A decentralized Information Service for IEEE 802.21 - Media Independent Handover (MIH)

Karel De Vogeleer, Selim Ickin, David Erman
School of Computing
Blekinge Institute of Technology
371 79 Karlskrona, Sweden

***Abstract*—Mobility frameworks nowadays are elaborate systems that involve many components that perform distinct operations. The IEEE 802.21 Media Independent Handover (MIH) standardized the functionalities of mobility frameworks and the interfaces between these components.**

The *Information Server*, a part of the MIH framework, is a system that contains information that is used for network intelligence purposes. Current mobility frameworks, especially the *Information Server*, are highly depended on the network infrastructure. Network operators are, however, reluctant to provide true mobility support to end users. To enable true mobility and MIH for end users who want to be independent, a decentralized *Information Server* is required that is maintained by the users themselves.

This paper presents a decentralized approach to MIH to enable end users with robust handover support. We analyze a decentralized hierarchical architecture and examine its performance. We investigate the scalability with regards to users in the system, entries stored in the database, the construction of the decentralized design, and the information distribution in the service.

We conclude that the proposed system is feasible to implement MIH as a decentralized system and the proposed architectures indicates to be scalable.

I. INTRODUCTION

The preferences and expectations of Internet users evolve continuously. Without doubt, users become increasingly mobile, as we have seen the past years. Since two decades research has been conducted on mobility frameworks that enables users to participate in seamless mobility throughout heterogeneous networks, including but not limited to WiMAX and WiFi. Seamless mobility allows users to roam between Internet access technologies without interrupting ongoing services and to maximize the Quality of Service (QoS). Network-centric roaming, or handover, approaches rely on the network infrastructure whereas user-centric roaming approaches are managed by end users. As for today, end users cannot benefit from seamless mobility, as most network operators are reluctant to invest in true seamless mobility enablers. Therefore a user-centric approach would allow end users to profit from seamless mobility today, according to their preferences, without third-party bias or governed business considerations.

The IEEE 802.21, Media Independent Handover (MIH) standard, provides a framework to support seamless handover. A subset of MIH's benefits are optimal network selection, seamless roaming to maintain connections, and lower power operation for multi-radio devices [1]. MIH consists of three services: the *Command Service*, *Event Service*, and the *Information Service*. IEEE 802.21 implementations are usually implemented in a network-centric manner, as the *Information Service* is located in the core network. The aim of this project is to build an independent MIH system where mobile users have the freedom to move where they want without loosing connectivity, irrespective of operators preferences.

The objective of this project is to develop a user-centric derivate of the *Information Service* for IEEE 802.21. The envisioned architecture of this service is based on a hierarchical Distributed Hash Table (DHT); each user maintains its personal *Information Service* node, all nodes are interconnected. The hierarchical DHT has multiple tiers or layers, where each tier unifies the lower-tier's DHTs. This architecture allows us to introduce locality in DHT and reduces signaling latency under certain assumptions. By exploiting the fact that communication has a localized nature, the hierarchical DHT will increase lookup performance in the database. As the DHT is fully distributed the hierarchical structure introduces resilience, fault tolerance and is an appropriate candidate for a user-centric implementation of the IEEE 802.21 Information Service.

II. BACKGROUND

We provide in this section a general background on the used IEEE standard, MIH, the CHORD DHT, and we also provide a literature overview on hierarchical DHTs.

A. IEEE 802.21 - Media Independent Handover (MIH)

This work is based on the IEEE 802.21 - Media Independent Handover (MIH) [2] standard. MIH facilitates the handover process of Internet Protocol (IP) communication form one technology to another by abstracting link layer intelligence to higher layers in a standardized manner [3]. Technologies of concern concern, but are not limited to, WiMAX (IEEE 802.16), WiFi (IEEE 802.11), Ethernet, or UMTS.

MIH is a middleware located between the applications and the radio modems, in particular between the Link layer and the IP layer, and is referred to as the MIH Function (MIHF). The MIHF contains three functions

- The *Event Service* is a standardized callback mechanism that anticipates on events or changes that occur in the link layer.

- The *Command Service* is a control system that initiates processes to change Link layer behavior or initiate handovers as a response to events that manifested in the network or at the end user's side.
- The *Information Service* is a database that contains network information that can be used to support decision engines.

