Production Systems and Information Engineering
Volume 2 (2004), pp. 143-158

SCALABLE ADDRESSING AND ROUTING IN LARGE-SCALE WIRELESS NETWORKS

GERGELY BICZÓK
Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics
biczok@tmit.bme.hu

NORBERT ÉGI
Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics
egi@tmit.bme.hu

PÉTER FODOR
Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics
fodorp@tmit.bme.hu

BALÁZS KOVÁCS
Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics
kovacsb@tmit.bme.hu

ROLLAND VIDA
Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics
vida@tmit.bme.hu

[Received December 2004 and accepted February 2005]

Abstract. One of the main characteristics of future networks will be the considerable increase in the number of communicating entities; PCs and laptops, but also PDAs, 3G phones, sensors, wearable devices, etc., will all connect to and communicate with each other, to form large-scale intelligent networks, integrating different technologies. On the other hand, there will be a considerable shift from fixed, wired devices to mobile, wireless ones. In these conditions, the current IP based mechanisms are foreseen not to fit these new network structures. Therefore, alternative addressing and routing solutions are needed to handle these large and highly mobile, dynamic, wireless networks. In this paper we propose a new, scalable approach to the addressing and routing problems that emerge in the context of the envisioned future network architecture.

Keywords: stateless routing, addressing, mobility, scalability, wireless communication

1. INTRODUCTION

Researchers in the field of networking have to continuously monitor the current trends related to the evolution of technologies, the changing user requirements or the emergence of new services, in order to design efficient new solutions that take into account these tendencies. In some cases, small modifications of existing approaches are enough to adapt to the evolving requirements. On the other hand, from time to time radical paradigm changes are needed. Currently there are several factors that indicate that such a radical change is inevitable, if we are to cope with the foreseen evolution of future networks.

One of the most important issues that will have to be handled in these networks is the significant increase of the number of communicating entities. We do not refer here to traditional, fix-installed nodes, but rather to new generation cell phones, wirelessly communicating notebooks, PDAs, personal identification cards, sensors, wearable devices etc. Integrating all these entities into a common networking infrastructure will generate serious scalability concerns regarding the current IP addressing mechanism and the routing schemes that are based on it.

Besides these scalability concerns that are due to the increasing number of communicating devices, it is important to note the associated shift of the ratio of mobile and fixed entities that will be involved in future network architectures. The main part of today's Internet is composed of fixed, wired devices, while wireless, mobile entities appear only at the periphery of the network. Current technologies, such as Mobile IP for example, are able to adequately handle small-scale mobility that appears at the periphery; nevertheless, they are not able to provide scalable and efficient mobility and addressing support, should the before-mentioned radical ratio change become a reality.

On the other hand, if mobility appears not only at the periphery of the network, a re-thinking of routing mechanisms will be needed. The current routing protocols were mainly designed to highly static, wired environments, where communication paths are broken rarely, and mainly due to errors, such as node failures or congestion. Therefore, they react very slowly to network changes, with long convergence times. As opposed to this, in a dynamic environment, where mobile hosts might reach the network through other mobile nodes and communication paths have to be continuously reshaped due to the enhanced mobility of the participants, it is hard to see these routing protocols performing well.

In order to underline these tendencies, let us provide now some concrete examples of future applications and communication environments. On the one hand, let us consider the case of a shopping mall, or an airport, where there are thousand of people that might have tens of thousands of wireless electronic devices able to communicate with each other. These devices, together with embedded sensors or other intelligent equipments, will group together in large and highly dynamic,

mobile ad-hoc networks. Similarly, let us consider the case of a highway, with sensors embedded in the road or the traffic signs, and with cars equipped with different wireless communication devices. In such an environment, a road sensor can alert the driver of a car about icy road conditions, information which can be then relayed through the ad-hoc wireless network to the other cars approaching. In all these cases nodes are highly mobile, the network has a rather large size, and the traditional addressing and routing techniques seem to be unable to ensure an efficient operation.

Thus, according to the tendencies that are to shape the networks of the future, new, scalable addressing and routing mechanisms are needed to provide continuous and consistent connectivity in a large and highly mobile network that uses different telecommunication technologies and devices.

