An Architecture for Seamless Mobility in Spontaneous Wireless Mesh Networks

Franck Rousseau, Yan Grunenberger, Vincent Untz, Eryk Schiller,
Paul Starzetz, Fabrice Theoleyre, Martin Heusse, Olivier Alphand, Andrzej Duda
*
LIG - Grenoble Informatics Laboratory
Grenoble, France
{rousseau,ygrunenb,untz,schiller,starzetz,theoleyr,heusse,alphand,duda}@imag.fr

ABSTRACT

In this paper, we consider *spontaneous wireless mesh networks* that can provide wide coverage connectivity to mobile nodes. Our mobility scheme builds upon separation between a persistent node identifier and its current address. When joining the mesh, a mobile node associates with a mesh router that updates a location service managed in the mesh as a distributed hash table. Mobility implies changing addresses while a node moves in the mesh. To keep the rate of location updates and correspondent node notifications low, the address of the new mesh router with which the mobile node is associated needs to be topologically close to the previous one. Thus, such a mobility scheme requires an addressing space with specific properties. We achieve this by defining an algorithm for constructing a pseudo-geographical addressing space: a few nodes know their exact locations and others estimate their relative positions to form a topologically consistent addressing space. Such an addressing space also enables scalable and low overhead routing in the wireless mesh—we propose a trajectory based long distance ballistic geographical routing.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Spontaneous wireless mesh networks, routing protocols, mobility, separation of identities and addresses

*LIG is a joint research laboratory of CNRS (*Centre National de la Recherche Scientifique*), INRIA (*Institut National de Recherche en Informatique et Automatique*), INP Grenoble (*Institut Polytechnique de Grenoble*), UJF (*Université Joseph Fourier*), and UPMF (*Université Pierre-Mendès-France*).

MobiArch'07, August 27–31, 2007, Kyoto, Japan.

Keywords

Coordinate addressing space, geographical routing

1. INTRODUCTION

We focus on *spontaneous wireless mesh networks* that begin to emerge to provide wide coverage connectivity to mobile nodes. One of their characteristics is *self-forming*—users just add another mesh router in some place to increase the coverage. The organization of the mesh networks needs to be *autonomic* to a great extent, because unlike the current Internet, such networks are not managed so they cannot rely on highly skilled personnel for configuring, connecting, and running mesh routers. However, we assume that they are structured according to *communities* formed by nodes sharing a common relationship of trust.

We consider that such mesh networks may be composed of a large number of routers providing connectivity to mobile nodes. We assume that mesh routers benefit from abundant resources (memory, energy, computation power, GPS devices in some cases) and may only move occasionally. To provide sufficient capacity to mobile stations, they may use multiple wireless network interfaces and different types of antennas such as sectorial or directional for increased space coverage. Neighbor mesh routers directly communicate over wireless links. To support end-to-end connectivity between any source and destination in the mesh, communication may require intermediate nodes that forward packets in a multihop way.

As a first approach, we consider the standard 802.11 technologies for wireless links to develop operational prototypes. However, we also work on the problems of integrating efficient physical and link layer mechanisms for the wireless mesh coupled with upper layer protocols. The idea is to explore a cross-layer approach in which the MAC layer gives access to the radio channel when a router needs to forward a packet. We think that the traditional view of separating functionalities into independent layers needs to be reconsidered, because of complex interactions between physical transmission of radio signals, access methods, and packet forwarding. Either all these functions are collapsed into one layer, or three lower layers are redesigned through cross-layer optimization. Thus, if necessary, we consider modifications at all the lower layers. In this paper, we only concentrate on network layer mechanisms for dealing with mobility: addressing and locating mobile nodes as well as routing and forwarding packets.

Our mobility scheme builds upon separation between a

persistent node identifier, EID (*end-point identifier*) and its current address of attachment to the mesh network. We store their binding in a *location service* managed by the mesh network as a distributed hash table (DHT). We assume that the main entities (mobile nodes and mesh routers) are different—mesh routers are stable and may only move occasionally whereas mobile nodes move in the mesh and benefit from connectivity through a neighboring mesh router. When joining the mesh, a mobile node associates with a mesh router that updates the location service with the EID of the mobile node and its address (that of the router). Mobility implies changing addresses of neighbor mesh routers when the mobile node moves in the mesh while keeping its EID. At some instants, the mesh router needs to update the location service and the correspondent nodes with a new address. If we want to keep the rate of location updates and correspondent node notifications low, the address of the new router with which the mobile node is associated needs to be topologically close to the previous one. Moreover, distances in the addressing space need to be related in some way to geographical distances, because of the movements of the mobile node. Thus, such a mobility scheme requires an addressing space with specific properties.

There is also another reason for constructing such a specific addressing space. As we want to achieve scalable and low overhead routing suitable for spontaneous wireless mesh networks, we explore other approaches to routing and forwarding than the standard IP. Unlike traditional approaches, *geographical routing* presents interesting properties: it does not require any information on the global topology so that there is no need for creating and maintaining routing tables. Moreover, it also perfectly fits intrinsic spatial characteristics of mesh networks—for instance, under geographical routing a mesh router with several sectorial antennas can easily forward packets in the direction towards a given destination. Geographical routing requires a coordinate addressing space, which in our case needs to be managed in an autonomous and fully distributed way. We thus propose an algorithm for constructing a pseudo-geographical addressing space based on a small number of nodes that know their exact locations. Such an addressing space enables a trajectory based long distance geographical routing that we call *ballistic*.

