introduces the original replication concept by Shah et al. [7] and explains how it can be extended to handle more QoS attributes. An analysis of the problem is given Section 3 and then we present our solution. Section 4 shows the experimental results of our methods and section 5 concludes our results and opens to new problems hidden behind service replication.

*This work was partially supported by a grant from Gemplus Research Labs, ACI *Jeunes Chercheurs* "Objets Mobiles Communicants" (205+CDR1) from the Ministry of Education and Scientifi c Research, France and the CPER Nord-Pas-de-Calais TACT LOMC C21

†Contact author `hauspie@lifl.fr`

2 The base concept

In low density networks, the main problem is network partitioning. As nodes move, some services can become unavailable since there is no more link allowing the clients to reach them. Figure 1 shows an example of the problem.

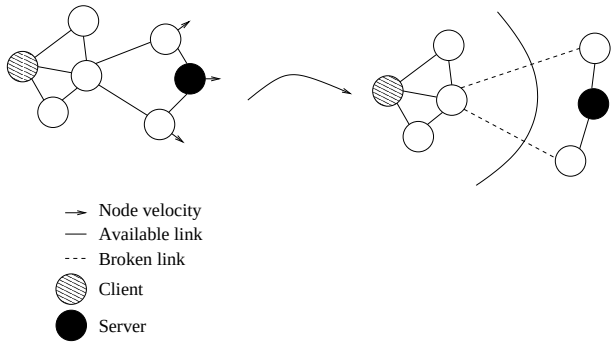

Figure 1: Network partition problem

To resolve it, we must detect when the partition occurs and *copy* the service from the server to a reachable node for the client. In [7], the authors detect partitioning using a positioning system. Each node collects position and velocity information from his connected group of nodes and then uses them to predict group partitioning. When a node predicts a partition, he sends a duplication order and a candidate is elected for hosting it in the subgroup. This method is efficient to enhance data access across the network but at a high cost. Indeed, each nodes needs a positioning system (expensive and bulky) and needs to interpolate position of the other nodes which implies heavy computation for small device. Moreover, network partitioning may not be the only trigger for a service replication.

In high density networks, the connection will be as good as the client is near the server since there will be fewer hops for routing messages between them and so, less jam caused by other nodes. Then, replicating a service on a closer host can be a good solution for enhancing the connection quality between a client and a server. These new replication features implies new things to detect before selecting a new host and replicating the service. We must find a way to detect whenever the connection should become bad by a non-expensive way in term of network occupation and equipment price. For the last thing, we don't want to use any positioning system and our solution must work in a full ad-hoc environment (*i.e.* without any fixed infrastructure). The next section of the paper describes our approach.

3 The service replication problem

3.1 Theoretical approach

As said in Section 2, we need to know when to trigger the service replication. In our case (high density networks), detecting network partitioning is not the only thing to do. Indeed, we must also detect whenever the connection becomes bad in term of reliability, bandwidth, delay... For this purpose, we based our work on two simples ideas.

Assertion 1 *If in a given route there are a lot of essential nodes (*i.e. *if an essential node disappear, our two nodes would not be able to communicate anymore), the route between nodes is weak because it can easily break.*

Assertion 2 *If two nodes are far away from each other, their communications would disturb and be disturbed by the network more than if they are near. Indeed, routing messages between them will generate more duplicated packets than if they were closed from each other even with a good routing protocol. If there are more duplicated packets, the overload generated will reduce available bandwidth and grow up the delay.*

Before going on, we need to introduce some definitions of ad hoc networks notions.

Definition 1 *Let v be a node. The set $N(v)$ represents the nodes that v can directly reach and is called the neighborhood of v. The binary relation that represents the communication link is supposed to be symmetric,* i.e. $v' \in N(v) \Leftrightarrow v \in N(v')$.

Definition 2 *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 $\forall i \in [\![1, n[\![,\ o_{i+1} \in N(o_i)$. 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" are few, weak and sensible to topological modifications [10]. In our measures, for two given nodes v and w, we will consider a subset of $P(v, w)$ called "sub-optimal loop-free paths".