The MIH standard does not define where the services should be implemented. The *Command Service* for example, that could include a decision engine for handover, could be implemented either in a mobile device but could also be offered as a handover service/server in the core network. The *Information Service* stores data that is useful for end user mobility. MIH *Information Service* defines a set of Information Elements that are indispensable for network selection, classified into three groups: *General Information and Access Network-Specific Information*, *PoA-Specific Information*, and *other Information*, which includes vendor- and network-specific details [4]. Examples include information about heterogeneous geographical network maps, service costs, QoS functionalities and roaming partners [3].

The *Information Service* contains shared information for mobile users and highly likely to be maintained by operators in the core network. As a result the data stored could probably be biased in favor of some network operator. The goal of our project is to decentralize this *Information Service* where all mobile users take the responsibility of maintaining their contributing information in the *Information Service*.

MIH entities communicate between each other with the Media-Independent Handover Protocol (MIHP). A detailed description of the protocol can be found in [4]. A protocol header is used to encapsulate MIH related payload. This payload can be of the *Command*, *Event*, or *Service* type, depending on the *Service Identifier (SID)* and the *Operation code (Opcode)* field. Data is retrieved from the *Information Service* via a *request* and replied with a *response* (SID = MIIS).

B. Introduction to CHORD

We now explain briefly the CHORD Peer-to-Peer (P2P) system. CHORD is a structured DHT [10] where all peers are placed in a ring. Each node has a uniquer ID and maintains values in the DHT of which the hashes are between itself and the proceeding node in the identifier space. Due to this structure, looking up a value is a deterministic algorithm. A list of peers in the system is stored at each peer. The distance in identifier space between the nodes in the list increase exponentially. Refer to [10] for a more detailed explanation.

C. Hierarchical Distributed Hash Tables (DHTs) Related Work

This section provides an overview of hierarchical DHTs that are developed over the past years. We will used DHTs as a basis for our *Information Service*. The presented work has its merits but also show disadvantages.

Garcés-Eric et al. presents in [5] a two-tier hierarchical P2P system. The lower-tier pertains both super-nodes and normal peers and the upper-tier interconnects reliable super-nodes. The lower-tier is meant to be a cluster of peers located in each other proximity, e.g. a university campus. Upper-tier and lower need not to have the same DHT algorithm. Though, Garcés-Eric et al. proposes CHORD for the upper-tier. Multiple super-nodes can reside in each lower-tier. An entry in the finger table of the upper-tier may contain multiple supernodes of a single lower-tier. The lower-tier groups are responsible for a certain portion of the address space of the CHORD ring. The authors so claim to improve the lookup time by $\log P / \log I$, where I is the number of groups and P the number of peers.

In [6], Hostätter et al. presents a fusion of CHORD and GNUTELLA called *Chordella*. The P2P system interconnects super-peers with a CHORD ring. Non-superpeers, or leaf nodes, can only query the DHT by forwarding requests to the superpeers. Leaf nodes are assumed to be resource-constrained devices whereas superpeers have less limitations in resources and are more reliable. In such a way, Hostätter et al. tries to achieve a reliable P2P system to be used in heterogeneous mobile environments.

Ganesan et al. proposed on [6] a procedure to build canonical version of existing DHTs. Ganesan et al. imposes a recursive routing structure by stacking DHT in a hierarchy, reflecting real-world organization. A 3-tier structure is imposed where the lowest tier is a group of nodes with logical connections. The second tier groups 3rd tier nodes and the top tier connects all second tier groups. The different groups of a lower-tier are connected in a upper-tier by carefully adding cross links between groups in a way that the total number of links per node doesn't change. The authors claim that they can combine the advantages of hierarchical systems and flat DHTs.

Freedman et al. presents in [7] Distributed Sloppy Hash Tables (DSHT). DSHT is a three layer hierarchical stack of DHTs, in the authors case, CHORD or Kademlia. Nodes are grouped as in [6] but according to their proximity in the virtual landscape, measured by *ping* values. A given value is stored in all three layers of the system. When a key is retrieved then first the lower layer DHTs queried, when the key is not found the middle is queried. Similarly, the upper layer is queried if the search in the middle layer was unsuccessful.

The later mentioned DHT structures can easily be converted into location management systems when the key-value pair is a mapping of a user Identifier (ID) to its location, useful for the MIH *Information Service*. The following papers presents such an approach.