The paper is organized as follows. Section 2 defines our mobility architecture. Section 3 briefly describes the principles of the pseudo-geographical addressing space. Section 4 presents the trajectory based long distance ballistic geographical routing. Section 5 integrates all principles and elements to define mobility management. Section 6 reports on the current prototype development and Section 7 concludes the paper.

2. WIRELESS MESH ARCHITECTURE

We start with the definition of the main elements in our architecture. As in many fundamental contributions in this domain [3–5, 15, 17], we distinguish between the following entities:

node — it is an entity capable of communicating and computing. It may be mobile (mobile node) or stable (mesh router).

end-point — it is a logical communicating entity corresponding to a single node.

end-point identifier (EID)—it is a short binary identifier for an end-point. It is persistent, i.e. it does not change when a node changes its position.

name — it is a human readable unique identifier associated with an EID.

address — it is a locator: a short binary identifier used for locating an end-point in the mesh network so that routers can forward packets to it. The address may change when a node changes its position and its network interface.

community — it is a group of nodes that share a common trust relationship.

Figure 1 presents the architecture of the wireless mesh. Mesh routers form the interconnection infrastructure for mobile nodes moving inside and getting connectivity from the nearest mesh router. The mesh network runs three core services: the *naming service*, the *location service*, and the *community service* (this last one is not presented in the figure). The naming service provides the mapping between the name and the EID of a node like in the standard DNS service. The location service handles the information about the mapping between EIDs and addresses. Our approach to mobility is based on the assumption that an address reflects the position of a mesh router and of any other mobile nodes handled by the router. When a mobile node joins the mesh, it establishes a relation with a mesh router that knows its EID. The mesh router updates the location service with the mapping between its address and the EID of the mobile node. When another mobile node wants to send packets to the remote mobile node, it first resolves its name to get the corresponding EID (operation 1) and sends the packet with the destination EID to the neighbor mesh router (operation 2), which in turn locates the address of the destination mesh router (operation 3). Then, the packet takes a route to the destination address established by the ballistic routing described later (operation 4). Finally, the destination mesh router forwards the packet to the destination mobile node (operation 5). So, the communicating nodes only see forwarding packets between their respective EIDs and mesh routers make use of addresses to actually forward packets to the destination.

As EIDs are persistent, the transport layer can use them across different network interfaces and across different locations to maintain long-lived transport connections in spite of mobility. We can construct EIDs in several ways: we can derive them from a public key like in HIP [15] or we can take into account a person that uses a node and derive EIDs from the public key of the node user. A node may have multiple network interfaces, but a single EID.

The community service manages community membership and authentication information needed for a mobile node to join the mesh. Inside a community, communication is simplified and similar to the traditional view of a LAN network or a VPN. We propose to manage all core services as distributed hash tables (DHT). The idea is to build upon all good properties of peer-to-peer (P2P) systems such as scalability and the absence of a centrally coordinated service. Even if we simplify our figure to represent services as something centralized, they are in fact distributed over all mesh routers.

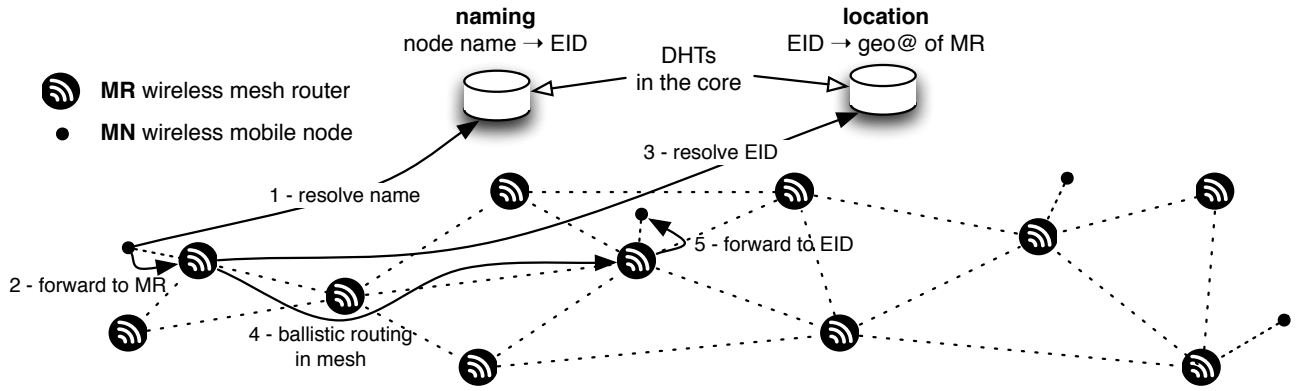

Figure 1: Wireless mesh architecture.