In our paper we present a new, scalable addressing and routing architecture, designed so as to cope with these requirements. The rest of this paper is organized as follows. First, in section II, we present the main components of the proposed architecture and describe the corresponding addressing schemes. Then, in section III, we explain the node lookup and routing mechanisms. Section IV analyzes the advantages and drawbacks of our novel system, while section V presents some related work that our proposal can be compared with. Finally, we describe some future directions we intend to follow, and draw conclusions.

2. THE ARCHITECTURE

The goal of this paper is to propose an addressing and routing solution that fits the envisioned future network architecture. Our aim was not to develop another fully ad-hoc routing protocol, but to design an efficient and scalable solution that exploits the fact that in future networks, fixed network infrastructure will be available to support mobile and ad hoc wireless communication. In the following we briefly present the components of the architecture.

2.1. Components

There are three main architecture components in our approach. Wireless domains (WD) are meant to be ad hoc networks that include a fixed access point to the core network; these access points are called domain edges. Besides these edges, WDs are composed of nodes, which connect to the core network either directly, through the access points, or indirectly, through other nodes. WDs are radio networks featuring a wireless broadcast medium, which can be well utilized in inter-neighbor communication instead of sequence of unicast messages. Domain edges communicate with each other on a wired core network. In this paper, we do not focus on the core network architecture; we only assume that it supports the

necessary addressing and routing primitives to enable communication among its components.

2.2. Addressing

Every wireless domain has a fixed domain ID in the network. Each node has a unique identifier, called Global Node Identifier (GNI), which is fixed as well. Moreover, each node has a Local Node Address (LNA), which reflects the position of the node in a domain. Inside a domain, nodes create a parent child hierarchy which is based on the hop distance from the domain edge. The LNA is a chain of numbers that correspond to the parent nodes of a certain node. For example, the parent of a node with an LNA of 10.2.4.5 has an LNA of 10.2.4, his parent has an LNA of 10.2, and so on. The addresses can be structured into a tree that is derived from the network graph. The vertices of the graph are nodes in the network, while an edge represents a parent-child relation between two nodes.

When a node arrives in a domain, first it broadcasts an address query message, to select its parent node. Each node that hears the query proposes an address to the newcomer. This address is composed from the LNA of the responding parent, extended with an arbitrary identifier, which differentiates the requesting new child from the other children of the parent node. From the received addresses the node selects its new address and therefore its parent (Figure 1).

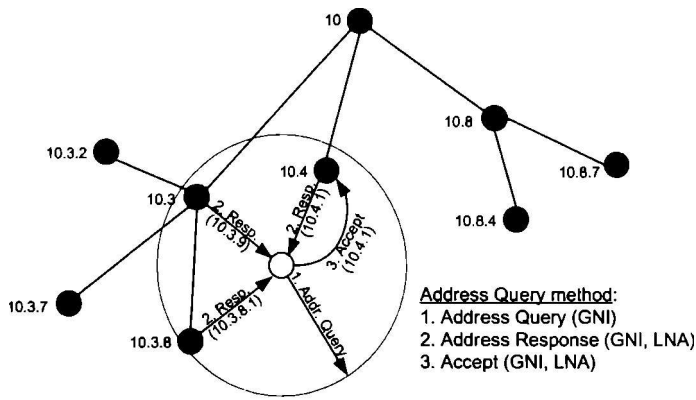

Figure 1: The address request procedure

The joining node always chooses the shortest address; by doing so, it tries to be as near as possible to the domain edge in the address tree. If the node receives several addresses with the same prefix length, it chooses the parent with the least children. This mechanism ensures a balanced address tree, which is important to achieve an efficient routing.

3. THE ROUTING MECHANISM

Routing in our scheme is a process composed of several steps. Initially, the source node knows only the GNI of the destination. In order to be able to send packets, the source has to find out the other addresses (domain ID and LNA) that characterize the destination's current location. Then, routing can be completed. In the following we present the details of this process.

In the core network there is a distributed Global Node Locator (GNL) called Location Agency; it stores (GNI, domain ID) address pairs for each node in the network. These address pairs reflect a node's current domain location. The address lookup process is based on an algorithm that uses distributed hash tables (DHT); it stores GNIs and location related domain IDs in a distributed manner. The domain IDs are stored in so-called Location Agents (LA). Each Location Agent is responsible for a given interval from the global value space of the hash function; as such, it stores information (i.e., the current domain ID) about all nodes whose GNI hashes into its interval. These Location Agents form the distributed Location Agency.

