ROUTING IN EXTREMELY MOBILE NETWORKS

Géraud Allard
INRIA - Domaine de Voluceau
Rocquencourt, France
geraud.allard@inria.fr

Philippe Jacquet
INRIA - Domaine de Voluceau
Rocquencourt, France
philippe.jacquet@inria.fr

Bernard Mans
Dpt. of computing - Macquarie University
Sydney, NSW 2109, Australia
bmans@ics.mq.edu.au

Abstract   To become realistically untethered, wireless communication networks need to be self-organised, rapidly deployable, infrastructureless and mobile. Existing protocols are efficient in routing data dynamically between mobile nodes that belong to the same connected component. Concrete applications such as Defence and Disaster-Relief cannot always assume that the network is connected (*i.e.,* not partitionned). However, even if the network is continuously partitioned, a "communication path" may be available through time and mobility using intermediate mobile nodes (temporally within reach of each other) - we have coined these "Extremely Mobile Networks". We consider the problem of routing in a highly mobile network which, possibly, may never be fully connected. We introduce new algorithms that always allow to route a packet toward a remote destination. The packet bounces from connected components to connected components, thanks to node mobility.

1.   Introduction

Existing MANET protocols are highly efficient in routing data between mobile nodes that belong to the same connected component. What about a disconnected network where source and destination may be located in two different

connected components? In this case usual routing protocols drop packet due to `host unreachable`. A simple idea is to allow the router that has no available route to the destination to keep the packet in buffer until the conditions become more appropriate for forwarding. The forwarding conditions will change because of mobility: the router can move closer to the destination so that they belong to the same connected component and the packet can be delivered. Realistically, the network may be continuously partitionned due to high mobility, and the traditional approach to allow a mobile node to wait for the network to be fully connected (*i.e.,* form a unique component) or to wait to be in range of the destination may lead to unacceptable delays. Furthermore, concrete applications, such as Defence and Disaster-Relief, cannot always rely on such assumptions. Nevertheless, even if the communicating nodes may *never* be within the same connected component, it is important to observe that a "communication path" may be available through time using intermediate nodes that are temporarily within reach of each other while moving, hence making such networks viable for critical applications - we have coined these *Extremely Mobile Networks*. Initial works have highlighted the importance of the problem by demonstrating the difficulty at hand. For torus-like networks, the authors in [9] propose algorithms using counting timers combining a simple random policy with the sophistication of utility-based policies and evaluate their performance. In [5], Li and Rus proposed to actively modify the trajectories of the mobile nodes to guarantee transmissions in minimal time while minimising the trajectory modifications, but this requires that the movement and the positions of the nodes are known (*e.g.,* using GPS). In [8], Shah proposed to deliver messages in disconnected MANET by broadcasting multiple copies of the message in the connected component (while limiting the number of multiple copies among neighbours and providing time-out for holding messages). However, for small networks, the replication of the messages increases the traffic substantially as well as reducing the capacity. Earlier studies [3, 10] have developed protocols for networks that exhibit correlated movement patterns. The rest of previous studies on mobility have assumed the continuous connectedness of the networks (in order to find adequate routing paths). This assumption has deflected the effort of researchers towards designing consensual mobility models for simulations in order to provide consistent and comparable conclusions [1]. The problem of bandwidth optimization by using space and mobility in a fully connected network is a different problem and is addressed in [2][4].

In this paper, we make the important assumption that the node motions are unpredictable. We introduce a novel routing algorithm based on information aging. We compare its efficiency with the basic routing strategies by providing comprehensive simulations and characterising the desired properties.

2. Simple Routing Strategies

We first review three simple strategies: Source Only (SO); Random Jumps (RJ); Flood Routers (FR). For each strategy we briefly highlight the pros and cons in particular with regard to the mobility models.

Source only. With this algorithm the packet stays on the source router until a route to destination is available, *i.e.,* when the router and the destination are in the same connected component. *Advantages:* The number of transmission is limited to the number of hops in the same connected component. This hopefully makes better use of the capacity. *Drawbacks:* The delivery delay may be extremely large when routers move randomly like in random walk model. It can be infinite when the source and destination are far apart and do not move.