3. ADDRESSING SPACE

We want to define a *coordinate address space* used for assigning addresses to EIDs in an autonomous and a distributed way. Such a coordinate space may be either virtual or real one based for example on the GPS position of a node. Many proposals considered addressing and routing in virtual coordinate spaces [2, 11, 16]. Their advantage is a relative ease of generation compared to the requirement of GPS positioning. However, they present several drawbacks. The first one is related to merging two subnetworks. Imagine that two regions of the networks develop in a independent and unplanned way. Each part creates its addresses in a given portion of the virtual subspace (for instance, P2P approaches for constructing a virtual space such as CAN can be used). The problem is how we can merge two parts when we place a mesh router interconnecting two parts. One part needs to change its addresses to accommodate for the addresses of the other part. We can also face address clashes if parts allocate addresses from intersecting portions of the coordinate space. The second problem is that if we want to handle mobility, the distance in the addressing space needs to be correlated with the real world distance, because nodes physically moves in the real space. When the distances in the addressing space reflect real movements, the most common mobility of a node will be in its neighborhood, so changing addresses will only be limited to this part of space.

If addresses are derived from geographical positions of nodes, the problem of merging does not exist, because nodes in different locations use different addresses. However, this approach requires the knowledge of the exact positions of all nodes, which may be too difficult or too expensive to obtain. We propose to build a *topologically consistent addressing space* by only requiring that a small portion of nodes (e.g. of the order of 10%) know their exact geographical positions. We can easily achieve this, if some roof routers operate GPS devices and a few others, for instance indoors, learn their positions through manual configuration at deployment. All other routers estimate their relative positions to form a global topologically consistent addressing space. The resulting addressing space is *pseudo-geographical* in the sense that the coordinate space is virtual and relative, but anchored in the real world through the exact geographical positions of some nodes.

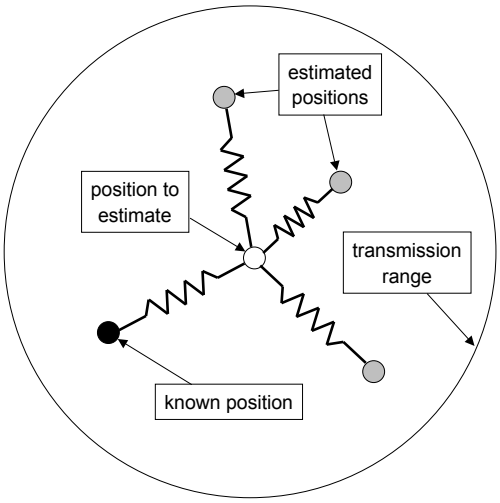

Figure 2: Bond model for positioning.

We propose an algorithm for address allocation based on a local energy function. It can allocate a new address to a node joining the mesh network or if needed, find new addresses of all nodes in case of network renumbering. We assume that a certain percentage of nodes know their exact geographical positions from GPS devices or by assignment.

A node that wants to find its coordinates considers a local potential function depending only on the relative positions of all its neighbors:

$$V_n = \sum_i f(\vec{x_n}, \vec{x_i}), \tag{1}$$

where V_n denotes the local potential value of the n-th node in the mesh located at the geographic position $\vec{x_n}$ with $\vec{x_i}$ being the coordinates of neighbor nodes in the radio range and $f(\vec{x_n}, \vec{x_i})$ being a pairwise potential function. The node needs to minimize the potential function V_n to find the optimal position $\vec{x_n}$ that becomes its coordinate.

We have implemented the method with a fast and easy to implement Nelder-Mead simplex minimizer and tested several potential functions. To evaluate their performance, we

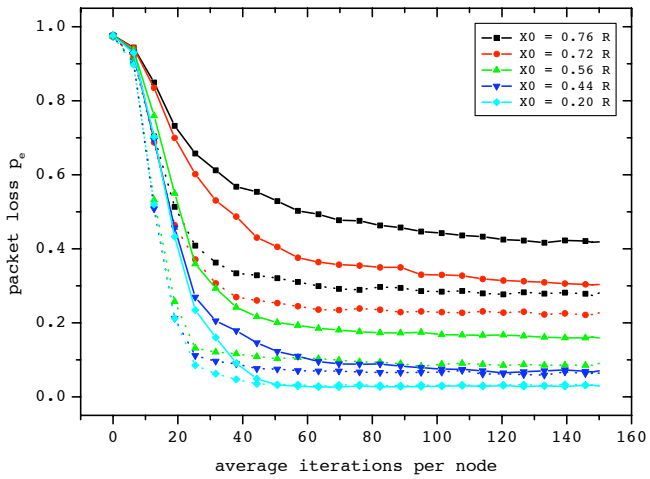

Figure 3: Elastic springs model, convergence speed ($\bar{n}$=11.5), p(GPS)=10% (solid line), p(GPS)=15% (dotted line).