As opposed to this, Domain Edges have a Local Address Directory function (LAD); they store (GNI, LNA) address pairs for every node in their wireless domain. The local address lookup function provided by a Domain Edge and the distributed global node locator function provided by a Location Agent are separate functionalities from the logical point of view. However, they can be physically collocated on the same machine. Therefore, in order to simplify our architecture, we consider in the following that a Domain Edge performs both functions.

Nodes in a domain maintain a Neighbor List (NL), where (GNI, LNA) address pairs are stored for neighbors within a given range. The Neighbor List is created and updated by periodical Hello messages exchanged among neighbors. Besides the direct neighbors, a node can store information related to nodes that are further away; these information can be exchanged by piggybacking them on the Hello messages. The range of the Neighbor List might vary, depending on the needs and the storing capabilities of the different nodes. The mandatory range of the list for the algorithm to operate correctly is one hop.

3.1. Address lookup

When a node wants to send a packet, it has to determine the current address of the target node from its GNI. First, it looks into its NL; if it cannot find the LNA of the destination, it initiates an address lookup message to its Domain Edge (Figure 2). In case of an unsuccessful lookup in the Edge's LAD table the edge node asks the global Location Agency about the current domain ID of the target. The address of the Location Agent where the requested domain ID may be stored is retrieved by hashing the GNI of the destination node. The responsible LA tells the domain ID of

target node; the ID is sent back to the source node, and can be used by core network routing to reach the target WD, where the destination node resides. Figure 2 presents the steps of this lookup process.

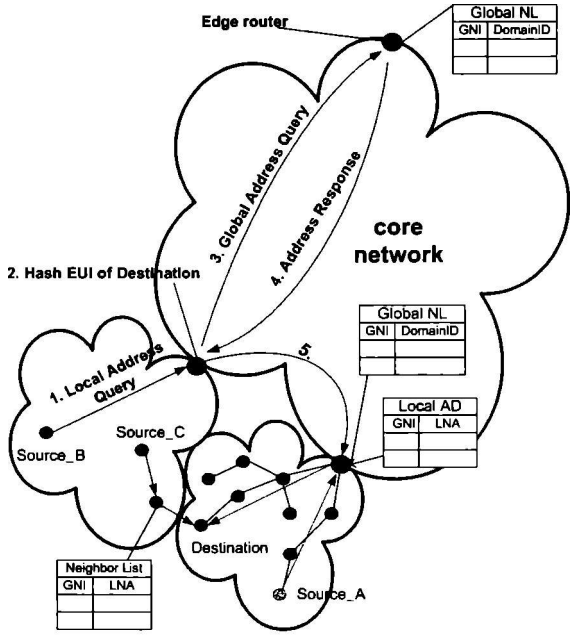

Figure 2: Node lookup and routing

3.2. Routing

According to the location of the source and the target node, intra- and inter domain routing can be differentiated. When intra domain routing is performed (Figure 2, Source_A), the source first looks for the target node's GNI in its neighbor list. If it finds a corresponding entry, it sends its packet directly to the destination. If not, the LNA of the target node can be obtained from the LAD service of the Domain Edge, by sending a LAD query upwards on the LNA tree.

The LAD lookup can be optimized by using the neighbor lists to reduce the number of hops the query has to travel. When initiating or forwarding a LAD lookup, a node looks into its NL to find an intra-domain neighbor that is closer to the domain edge (i.e., has a shorter LNA) than its own parent. If there is such a node, the lookup request is forwarded to it. As the root of the intra-domain LNA tree is the Domain Edge, it is assured that the request reaches it. In addition, some extra hops might be saved by shortcutting the LNA tree.

If we are to deal with wireless domains that contain a large number of nodes, it might be reasonable to handle the LAD tables in a distributed manner as well, among several local nodes in the domain. Nevertheless, this functionality is not included yet in the current architecture.

If the source knows the LNA of the destination node, it searches for a prefix match between the target LNA and an LNA in its neighbor list. In case of success, the packet is sent to this neighbor. By doing so, a much shorter route can be found than the route obtained from the local address tree.

If there is no address prefix match, the packet is sent to the parent node. Each node obtains the LNA of its parent by simply cutting off the last component of its own LNA; no routing table has to be maintained.