Definition 3 *Let v and w be two nodes and $p = o_1, o_2, \ldots, o_n \in P(v, w)$ a path between v and w.*

1. *p is called optimal if and only if $\forall p' \in P(v, w) \quad |p'| \geq |p|$. If p is optimal, $|p|$ is called distance between v and w and is denoted by $d(v, w)$.*
2. *p is called loop-free if and only if $\forall i, j \in [\![1, n]\!]$, $o_i = o_j \Rightarrow i = j$.*
3. *p is called k-sub-optimal (with $k \geq 1$) if and only if p is loop-free and $|p| < d(v, w) + k$. We denote by $SOP_k(v, w)$ the set of k-sub-optimal paths between v and w.*

To illustrate the two assertions given above, let's have a look on Figure 2.a and Figure 2.b. On both, dashed lines represent a physically possible connection and solid ones represent available disjoined path from the client to the server.

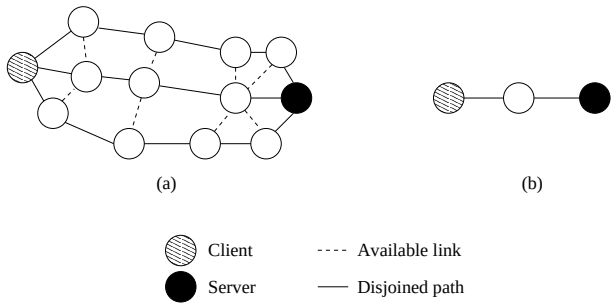

Figure 2: Network samples

The first one (*a*) shows how a network configuration can be reliable. Indeed, there are three disjoined paths to link the server and the client, if one of those paths broke, packets would be able to use one of the two others and so, the connection is not interrupted. The second one (*b*) illustrates the efficiency of a configuration. As the nodes are close from each other, the packets' route is short and they do not loose time by multi hopping from node to node. Moreover, the number of nodes involved is little, so as the overhead generated.

The problem is that those two quality criteria are rather contradictory, one requires to involve a lot of nodes, the other, a few. So, we need to find a compromise between those criteria. Following this idea, we define a metric for link robustness that takes into account reliability and efficiency.

Definition 4 *Let v and w be two nodes. The set of parts of $SOP_k(v, w)$ containing only disjoined paths is denoted by $DSP_k(v, w)$ and is defined by:*

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

Definition 5 *Let v and w be two nodes. The robustness of the link between v and w is defined by:*

$$R_k(v,w) = \max_{C \in DSP_k(v,w)} \left\{ \sum_{p \in C} \frac{1}{|p|} \right\}.$$

This formula handles reliability by counting the number of disjoined paths from the server to the client and efficiency by giving more importance to a short path than to a long path. The main problem of this theoretical definition is that it is rather impossible to compute its exact value with a small device as we need to know all possible disjoined paths leading to the server. We must then find an algorithm that gives us an approximative value of the robustness at low cost in term of network congestion.

3.2 Practical approach

To answer the question of practical measurement of the robustness, we first need to know who computes this value, the server or the client? As the server should have a lot of clients, we can't let him loose time and resource by computing robustness for all its clients. So, the client will compute it.

3.2.1 Efficient broadcast based protocol (mode 0)

Here is a first protocol proposition, we respectively denote by v and w the client and the server:

1. v sends a link evaluation request message to w using a routed message (here, we can use an efficient routing protocol) ;
2. w replies by a link evaluation reply message using a flooding protocol with a TTL of $d(v,w)+k$ (k depends on the $DSP_k(v,w)$ set we want to use) ;
3. each nodes forwarding the messages appends its ID in the message in order to construct the path used by the message ;
4. v receives the responses and check it with the paths it has already stored. If this path is disjoined of all the others, the node stores it. Otherwise, the path is discarded (as this path came later to the node, it is supposed to be longer so we do no need it) ;
5. when a given time out has elapsed, v can evaluate $R_k(v,w)$.

Efficient flooding algorithms are known to be unfair-loading [9, 12, 5] as they try to select a small set of nodes for forwarding packets. This leads to routes involving always the same nodes and though, not all the routes. As few disjoined paths leads to very approximative robustness, we need to generate more and therefore to spawn more packets in the network. So, we propose to extends the broadcast method.