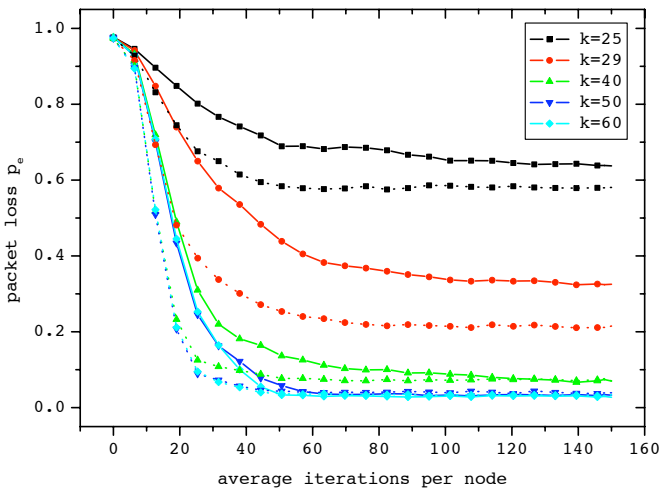

Figure 4: Elastic bond model with $exp()$ core, convergence speed ($\bar{n}$=11.5), p(GPS)=10% (solid line), p(GPS)=15% (dotted line).

have simulated a wireless mesh with all coordinates generated according to the proposed method and observed the convergence speed of the algorithm and the packet loss rate of geographical greedy routing. Figure 3 shows these indices for a given percentage of nodes with the exact positions and a simple elastic-type spring potential model given by:

$$V_n = \sum_i (||\vec{x_n} - \vec{x_i}|| - X_0)^2, \qquad (2)$$

where $||\cdot||$ denotes the Euclidean distance and X_0 is a parameter with a value between 0 and R defining the pairwise distance for which the potential function $f()$ becomes repulsive. $\bar{n}$ denotes the mean number of neighbors. We have found that it is crucial for the convergence of our algorithm that $f()$ contains a repulsive term for $||\vec{x_n} - \vec{x_i}|| \to 0$, otherwise we do not observe the convergence of packet loss rate for an increasing number of minimization steps.

We have observed that the convergence speed of the algorithm is sensitive to the amount of nodes with the exact position in the mesh—more such nodes means of course faster convergence, but at the same time, the algorithm is even more sensitive to the choice of X_0: in an example simulation, we have observed that the choice of $X_0 = 0.1R \ldots 0.40R$ gives very similar results, but larger values of X_0 have quickly a negative impact on the convergence speed. Thus, we can choose X_0 from a fairly large interval of values, but it should not exceed some threshold.

We have also examined a different potential function with an exponential part defined as:

$$V_n = \sum_i \exp\left(\frac{1}{\kappa \cdot (||\vec{x_n} - \vec{x_i}|| + \epsilon)}\right) + \kappa \cdot (||\vec{x_n} - \vec{x_i}||)^2, \qquad (3)$$

where κ is a strength parameter and ϵ an arbitrary small constant. Parameter κ defines the strength of the repulsive term in relation to the elastic energy term. We obtain similar results using this potential function (cf. Figure 4). We can conclude that the exact form of the repulsive term is not essential for the convergence of the algorithm, but the repulsive term needs to start dominating the pairwise potential at an appropriate distance around $\sim 0.25R$ from the entering node. We have also observed that for a given percentage of nodes with the exact positions and a suitable value of the potential parameter (e.g. $\kappa = 60, X_0 = 0.2R$) the exact shape of the potential function has virtually no impact on the convergence speed.

Several authors have already considered the problem of relative positioning. In the absence of GPS, for instance indoors, the location information can be obtained from relative positioning based on estimation of the signal strength [2]. In another approach, Rao et al. have proposed a scalable coordinate-based routing algorithm that does not rely on location information [16], but requires the knowledge of the location of perimeter nodes. Moreover, the approach based on the projection of coordinates on a circle with origin at the center of gravity of the perimeter nodes does not converge if the number of relaxation steps tends to infinity.

4. ROUTING

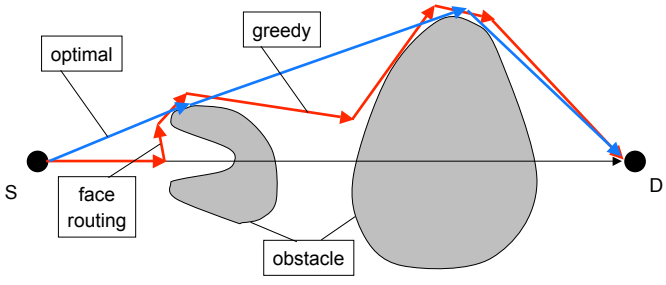

Figure 5: Greedy routing vs. optimal trajectory.

Routing in our pseudo-geographical addressing space extends existing greedy geographical approaches. Pure greedy geographical routing presents several drawbacks that we want to avoid [1,6,7,12]. For instance, when routing fails at topological defects such as voids or concave regions, a backtrack

method such as *face routing* should be used. Topological defects can be detected at concave nodes, which has only neighbors in the backward direction. Face routing can circumvent topological defects using a right hand rule, but requires the construction of a planar graph of links between neighbor nodes. Several recent papers reported on the difficulty of constructing planar graphs in real wireless environments, a non planar graph leading to a significant probability of packet loss [8–10, 13, 14]. Figure 5 illustrates the drawback of greedy routing compared to an optimal trajectory—for a given topology with obstacles, the greedy routing coupled with a backtrack algorithm for circumventing obstacles is much longer than the optimal route.