Random jumps. With this algorithm the current router transmits the packet to a random router of the connected component of the current router. The new router becomes the current router which will retransmit the packet (and so forth), until the current router is in the same connected component of the destination (thanks to mobility). *Advantages:* The process is not limited by the relative mobility of the source and the destination. *Drawback:* The frequency at which the packet is retransmitted may lead to a bandwidth waste. The frequency must be frequent enough so that the node can jump to another connected component when there is a temporary merge. In random mobility models the move of the packet is equivalent to a random walk which may take a huge amount of time to reach the connected component of the destination. *Remark:* The frequency of retransmissions could be limited by only allowing transmissions toward new nodes in the connected component. This is still a random walk anyhow.

Flood routers. The packet is flooded to all members of the connected component. Every time a new node comes into the connected component it receives directly the packet from a border node, then the node floods the packet to its neighbors, *i.e.,* only node which receives the packet for the first time retransmits it. The packet will eventually reach its destination (possibly, through multiple routes). *Advantages:* The packet will arrive to destination much faster than with a random walk. In fact, without considering physical constraints (*e.g.,* interference and collisions), this is clearly the fastest strategy as any route used by another strategy is one of the branch of the flooding. There is no need to manage the connected component membership (flooded packets do that by themselves). *Drawback:* The cost of maintaining and flooding a huge number of packet copies in the network may be prohibitive and may impede the network capacity. Flooding may be optimized using MultiPoint Relays (MPR) as in the OLSR protocol [6]. However, there is no way, except using timeout or sophisticated acknowledgment flooding, to kill redundant copies when the destination received its own.

3.   Forward to Best Gateway

We now introduce a strategy, that we call Forward to Best Gateway (FBG), based on information aging. This algorithm aims to get similar fast convergence of the flood router algorithm without the cost of multiple packet flooding. Like in FO and RJ, the packet is stored in only one router at a time. The router forwards to the best gateway to the destination when there is a new one. The current router is always the best gateway. The best gateway is the relay which has the most recent and shortest route to the destination. To compute the most recent and shortest route the algorithm relies on a link aging protocol.

Link aging protocol. Every node maintains a topology database of the links with the age. Age(L) is the age of link L in the database. The age of the link is obtained from the information obtained from other nodes. When two nodes are just connected by a rising link they compare they database. Then they flood (via MPR) their connected component with the most recent link age. In other words each of the two nodes retransmits a tuple (link, age) when the age is smaller than the age(link) which was previously in its database.

Most recent and shortest route calculation. We define age(route) as the minimum value of age(link) for all links in the route. A route is an increasing age route when the value age(link) increases on the route. There are two options: (1) shortest path of the less aged route, (2) shortest of the less aged increasing (link) age route. In the first case one computes the shortest route with the metric age-length, where age-length(route1)< age-length(route2) iff (i) age(route1)< age(route2), or (ii) length(route1)< length(route2) when age(route1) = age(route2). This is not a basic Dijskstra algorithm since the optimal route is not necessarily the concatenation of optimal routes. For example age(AC)=10, age(AB)=0, age(BC)=0, age(CD)=20, age(AD)=30. In this case the optimal route from A to C is (A,B,C), from C to D is (C,D), but from A to D it is (A,C,D). However this can be computed via a vector Dijkstra. In the second case one computes the shortest less aged increasing age route by a simple Dijkstra algorithm since any optimal increasing age route is the concatenation of optimal increasing age routes.

THEOREM 1 *For any route route1 there exists an increasing age route route2 such that age(route1)=age(route2).*

Proof. The proof comes from the fact that the information from the status of a link necessarily came to the node via a link less aged (a link which was active when the link age update went through). ∎

Aged distance vector variant. The shortest less aged increasing route can be computed as follow: for any destination d, each node keeps a distance vector (next hop, age, length). If its next hop is aged (not active) then the node is said to be gateway to d. Gateways to d continuously advertized their tuple (age,

length) in their connected component. The nodes in the connected component compute their next hop to their best gateway according to the metric age-length by adding to the length the distance to the gateway. In the process the node may lose its status of best gateway. *Rational:* The gateway nodes are likely to be on the border the closest to the destination. The closer a connected component is to the destination, the smaller should be the age to this destination (a node in the same connected component shares the same age for a given destination). When two connected components merge, the one with the smaller age should be closer and enforces its gateway nodes and the gateway nodes in the other connected component lose their status. Therefore the packet directly jumps towards a gateway node of the connected component with the smallest age. By following this strategy, the packet is expected to get closer to the destination.

