MŰEGYETEM 1782

DEPARTMENT OF TELECOMMUNICATIONS AND MEDIA INFORMATICS
BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS

SCALABLE METHODS TO IMPROVE THE COMMUNICATIONS IN PEER-TO-PEER NETWORKS

Collection of Ph.D. Theses
by
Balázs Kovács

Research Supervisor:

Dr. Rolland Vida
Department of Telecommunications and Media Informatics

BUDAPEST, HUNGARY
FEBRUARY 2009

1 Introduction

The spread of wired and wireless Internet and the ever widening variety of applications acquaint many users with the opportunities residing in high-speed communications. The growing number of Internet users, paired with the increasing number of communication devices per user requires a reconsideration of communication algorithms so as to ensure scalability.

Generally, by scalability we understand the ability of a system, network or process to handle growing amounts of work in a graceful manner, or to be readily enlarged [1]. In other words, a protocol should be designed to remain suitably efficient even under severe load in order to label it as a "scalable" solution. In terms of communication protocols this means that service quality may change only slightly in case of a drastic modification in the number of nodes, their mobility, the traffic parameters, or other network features. This can be achieved by keeping the states stored by network nodes low, which influences memory usage, processing times, and most importantly the information to be spread and collected in the network, i.e., signaling load.

Distributing functionalities and responsibilities is a feature that may prepare a system to become more scalable. One form of distributed systems is the peer-to-peer (P2P) architecture, i.e., a system consisting of entities that are functionally equal. In computer networks, P2P refers to a network of nodes where every node serves and uses the services of other nodes, i.e., every node can be a server and a client. One distinguishing feature of P2P is the contribution of peers to network resources as they join. Another benefit of P2P is the elimination of single-points-of-failure; the network can continue to operate after the failure of any of its participants. There are solutions that employ a so called super-peer architecture, which designates, based on some classification criteria, a set of nodes to manage the P2P network; however, the super-peer role can be filled in by any of the peers, if a failure of the current super-peer occurs.

Usually, the P2P architecture is defined in the OSI application layer; P2P is a popular solution for file-sharing, chat, phone, and media streaming applications. However, the P2P communication paradigm is much more than that. Generally speaking, a system can be of a P2P nature if the features mentioned in the previous paragraph are valid for that system. Routing of wireless ad hoc networks using 802.11 [4] technology is a special network-layer P2P service. In order to provide multi-hop radio communication in ad hoc networks, every node should offer routing services to others. 802.11 Distributed Coordination Function (DCF) with Carrier Sense Multiple Access Collision Avoidance (CSMA/CA) [4] being a contention-based medium access technology, removes any form of coordination from a 802.11 network, thus enabling 802.11 nodes to act as routers.

My dissertation focuses on the scalability of communication protocols in both the network layer, for ad hoc networks, and the application layer, for overlay networks formed according to specific application requirements.

In an ad hoc network, scalability problems in the network layer are mainly related to addressing and routing, in case of large network size, high mobility, or a combination of both of these factors. There are proactive, reactive, and hybrid/hierarchical approaches to handle routing in wireless ad hoc networks. Papers [3, 5, 7] present the most representative ones. For further details, please refer to the dissertation [2].

I propose a hierarchical addressing and routing method, a hybrid solution that is built on the assumption that the dynamism of the wireless ad hoc network can be tied to some relatively fix, geographic points close to the residence area of the mobile nodes. The architecture that I called SCARF (SCalable Addressing and Routing architecture for Future networks) deploys fix network nodes close to these residence areas and involves them in network administration tasks. Furthermore, I also present an extension of the architecture, which provides efficient multi-hop Internet access for the SCARF network.

On the other hand, my dissertation focuses on scalability issues in the application-layer as well, more specifically in the construction and operation of overlay networks based on structured P2P networks, namely, distributed hash tables (DHT). Structuring means that the P2P network is ordered according to an algorithm so as to accelerate lookups. DHTs map resource identifiers into a hash space that is distributed among the network nodes. If a resource is mapped into the hash subspace of a node, that node will be responsible for the resource; thus, lookups for this resource will end up at this node. Generally, DHTs can be used to construct structured P2P systems, keeping the cost of lookups low. Since these DHT protocols operate in the application layer, they are free to construct an arbitrary logical, i.e., overlay topology.

My dissertation exposes an enhancement for a well-known but not so popular DHT algorithm, called Content Addressable Network (CAN) [17]. The major shortcoming of CAN is that lookup cost in the network is relatively high, as it grows with $O(dN^{\frac{1}{d}})$, where N is the size of the network and d is the number of dimensions of the coordinate space, a network parameter that is costly to be changed during the operation of the system. As opposed to CAN, state-of-the-art DHT protocols such as Chord [18] and Kademlia [20] ensure lookups to finish in $O(logN)$ steps. Nevertheless, CAN has several merits to exploit; pairing these with the enhancements I propose can lead to an $O(logN)$ lookup cost, making the improved solution outperform Kademlia and Chord in several aspects.

In DHTs such as Chord and Kademlia the means to provide fast lookups is to have some contacts residing in “distant” parts of the network. Thus, for every such DHT, a distance metric is defined on the ID space. Generally, a DHT node stores at least one contact for exponentially increasing distances from its own ID. As a consequence, a lookup in a DHT is a logarithmic search in an ordered space. In order to support the efficient operation, DHTs deploy and update *long-range contacts*. The only role of these contacts is to speed up lookups. On the other hand, DHT nodes need precise information about *short-range contacts*, which is essential for maintaining a consistent ID space. Therefore, we can formulate different aims concerning the maintenance of short- and long-range contacts.

In my PhD work I focused on creating an enhanced algorithm for CAN called LR-CAN (Long Range CAN) that presents an adaptive solution for establishing overlay connections in the DHT.

My thesis booklet presents the most important results of my research work in two groups of theses.

2 Research Objectives

The aim of the first thesis group was to elaborate the SCARF (SCalable Addressing and Routing architecture for Future Networks) protocol for scalable, ad hoc communication where random movements of mobile nodes occur around some specific, relatively fix points. My research focuses on the way how the fix points, as network nodes, could accelerate node localization, reduce broadcast traffic, and increase mobility tolerance. On the other hand, the first thesis group also proposes an architecture for supporting multi-hop wireless Internet access through a SCARF ad hoc network.

The research work summarized by the second thesis group exposes an enhancement of the CAN DHT protocol. The aim of this research is to create an algorithm that can provide lower and more balanced lookup cost, paired with a lower signaling cost than state-of-the-art DHT protocols such as Chord and Kademlia can ensure. By doing so, it could set the focus of DHT related research back on CAN.

3 Methodology

I employed simulations to prove the viability and analyze the efficiency of the algorithms and protocols I proposed in the first thesis group. I implemented the simulation code in the ns-2 [13] discrete event simulator.

The investigation of the application layer DHT enhancement proposed in the second thesis group would be possible to approach analytically as we may neglect some or all connectivity issues when certain network metrics are evaluated. However, my assumptions for the analysis, and the intention to have an extensive comparison to other solutions, required to evaluate the protocol in a network simulator.

4 New Results

4.1 Scalable Routing of Wireless, Ad hoc Networks

Nowadays, the license-free IEEE 802.11 family is the most popular wireless technology for Internet access or in ad hoc networks. One serious drawback of 802.11 systems is the inefficient and unreliable propagation of broadcast messages, a service that is often employed by network layer routing protocols. Disseminating or requesting node information in a wireless ad hoc network by broadcasting the radio medium is costly in 802.11 networks.

There are many cases when ad hoc networks may be operated around buildings, rooms, campuses, and other manmade infrastructures, i.e., places where we can define relatively fix reference points the mobile nodes move around. In case we permit the constraint these reference points set for the dynamism of the network, we may benefit in network administration tasks in terms of reducing signaling load.

Thus, in this thesis group I propose a network architecture that eliminates the broadcast waves in node and route discoveries, and uses unicast queries instead. To do so, I construct a hierarchical architecture which is built and maintained with minimal signaling load. The architecture sets up a good basis for fault-tolerant routing.