Sethom et al. proposed in [8] a Tapestry based location management system for mobile users. Their idea is to map the hash of a service, device name, and a user identifier to a physical IP address that is stored in the DHT. Tapestry nodes are fixed nodes located somewhere in the cloud. Mobile users can retrieve location information of another user by submitting a query to one of the Tapestry nodes. When a node moves from one network to another, a binding update is submitted to the DHT to update its location information. The obsolete location information is replaced by a pointer to the new location of the updated information.

III. ARCHITECTURE

In this section we present the architecture of the decentralized *Information Service*. We propose a DHT based database that is maintained by a set mobile users. A DHT is a structured P2P system with a deterministic lookup procedure. As a DHT is fully distributed it is ideal to acts a counterpart to the centralized conventional *Information Service* maintained by a network operator. DHTs are also known to be scalable, fault tolerant, self-organizing, and provide hash table functionality [9]. For the *Information Service* we are looking in particular for a database system that provides low lookup latencies in a distributed fashion.

In our project we utilize CHORD, one of the first DHTs [10] developed. It is possible, however, to replace the DHT with any other DHT in the system. Peers, in regular DHTs or flat DHTs, locations are not known and can be anywhere in the world. This means that even though you are neighbors, your neighbor in the DHT can be physically residing at the other side of the world. This might induce more *lookup latency* and is undesirable. We know that calls and user mobility patterns have a localized nature [11]. Consequently, we want to keep DHT signaling in the callers proximity in a first stage. There are two ways of introducing locality in DHTs, by adapting the hashing procedure of keys and by employing a hierarchical architecture. Adapting the hashing, is in our believes not optimal as this affects the separation of user and location. We therefore opt for a hierarchical design, which yields higher resilience and scalability. In fact many of the largest system in the world today are hierarchically build, including the human functioning.

DHTs need maintenance, especially when peers *join* and *leave* the system and query traffic from other peers could potentially use any peer's resources. When you are willing to share your resources this is okay, but if you have unreliable channels, like wireless connections in mobile environments, P2P traffic might struggle to keep up with its maintenance. Additionally, mobile user device's have limited resource, like battery power and bandwidth. Therefore you will be less likely to participate in P2P activities on mobile devices. We opt to implement the *Information Service* at a fixed location with reliable connectivity. A dedicated server at the mobile user's home would be a perfect location or a thirty-party service could do the job. Current technology is able to provide low consumption servers dedicated for web-services and hence low environmental-footprint. Figure 1 shows a schematic overview of such a system. Each peer in the DHT maintains it own *Information Service*, all *Information Services* are interconnected so that everybody has access to any data stored in the distributed database. All the data that a user makes available in the database, the user maintains himself. This mean that the DHT in essence does not store data, but it does store a reference in which *Information Service* the data is stored. Multiple *Information Services* can also store a *value* under the same *key*. This is also known as multi-mapping.

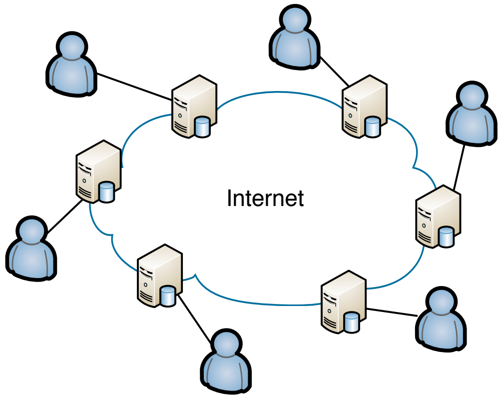

Fig. 1. Topological overview of the proposed system. The mobile users are connected the their personal MIH *Information System*. A hierarchical Distributed Hash Table (DHT) provides a distributed and decentralized database system between the personal *Information Servers*.

In the background section we have presented the most applicable state-of-the-art withe regards to hierarchical DHT design. Only one of the presented work provides built-in locality features and that is the DSHTs [7]. The authors believe that there is no other straightforward means of introducing locality in DHTs besides the DSHT concept. The DSHT has 3 tiers in the hierarchy, a simple layering of 3 different DHTs with incremental geographical range. Peers in the lowest tier are grouped if their mutual Round Trip Time (RTT) is less than 20 ms. For locality reasons, in our project we group nodes that reside in the same geographical region, in our case the same county. The second tier collects all counties of the same country, and the upper-tier groups the whole world. In such a system, the MIH *Information System lookup latency* T_{DB} seen by the mobile user is equal to

$$\begin{aligned} T_{DB} &= T_{MU\rightarrow IS} + T_Q + T_{IS\rightarrow MU} \\ &= 2\,T_{MU} + T_Q. \end{aligned} \tag{1}$$