We propose a two-tier routing architecture: long distance directional geographical routing coupled with short range topological routing in a limited neighborhood of a node. We describe this architecture below.

4.1 Ballistic geographical routing

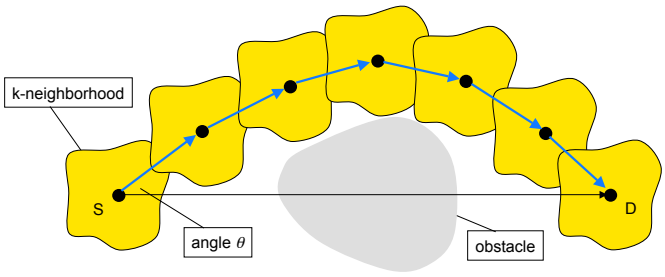

Figure 6: Ballistic geographical routing.

We propose an approach to reach far destinations that we call *ballistic geographical routing*. Instead of greedy geographical routing, we define directional routing based on an angular trajectory to long distance destinations (cf. Figure 6). A node forwards packets along a trajectory that forms an angle with respect to the direction towards a destination. The name for this routing comes from the fact that it is similar to the behavior of a thrown object at a given launch angle that moves under the influence of a gravitational force field and falls at some remote place. The angular trajectory is known globally and implicitly by only providing a rough direction to a destination: to go to destination D, take direction θ. The right trajectory is constructed and adapted based on the information from neighbor areas so to get around obstacles (topological defects such as voids or congested areas that should be avoided).

4.2 Topological routing in k-neighborhoods

Long-distance geographical routing is not enough. We couple it with *topological* routing in the *k-neighborhood*, a closed neighborhood of a node limited to k hops. There is one k-neighborhood per node, i.e. the notion of k-neighborhood is local to a node and do not rely on any kind of a clustering algorithm to create disjoint neighborhoods. Topological routing in a k-neighborhood operates over short distances and relies on a route known locally and explicitly based on the precise topology of k-hop neighbors.

Figure 7 presents the way in which we integrate the two types of routing. When forwarding a packet to a given remote destination, a node gets the direction to use from the

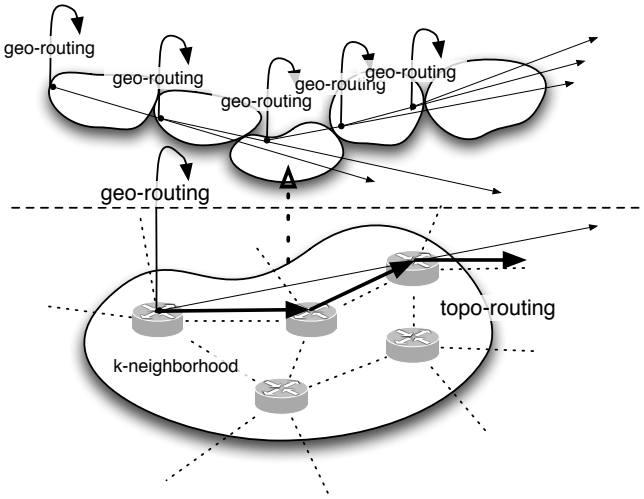

Figure 7: Integration of ballistic geographical routing with topological routing in k-neighborhoods.

ballistic geographical routing and finds the farthest node within its k-neighborhoods in the direction to follow. Then, it forwards it to the farthest node through some intermediate nodes in a LAN-like manner. Upon receiving the packet, the farthest node uses the ballistic geographical routing to find the next right direction. The whole process repeats for each k-neighborhood of the farthest nodes. In this way, a node has something like a fish-eye view of the network: detailed view on a short distance (a kind of a local subnet with direct packet forwarding) and rough directions towards long distance destinations. When a packet travels towards a destination, it benefits at each router from the precise view of the router and its rough direction to the destination.

For topological routing in k-neighborhoods, we can use any already proposed routing protocol for small scale ad hoc networks such as those proposed within the 802.11s standard: RM-AODV (Radio-Metric Ad hoc Distance Vector), RA-OLSR (Radio-Aware Optimized Link State Routing), or FSR (Fisheye State Routing), as well as other suitable protocols such as OFLSR (Optimized Fisheye Link State Routing).

4.3 Size of the k-neighborhood

An important operational parameter is the size of k-neighborhoods—what is the right value for k? We have run some simulation studies of a randomly created wireless mesh under the assumption of the unit disk graph: we place N nodes randomly distributed in a circular arena, each node being only able to communicate with its neighbors within a transmission radius. Under these assumptions, the resulting distribution of the node rank (the number of its neighbors) converges to the normal (Gaussian) distribution. Even if the assumptions are too simplistic for real wireless environments, this analysis gives us a first-order insight into macroscopic characteristics of large scale mesh networks. We have observed the behavior of packet forwarding under greedy geographical routing enhanced with k-neighborhoods: a node forwards a packet to the farthest node belonging to its k-neighborhood that makes the best progress towards the destination (according to the Euclidean metric). Inside a k-

neighborhood, a node forwards packets to the farthest node using greedy distance routing. In this experiment, packets may fail to reach the destination because of topological defects.