In the worst case, this mechanism continues until the packet reaches the Domain Edge; then, the packet is sent through the parents of the target node, until it reaches the destination. Again, no routing table is needed. All the children of a parent node receive the packet. Among them, only the node with the LNA included in the target LNA forwards it.

In inter domain routing (Figure 2, Source_B) the source node knows the domain ID of the target node's current area from the Location Agency. First, the packet is sent to the Domain Edge; then, it is forwarded to the target domain. Here, the Domain Edge determines the target LNA from its LAD table; the packet is then sent down in the address hierarchy to the target node.

In some cases the packet might not have to travel through the core network, even if the communicating parties are situated in different wireless domains. This can happen when the source and the target are near to each other (typically the source is near the border of its domain). In this case, efficient packet forwarding can be obtained by the use of the Neighbor List (Figure 2, Source_C).

3.3. Multiple LNA trees

The above described routing mechanism has the drawback of putting a significant load on upper part of the LNA tree inside a domain; if no shortcut is found in the neighbor lists of the intermediate nodes, an intra-domain packet has to travel all the way up to the Domain Edge, and then down again on the appropriate branch of the LNA tree, towards the destination. Therefore, a significant amount of traffic will be handled by the nodes neighboring the Domain Edge. Another drawback is the lack of robustness of the approach, which pops up in the case of node mobility; if one of the nodes on the LNA tree moves out of the range of its parent (or child) node, the LNA-based routing path is broken.

To alleviate these drawbacks, we propose to use multiple LNA addresses and multiple trees, instead of only one. When entering the wireless domain, a new node

requests LNA addresses on all of these trees. Each neighboring node that receives the query (and that already has an LNA address on all of the trees) proposes to be the parent of the newcomer on all these trees. Then, the new node can select an LNA address for each tree, and reject the other proposals. Note that a node can have LNA addresses of different lengths on the different trees. Therefore, some of its proposals might be attractive to the newcomer, some not. Thus, the new node can choose to have different parents on the different LNA trees.

Once the new node has chosen its addresses on the different LNA trees, an update message is sent to the LAD service; all these addresses will be stored along with the GNI of the node. Therefore, when an address lookup is initiated by a source node, all the LNA addresses of the target will be returned.

Using multiple LNA trees in the same wireless domain has several advantages. On the one hand, when routing a packet, all these trees can be used to find the best path to the target node; an intermediate node that receives a packet on one of the LNA trees can decide to forward it over a different tree, if on that latter tree it has a neighboring node that is close to the target's corresponding LNA. Thus, routing can be accelerated.

On the other hand, robustness is enhanced through the use of multiple trees; if a node looses one of its parents (or child) – e.g., because it moves out from it's communication range – it can still use the other LNA trees to forward the packets. Thus, the routing paths are not broken.

The drawback of having multiple LNA trees is the increased amount of signaling and stored states. Both in the LAD tables and in the neighbor lists nodes have to be stored with several LNAs. Also, creating and maintaining multiple LNA trees requires additional signaling. However, this might represent an acceptable tradeoff if routing efficiency and robustness can be enhanced.

3.4. Alternative Routing

Besides the LNA-based routing mechanism, we also propose an alternative routing scheme that can improve the performance of the approach in several cases. Although the address tree based routing model provides a stateless solution for packet delivery, it suffers of some significant drawbacks. First, it lacks protection against link failures caused by node mobility, which is not affordable in a large-scale, highly dynamic networking environment that we envisaged. Second, in case of intra-domain routing it barely supports the optimal routes, as the majority of the packets are sent upwards to the Domain Edge, before being redistributed along the LNA-based tree branches. Thus, network traffic concentrates near the Domain Edge, which could lead to congestion and packet loss. In these conditions, there is a need for distributing the traffic load, for finding optimal routes and for

eliminating faults that are due to node mobility. Alternative routes could provide an answer to these issues.

The alternative routing solution that we propose can be summarized as a hash based probability routing. Its main goal is to find alternative routes that lead to the destination node, or to one of the address sub-trees that contains the destination node. The basic idea of the algorithm is to split up, at every node, the global value space of a hash function that is applied on the GNIs of the target nodes, among its neighbors. Thus, the number of sub-spaces depends on the number of a node's neighbors; this can be obtained from the node's Neighbor List, and it will vary from node to node. The hash function used for alternative routing can be similar to, or different from the hash function used for the Global Node Locator service.