Here $T_{MU\rightarrow IS}$ is the one-way delay between the mobile user and its *Information Service*, $T_{MU\rightarrow IS}$ the one-way delay in the other direction, and T_Q is the *Information Service lookup latency*, the *lookup latency* of the DHT. Under general assumptions we assume that $T_{IS\rightarrow MU} \approx T_{MU\rightarrow IS} \approx T_{MU}$ and thus we obtain equation 1.

Section IV will discuss the performance analysis of the proposed hierarchical DHT that acts as the *Information Service* in MIH. The aim is to give an adequate estimate of T_{DB} in equation 1.

A. Additional Considerations

Deploying a DHT for the MIH *Information System* as a distributed database can also be used for other purposes as we present in the following subsection.

The DHT storage could be extended to store the location of the users in the virtual mobile landscape, keeping a mapping between the user's location. In a centralized system, e.g. Mobile IPv4 (MIPv4), the location of users are tracked by the Home Agents (HAs). As our system is dealing with a decentralized system, an independent location register needs to be maintained. A user ID is hashed to create the key for contact information in the DHT and must be a personal identifier that separates the location from the user. The idea is to find your correspondent node without the need of knowing where or how the person resides. Some research uses identifiers including device names, or technologies. In the author's opinion this is incorrect. More suitable would be an email address or a name. A random ID could yield better privacy, but must be a shared given before the start of communication.

Additionally, the DHT overlay can be used for overlay routing satisfying QoS constraints, e.g., available bandwidth, drop-rate, etc. This is an improvement with regards to the best-effort routing service that the current Internet provides today. Overlay routing, compared to best-effort routing, on a mobile device would allow for more efficient service and network utilization, e.g., spectral or transmission power, could be taken into account [12]. This implies that data from other users could potentially be routed over your mobile device. Consequently, resources such as bandwidth and battery power of one mobile user are used for other user's purposes. Levering the overlay routing from the mobile user plane to the DHT plane could then yield better performance. Despite the extra routing dimension that are introduced, the end-to-end (e2e) delay will be significantly less and more reliable for the latter DHT routing approach. Routing optimization techniques as Self-Adaptive Multiple Constraints Routing Algorithm (SAMCRA) [13] can then be applied to compute optimal paths in the overlay.

IV. PERFORMANCE EVALUATION

This section presents the performance analysis of the presented hierarchical system. In particular we try to find an estimate of the of T_{DB} in equation 1.

We showed that T_{DB} constitutes of T_{MU} and T_Q. We can define T_Q in a flat DHT design more precisely as

$$T_{Q_{FLAT}} = (\lambda + 3)\, T_l. \tag{2}$$

Where, λ is the average number of hops needed to locate a value in the DHT, and T_l is the average one-way delay between two peers in the DHT. The term $(\lambda + 3)$ results from the fact that once the value of a *key* is located after λ hops, the value is sent to the querying peer (one more hop), then the peer contacts the other peer that maintains the actual data. The later introduces two extra hops, which yields in total $(\lambda + 3)$ hops.

For our 3 tier DHT the T_Q is slightly more complicated, T_Q is a triple stack of $T_{Q_{FLAT}}$:

$$\begin{aligned} T_Q &= \alpha\,(\lambda+3)\,T_{l1} + \beta\,(\lambda+3)\,T_{l2} + \gamma\,(\lambda+3)\,T_{l3} \\ &= (\lambda+3)\,(\alpha\,T_{l1} + \beta\,T_{l2} + \gamma\,T_{l3}) \\ &= (\lambda+3)\,((\alpha+\beta)\,T_L + \gamma\,T_{l3}) = (\lambda+3)\,\Omega. \end{aligned} \tag{3}$$

Where α, β, γ are the probabilities that what you need is located in tier 1, 2, and 3 respectively. T_{l1}, T_{l2}, T_{l3} are the average one-way delays in the different tiers. Tier 1 unifies counties and tier 2 countries, represented by their average one-way delays T_{l1} and T_{l2}. We assume that $T_{l1} \approx T_{l2} \approx T_L$ when we know that T_{l1} and T_{l2} are approximately equal. This is a correct approximation given the fact that even though your neighbor is geographical close, if the peer is connected to a different Internet Service Provider (ISP) exchanging data might traverse a country. ISPs are only interconnect at very limited number of points in their network. Thus interchanging data in between geographically close peers might yield unexpected large RTTs. Given the lack of proper statistical evidence to define α, β and γ we will assume the Pareto principle for now and set the values to: $\alpha = 0.8,\ \beta = 0.16,\ \gamma = 0.04$.