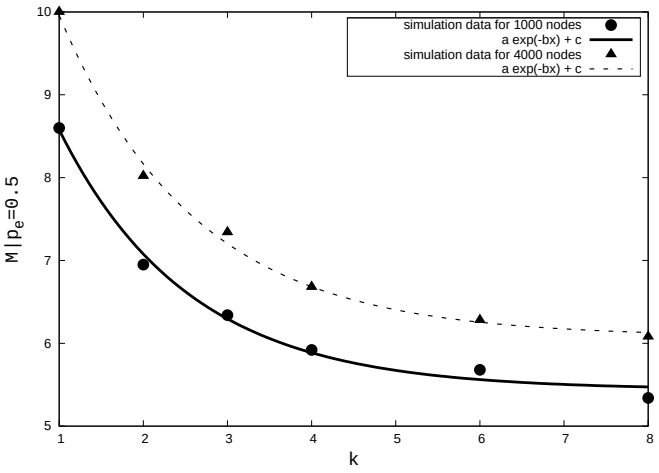

Figure 8: Critical average node rank for the increasing size of the k-neighborhood.

Figure 8 presents the critical average node rank M in function of k, the size of the k-neighborhood for $N = 1000$ and 4000. Critical means that it corresponds to the rank for which the packet loss probability p_e becomes 0.5. The figure shows experimental data obtained from simulation and their fit with an exponential function of type $M = a \exp(-bk) + c$. If the graph becomes dense (high average node rank), there are less topological defects and the packet loss probability tends to zero. We can see that the critical average node rank is lower for increasing values of k: when $k \to \infty$, the k-neighborhood covers the whole network and its topology is known. In this case, a packet reaches the destination as long as a route exists. We can also observe that the gain obtained by increasing the size of the k-neighborhood fades exponentially and the choice of a suitable value of k depends on the size of the network, for example the right choice for $N = 1000$ would be $k = 4$ and $k = 6$ for $N = 4000$.

4.4 Spatial prefix

To support the long-range ballistic geographical routing, we assume that each node maintains a routing table giving for each *spatial prefix* the right direction θ. A spatial prefix defines all destinations within some range, for example we can define prefix `D/x` as all nodes around position D with range $r = d \cdot 2^{-x}$, where d is a coefficient that reflects the mapping to real world distances and to the resolution of the addressing space. The prefix size of $x = 0$ means the whole universe—the default route. We expect that the number of prefixes stays low similarly to the behavior of routing tables at edge subnetworks in the current Internet—they only contain a small number of entries allowing for local routing and other destinations are aggregated in a default route. In our case, a node will maintain some local spatial prefixes and aggregate far away destinations into short prefixes corresponding for example to the directions to gateways with the current Internet. Thus, the proposed ballistic routing presents the advantage of keeping little state and limiting the size of routing tables due to easy spatial aggregation of prefixes.

4.5 Direction adaptation

The information in spatial routing tables needs to reflect the current knowledge of obstacles in the network so that packets take routes to avoid them. However, we still keep our fish-eye view of having only a rough indication on how to get to remote destinations and knowing the exact topology of the k-neighborhood. Ballistic routing thus requires a protocol for detecting obstacles and disseminating the information about right directions. It can also provide some indication on resource usage along some directions to allow spreading traffic towards remote destinations over different routes. We have some initial thoughts about such adaptation and we work on the design of a required protocol.

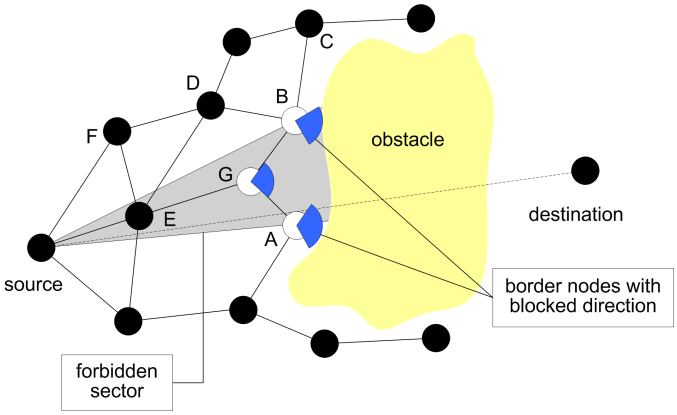

Figure 9: Detecting blocked nodes on the border of an obstacle.

A first idea that we explore concerns a reactive protocol for detecting obstacles. Reactive refers here to the strategy in which traffic towards a given destination initiates updating the information on available or impossible directions. When the state of the network changes, this should also force updates to the routing information. We assume that at the beginning the wireless mesh forwards packets to a destination like in greedy geographical routing. A node that receives a packet and cannot find any next-hop node considers itself as a *blocked node* for the destination—there is an obstacle in its direction. It advertises the blocked direction in its k-neighborhood and backtracks the packet to the previous hop that removes the blocked node from its list of possible next-hops for the destination. If the list is empty, the previous hop becomes blocked too. The operation repeats this way until a node finds the right direction or the packet fails to reach the destination.