3.2.2 Stormy broadcast based protocol (mode 1)

In a typical broadcast algorithm, when a node receives a packet, he checks its sequence number. If the node has already forwarded a packet with the same sequence number, he discards it. Our idea is to grow the number of emitted packets in the neighborhood of the server by simply asking its neighbors to make the broadcast for him. As it generates too much traffic, the server asks its neighbors and proposes them a small set of sequence number in which nodes randomly select one for doing their own broadcast. Here is a simplified explanation of this (the client and the server are still denoted by v and w respectively):

1. v sends a link evaluation request message to w using a routed message (here, we can use an efficient routing protocol) ;
2. w sends a message containing a set of sequence numbers to all its neighbors. When receiving it, a node randomly selects a sequence number in the set and sends a link evaluation request using the simple flooding algorithm depicted above with a TTL of $d(v,w)+k$;

3. each nodes forwarding the messages appends its ID in the message in order to construct the path used by the message ;

4. v receives the responses and stores only disjoined paths (by the same way as mode 0) ;

5. when a given time out has elapsed, v can evaluate $R_k(v, w)$.

The result of this method are good in regards of the number of disjoined paths generated which grows up in a significant way. The problem of this method is that the traffic generated blows up. The main problem of this method is that as a broadcast is used, nearly all the network is flooded by the packets exchange.

3.2.3 Directed response based protocol (mode 2)

The idea to enhance mode 1 is to limit it in the needed zone as most as possible. We defines a last algorithm to evaluate $R_k(v, w)$:

1. v sends a link evaluation request message to w using an efficient broadcast protocol [4]. Every node u receiving the packet stores $d(v, u)$;

2. w sends a message containing a set of sequence numbers to all its neighbors. When receiving it, a node randomly selects a sequence number in the set and sends a link evaluation request broadcast. Every node u receiving this packet forwards it if $|p(w, u)| + d(u, v) \leq d(v, w) + k$ and if it has not forward a packet with the same sequence number before ;

3. each nodes forwarding the messages appends its ID in the message in order to construct the path used by the message ;

4. v receives the responses and stores only disjoined paths (by the same way as mode 0) ;

5. when a given time out has elapsed, v can evaluate $R_k(v, w)$.

After simulations, this protocol appears to be far better than the others depicted above. The number of paths generated are quite acceptable and the traffic generated is even better than an efficient broadcast. Moreover, if the lower network layers implements an efficient "hello" protocol that allows the building of a table giving an approximation of the $d(u, v)$, the link evaluation request can be done by a simple routing message using an efficient protocol.

4 Experiments

All the three protocols described in Section 3 were simulated using a discrete-event simulator that we developed in C++. We assumed an ideal MAC protocol which provides collision-free broadcasting at 11Mb/s (the average packet transmission time is then 1ms). The parameters that are fixed in our simulations are the transmission radius (10 meters), the size of the overall area (250m x 250m) and the nodes' speed (10 meters by second). The density (*i.e* the average number of nodes in each node's neighborhood) can be 5, 10, 15, 20, 25 and 30 (equivalent to 995, 1990, 2985, 3980, 4976 and 5971 nodes respectively). The average distance between client and server can be 5, 10 and 15 hops. We only focused our attention on the reply because the request is done by rather the same protocols. In each case, 1000 link evaluation are done and we looked for three parameters:

- number of disjoined paths found ;
- number of nodes involved in the link evaluation reply ;
- number of packets generated by the link evaluation reply.

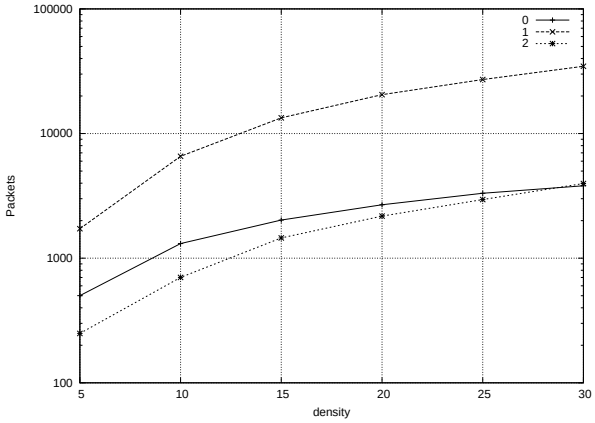

Figure 3: Number of emitted packets (distance 10)