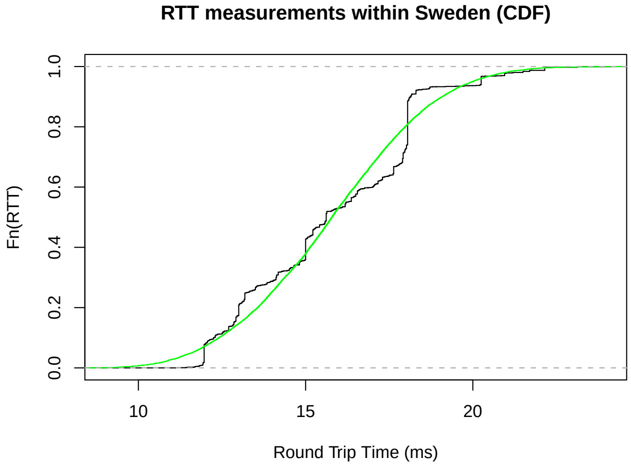

Fig. 2. The Empirical Cumulative Distribution Function (ECDF) of RTT measurements in Sweden together with the fitted Weibull Cumulative Distribution Function (CDF) with parameters $\lambda = 5.31$ and $k = 1.91$ and *offset* $= 11.18$.

A. Point-to-Point Delays

In this subsection we estimate the values of T_{l1}, T_{l2}, and T_{l3}. These are the average one-way delays in the different tiers of our hierarchical DHT which span over counties, countries, and the world respectively. To measure these values we have set up an experiment where we have pinged computer devices all around the world with the aid of the *ping* tool. *ping* measures Round Trip Times (RTTs), we can approximate our Ts by splitting the RTT in half

$$\frac{RTT_{lx}}{2} \approx T_{lx}. \tag{4}$$

The authors are currently located in Sweden, in Northern Europe. To estimate $T_{l1} \approx T_{l2} \approx T_L$ we have pinged a random subset of the Swedish IP address range, about 10^3 peers, from the academic network at Blekinge Institute of Technology (BTH) in Karlskrona. The Empirical Cumulative Distribution Function (ECDF) of the measured data is shown in Figure 2. The ECDF of the curve is not so smooth, but regardless of that the Weibull Cumulative Distribution Function (CDF)

$$1 - e^{-((x-\alpha)/\lambda)k}, \tag{5}$$

yields the lowest standard error compared to other prominent distributions. Here, λ is the scale factor, k the shape factor,

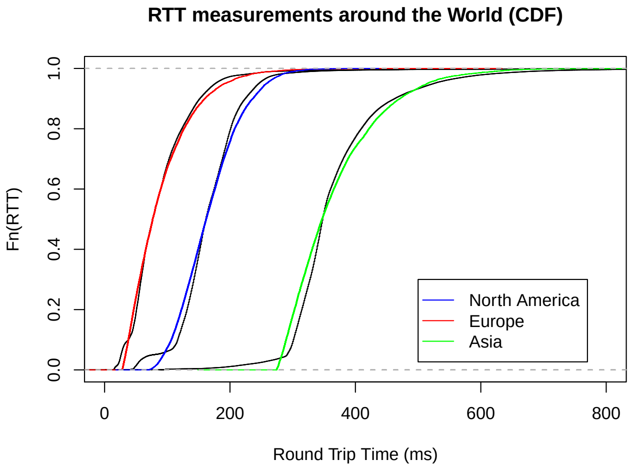

Fig. 3. The ECDF of RTT measurements around the world together with the fitted Weibull CDF. The parameters for the fittings can be found in Table I.