When a packet arrives to a given node (A), the hash function is used to map the GNI of the destination into a sub-space that is assigned to one of node A's neighbors (node B). Then, B will be the next node to forward the packet from node A. It is important to mention that node B is not aware of the local assignment of sub-spaces done at node A; B will locally split the global value space among its own neighbors, apply the hash function on the destination's GNI, and choose the next hop (node C) to forward the packet according to its own assignment table.

The advantage of the hash based forwarding is that it performs stateless, semi-random packet forwarding, since only the GNI of nodes are taken into account; the packets of the same flow, sent to the same destination node, will be forwarded on the same alternative route, through the same intermediate nodes, without using any routing tables. The LNA, and therefore the current location of the target node, are not taken into account. Thus, when choosing the next hop on the alternative route, there is no guarantee that the packet will get physically closer to the destination. However, this might happen. In order to detect such a case, a node that receives a packet on an alternative route will always check its Neighbor List for the destination GNI; the packet will be further forwarded on the alternative route only if no matching entry is found in the NL. Moreover, the alternative route might cross the LNA-based routing path, in which case the cross point node can choose to stop forwarding on the alternative path, and send the packets down the LNA-branch. To do so, the receiving node first checks whether it is in the destination's LNA branch itself (its own LNA is included in the destination's LNA) or not. The hash-based forwarding is continued only if there's no match in the LNAs.

The two optimization possibilities can be applied together as well. That is, a node on the alternative route might fail in finding the destination node in its Neighbor List, but might discover a neighbor whose LNA is included in the LNA of target node. In that case, that neighbor will be chosen as next hop, without applying the hash function again on the target GNI. The efficiency of this mechanism depends on the range of the Neighbor List; the larger the range, the higher the probability of

finding a node in the neighborhood of the target. On the other hand, there is a tradeoff between the low-state nature of the routing scheme, expressed by the allowed size of the Neighbor List, and the possible optimization gains. Simulations in future work will give answers on the positive and negative effects of Neighbor List size.

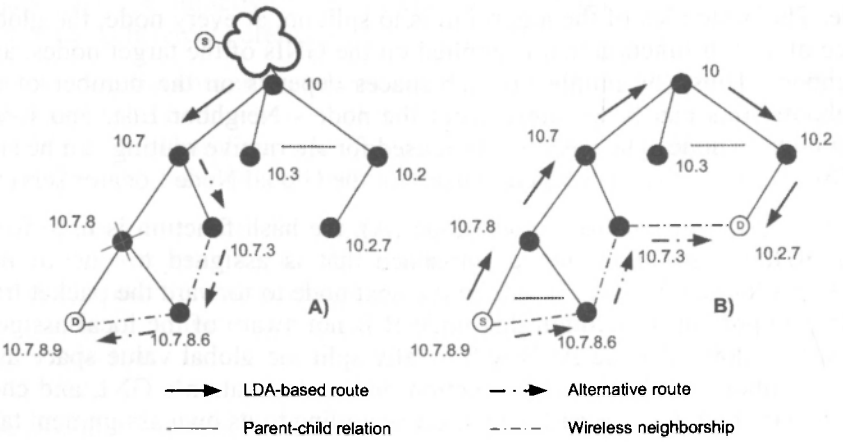

Figure 3: Using alternative routes in case of A) route failure B) route optimization

As stated before, the alternative routing scheme has a semi-random nature, which might result in routes that do not converge towards the destination. In order to limit the length of an alternative route, a TTL (Time To Live) field can be used; when the packet has crossed a number of alternative hops without reaching the target or without crossing the corresponding LNA-branch, it is dropped. Also, the lack of traditional routing tables and associated routing primitives (e.g., reverse path forwarding check) can result in the alternative routes containing loops. They can be avoided by using sequence numbers for the packets; a packet that was already handled by a node will not be processed again.

The alternative routing scheme can be used to handle several possible issues, such as fault tolerance, load balancing, or route optimization. For example, in case of a node failure in the middle of an LNA-branch, due to battery depletion or mobility, alternative routing can be used to get around the failure. This situation can typically arise when routing packets of a source located in a different wireless domain; after entering the destination's domain through the Domain Edge, packets are normally forwarded along the LNA-based tree. Figure 3.A presents an example for this scenario.