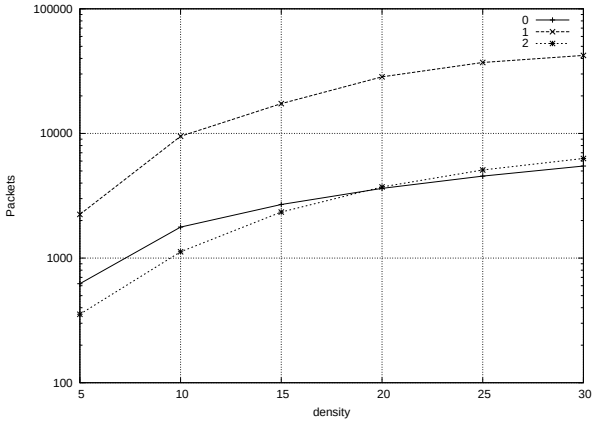

Figure 4: Number of emitted packets (distance 15)

5 Conclusion

This paper propose a new metric for link robustness in mobile wireless environment and presents a efficient and reliable algorithm to evaluate it. Our experiments have demonstrated through simulations and analysis that our methods gives a good metric of a link's robustness without blowing down the network efficiency by a kind of "broadcast storm". As the computation needed by the nodes are really light, we expect this methods to work on small devices. Moreover, as the algorithm is totally decentralized, it handles topology changes well.

We now need to test this algorithm with a non perfect MAC protocol (802.11b for instance) as doing two or three broadcast in the same time can result in a high collision rate and so to less efficient results. Moreover, we are already working on a protocol using this one to elect the host of service replication at low cost.

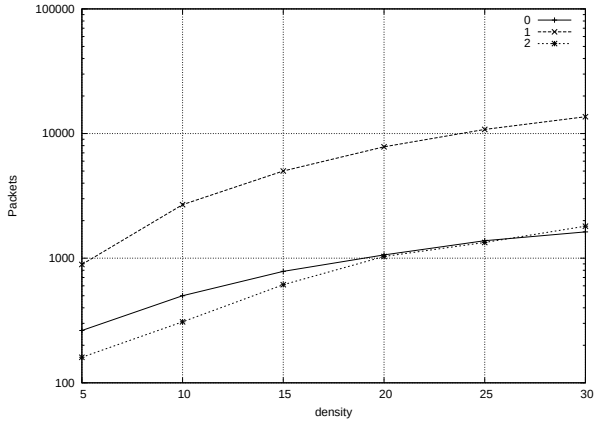

Figure 5: Number of emitted packets (distance 5)

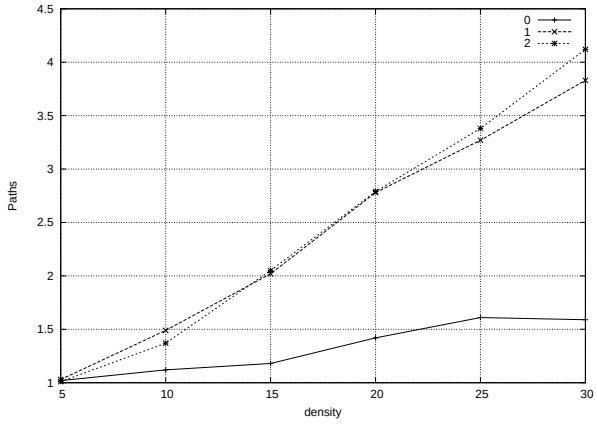

Figure 6: Number of disjoined paths found (distance 10)

References

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

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

[3] IEEE standards boards part 11: Wireless LAN medium access control and physical layer specifications, 1999.

[4] J. Cartigny, D. Simplot, and J. Carle. Stochastic flooding broadcast protocols in mobile wireless networks. *submitted*, 2002.

[5] A. Laouiti, A. Qayyum, and L. Viennot. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. *35th Annual Hawaii International Conference on System Sciences (HICSS'2001)*, 2001.

[6] E. M. Royer 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.

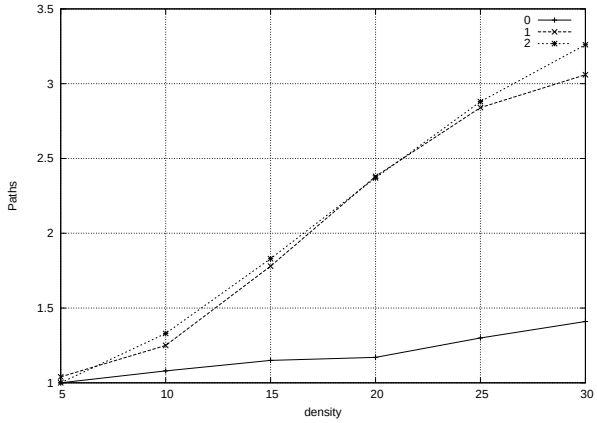

Figure 7: Number of disjoined paths found (distance 15)