Thesis group 1: *I proposed an addressing and routing method for wireless, ad hoc networks that performs routing without using routing tables, and thus keeps network layer signaling low.*

4.1.1 Addressing

Thesis 1.1 (Address delegation for a wide and balanced tree [J2, J6, C2, C3]): *I proposed a hierarchical address delegation strategy for wireless, ad hoc networks. Mobile, ad hoc nodes construct an address tree, which is a rooted tree graph that singles out a static root node, its edges reflecting the physical connections between nodes. Nodes build parent-child relationships in the address tree.*

Parent-child relationships are constructed by an address-selection algorithm. For the joining node, the algorithm selects a parent that is closest to the root and that has the least children.

The tree structure is the most important element of the algorithm. Every newcomer node learns its neighbors and their hierarchical identifier, called Local Node Address (LNA), from periodical Hello messages. These messages bear adequate information for the newcomer to select its parent from the neighboring parent candidates. Then, the selected parent node will delegate an LNA to the newcomer, formed by appending a suffix to the parent's LNA. As an example, the newcomer node in Fig. 1 will connect to node 1.2.2 and will be handed the extended LNA 1.2.2.1.

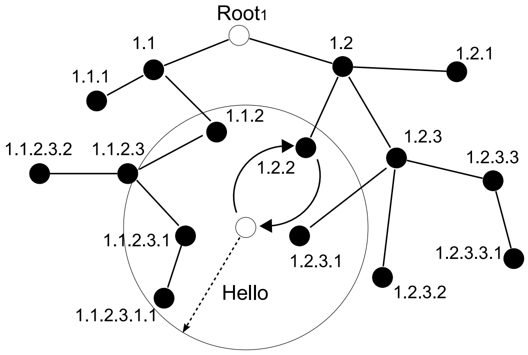


Figure 1: *Address delegation*

The address-delegation strategy, together with the address-selection algorithm keeps the address tree wide and balanced around the root, which is beneficial in routing data packets.

Corollary 4.1. *The wide and balanced address tree ensures the wireless topology to be centered around the root node, and provides horizontal load balancing, i.e., balancing between nodes on the same level in the address tree.*

In contrast to other solutions that exist in the literature ([6, 7, 8, 9, 10]), this hierarchical addressing strategy assumes a trusted, static node as the root. Incorporating such a node sets constraint for the dynamism of the network, but it can facilitate addressing and node localization if this architecture is handled carefully (discussion after Thesis 1.3). Ensuring wide and balanced characteristics for an address tree was not an objective of previous solutions described in the relevant literature.

4.1.2 Routing

The proposed addressing strategy enables the operation of a routing algorithm without the need for routing tables; the following definitions help to understand the concept of the routing algorithm. Thesis 1.2 presents how routing is solved in my proposed architecture.

Definition 4.1. A *routing table* of a node is a database which consists of entries that name a neighboring node or a list of nodes that take closer or lead to certain destination nodes.

In wireless ad hoc networks, for faster routing path availability a node needs to store entries not only for its own communications, and thus the routing table grows with the size of the network $O(N)$ or $O(logN)$ in the best case, where N is the number of nodes.

Definition 4.2. In a wireless network, a node's *neighbor list* is a database which consists of entries corresponding to nodes within a certain "distance" (generally in the radio range).

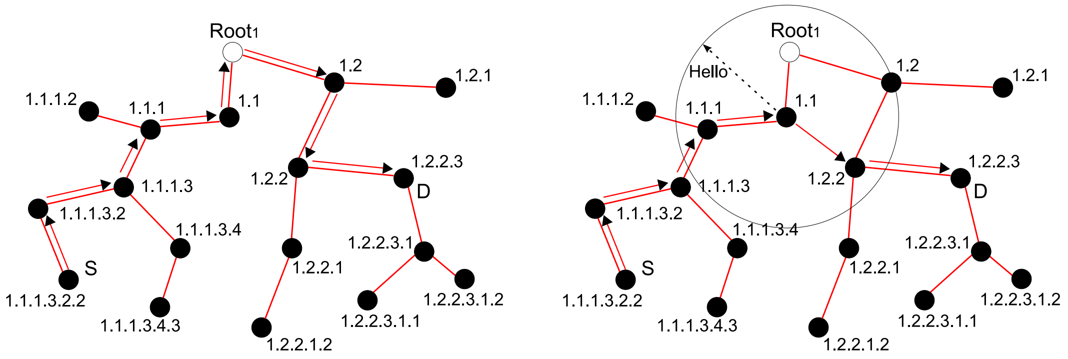
In general the size of a neighbor list is $O(R^2\rho)$ where R is the radio range, and ρ reflects the density of the network that can be considered as constant in reasonable scenarios. Since R can be also considered as constant, the size of the neighbor list is constant as well.

Let $S = \{m_i\}_{i=1}^{M}$ denote the set of available M nodes in the neighbor list. The length of an LNA address a is $|a|$. In case a node i has address a_i and node j has address a_j, the common prefix length of a_i and a_j is $|p_{ij}|$. The tree-based distance between nodes i and j is $d_{ij} = |a_i| + |a_j| - 2 \cdot |p_{ij}|$.

THESIS 1.2 (Address-tree based routing [J2, J6, C2, C3]): *Based on the hierarchical addressing scheme presented in Thesis 1.1, I proposed a routing solution that uses Local Node Addresses and neighbor lists solely, without the need for routing tables. For a given destination node D, the nexthop will be the node* $\operatorname{argmin}_{m \in S}(d_{mD})$.

If a node knows the addresses of its neighbors and the address of the destination, a loop-free forwarding can be assured.

The proposed address based routing method, complemented by neighbor lists, does not need to propagate information to distant parts of the network, and does not need to burden the network with broadcast route discoveries.



(a) Routing by address tree

(b) Routing by neighbor list

Figure 2: Basic routing

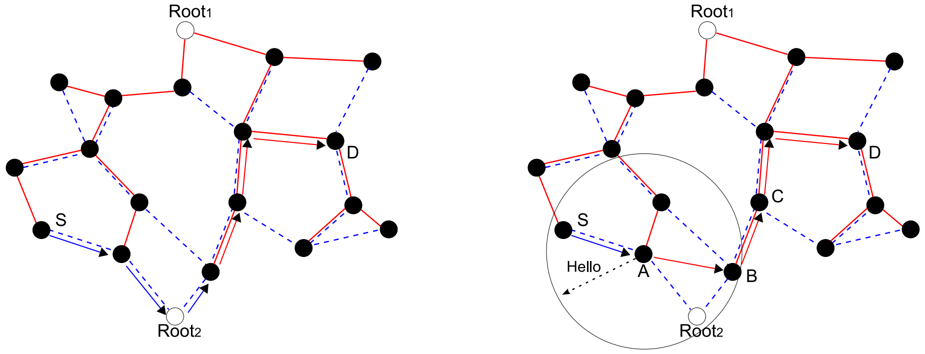
In Fig. 2(a) we can see an example on how routing may be performed using one address tree. However, it is not mandatory for the packets to travel along the links of the address tree. If the network topology allows it, nodes can find shortcuts between tree branches by the aid of their neighbor lists, as shown in Fig. 2(b). Besides shortening the routing paths, these neighbor list based shortcuts have also the advantage of relieving the root node from the burden of forwarding each and every data packet that is injected in the network by one of the communicating nodes.

There are other hierarchical solutions as well ([7, 8, 9, 10]) that may reduce routing table size to $O(\log N)$. However, these solutions still need to disseminate routing information of some complementary proactive or reactive ad hoc routing protocol; moreover, for these protocols there is a trade-off between mobility tolerance and the signaling of mobility events, which is hard to control.

The routing solution proposed above in Thesis 1.2 may overwhelm the nodes that are close to the root of the address tree. In order to relieve these nodes, it is important to provide vertical load balancing. The next thesis addresses this issue.