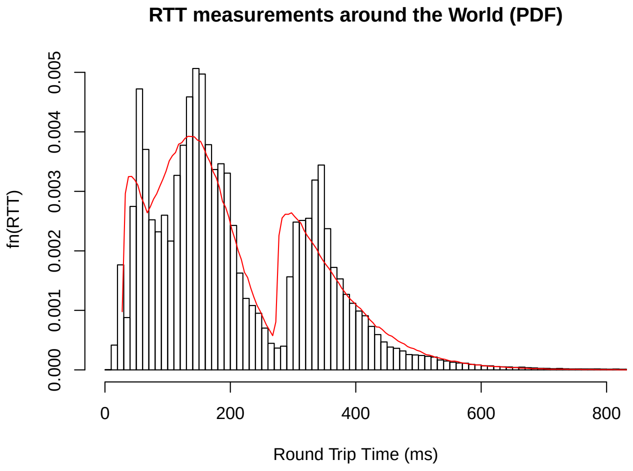

Fig. 4. The EPDF of RTT measurements around the world together with the fitted Weibull PDF. The parameters for the fittings can be found in Table I.

and α is an offset. For the values of the parameters refer to table Table I.

TABLE I
PARAMETER OVERVIEW OF THE MODELED DISTRIBUTIONS OVER OUR CONTINENTAL AND NATIONAL MEASUREMENT DATA.

Weibull distr.	λ	k	α	ratio	mean
Sweden	8.56	3.34	8.12	N/A	11.28
Asia	100.37	1.21	273.78	$\pi = 0.32$	367.92
Europe	65.68	1.17	27.89	$\mu = 0.28$	90.07
North-America	112.97	2.09	67.57	$\kappa = 0.4$	167.63

Similarly we have pinged peers all over the world. As the surface of the earth is not uniformly distributed with computer over its surface the model is rather complicated. Therefore we will make an approximation to simplify the model by using a mix of three distribution including the RTT measurements from Europe, North-America, and Asia as seen from Sweden. These constitute 91% of all the gathered data, and thus should be adequate to make an appropriate model of world-wide RTT. This includes about $300 \cdot 10^3$ devices for North America, and around $200 \cdot 10^3$ measures for Europe and Asia.

We can model these three continents with Weibull distribution as well, we observe that the head of the CDFs are unusually large for Weibull distributions. We are seeking a fairly simple model and will therefore disregard the heads. Asia, Europe, and North America can then be approximately modeled with a Weibull distribution, see equation 5. The parameters are listed in Table I. Figure 3 depicts the individual CDFs of the continents of concern. The RTT model of the world is the combined model of the three continents each multiplied a given probability factor. These probability factor is computed by measurements points occurrence ratio in the whole data set

$$T_{WORLD} = T_{l3} = \kappa T_{NA} + \mu T_{EUROPE} + \pi T_{ASIA}, \quad (6)$$

where $\kappa = 0.4$, $\mu = 0.28$, and $\pi = 0.32$. Figure 4 depicts the Probability Distribution Function (PDF) of our model against the Empirical Probability Distribution Function (EPDF) of the measured RTTs. Given that we assumed some simplifications our model of world-wide RTT is a fair approximation with a mixture of three Weibull distributions.

Knowing that the average of a Weibull distribution is

$$E[\text{Weibull}] = \lambda \Gamma\left(1 + \frac{1}{k}\right) + \alpha, \quad (7)$$

we can compute the expected values of Ω, T_L, and T_{t3}, in units of ms, using the equations 3 and 4:

$$E[\Omega] = 9.605,\ E[T_L] = 5.64,\ E[T_{l3}] = 104.99 \quad (8)$$

These models and values we will use in the random processes of our simulation to get more realistic results.

V. SIMULATION OF THE MIH INFORMATION SERVICE DATABASE

A. *Simulation Environment*

For the simulation the MIH database we implemented the hierarchical DHT in the Oversim simulator environment [14]. Oversim is an open-source package for the OMNeT++ discrete event simulation [15] and runs on top of the INET framework. The latter provides wired and wireless network protocol support in OMNeT++. Oversim supplies facilities especially for overlay technologies. Accordingly, Oversim provides good support for DHT simulation. We adapted the Oversim code to support hierarchical DHT design. The simulations were run on a Dell Inspiron 8500, 2.5 GHz, 1 GB RAM, with Ubuntu 10.04 LTE installed running on Linux version 2.6.32.