Let us consider the example in Figure 9 with a packet going through nodes E and G to B. When node B detects its blocked situation, it advertises the blocked direction in its k-neighborhood and backtracks the packet to G, which tries to go through A. This last node does the same and G also declares itself blocked based on the failure of its next-hop nodes. Trying different directions results in the discovery of blocked nodes on the border of an obstacle and of the *forbidden sector* containing blocked nodes (the cone in the figure with the source and nodes E, G, A, B), for instance node E learns in this way that it should avoid forwarding through

node G. Blocked nodes can also advertise this situation beyond their k-neighborhood in a message towards the source so that the nodes in the forbidden sector learn about the directions to avoid. In this way, after the learning period the network acquires some knowledge of the right directions to use.

Note that such advertisement and adaptation can take into account not only topological defects such as voids, but also congested areas or other regions that needs to be avoided.

5. MOBILITY MANAGEMENT

Mobility management builds upon the principles and elements of the architecture described in the previous sections: separation between EIDs and addresses, location service, pseudo-geographical address space, geographical ballistic routing, topological routing in k-neighborhood. Separating EIDs from addresses means that a mobile node always uses its stable identifier to send and receive packets wherever it is located in the wireless mesh. Packet forwarding in the mesh involves addresses of mesh routers that translate to EIDs of mobile nodes. Such a scheme makes communication of mobile nodes independent of their actual positions in the mesh. Moreover, it guarantees anonymity and privacy, because mobile nodes do not know the addresses of their correspondent nodes and they may only manipulate EIDs. Obviously, these properties rely on trusted mesh routers, but we think that such a requirement is easily achievable. Below we explain how to put all the pieces of the puzzle together to achieve seamless and efficient mobility.

5.1 Joining the mesh

An efficient mobility management scheme requires a lightweight process of joining the mesh. We assume that each mesh router has an address in the pseudo-geographical space constructed as described above and neighbor routers know each other within their k-neighborhood. They also know the mobile nodes identified by their EID and associated with all neighbor mesh routers. In addition to that, they cooperate and exchange all the information needed to communicate at layer 2 over an 802.11 type of wireless links: channels to use, load of each mesh router, public keys for encrypting packets, and MAC addresses. The information is merged and broadcast in periodic beacons.

A mobile node that wants to joint the mesh takes advantage of this information to choose a suitable mesh router as its network point of attachment. At this instant, the mobile node can send packets to other nodes via the mesh router with some minimal properties and rights (e.g. basic quality of service). When the mesh router receives a packet to forward, it updates the location service by storing the binding *mobile node EID–mesh router address*. This means that the packets with the EID as the destination address will in fact go to the address of the router. After this update, the mobile node can receive packets sent to its EID. This first communication possibility is at the basis of the lowest level *connectivity community*.

This scheme for association with a mesh router may require modifying the 802.11 way of operation in the infrastructure mode. Currently, it includes the phases of scanning and authentication for initializing bridge operation and enforcing security at layer 2, which takes a significant amount of time. Our goal is to prepare and disseminate all needed association information in advance to provide basic connectivity.

After joining the mesh, the mobile node can benefit from other more sophisticated properties and rights (e.g. real-time quality of service, requesting particular routes, support for end-to-end security etc.) by explicitly joining other logical communities. To do this, it can send an explicit request for `joining` another community, a logical one. The mesh router queries the *community service* to check if it may accept the node in a given community. Joining a community requires authentication to benefit from the services offered by the community.

5.2 Handoff

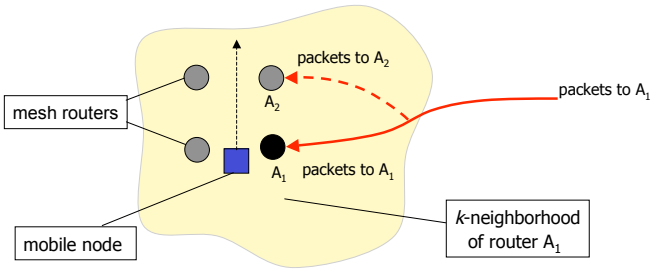

Figure 10: Handoff.

When a mobile node moves, it can prepare to change mesh routers based on the information about the layer 2 characteristics of neighbor routers, for example choose a lightly loaded one. As the mobile node always uses the same EID independently of its position, it is up to mesh routers to perform all the required modifications during handoff, in particular, update the location service with the address of the new mesh router that will handle the mobile node after handoff.

Handoff performance depends on the update rate of the location service—this rate should be kept low for an efficient handoff. We can achieve this objective by updating the location service in a lazy manner when a mobile node moves inside a k-neighborhood. As the mobile node moves in the real geographical space, the address of the next mesh router will be topologically close to the address of the previous one, the proximity depending on the distance and the speed of the movement. Imagine that the mobile node associated with router A_1 moves near router A_2 (cf. Figure 10). It chooses A_2 based on the available information and sends a `handoff` request to its current router A_1 that relays it to router A_2. Router A_2 registers the association with the mobile node and starts forwarding its packets. At the beginning, router A_1 can forward packets destined for the mobile host to A_2. At the same time, it will disseminate the information about the handoff in the k-neighborhood so that a mesh router at the border of the k-neighborhood can divert packets to the new address as shown in the figure. In this way, packets travel through the optimal route in the k-neighborhood towards the mobile node. Such a scheme does not require any notification of corresponding nodes about changed addresses, because they still send packets to the address of the previous mesh router A_1.