THESIS 1.3 (Multi-tree routing [J2, C2]): *In order to increase the efficiency of the routing algorithm and ensure vertical load balancing, I proposed to deploy multiple, physically carefully separated, static root nodes, and to build multiple hierarchical address trees that span the wireless network. Nodes should connect to each of these address trees to obtain multiple addresses (LNA) and should have different parents in different trees, if possible. Then, the forwarding process will select the next hop node from the neighboring nodes on any of the address trees, so as to ensure the shortest route toward the destination.*

Let $T = \{t_k\}_{i=1}^{K}$ *denote the set of* K *address trees spanned in the network. For a given destination node* D*, the nexthop node is* $\mathrm{argmin}_{m\in S}\left(\min_{t\in T} d^t_{mD}\right)$*, where* d^t_{mD} *is the distance between node* m *and destination node* D *on tree* t*.*



(a) Routing by two address trees

(b) Routing by neighbor list

Figure 3: Multi-tree routing. Solid lines: tree of $Root_1$. Dashed lines: tree of $Root_2$.

As we can see in Fig. 3(a), we can find more alternative and even shorter routes by the combination of two address trees between the same source–destination pair. Instead of 8 hops on the tree of $Root_1$, node S finds that it is only 7 hops away from D on the tree of $Root_2$, so it starts to forward the packet that way. Node A uses its neighbor list to cut the path and jump immediately to node B. A also notices that forwarding should switch to the tree of $Root_1$ at node B, as D is closer on that tree; thus, it uses the address of B in the tree of $Root_1$.

The physical separation of the root nodes is of high significance in terms of routing performance. The more trees we use, and the more diverse trees the roots can span, the better vertical load balancing we may experience, as the probability of finding alternative routes increases. Thus, root nodes are less probable to be involved in forwarding. Determining the number of roots and their position are decisions we have to make during the design phase of our managed wireless network. Certainly, using more address trees means additional signaling; the size of a control packet grows, and the number of control messages related to tree construction multiplies with the number of address trees in use. Note however that the rate of hello messaging, which constitutes the main part of signaling, does not change.

The SCARF architecture supports the installation of root nodes on demand, to facilitate the design phase; in my dissertation, I do not investigate this feature.

The multi-tree forwarding mechanism can reveal forwarding failures either by logical topology—it can not find a node that has lower distance to the destination than the current node—or by the physical network mechanisms—the nexthop-node is found but the link-layer stack indicates a forwarding error.

Corollary 4.2. *In case of detecting a forwarding failure, the multi-tree forwarding mechanism can redirect a packet to a routing hop that was already visited, without generating a loop, since for the current packet, the algorithm permanently excludes from forwarding address trees on which a failure was detected.*

Furthermore, this feature enhances adaptivity to link failures as it tries to progress along alternative routes before generating route error messages. Once reaching the destination, information about node addresses can be updated.

Note that for a desirable performance the wireless network spanned by the k address trees needs to be at least k-connected.

4.1.3 Local Address Directory

In order to perform routing in the ad hoc network, it is first necessary to identify the destination nodes. Since the LNA addresses of the nodes might change over time as they move around in the ad hoc network, a global, unique identifier must be assigned to each node. I called this Global Node Identifier (GNI). As a consequence, when routing toward a mobile destination node, we have to map first its GNI to probably multiple current LNAs, corresponding to the different address trees.

This mapping is stored in a so-called Local Address Directory (LAD). The mapping service is run only by the static root nodes. Each root node stores the GNI of those ad hoc nodes that are attached to its address tree; a GNI is mapped to an LNA belonging to this address tree. The LAD service enables nodes to initiate communication to any other node in the network, by sending unicast queries to the LAD.

Mechanisms of the routing algorithm ensure that the LAD has to be contacted only if routing is unable to recover from a severe topology change due to mobility. For further details, please refer to the dissertation.

The addressing and routing method that I proposed requires the LAD to store up-to-date LNA addresses. To fulfill this requirement, mobile nodes need to notify the LAD each time they acquire a new LNA address. Moreover, the LAD needs to be aware of the presence of a mobile node, either by using soft-state entries that expire if not refreshed, or by employing hard-state entries that are deleted if the network notifies the LAD about nodes that became disconnected.

4.1.4 Intra-domain Mobility

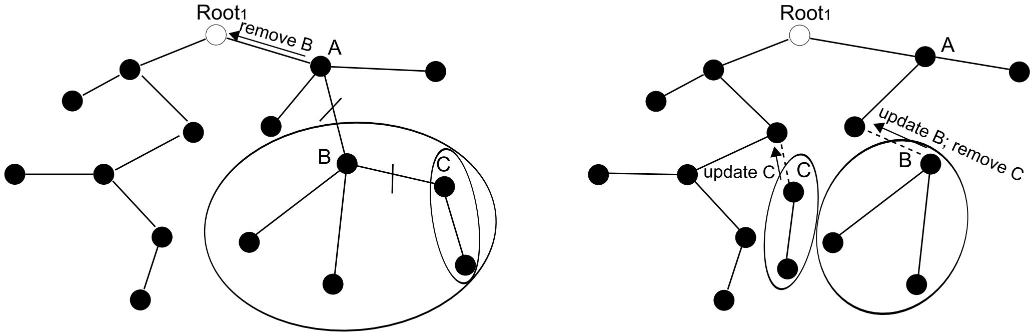
Mobility of nodes may cause link failures in the address tree.

Corollary 4.3. *The address-delegation strategy presented in Thesis 1.1 enables the subtrees of the topology to be handled as one entity in case of mobility events. Thus, the re-addressing of subtrees is of low-cost.*

Re-addressing a subtree consists of two network interactions. First, the root i of the moving subtree needs to notify the LAD about its new address obtained from its new parent. The LAD runs its address-replacement algorithm that changes all addresses a_j of nodes $j \in T_i$ to $\langle\langle a'_i\rangle.\langle id_j^{(i)}\rangle\rangle$, where T_i is the subtree of node i, a'_i is the new address of i, and $id_j^{(i)}$ is the identifier of node j inside T_i. Second, Node i issues a triggered hello message so as to notify its children about the prefix change of their address; the children do the same, until the update reaches the leaves of the subtree.

In Fig. 4 we can see the subtree T_B of node B loosing its parent as the parent moves away; at the same time, the subtree T_C of node C gets disconnected as well from node B. Both B and C request a new address. As a result, the LNAs of all the nodes participating in the subtrees of nodes B and C are refreshed in the LAD. This is achieved by only two unicast update messages, one sent by node B and one by node C. Thus, the number of messages generated by mobility events is proportional to the number of broken links.

THESIS 1.4 (Preserving tree-balance [J2, J6, C2, C3]): *Based on the method presented in Corollary 4.3, I proposed an algorithm that adapts the logical network topology (i.e., the shape of the address tree) to the changes due to network dynamism in order to converge to the balanced tree obtained by the algorithm of Thesis 1.1.*



(a) Subtrees disconnect

(b) Subtrees reconnect

Figure 4: Handling node mobility

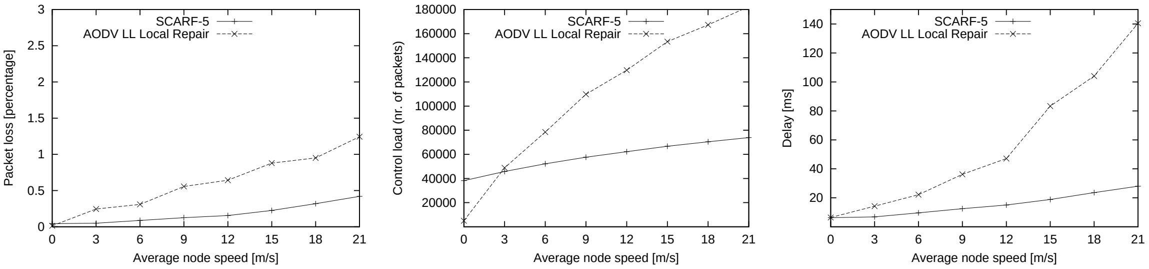
The purpose of this algorithm is to preserve the quality of connections; deep address trees are not preferred. The algorithm permanently scans the surroundings of nodes to find better points of attachment in the address trees that may appear due to the dynamism of the network. Based on Corollary 4.3, the re-attachment of a node results in improved connectivity for its whole subtree. To avoid overwhelming the network by parent-child negotiations, I used a tuning parameter l ($1 < l \leq l_{max}$), to mark the tree level above which nodes might start to look for a new parent. If a node descends below level l, probably because of a mobility event related to one of its ancestors, it is obliged to do so. Note that the root node is at level 1 and l_{max} is a fix parameter representing the maximum allowed tree depth.