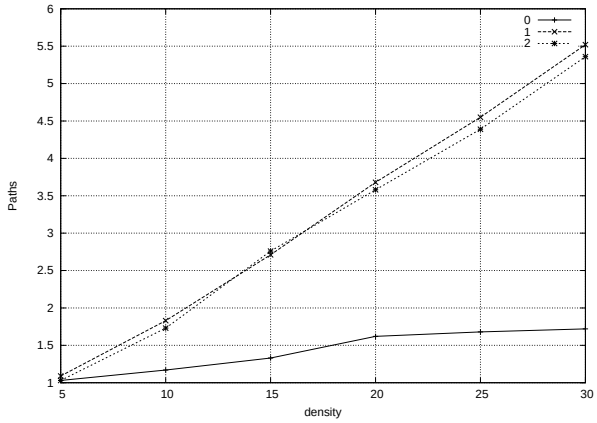

Figure 8: Number of disjoined paths found (distance 5)

[7] K. Chen S.H. Shah 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.

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

[9] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(12), December 2001.

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

[12] J. Wu and H. Li. A dominating-set-based routing scheme in ad hoc wireless networks. In *Proceedings of the Third Int'l Workshop Discrete Algorithms and Methos for Mobile Computing and Comm (DIALM)*, pages 7–14, August 1999.

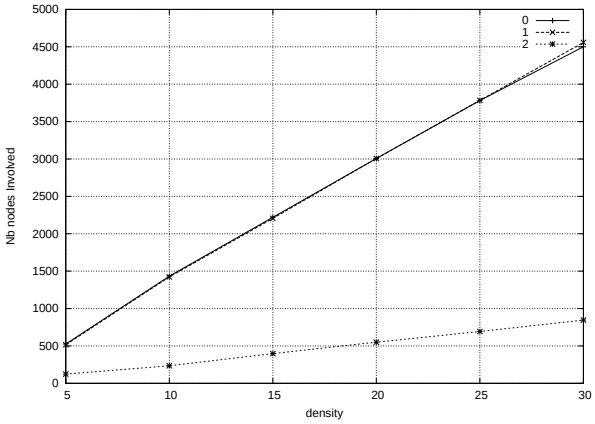

Figure 9: Number of objects involved (distance 10)

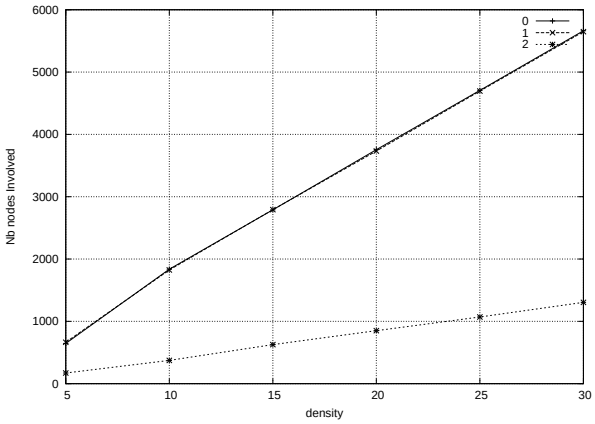

Figure 10: Number of objects involved (distance 15)

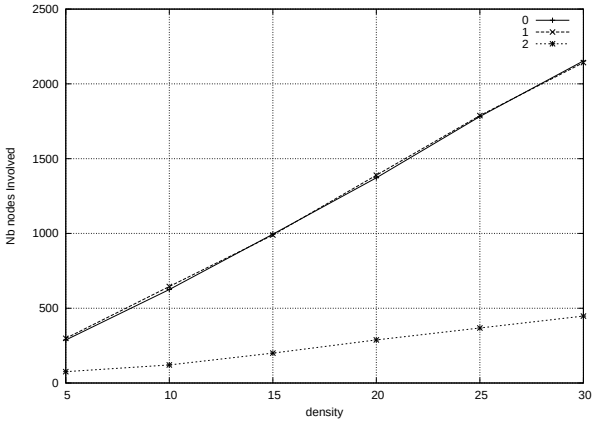

Figure 11: Number of objects involved (distance 5)