4. Models and parameters of interests

First of all, we assume that the speed of packet propagation inside a connected component is infinite (very large compared to the node speed). We need to take into account the three following parameters v_m, v_a and v_p. Quantity v_m is the average speed of the destination node, v_a is the speed at which a flooding via FR protocol propagates. Quantity v_p is the speed at which the packet heads towards its destination in the best gateway FBG. We expect the following inequality: $v_p \leq v_a$. The flooding should propagate faster than the best gateway (which can be seen as one branch of the flooding tree). Quantity v_a should also be the speed at which the age toward a given destination propagates. In other words if a node was at position z at time t then the nodes which have route of age θ at time $t+\theta$ should be at average distance $\theta \times v_a$. Therefore it is expected that the condition for the aging protocol to give a good indication of node position (without assuming the use of devices such as GPS) be: $v_m < v_a$. Looking carefully it looks also as a condition for the FR algorithm to work properly (*i.e.,* better than Random Jump). If v_a is about the speed of sound then the destination should move at subsonic speed. Furthermore, for the FBG algorithm to work properly, we need that the packet moves at speed faster than v_m in order to eventually reach the destination and to let the algorithm work better than a basic random walk: $v_m < v_p$. Notice that we must consider the asymptotic speed of the mobile, *i.e.,* the average distance between initial position and final position when the time tends to infinity. In the case of a random walk this speed tends to zero. Therefore the algorithm will work with any random walk model even if the domain of the network is unbounded.

Therefore the performance of the FBG algorithm compared to FR algorithm can be described with the ratio $\frac{v_p}{v_a}$. The quantity v_a is a function of the node mobility model and the node density (*e.g.,* [4]). The faster the nodes move the larger is v_a. It should be interesting to consider the random walk model under

the three alternative conditions: (i) the free space distance is smaller or of same
order than the radio ranges; (ii) the free space distance is smaller or of same
order than the connected component average radius (shapes may be intricate);
(iii) the free space distance is much larger than the connected component aver-
age radius. In the later case the performance should be similar to the random
way point model. *Conjecture:* if the average speed v_m for traveling across a
connected component is the same in every component then the (v_a, v_p) tuple
should be similar accross any model.

5.   Simplification and improvement

Jump to best age. When a connected component with lesser age merges
with the current connected component, the packet could simply jump to any
node in the less aged connected component (instead of traveling to the new best
gateway). In this case there is no need to maintain the complicated gateway
system. Only the age to the destination (and not the path) in the connected
component would suffice (in addition to the membership monitoring inside the
connected component).

Jump to the oldest of the same age. Instead of randomly jumping on
a mobile router that have the least age before the merge, one can make the
packet jumping to the router whose actual age has changed the less recently,
i.e., to the oldest in the age in the connected component (since all nodes in the
connected component got the same age).

Use of the Doppler effect. We introduce a new improvement using the well
known phenomenon of frequency shift named after Doppler (1803-1853). We
define the Doppler effect of a node A as the average rate at which the age to the
destination varies. A large Doppler effect indicates that the node is traveling
toward the destination. The Doppler effect could be used for the protocol: the
packet likely moves to the nodes which have the largest Doppler effect. For
example, when the connected component merges with a connected component
with a younger age then the packet moves to the node with highest Doppler
effect in the new connected component.

Use of the average age of the nodes. Let A and B be two nodes and let
$age(AB)$ be the age of the link AB stored in A's database. The *average age*
of AB, denoted as $\overline{age}(AB)$, is recomputed as follows each time $age(AB)$
changes: the new value of $\overline{age}(AB)$ is $\rho \times \overline{age}(AB) + (1 - \rho) \times age(AB)$
with $0 < \rho < 1$ as a fixed parameter. Thereby a node can keep track of its past
ages and use this parameter for routing by sending packets to the node that
has the smallest average age. The number of retransmissions is limited since a
packet is only forwarded when an age modification is detected.