4.1.5 Simulation

THESIS 1.5 (Scalability simulations [J2]): *Through extensive simulations I showed that the proposed SCARF addressing and routing method is able to conduct the same offered user traffic load with significantly lower signaling overhead than AODV ad hoc routing, in case the network is densely populated, mobile nodes are highly dynamic, and several parallel user flows are to be maintained. Furthermore, user data packets suffer lower end-to-end delays.*

Simulations were performed in the ns-2 [13] network simulator. A significant number of scenarios were analyzed for each set of mobility, network load, and network population parameters. The SCARF architecture was analyzed with one, three, and five root nodes deployed. I compared the efficiency of SCARF with AODV [11], one of the most efficient and scalable on demand ad hoc routing protocols. In my simulation I used the officially RFC compliant version of the protocol implemented by the Uppsala University [12]. I analyzed several different parameters and options for AODV and concluded that AODV-Link Layer with Local Repair (AODV-LL-LR) is the most efficient AODV variant. AODV-LL obtains link availability from link-layer feedbacks when data packets are sent; it thus eliminates the need for hello messages. On the other hand, Local Repair means that the protocol tends to keep packets on an intermediate node until a route is repaired after a link failure. I used link layer feedbacks in SCARF as well, in order to sense link failures as soon as possible. The same MAC layer parameters were set for both protocols.

For the simulations I employed 30, 50, or 70 parallel Constant Bit Rate (CBR), UDP traffic flows, started between the 50^{th} and the 100^{th} second of the 400-second simulations. The flows had low traffic rate (1 kbit/sec, 2 packets/sec) in order not to overwhelm the MAC layer and to enable it thus to support scenarios with many flows having different source–destination pairs. Such a scenario burdens the routing algorithm more than the MAC layer.



(a) Packet loss

(b) Signaling load

(c) End-to-end delay

Figure 5: ns-2 simulations with 80 nodes and 70 parallel flows

Here in the thesis booklet I only present the figures with the highest simulated load. As mobility, traffic load and network size increases, SCARF becomes more and more efficient, clearly outperforming AODV-LL-LR. In Fig. 5 I present a scenario where 80 nodes were deployed inside a 800m x 800m simulation area; the nodes had a 250m radio range. The results show that for a 15 m/s average speed my solution with five root nodes generates up to 50% less signaling than AODV-LL-LR, while having better delivery ratio, and lower end-to-end delays. Note, that packet delivery ratio of both AODV-LL-LR and SCARF is remarkable for this network size and mobility, compared to other ad hoc routing protocols.

4.1.6 Internet Connectivity for SCARF

So far, I proposed an ad hoc routing solution that has lower signaling overhead, lower delay, and higher packet delivery rate than other ad hoc routing solutions, if movements in the ad hoc network occur around relatively fix reference points. But what if these fix points are dedicated special nodes, for example wireless access points?

802.11 wireless networks are usually operated in *managed* mode, where an Internet Access Point (AP) is permanently available. One of the main drawbacks of 802.11 a/b/g solutions is their short radio range. Fading of radio signals is more drastic in buildings, as different obstacles reflect and absorb radio waves. A solution for this shortcoming can be a layer-3 extension of the hotspot coverage area, i.e., providing a multi-hop access of the AP for the mobile nodes that do not see it directly. The SCARF architecture could be used to support such a service.

THESIS 1.6 (Architecture [J2]): *I proposed an extension of the SCARF architecture to support multi-hop Internet access that requires the following steps:*

1. *The root points of the SCARF address trees have to be attached to Internet Access Points (APs);*
2. *The multiple APs have to be grouped into a Wireless Domain (WD) to enable the advantages of multiple root nodes;*
3. *Inside a WD, APs should be connected by a trusted, low-latency, wired, mesh network to further reduce the signaling overhead that the wireless network generates;*
4. *A Wireless Domain Gateway (WDG) has to be dedicated to hide the wireless domain from the Internet, and to provide transparent and efficient mobility handling inside the domain.*

Furthermore, I showed that SCARF supports Internet access by implementing Internet Gateway discovery, IP address configuration, and Internet connection setup.

The architecture can be inserted as a Locally Enhanced Routing Scheme into the IP mobility framework [16]. Multi-hop Internet access for wireless nodes through SCARF can be assured with such modifications of the protocol that do not deteriorate its wireless performance presented in Thesis 1.5.

The mesh connectivity between APs enables the root nodes to communicate with each other when LAD related signaling is received; as a consequence, mobile nodes do not need to address each AP if the purpose is simply to access the LAD service. LAD related signaling can be addressed arbitrarily to any of the APs.

The purpose of the WDG is to attach the multi-root architecture to the Internet and to support inter-domain mobility. For Internet access, it is practical to ensure interoperability with Mobile IP [14, 15]; thus, nodes that are not SCARF enabled can also set up and maintain communication to mobile nodes residing in the SCARF domain. To attach SCARF nodes to the IP system each node should configure its home IP address as its GNI. The main tasks of Internet access are solved by SCARF as follows:

- *Internet gateway discovery*: as in a SCARF network the propagation of root node information is a basic feature that is used for maintaining the address trees, my solution does not need to implement any further mechanism to incorporate Internet gateway advertisements or solicited gateway request;

- *Address configuration*: the Wireless Domain Gateway assigns its IP address as a care-of address (CoA) for all mobile nodes in its domain; once an incoming packet reaches the WDG, the SCARF architecture handles the further delivery of the packet, with the help of the LAD service that maps the GNI (i.e., the home IP address) to the up-to-date LNAs;

- *Connection set up to an Internet node*: the address query phase of the connections setup is explained in Fig. 6. Mobile node MOB_a residing in the wireless domain WD_i tries to establish a connection to node b, with the identifier GNI_b. As the node cannot be found through the LAD in domain i, the request is forwarded to the home agent HA_b responsible for node b, which then redirects it to the wireless domain WD_j where the node currently resides. The LAD in this domain does the mapping, and the reply is propagated back to the initiator.

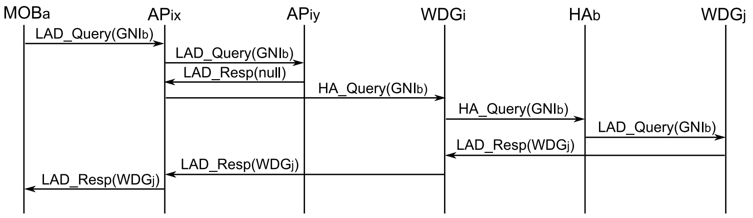


Figure 6: Inter-domain communication: finding a SCARF node through the Internet

For a detailed discussion of inbound communication toward a domain and interdomain mobility please refer to the dissertation.

4.2 An Adaptive Distributed Hash Table

THESIS GROUP 2: *I proposed an enhancement of the CAN DHT which enables it to outperform state-of-the-art P2P DHT lookup protocols in lookup performance and efficiency.*

One of the first published DHTs is CAN (Content Addressable Network) [17]. CAN maps nodes and resources onto a d-dimensional ID space that wraps to a d-torus. The ID space is split into d-dimensional zones which are assigned to the nodes.

Long-Range CAN (LR-CAN), an algorithm that enhances CAN by utilizing long-range contacts, a popular solution for scalable DHTs. Second, the algorithm introduces adaptivity to network size through a cost-limit function (denoted as $SR(N)$—from Short Range—further on). In current DHTs the number of long-range contacts implicitly changes as network size changes, and their number is proportional to the logarithm of network size. On the contrary, LR-CAN controls the number of long-range contacts explicitly through the above mentioned definite cost-limit function. With the appropriate cost-limit function we can set LR-CAN so as to outperform Chord and Kademlia. Nevertheless, its multi-dimensional ID space, bidirectional along each dimension, allow LR-CAN to keep the necessary maintenance traffic lower than in the case of the above mentioned state-of-the-art DHT protocols.