At some time however, router A_1 needs to update the location service. It can make the decision based on the pro-

gress of the mobile node, for instance when it approaches the border of the k-neighborhood. At that instant, router A_1 updates the EID binding with the address of the current mesh router with which the mobile node is associated and notifies the mesh routers of corresponding nodes about the change in the destination adress. In this way, packets sent by corresponding nodes start coming to the current mesh router.

Handoff to another router in a given k-neighborhood does not require querying the community service, because the mobile node has already joined the network.

6. IMPLEMENTATION

We have begin to develop a prototype of the proposed architecture. We use Linux IPv6 as a substrate for the development. We wanted to integrate geographical addresses with the standard ones so that both may coexist. A special prefix denotes geographical addresses and the TUN interface (virtual network card) encapsulates packets with static addresses (eg. `2001::`) into packets with geographical ones (eg. `2002::`). Packets with geographical addresses are redirected inside the kernel in the prerouting table to pass by the user-space queue module. Then, an application obtains a user-space copy of the packet from the geographical address space and decides of the next-hop by setting via a special function of `libipq` an appropriate `NF-Mark` in the in-kernel copy. This is done only if the destination is other than the local node in the geographical address space—the packet is decapsulated and sent back to the TUN device. In the kernel, we use routing rules (in the rule table) according to the `NF-Mark` set in the packet to select a routing table containing just one entry: a direct neighbor as the default gateway. Thus, we choose by means of NF-Mark which routing table to use and the routing table selects one of direct neighbors. The implementation required a slight modification of the kernel—5 lines of code.

7. CONCLUSION

In this paper, we have described our ideas on an addressing and routing architecture for seamless mobility in spontaneous wireless mesh networks. The main contributions include: the concept of the pseudo-geographical addressing space, a new approach to geographical routing based on ballistic trajectories, a location service based on DHT, and seamless mobility management through a transparent handoff in a k-neighborhood.

We propose this paper to the workshop for discussion and early feedback. We are aware that the ideas presented below are only paper design, however we are working on more extensive validation and implementation—we develop an operational prototype of a mesh router that uses the proposed addressing space, ballistic routing, core services, and supports handoff of mobile nodes. We also experiment with 802.11 cards with a modified access method suitable for multi-hop forwarding and we develop a lightweight protocol for joining the mesh.

Acknowledgments

This work was partially supported by the European Commission project WIP under contract 27402, the French Ministry of Research projects AIRNET under contract ANR-05-RNRT-012-01 and ARESA under contract ANR-05-RNRT-01703.

8. REFERENCES

[1] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with Guaranteed Delivery in Ad Hoc Wireless Networks. In *Proc. DIALM*, Seattle, USA, August 1999.

[2] S. Capkun, M. Hamdi, and J. P. Hubaux. GPS-Free Positioning in Mobile Ad-Hoc Networks. *Cluster Computing*, 5, April 2002.

[3] I. Castineyra, N. Chiappa, and M. Steenstrup. The Nimrod Routing Architecture. RFC 1992, 1996.

[4] M. Crawford, A. Mankin, T. Narten, I. Stewart, and L. Zhang. Separating Identifiers and Locators in Addresses: an Analysis of the GSE Proposal for IPv6. Internet draft, 1999.

[5] P. Francis. Addressing in Internetwork Protocols. PhD thesis, University College London, UK., 1994.

[6] H. Frey. Scalable Geographic Routing Algorithms for Wireless Ad-Hoc Networks. *IEEE Network*, July/August 2004.

[7] B. Karp and H. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proc. MOBICOM*, Boston, USA, August 2000.

[8] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. Geographic Routing Made Practical. In *Proc. NSDI*, pages 112–124, 2005.

[9] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. On the Pitfalls of Geographic Face Routing. In *Proc. DIALM-POMC '05*, 2005.

[10] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. Lazy Cross-Link Removal for Geographic Routing. In *Proc. SENSYS*, 2006.

[11] R. Kleinberg. Geographic Routing Using Hyperbolic Space. In *Proc. INFOCOM*, 2007.

[12] E. Kranakis, H. Singh, and J. Urrutia. Compass Routing on Geometric Networks. In *Proceedings of the 11th Canadian Conference on Computational Geometry*, 1999.

[13] B. Leong, B. Liskov, and R. Morris. Geographic Routing Without Planarization. In *Proc. NSDI*, 2006.

[14] B. Leong, S. Mitra, and B. Liskov. Path Vector Face Routing: Geographic Routing with Local Face Information. In *Proc. 13th IEEE ICNP*, 2005.

[15] R. Moskowitz and P. Nikander. Host Identity Protocol (HIP) Architecture. RFC 4423., 2006.

[16] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic Routing without Location Information. In *Proc. MOBICOM*, 2003.

[17] J. Saltzer. On the Naming and Binding of Network Destinations. RFC 1498, 1993.