6.   The road model

In this model we assume an unbounded highway with a stream of cars moving to the left and another stream of cars moving to the reverse direction, all with uniform speed v_m. We assume that on both lanes the car distribution follows the law of Poisson with density D. We use the unit disk graph model (the radio range is exactly one unit). Using basic algebra, we can prove the following theorem.

THEOREM 2 *When D is the density on a single lane, $v_a = \frac{e^{2D}+1}{2} v_m$.*

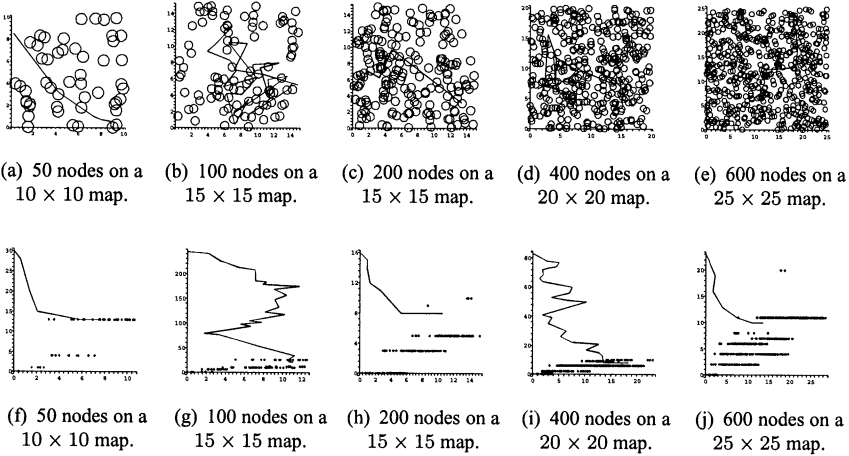If we properly use the Doppler effect then we can get $v_p = v_a$.
Proof. We consider a packet that is routed toward the right. The average length L of the connected component containing the car that holds the packet and made of cars of both lanes satisfies the identity: $L = \ell + (1 - e^{-2D})L$ with $\ell = \int_0^1 2Dx e^{-2Dx} dx = \frac{1-(2D+1)e^{-2D}}{2D}$. The age information instantaneously travels in the connected component. When the packet arrives at the end of the connected component it has to wait $\frac{1}{Dv_m}$ before jumping of one unit distance and then entering a new connected component. Therefore the average age propagation speed on this lane in the referential of the car is $(L+1)v_m D = \frac{e^{2D}-1}{2} v_m$. Adding the car speed v_m yields the desired result. ∎

Remark: the speed tends to v_m when $D \to 0$ since the connected components tend to vanish. The speed is proportional to v_m which is expected but also proportional to the exponential of car density.

7.   Simulations

We first present some simulations that show the impact of our algorithm on the behaviour of the routers and packets. We then compare the performances of our algorithm, and its variants, with the basic strategies.

Routing and packet paths. We consider the random walk model where at each step nodes move in average of unit distance (the radio range). We display for different maps size and router number: (1) the router map and the packet path: *i.e.,* the actual location where the packet has jumped until it reaches its destination The router map corresponds to the positions of the routers when the packet reached his destination. (2)The router age histogram and the packet history: the router age histogram gives the distribution of age versus distance to destination, the packet history gives the time at which the packet has jumped to a new router versus the distance of this router to the destination. The simulations show that the FBG algorithm works very well in dense networks but show great variations in the behavior of the packet routing in sparse networks. For instance the packet may travel quickly to its destination or may stay a very long time far from the destination even if it jumps frequently from routers to routers.

(a) 50 nodes on a   (b) 100 nodes on a   (c) 200 nodes on a   (d) 400 nodes on a   (e) 600 nodes on a
    10 × 10 map.   15 × 15 map.   15 × 15 map.   20 × 20 map.   25 × 25 map.