THESIS 2.1 (The adaptive LR-CAN algorithm [J1, C1]): *I proposed an algorithm, called LR-CAN, which uses long-range contacts in CAN to reduce lookup cost. The number of long-range contacts of a node changes adaptively according to a desired lookup cost expressed by a cost-limit function, as defined in Thesis 2.2. The algorithm on a peer performs the following steps:*

1. *the peer node estimates average routing cost ($\widehat{R}_{avg}$) and the number of nodes ($\widehat{N}$) in the network;*
2. *the peer node determines the cost-limit for $\widehat{N}$ ($SR(\widehat{N})$);*
3. *if $\widehat{R}_{avg} \geq SR(\widehat{N})$, the peer node sets up a new long-range level with a certain number of contacts.*
4. *the peer goes to step 1 until $\widehat{R}_{avg} < SR(\widehat{N})$*

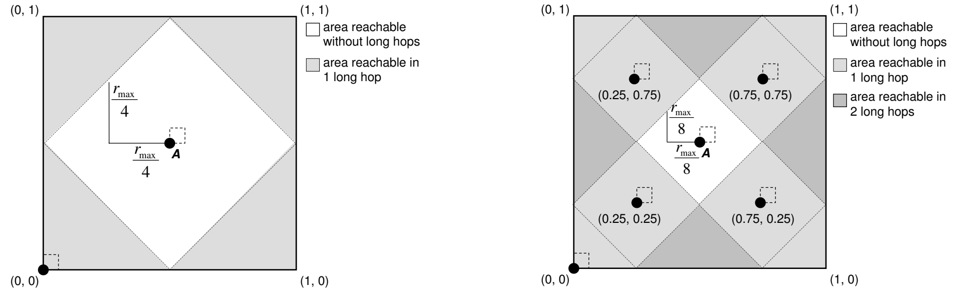
Each peer performs the same steps in one "stabilization" period, which maintains the contacts as well and removes them if necessary.

The main idea behind the LR-CAN algorithm is that a node is able to assess the network size (number of nodes) individually, without the help of an information server. This is due to the fact that the node ID and the zone assigned to the node in the CAN ID space are independent. In fact, when a node joins, it draws a random d-dimensional coordinate, performs a lookup for this coordinate, and asks its owner to split its zone into two equal parts. On stochastic basis, always the largest zones will be split; hence, the distribution of the ID space will converge to a distribution with equal zones.

Based on this assumption, network size ($\widehat{N}$) can be assessed in several ways. One of the most accurate solutions is by routing to the most distant point in the ID space and measuring the hop distance ($\widehat{R}_{max}$); note that routing can only progress horizontally or vertically, and thus the most distant point can be easily determined accordingly. If the average lookup cost ($\widehat{R}_{avg}$) is higher than the value that the cost-limit function ($SR(N)$) allows, a new level of long-range neighbors is deployed. By this method, LR-CAN keeps the average lookup cost below the value of the cost-limit function.

$$\widehat{R}_{avg} = \widehat{R}_{max} \cdot m_{\frac{avg}{max}} < SR(\widehat{N}), \tag{4.1}$$

where $m_{\frac{avg}{max}}$ is the empirical ratio of R_{avg} and R_{max} (an accurate estimation proved by simulations). Network nodes behave consistently; thus, their individual decisions will be valid in a global scope, improving the performance of the whole network.



(a) One long-range level is active

(b) Two long-range levels are active

Figure 7: A two-dimensional ID space when long-range contacts are used

From now on I will call the traditional CAN routing as *short-range routing* because it uses only short-range contacts in forwarding; accordingly, I call *long-range routing* the case when long-range contacts forward a lookup. The routing of LR-CAN can be split in two phases: in the first one a certain key is approached by long-range routing; in the second phase short-range routing leads to the owner of the key. In Fig. 7 we can see how the deployment of long-range contacts affects LR-CAN routing. The white square shows the subspace where only short-range routing is employed, in case node A starts a query. If the query is initiated to some distant part of the ID space, A uses its long-range contacts first, and only afterwards the traditional CAN routing, based only on short-range contacts; thus, the scope of traditional CAN routing is reduced. Fig. 7(a) shows the case when only one long-range level is activated, whereas in Fig. 7(b) two long-range levels are active. We can see that on the first level, there is only one LR contact, while on all the other levels there are 2^d contacts. The reason is the torus property: the "corners" of the ID space are equivalent to one point in the torus.

The lookup cost of the original CAN is $O(dN^{\frac{1}{d}})$, while the number of contacts is $O(d)$ contacts. In the following thesis I analyse the performance of LR-CAN, compared the original solution.

THESIS 2.2 (LR-CAN lookup-cost analysis [J1, C1]): *I showed that the LR-CAN algorithm reduces the lookup cost of CAN to $O(\log N)$ by using $O(\log N)$ contacts.*

Although in the traditional CAN architecture d can be set so as to provide $\log N$ lookup cost, the value of d will be appropriate only for the proper N; if N changes by orders of magnitude, d must be changed, which needs costly rehashing of the whole ID space. The number of long-range levels in LR-CAN, denoted as L, introduces a way to efficiently provide $O(\log N)$ cost. As a result, the introduction of L eliminates the significance of d in lookup cost of LR-CAN. It is important to note that the ability to build a multi-dimensional space still has significant benefits, since it vests the system with better fault-tolerance (more short-range contacts); moreover, the imprecision of long-range contacts is less detrimental for lookup performance than in a single-dimensional ID space.

If we assume that routing can progress only horizontally or vertically, the maximum distance in one dimension between two points $\mathbf{p}$ and $\mathbf{q}$ in the ID space, $(\mathbf{p}, \mathbf{q} \in \{\mathbb{R}^d | [0, 1)\}$) can be $\frac{1}{2}$. In d dimensions this maximum distance is $r_{max} = \frac{d}{2}$ (see Fig. 7). In order to obtain the distance in hops, the maximum distance has to be multiplied with the "resolution" $(N^{\frac{1}{d}})$ of the ID space. Each new level of long-range contacts halves the maximum distance. Hence the average route length of short-range routing (not including jumps on long-range contacts) is:

$$R_{avg} = \frac{r_{max}}{2^{L+1}} \cdot N^{\frac{1}{d}} \cdot m_{\frac{avg}{max}}. \tag{4.2}$$

As I want LR-CAN to provide $O(\log N)$ lookup cost, similarly to other DHT solutions, I propose to use $SR(N) = \frac{1}{c} \cdot \log_2 N$ as the upper bound of short-range routing, where c is the cost-limit factor, an arbitrary positive real number. To express lookup cost we need to deduce how L depends on N. Substituting R_{avg} by $\frac{1}{c} \log_2 N$ in equation 4.2 yields to the following:

$$\begin{aligned} L &= \log_2 \frac{r_{max} N^{\frac{1}{d}} m_{\frac{avg}{max}} c}{2 \log_2 N} = \log_2 r_{max} N^{\frac{1}{d}} m_{\frac{avg}{max}} c - \log_2 2 \log_2 N \\ &= O(\log N) - O(\log \log N) = O(\log N) \end{aligned} \tag{4.3}$$

as d can be considered as constant in my algorithm. The average message cost of lookups can be expressed by the sum of the long-range and short-range routing costs:

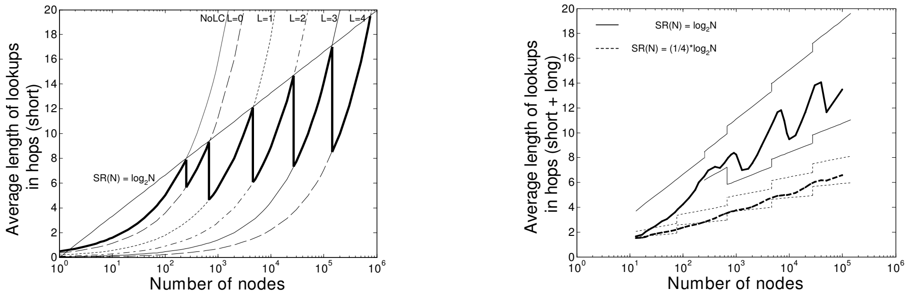
$$LR_{avg} + R_{avg} = O(L) + O(\log N) = O(\log N) \tag{4.4}$$

The reduced lookup cost of LR-CAN comes at the price of having more contacts to maintain than in CAN:

$$LC_{avg} + SC_{avg} = 2d + 1 + L2^d = O(d) + O(L2^d) = O(L) = O(\log N) \qquad (4.5)$$

Certainly, L has an implicit upper bound; its margin depends on the "resolution" of the ID space. There is no point in defining new long-range coordinates if they map to the same nodes. Consequently L should be defined on the $(-1 \leq L \leq \lfloor \log_2 N^{\frac{1}{d}}/2 \rfloor, L \in \mathbb{Z})$ domain; -1 indicates the case when no long-range contacts are used. The desired cost-limit function is expressed through the c parameter, although the domain of L implicitly defines also the reasonable domain of c, i.e., there is no point in defining a too high c value. Nevertheless, the protocol presented in Thesis 2.1 can safely handle any arbitrarily high c and deploy long-range contacts only when beneficial.