Alternative routing can also be applied to optimize routing paths in case of intra-domain communication. The source node starts to send packets along two possible paths (Figure 3.B). One corresponds to the LNA address tree, while the other is an

alternative path obtained through the above described hash routing scheme. In the second case route lookup can be constrained by the TTL field of the packet. The maximum number of alternative hops we allow can be obtained from the lengths of the source and destination addresses. For example, the route from 10.7.8.9 to 10.2.7 will consists of 4 intermediate nodes along the LNA address tree. As the destination is 5 hops away along this tree, alternative routes shorter than 6 hops are desirable.

Even if there are no shorter routes than the one based on the LNA-tree, we could use alternative routes to provide load balancing. Thus, considering the scenario in Figure 3.B we could accept 5–hop-long alternative routes to shift network load off the root nodes.

Since alternative routes continuously change according to node mobility and stateless packet forwarding, routes must be monitored and compared periodically to the reference route obtained from the LNA address tree. Upon deciding on the optimal route between the alternative and the LNA-based paths, packets will be sent only along that optimal route, until the next periodic evaluation.

The proposed alternative routing solution does not ensure shortest-path delivery; packet forwarding is rather semi-randomized, the actual location (LNA) of the target node not being taken into account. However, it provides a stateless solution to support fault tolerance, load balancing, or route optimization, with a certain probability, which depends on the range of the Neighbor List or the density of the neighboring nodes inside a wireless domain. The stateless nature of the approach is due to the hash-based forwarding that avoids the use of routing tables.

3.5. Mobility Handling

Since future networks are thought to be highly mobile, efficient mobility handling is a key issue in novel routing mechanisms.

There are a number of issues that should be addressed when handling mobility. If a node moves to another physical location, the topology should heal itself, and the addressing should remain consistent. Moreover, if a new node arrives at a domain, it should be easily attached to the system, and the corresponding lookup tables should be properly modified. In the following we give a brief overview on how the architecture deals with the mobility of nodes.

There are two basic kinds of mobility in our system. The first type is intra-domain mobility (the so-called "micro-mobility"): the node moves to another location inside the domain. The second type is inter-domain mobility (the so-called "macro-mobility"): the node moves outside the boundaries of its former domain, and joins a different ad hoc network (wireless domain).

In case of intra-domain mobility, a mobile node should acquire a new LNA, according to its current location inside the wireless domain. However, address changes do not have to spread outside the specific domain. Since the distributed Location Agency only deals with the domain memberships of the nodes (it stores (GNI, domain ID) tuples), the change in the LNA of the mobile node does not affect the lookup tables in the LAs. Therefore, only the Domain Edge should be informed about the new LNA, in order to update its LAD table accordingly. Further advantage of keeping address changes local is that in case of ongoing inter-domain communication, neither the correspondent node of the moving node nor the Location Agency has to be informed of the address change, as Location Agency only bothers with domain ID changes. Message redirection due to node micro-mobility is done at the Domain Edges.

If a node moves out of the range of its parent node, the parent should delete it from its Neighbor List and redistribute the global GNI-based hash value space among its remaining neighbors. On the other hand, at its new location the mobile node acts as a newcomer, as described in Section II/B; it acquires a new LNA from its new neighbors, and registers it at the Domain Edge.

Inter-domain mobility requires different actions to be taken, as shown in Figure 5. If the moving node requests attachment in a new domain, its new parent node should inform the Domain Edge about the newcomer. The Domain Edge notices that there is no entry for the newcomer's GNI in its LAD table; thus, it creates an entry for it, it hashes the new node's GNI, and finds out which LA is responsible for it. Then, it initiates an update message towards that LA.

Upon reception of the update, the LA refreshes its GNL table. Moreover, if there was already an entry for that specific GNI associated with a different domain ID (i.e., the mobile node is not totally new to the network), it informs the Domain Edge of the node's former domain about the change. At last, the former Domain Edge refreshes its local lookup table, i.e., it deletes the corresponding entry. In case of ongoing communication, the former domain may notify the correspondent node of the mobile node, that a domain change occurred. To ensure a more sophisticated domain handoff procedure, soft-handover and buffering methods can be used.

Since nodes in a wireless domain are likely to communicate and move together in groups, the addressing method may be driven by policy considerations. These groups are controlled by a master device that represents the parent of the group sub-tree. In case of group mobility the master is the only one that has to interact with the environment on behalf of group address change. Only the group identifier address prefix has to be changed and updated at the Domain Edge when acquiring a new group parent.