(f) 50 nodes on a   (g) 100 nodes on a   (h) 200 nodes on a   (i) 400 nodes on a   (j) 600 nodes on a
    10 × 10 map.   15 × 15 map.   15 × 15 map.   20 × 20 map.   25 × 25 map.

FBG-based routing algorithms performance. We conducted 400 simulations using five different algorithms: SO, RJ, FBG and two algorithms based on FBG denoted as *FBG with Oldest Same Age* (FBG-OSA) and *FBG with Average Age* (FBG-AA). We considered 600 mobile routers located on a $250m \times 250m$ area. Transmission ranges are supposed to be equal to $10m$ for all the routers. We used the Random Waypoint mobility model with speeds randomly chosen between $5m.s^{-1}$ and $10m.s^{-1}$. For each scenario, we randomly choose a source node and a destination node respectively denoted as *orig_node* and *dest_node*. Let us consider a packet issued by *orig_node* that must be delivered to *dest_node*. Let t_a be the time elapsed between the packet's first transmission by *orig_node* and its reception at *dest_node* using algorithm a. For a given scenario, let us define $T_{a,b}$ as $T_{a,b} = t_a - t_b$, with a and b two different routing algorithms. If $T_{a,b} > 0$ (*resp.* $T_{a,b} < 0$) then the algorithm b (*resp.* a) has delivered the packet to *dest_node* faster than the algorithm a (*resp.* b). In the sequel, we will use $T_{a,b}$ to compare the packet delivery time of two algorithms.

We show pairwise comparison between algorithms. Each *Algo1 vs. Algo2* comparison depicted shows the probability that either *Algo1* or *Algo2* leads to the fastest packet delivery. It also shows the probability that delivery time is the same for both algorithms (denoted as "=" on the figure). First, *RJ vs. SO* indicates that RJ and SO have almost the same probability of fastest packet delivery ($\approx 40\%$). Then, analysing *FBG-based* algorithms performance shows that FBG-OSA and FBG-AA allow a faster packet delivery than SO and RJ with a significant probability ($\approx 60\%$ for FBG-AA). Furthermore, although *FBG-OSA vs. FBG-AA* have 40% of identical delivery time, FBG-AA's per-
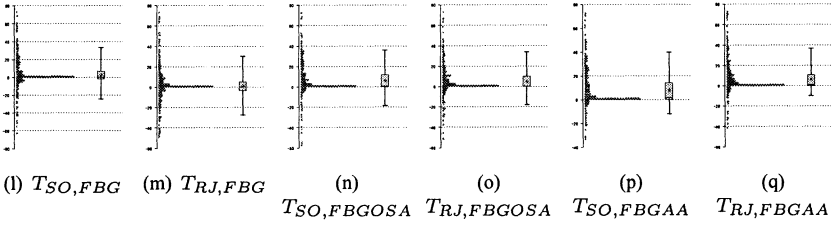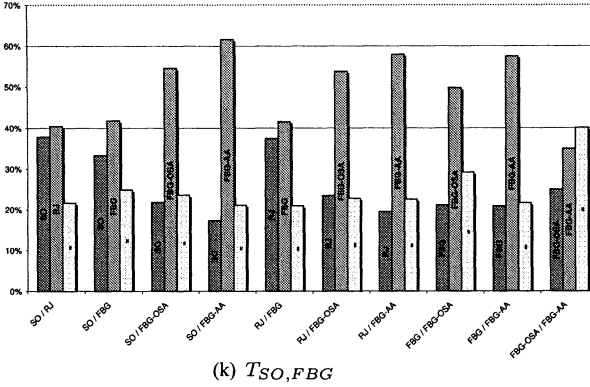
(k) $T_{SO,FBG}$

(l) $T_{SO,FBG}$ (m) $T_{RJ,FBG}$ (n) (o) (p) (q)

$T_{SO,FBGOSA}$ $T_{RJ,FBGOSA}$ $T_{SO,FBGAA}$ $T_{RJ,FBGAA}$

Figure 1. Probability of Packet delivery time and Pairwise Comparison of Algorithms.