In Fig. 8(a), we can see how short-range route lengths grow in function of network size, in a two-dimensional CAN space, according to the above presented analytical model. A new level is switched on when the $f(N) = dN^{\frac{1}{d}}$ curves cross the desired average of short-range routing, namely $SR(N) = \frac{1}{c} \cdot \log_2 N$. In Fig. 8(a), $c = 1$. When $SR(N)$ is reached, route length drops, while the number of contacts increases. The number of levels corresponding for the different curves is denoted by L. The average, theoretical route length of the short-range routing grows along the thick line.



(a) Analytical results on the growth of $\mathbf{R_{avg}}$ in function of $\mathbf{N}$

(b) Simulation results on the average route length for different SR functions

Figure 8: Comparison of analytical and simulation results

The mathematical analysis contains assumptions that cannot be fully met in simulations and real life. We can see the simulated results on the thick curves in Fig. 8(b) with two different values of c ($c_1 = 1$ and $c_2 = 4$) that define two different cost-limit functions. On this figure we can compare the simulated results to the analytical ones, as the thin lines represent a theoretical lower and upper bound for the simulation results; they enclose the respective thick curve. The jumps in the thin curves are due to the change in long-range routing cost (LR_{avg}) that appears when a new long-range level is introduced; this routing cost depends only on the value of L and d (see Thesis 2.4). The theoretical thin curves assume equal-size, 2-dimensional zones and the same L value for all nodes, assumptions which obviously do not hold in real setups.

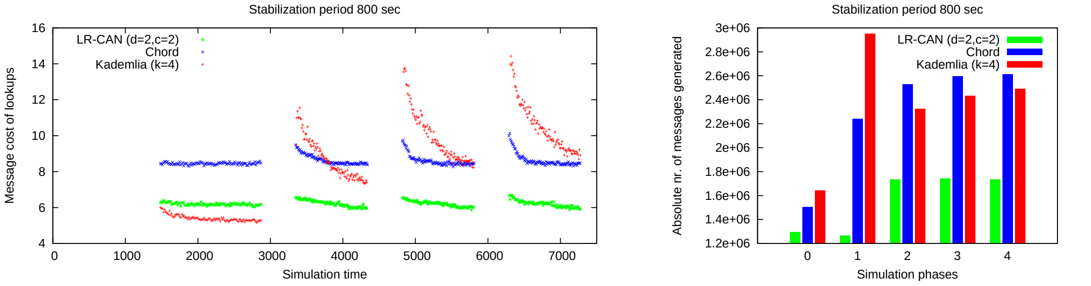
THESIS 2.3 (LR-CAN, Chord, and Kademlia performance comparison [J1]): *Through extensive simulations, I have shown that compared to acknowledged, state-of-the-art P2P lookup algorithms (such as Chord and Kademlia), in case of highly dynamic networks LR-CAN can provide faster lookups and lower deviation in lookup lengths. In addition, it generates less signaling traffic.*

Having multiple dimensions is a solution to speed up lookups. Linear search in a one-dimensional space generates $O(N)$ lookup lengths, compared to the $O(dN^{1/d})$ lengths that CAN provides in a d-dimensional space; higher d values yield lower lookup cost. Consequently, when a multi-dimensional solution, such as LR-CAN, has to provide a given lookup cost, it may have less strict requirements for the precision of long-range contacts than in single-dimension solutions, such as Chord. As a result, LR-CAN may generate less signaling for the maintenance of long-range contacts.

In my simulations I focused on performing a *fair comparison* of these DHTs. The goal of the simulations was to investigate how dynamism affects the message cost of lookups, which is basically influenced by the precision of long-range contacts. Since the effects of failures and graceful departures of nodes are similar in terms of long-range contact maintenance, I did not implement failure handling mechanisms in Chord and LR-CAN. As the on-demand nature of Kademlia enables it to handle failures without any additional mechanisms, the comparison in Fig. 9 is only partly relevant for Kademlia because, as mentioned, Chord and LR-CAN lack the corresponding mechanisms. The investigation of failure-tolerance can be another important topic of DHT-related research [19].

I performed these simulations with the help of an event-based P2P simulator for search/lookup algorithms [21, 22]. At the beginning of the simulation 10.000 nodes joined the network and stored random content. The remaining simulation period was divided in four equal 1470 second long parts. In the first part, nodes performed 10 lookups each for stored random keys. In each of the remaining three parts 32% of the nodes were replaced in a churn window; then, each node performed 10 lookups. The equal length of the simulation sections and the similar load in each section allows us to observe how the dynamism of the network affects the system performance.

There are several parameters to tune the performance of these DHTs. The stabilization period is one of the most important ones: it defines how often a node checks the correctness of its contacts. A larger stabilization period leads to weak tolerance of network dynamism, while lower stabilization periods result in increasing signaling traffic. The figures presented herein show results corresponding to a 800-second stabilization period, which is the double of the churn window. The parameter k in Kademlia has also a not negligible effect on dynamism tolerance; the higher the k, the better the dynamism tolerance and the higher the signaling. For LR-CAN, both the number of dimensions and the c parameter of the cost-limit function were set to 2.



(a) Lookup performance

(b) Signaling (one phase is 1470 sec)

Figure 9: Lookup performance and corresponding signaling traffic in a dynamic network

In Fig. 9, we can see that network dynamism has the weakest effect on the deviation of lookup cost in LR-CAN. Moreover, LR-CAN is able to provide the fastest lookups among these protocols; additionally, lookup performance can be tuned by replacing the cost-limit function (e.g., by modifying the c value). On the other hand, LR-CAN generates the least overall signaling traffic, as long-range contact management can be more effective due to the special ID space structure of CAN.

Further effects of the different system parameters and the according behavior of the algorithms are investigated in details in the dissertation.

Thesis 2.4 (Signaling minimization [J1]): *Based on the results of Thesis 2.2, I determined the optimum for L for the whole network so as to minimize the aggregated cost of lookup traffic and signaling related to contact maintenance.*

The proposed calculation needs information about the frequency of long-range contact maintenance (λ_p), network size (N), and average lookup rate (λ_l) in the whole network. Using these three input parameters, I can model the message cost of maintenance ($M_{cost}(L)$) and lookups ($L_{cost}(L)$) per time unit:

$$M_{cost}(L) = \begin{cases} 2 \cdot N \cdot \lambda_p \cdot (1 + L), & \text{\textit{if}} \ -1 \leq L < 0 \\ 2 \cdot N \cdot \lambda_p \cdot (1 + L \cdot 2^d), & \text{\textit{if}} \ L \geq 0 \end{cases} \tag{4.6}$$

$$L_{cost}(L) = N \cdot \lambda_l \cdot (LR_{avg}(L) + \frac{\frac{d}{2} N^{\frac{1}{d}}}{2^{L+1}} \cdot m_{\frac{avg}{max}}), \quad \text{\textit{if}} \ L \geq -1 \tag{4.7}$$

$$LR_{avg}(L) \approx \begin{cases} 0.5 \cdot (1 + L), & \text{\textit{if}} \ -1 \leq L < 0 \\ 0.5 + d \cdot 0.343 \cdot L, & \text{\textit{if}} \ L \geq 0, d = 1, 2 \end{cases} \tag{4.8}$$

The optimum for L is:

$$L_{opt} \approx \arg \min_{\left(\left[-1,\log_2 N^{\frac{1}{d}}/2\right],L\in\mathbb{R}\right)} \left(M_{cost}(L) + L_{cost}(L)\right) \tag{4.9}$$

Fast lookups are desirable in a DHT, but their provision requires a not negligible amount of signaling traffic. There might be cases when the low number of lookups injected in the system results in some long-range contacts becoming superfluous, i.e., their maintenance costs more than what can be gained on their availability. With a few assumptions and modifications to the original LR-CAN algorithm, we can construct an algorithm (SIGMIN) that adapts the number of long-range contacts not only to network size (N) but lookup rate (λ_l) as well, by using the method of Thesis 2.4 to determine L_{opt}.