4. PERFORMANCE ANALYSIS

One of the most important goals of the proposed architecture is to be scalable, even in the presence of a large number of nodes in the network. With regard to scalability, it is not advisable to maintain routing tables in every router. Moreover, in ad hoc networks, all nodes act as routers; if there are a large number of nodes in the network, the routing tables consist of a large number of entries, which compromises scalability.

As opposed to traditional solutions, our approach provides stateless routing: the LNA-based hierarchical addressing and the hash-based forwarding mechanisms eliminate the need for states in the nodes. Besides this being a fully scalable method, the lack of routing tables can also speed up the routing decision process, as nodes do not have to perform searches in large tables.

Our stateless routing scheme is also fault tolerant, in the sense that there are no corrupted states in the routers (since there are no routing states at all). From the architecture's point of view, fault tolerance can also be achieved both at the upper and the lower levels of the hierarchy. Considering the upper level, the distributed Global Node Locator service in the core network ensures that if one of the Location Agents goes down there will be no stoppage in the network's operation. On the other hand, at the lower level the Neighbor Lists provide some flexibility in routing; if a node on the LNA-based path goes down, routing can be performed based on NLs.

The performance of our algorithm can also be enhanced through the use of multiple LNA trees, as presented earlier. If several trees are deployed, routing efficiency and robustness increases. However, multiple trees generate an increased signaling overhead as well. Therefore, the right balance should be found between the number of used LNA addresses and the overhead that is still acceptable to the wireless network. It is straightforward to think that the relative performance enhancements will decrease with the number of LNA trees (i.e., using two LNA trees instead of only one, changes the performance of the algorithm in much more drastically than using ten LNA trees instead of nine). However, this will depend on several factors, such as the size of the domain or the mobility patterns of the nodes. Evaluating the usefulness of the multiple LNA trees for different scenarios will necessitate a thorough simulation-based analysis.

Alternative routes back up the regular, LNA-based routing. The significance of the alternative routing mechanism is twofold: it does not only provide routes in case of node failures, radio transmission problems (e.g., noise, interference) and mobility, but can also be used as a load balancing method, shifting traffic load off the nodes close to the root.

Handling mobility is of great importance in future networking scenarios, as the number of mobile nodes is strongly increasing. The architecture supports both intra- and inter-domain mobility. Note that the alternative routing mechanism is suitable for a highly mobile scenario as well.

5. RELATED WORK

In this section we overview some of the related work in the literature that uses similar concepts to solve addressing or routing problems. We will briefly present the advantages and drawbacks of our solution when compared to these existing proposals.

Landmark [1] applies hierarchical addressing to build scalable network architecture. It defines Landmark nodes, which are routers whose neighbors within a certain number of hops contain routing entries for that router. The address of a node is a chain of different hierarchical level Landmarks. Every node stores the Landmarks in a hierarchical manner; and stores a neighbor list. The address distribution and storage method used by the Landmark Hierarchy is also a scalable solution, and resembles to our addressing method. However, Landmark was mainly proposed for wired networks; therefore, it does not consider the physical properties of the wireless networks, as opposed to our proposal.

If we are to distribute many different values in a balanced form, hash functions can be used to map the addresses into a hash value interval. This interval can be partitioned, with the portions of the interval being assigned to the different nodes; each node is responsible for the addresses that are mapped into its hash interval portion. This method is especially used in DHT based routing systems, such as Content Addressable Networks (CAN) [2], Chord [3], or Tribe [4].

In our solution, the distribution of the GNL database uses a similar method; the global GNI-based hash value space is partitioned among a set of Location Agents (which we considered to be collocated with the Domain Edges, to simplify the architecture). Moreover, during alternative routing, a given node partitions the global GNI hash value space among all the nodes in its immediate vicinity.

In overlay networks, such as those created through the use of DHTs, it is useful for nodes to know the addresses of the nodes in their immediate vicinity in the overlay. The same holds for wireless environments, where nodes might store information about their neighbors in the wireless network space. To store a so-called neighbor list needs some extra resources in nodes, but it can provide a more efficient routing. A well known system, in which the nodes use a list to store the addresses of the nodes in their vicinity, is Pastry [5]; it provides a scalable, decentralized object location and routing solution.