formance seems to be slightly better than FBG-OSA. On the other hand, FBG does not really improve the probability of fastest time delivery compared to SO and RJ since the performance is quite similar for these algorithms. In addition, FBG-OSA and FBG-AA are better than FBG for almost 50% and 55% of the simulations while they lead to same results for 30% and 25%. We depict the distributions of $T_{Algo_1,Algo_2}$ with $Algo_1 = \{SO, RJ\}$ and $Algo_2 = \{FBG, FBG - OSA, FBG - AA\}$. We leave aside the FR algorithm since we do not want the network to carry several copies of the same packet, which could lead to heavy network load. The left part of each figure describes the distribution of $T_{Algo_1,Algo_2}$ in seconds for the 400 simulations. The boxplot on the left indicates some noteworthy values of the distribution. The lower and the upper tails represent respectively the 5th and the 95th percentile (*i.e.,* respectively 5% and 95% of the results are lower than this value). The lower and the upper bound of the rectangle represents respectively the 25th and the 75th percentile of the data set. The line and the point inside the rectangle indicates respectively the median (*i.e.,* the 50th percentile) and the mean value of the distribution. First, it is important to note that the 25th percentile and the median

are close to zero for all the distributions: 25% of the values are quite similar for the compared algorithms. Next, let us focus on the FBG-AA algorithm. Since the mean values of $T_{SO,FBG-AA}$ and $T_{RJ,FBG-AA}$ are respectively 7.59s and 5.63s we can expect a good average gain using FBG-AA. Furthermore, the 5th percentile of $T_{SO,FBG-AA}$ and $T_{RJ,FBG-AA}$ are respectively equal to $-11.7s$ and $-8.4s$, hence we can deduce that SO and RJ best values are very closed to FBG-AA worst ones. In other words, when SO or RJ are better than FBG-AA, the gain brought on by SO or RJ is rather small. On the other hand, the 75th percentile (13.75s) and the 95th percentile (36.1s) indicates that FBG-AA leads to a high gain with a significant probability. We can make a similar analysis for FBG-OSA however FBG-AA presents better results. Note that, the FBG algorithm does not achieve really good performance especially because of the low 5th, 25th percentiles. The distribution tends to be center at 0 especially for $T_{RJ,FBG}$. It shows how significant improvements provided by FBG-OSA and FBG-AA are. To sum up, FBG-OSA and especially FBG-AA not only avoid SO and RJ drawbacks (*i.e.,* source node immobility and large number of retransmissions) but also achieve better packet delivery time with high gain.

References

[1] T. Camp, J. Boleng, and V. Davies, 'A Survey of Mobility Models for Ad Hoc Network Research', *Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, **2(5)**, (2002), pp. 483-502, 2002

[2] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad hoc wireless netorks", IEEE infocom 2001.

[3] X. Hong, M. Gerla, G. Pei and C. Chiang, 'A Group Mobility Model for Ad Hoc Wireless Networks', *ACM Workshop on Modeling and Simulation of Wireless and Mobile Systems*, 1999.

[4] P. Jacquet, "Space-time information propagation in Mobile ad hoc Networks", IEEE Information Theory Workshop, Texas, October 2004.

[5] Q. Li and D. Rus, *Communication in disconnected ad hoc networks using message relay*, Journal of Parallel and Distributed Computing, vol. 63, pp. 75-86, 2003.

[6] T. Clausen, P. Jacquet, et al., *Optimized Link State Routing Protocol*, IETF-RFC-3626, http://www.ietf.org/

[7] R. Rajaraman, 'Topology Control and Routing in Ad hoc Networks: A Survey', *SIGACT News*, **33**, (2002),pp. 60-73.

[8] R. Shah and N.C. Hutchinson, 'Delivering Messages in Disconnected Mobile Ad-Hoc Networks', *ADHOC-NOW*, LNCS 2865, Springer-Verlag, pp. 72-83, 2003.

[9] T. Spyropoulos, K. Psounis, C. Raghavendra, "Single-Copy Routing in Intermittently Connected Mobile Networks," IEEE SECON, October 2004.

[10] K. Wang and B. Li, 'Efficient and guaranteed service coverage in partitionable mobile Ad Hoc Networks', *IEEE-Infocom*, vol. 2, 2002, pp. 1089-1098.