The problem is thus to find an optimal value for the number of long-range levels (L) in function of network size (N), long-range maintenance frequency (λ_p), and lookup rate (λ_l). A method for gathering information about N was described earlier in Thesis 2.1. The frequency of long-range maintenance, i.e., the stabilization period λ_p is a global network parameter. However, for λ_l we need to implement an aggregated-data-collection technique which collects lookup rate data from the entire network, or at least from a significant part of it. In my solution, data collection is initiated by the bottom-leftmost node of the ID space, which can be a dedicated anchor node in the LR-CAN if it is persistent since the launch of the system; the collection has a cost of $O(N)$. After the data collection, this node can evaluate the input parameters and find the optimal value of L, which is then broadcast to the network. Consequently, the earlier definition of L, which was presented in Thesis 2.1 and was based on individual decisions of peers, has to be changed to a method where one node derives the optimal value of L based on aggregated global information.

$M_{cost}(L)$ includes the message cost of checking one long-range contact, the network size N, the maintenance frequency (λ_p), and the number of long-range contacts when L levels and d dimensions are used. $L_{cost}(L)$ includes the network size, the lookup rate (λ_l), and the routing cost on long- and short-range contacts (for the latter see Thesis 2.2). For the long-range cost $LR_{avg}(L)$, I gave an accurate approximation in equation 4.8 that is derived from the average cost obtained from the following formulas:

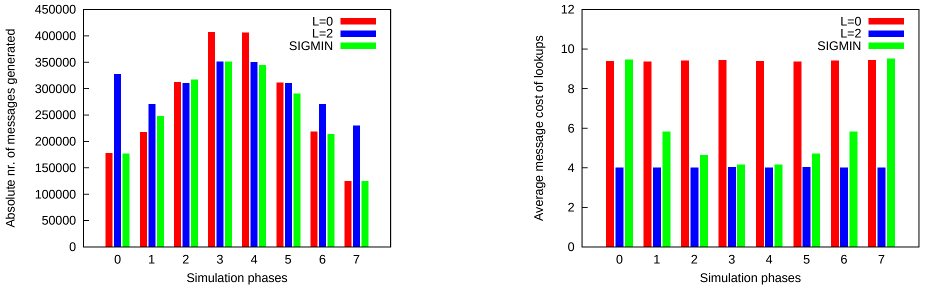
$$c(L) = 2^L \cdot \left(\frac{1}{2} + \frac{L}{2} - \sum_{i=2}^{L}\left(\frac{1}{2^i} + \sum_{j=1}^{\lfloor \log_2 (i-1) \rfloor} \frac{1}{8^j}\right)\right), \qquad \text{if } L \geq 1 \tag{4.10}$$

$$LR_{avg}(L) = \begin{cases} \frac{1}{2}, & \text{if } L = 0 \\ \frac{(c(L)-1)}{2^{L}} + \frac{1}{2^{L+1}}, & \text{if } d = 1, L \geq 1 \\ \frac{(c(L)-1)}{2^{L-1}} + \frac{1}{2^{2L+1}} - \\ \qquad -4 \cdot \sum_{i=1}^{\lfloor \log_2 (L+1) \rfloor} \frac{1}{4^i} \cdot \sum_{i=1}^{\lfloor \log_2 (L) \rfloor} \frac{1}{4^i} & \text{if } d = 2,\ L \geq 1 \end{cases} \tag{4.11}$$

The validity of these formulas can be checked against the weighted average of routing zones as presented in Fig. 7. Equations 4.10 and 4.11 prove the validity of equation 4.8 only for maximum two dimensions; however, two dimensions are adequate for LR-CAN and SIGMIN, since the lookup performance does not depend on d in these two algorithms, as mentioned in Thesis 2.2.

Solving equation 4.9 yields the optimal value of L, which will be a real value. However, on a given node, L must have an integer value. Thus, the nodes will individually choose the integer value either immediately below or immediately above the optimal value, using a simple stochastic method. As a result, the global average of L will be close to the optimal value.

Setting an optimal value for L in the P2P network is a forward decision, i.e., we need to optimize the performance of the network in the future based on past data. Hence, peers should provide detailed information about their lookup activity on a regular basis (e.g., daily), not only a simple average of a certain larger timeframe.



(a) Signaling

(b) Corresponding lookup cost

Figure 10: Simulation of lookup rate influence on the SIGMIN algorithm

The simulation that produced the graphs in Fig. 10 consisted of equal timeframes with constant node population. Only lookup rate changed in each phase, linearly increasing from phase 0 to 3, being constant in phases 3 to 4, and decreasing from phase 4 to 7. In one case L was set to 0, in another case L was set to 2 in a static manner for the entire network, while in the third case nodes used the SIGMIN algorithm to determine the value of L. SIGMIN was started with no long-range contacts. Note that setting up new levels of long-range contacts has some non-negligible cost, which is clearly shown in the first phase for the $L = 2$ variant; if $L = 0$, there are much less contacts to set up and maintain. SIGMIN might also be affected by this additional signaling cost, as an increasing lookup rate might trigger the introduction of new long-range contacts, according to equations 4.6 – 4.9. This effect can be seen in phases 1 and 2; hence, in phase 2 SIGMIN generates a higher signaling load than the other solutions, although only by a slight margin (this casual cost is not expressed by the formulas 4.6 – 4.9). The remarkable ability of the SIGMIN algorithm is to always converge to the more efficient variant in terms of signaling.

5 Applicability of the Results

In the first part of this thesis booklet I presented a routing algorithm that builds on the existence of some fix nodes around the residence area of the ad hoc network. The algorithm provides robust ad hoc communication, low packet delays, and low signaling if network population is dense, mobility is significant, and many communication requests are injected into the system. The fix nodes can be attached to Internet Access Points and get connected through a wired mesh to improve routing efficiency in the wireless network. As a result, a WiFi enabled building may become fully covered for Internet access, even if only a reduced number of access points are deployed; moreover, local, in-building communication can be solved in a P2P manner through mobile devices, without involving the access points.

The second thesis group presents the LR-CAN algorithm that amends the original CAN DHT in several aspects. First, it provides adaptivity to network size. With the original CAN algorithm, severely changing network population was very costly to handle. Adaptivity ensures that nodes in the network always maintain the appropriate number of contacts as network size changes; this keeps lookups fast and their message cost low. Second, in LR-CAN we can use different "cost-limit" functions for lookup cost and we can easily replace them according to network requirements. These requirements may be driven by service quality provisioning, or a simple consideration about the volume of lookup traffic; for instance, if the lookup rate in the network is low, then LR-CAN should not maintain too many long-range contacts as their maintenance costs more than what can be gained on their availability. The decision about replacing the cost-limit function can be made by a critical mass of peers or an ancestor peer who has administration privileges. Third, with a few reasonable modifications and assumptions, LR-CAN is able to find the optimal number of contacts in the whole network based on measurements of lookup intensity, and dynamically modify them in order to minimize the sum of its lookup and signaling traffic.

References

[1] A.B. Bondi. Characteristics of Scalability and Their Impact on Performance. In *Proceedings of the 2nd International Workshop on Software and Performance*, pp.195–203, Ottawa, Ontario, Canada, 2000.