In Pastry, the size N of the neighborhood set is determined; at all times, the neighborhood set is completely filled with the address of the N closest nodes. As opposed to this, in our proposal the size of the Neighbor List is not determined. It contains entries for all the nodes that are one or a given number of hops away from the node; thus, it has a variable size that depends on the size of the network, the mobility of its elements, etc.

Finally, a routing method that is in a way similar to our alternative routing scheme is presented in [6]. Rumor routing is based on a routing query being sent on a random walk, until it reaches an already established path. If such a path is found, further routing is done along that path. If not, the routing query is resubmitted, in the hope of the new random walk leading to a useful path. The alternative routing scheme we propose is different in the sense that the hash-based forwarding scheme ensures the packets for the same destination being sent over the same alternative path, as opposed to the ever changing random walks in Rumor routing.

6. CONCLUSIONS AND FUTURE WORK

The increase in the number of communicating entities in today's networks, the need for supporting small and large scale mobility, and the justification of ad hoc networks have raised new challenges that routing protocols have to face. In the present paper we introduced a novel architecture that aims to provide IP-independent addressing and a scalable routing mechanism, which operates in a stateless manner. An alternative routing scheme is proposed to handle node failures, load balancing, and mobility support.

The goal of the paper was only to demonstrate the need for new, scalable addressing and routing mechanism, and to lay down the basic principles of a novel architecture that can efficiently handle the requirements of future networks. However, there are a lot of points in the proposed approach, where optimization is needed.

One important issue to optimize is the balancing of the address tree. A well balanced LNA assignment can provide better connectivity, and enable shorter and faster routes. Keeping the address tree balanced implies re-addressing functions that have to be designed in a scalable manner as well.

Moreover, re-addressing is needed when a parent node of a sub-tree moves away; the "orphan" sub-tree has to be "adopted" by another node. Our hierarchical addressing scheme provides aggregation opportunities, as the entire sub-tree can be easily re-addressed. If the root node of the sub-tree finds a new possible parent, re-addressing means replacing the LNA of the former parent node with the LNA of the new parent in the LNAs of all the sub-tree nodes. Moreover, a single update message is enough to refresh the LAD entries that correspond to all the sub-tree nodes in the Domain Edge. However, these aggregated re-addressing steps can lead

to unbalanced LNA trees. Finding an efficient optimization scheme for this issue is subject of future work.

The issue of security is also a problem that needs to be addressed. Our goal in this article was to introduce an architecture that simplifies node lookup, and a routing solution that exhibits the properties of wireless medium and keeps stored states low. Security problems arising in our networking context can be caused by address spoofing, that effect on the structure of address-tree, or denial of packet forwarding that may block best-effort forwarding sequences. Our goal in the future is to design the functions of our architecture keeping in mind the mentioned problems, but security extends beyond the scope of this paper.

Finally, implementing a prototype system, and performing simulations to test the efficiency of the architecture is also subject of future work.

REFERENCES

[1] TSUCHIJA, P. F.: *The Landmark hierarchy: a new hierarchy for routing in very large networks.* In Proceedings of the ACM SIGCOMM, Stanford, CA, August 1988, pp. 35-42.

[2] RATNASAMY, S., FRANCIS, P., HANDLEY, M. KARP, R., SHENKER, S.: *A Scalable Content Addressable Network.* In Proceedings of the ACM SIGCOMM, San Diego, August 2001, pp. 161-172.

[3] STOICA, I., MORRIS, R., KARGER, D. KAASHOEK, M F., BALAKRISHNAN, H.: *Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications.* In Proceedings of the ACM SIGCOMM, San Diego, August 2001

[4] VIANA, A. C.., Amorim, M. D., Fdida, S., Rezende, J. F..: *Indirect Routing Using Distributed Location Information.* In Proceedings of IEEE International Conference on Pervasive Computing and Communications, PerCom'03

[5] DRUSCHEL, P., ROWSTORN, A.: *Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems.* In Proceedings of the 18th IFIP/ACM International on Distributed Systems Platform, (Middleware 2001), November 2001.

[6] BRAGINSKY, D., ESTRIN, D.: *Rumor routing algorithm for sensor networks.* In Proceedings of the First ACM Workshop on Sensor Networks and Applications, Atlanta, GA, USA, October 2002, pp. 22-31.