[2] B. Kovács. Scalable Methods to Improve the Communications in Peer-to-Peer Networks *Ph.D. Dissertation*, Budapest University of Technology and Economics, Budapest, Hungary, 2009.

[3] C.E. Perkins and P. Bhagwat. Highly Dynamic Destination–Sequenced Distance–Vector Routing (DSDV) for Mobile Computers. In *Proceedings of ACM SIGCOMM*, pp.234–244, London, England, 1994.

[4] IEEE Computer Society LAN MAN Standards Committee. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11–1997, The Institue of Electrical and Electronics Engineers, New York, NY, 1997.

[5] C.E. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, July, 2003.

[6] G. Pei, M. Gerla, X. Hong, and C. Chiang. A Wireless Hierarchical Routing Protocol with Group Mobility. In *Proceedings of IEEE Wireless Communications and Networking Conference*, pp.1538–1542, Boston, MA, 1999.

[7] P.F. Tsuchiya. The Landmark Hierarchy: A New Hierarchy for Routing in Very Large Networks. In *Proceedings of the ACM SIGCOMM*, pp.35–42, Stanford, CA, 1988.

[8] B. Chen and R. Morris. L+: Scalable Landmark Routing and Address Lookup for Multi–Hop Wireless Networks. *Technical Report 837*, Massachusetts Institute of Technology Laboratory for Computer Science, 2002.

[9] S. Du, A. Khan, S. PalChaudhuri, A. Post, A. K. Saha, P. Druschel, D. B. Johnson, and R. Riedi. Self-Organizing Hierarchical Routing for Scalable Ad Hoc Networking *Technical Report TR04-433*, Department of Computer Science, Rice University, Houston, TX, USA, 2004.

[10] J. Eriksson, M. Faloutsos, and S. KrishnaMurthy. Peernet: Pushing Peer–to–Peer Down the Stack. In *Peer-to-peer Systems II*, Vol.2735, pp.268–277, ISBN.978-3-540-40724-9, Springer-Berlin, Heidelberg, 2003.

[11] R.D. Samir, C.E. Perkins, E.M. Royer. Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks. In *Proceedings of INFOCOM*, pp.3–12, Tel Aviv, Israel, 2000.

[12] Uppsala University. Ad Hoc On Demand Distance Vector Routing - For Real World and Simulation. URL: http://core.it.uu.se/AdHoc/AodvUUImpl.

[13] NS-2. URL: http://www.isi.edu/nsnam/ns/

[14] C.E. Perkins. IP Mobility Support for IPv4. RFC 3344, August, 2002.

[15] D. Johnson, C.E. Perkins, and J. Arkko. IP Mobility Support in IPv6. RFC 3775, June, 2004.

[16] M. Miklós. Hierarchikus mobilitás-kezelés skálázható vezetéknélküli hálózati architektúra felhasználásával. Masters' Thesis, Budapest, 2006.

[17] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content–Addressable Network. In *Proceedings of the ACM SIGCOMM*, pp.161–172, San Diego, CA, 2001.

[18] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: Scalable Peer–to–Peer Lookup Service for Internet Applications. In *Proceedings of the ACM SIGCOMM*, pp.149–160, San Diego, CA, 2001.

[19] D. Liben–Nowell, H. Balakrishnan, and D. R. Karger. Analysis of the Evolution of P2P Systems. In *Proceedings of the* 21^{st} *ACM Symp. Principles of Distributed Computing (PODC)*, pp.233–242, Monterey, CA, 2002.

[20] P. Maymounkov and D. Maziéres. Kademlia: A Peer–to–Peer Information System Based on the XOR Metric. In *Proceedings of the 1st International Workshop on Peer–to–Peer Systems (IPTPS'02)*, pp.53–65, Cambridge, MA, 2002.

[21] V. Darlagiannis, A. Mauthe, N. Liebau, and R. Steinmetz. An Adaptable, Role–Based Simulator for P2P Networks. In *Proceedings of International Conference on Modeling, Simulation, and Visualization Methods*, pp.52–59. Las Vegas, NV, 2004.

[22] Peerfactsim.kom. [Online]. Available: http://www.peerfact.org

Publications

[J] JOURNALS

[J1] **Balázs Kovács** and Rolland Vida. Long-Range CAN: to Enhance the Performance of Content-Addressable Networks. *Int. Journal on Advances in Networks and Services, IARIA Journals*, accepted, 2009.

[J2] **Balázs Kovács** and Rolland Vida. Hierarchical, Multi-spanning Architecture for Managed Wireless Networks. *Int. Journal of Wireless and Mobile Computing, Inderscience Publisher*, accepted, 2007.

[J3] **Balázs Kovács** and Csaba Simon. „Ambient" hálózatok - áttekintés. *Híradástechnika*, Vol. LX., pp. 39–44, Budapest, July 2005.

[J4] Gergely Biczók, Norbert Égi, Péter Fodor, **Balázs Kovács**, and Rolland Vida. Skálázható útválasztás mobil környezetben. *Híradástechnika*, Vol. LIX., pp. 26–31, Budapest, September 2004.

[J5] **Balázs Kovács** and Rolland Vida. A Zigbee technológia. *Híradástechnika*, Vol. LVIII., pp. 9–12, Budapest, November 2004.

[J6] Gergely Biczók, Norbert Égi, Péter Fodor, **Balázs Kovács**, and Rolland Vida. Scalable Addressing and Routing in Large-Scale Wireless Networks. *Production Systems and Information Engineering*, Volume 2 (2004), pp. 143–158, HU ISSN 1785–1270, 2004.

[J7] Gergely Biczók, Kristóf Fodor, and **Balázs Kovács**. Handover Latencies in BCMP Networks. *Communications: Scientific Letters of the University of Zilina*, No.1, pp.34–37, Zilina, Slovakia, 2004.

[J8] Gergely Biczók, Kristóf Fodor, **Balázs Kovács**, and Ágoston Szabó. Pervasive computing - An Overview. *Híradástechnika (selected papers)*, Vol. LVIII., pp. 2–7, Budapest, June 2003.

[C] CONFERENCES

[C1] **Balázs Kovács** and Rolland Vida. An Adaptive Approach to Enhance the Performance of Content-Addressable Networks. In *Proceedings of the IEEE* 3^{rd} *International Conference on Networking and Services ICNS 2007*, Vol. 1, pp.93–98, Athens, Greece, June 19–24 2007. Available electronically on: *IEEEexplore*, **Best Paper Award**.

[C2] **Balázs Kovács** and Rolland Vida. Hierarchical, Multi-spanning Architecture for Managed Wireless Networks. In *Proceedings of the IEEE International Workshop on Pervasive Computing and Ad hoc Communications (PCAC-AINA'06)*, Vol. 2, pp.111–115, Vienna, Austria, April 18–20 2006. Available electronically on: *IEEEexplore*

[C3] Gergely Biczók, Norbert Égi, Péter Fodor, **Balázs Kovács**, and Rolland Vida. Scalable Addressing and Routing Architecture for Future Networks. In *Proceedings of the Polish-Czech-Hungarian Workshop*, Volume 1, pp.19–28, Budapest, September 2004.

[C4] Róbert Szabó, Péter Kersch, **Balázs Kovács**, Csaba Simon, Márk Erdei, and Ambrus Wagner Dynamic Network Composition for Ambient Networks: a Management View. In *Proceedings of Eurescom Summit 2005*, pp.35–42, Heidelberg, Germany, April 27–29 2005.

[C5] **Balázs Kovács**, Rolland Vida, and Gergely Biczók. Hierarchical Architecture for Managed Wireless Networks. In *Proceedings of Student Workshop Conext 2005*, pp.260–261, Toulouse, France, October 24–27 2005.

[C6] Kristóf Fodor and **Balázs Kovács**. Blown-up rendszer megvalósítása. In *Proceedings of HTE-BME 2003*, Budapest, Hungary, 2003.

[O] **Other Publications**

[O1] Gergely Biczók, Kristóf Fodor, **Balázs Kovács**, and Ágoston Szabó. Blown-up rendszer tervezése és megvalósítása. *TDK Paper*, Budapest, Hungary, 2002.