Figure 5 shows the implemented simulator given a 3 layer hierarchical DHT structure. In this example a dummy simulation is run with 10 nodes on the top tier (*tier 0*), 5 in the middle (*tier 1*), and 3 in the lower-tier (*tier 2*). For simplicity and performance reasons we only simulate a single tier 2

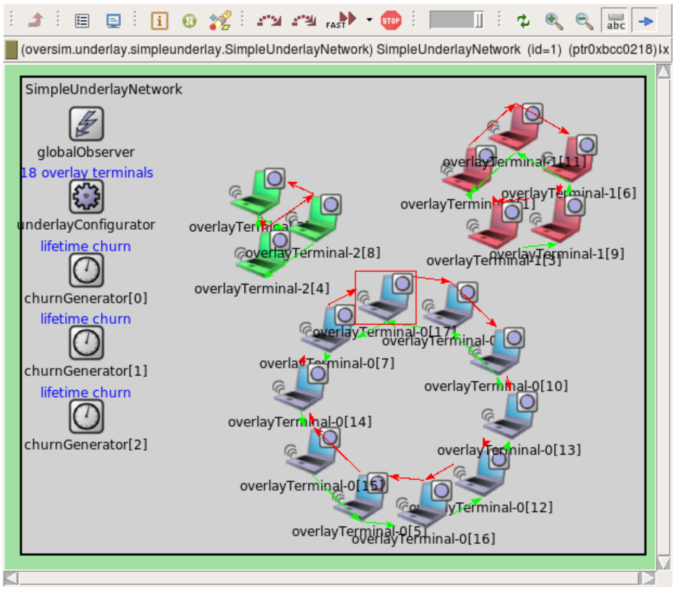

Fig. 5. Overview of the implemented hierarchical DHT simulator. A simulation is shown with 10 nodes on the top tier, 5 in the middle, and 3 in the lower-tier.

layer, together with its tier 1, and tier 0 layer. The maximum amount of total nodes simulated is 11100, which is close to the physical memory limit of the device that was used for simulation. For most simulations we simulated a hierarchical system with one DHT on each tier. This is a simplification to address the physical limitations of the used hardware. We assumed that each tier is ten times larger, unless stated differently, then the tier below.

B. *Performance Metrics*

We are interested in the performance of the MIH Information Service database from the end user's point of view. Therefore the metric of concern is the lookup time of values in the database, which we refer to as *lookup latency*. In the series of simulations we will analyze the *lookup latency* under different configurations of the hierarchical DHT. The default settings of concern used in the simulator are given in Table II. The second column of the table presents the default nodes per layer, the third shows the ratio by which the information that you are looking for is found or not, the forth column provides the amount of information stored per user and per layer, and the last column gives the time interval by which each node looks up an entry in the DHT. All simulations were simulated for one hour of simulation time. Given that all nodes are performing the same operations, one hour simulations are enough to yield a representable sample of the system's behavior.

TABLE II
DEFAULT SETTINGS OF THE SIMULATION OF THE HIERARCHICAL DHT FOR THE MIH INFORMATION SERVICE DATABASE.

Layer	# nodes	hit-ratio	# entries	Lookup int.
tier 0	3000	1	10	N/A
tier 1	300	0.8	10	N/A
tier 2	30	0.8	10	60 s

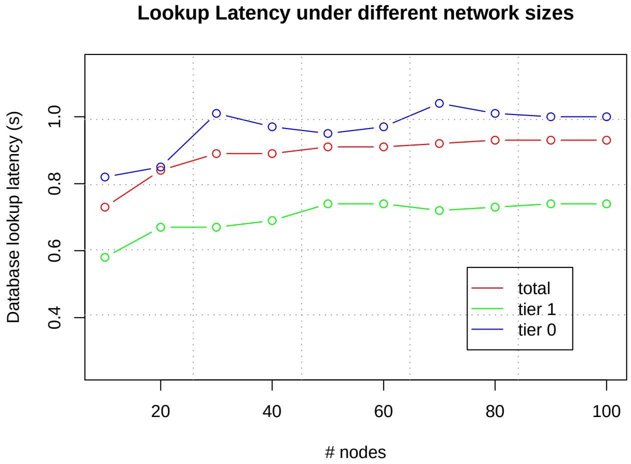

Fig. 6. The *lookup latency* with regards to the network size. The database *lookup latency* is given including *tier 1* and *tier 0*. The *lookup latency* converges for larger networks size, indicating scalability.

During the different simulation runs, presented in the next section, we will examine the *lookup latency* under different configurations compared to the values in Table II.

C. *MIH Information Service database simulation results*

1) The hierarchical DHT size: This batch of simulations analyse the influence of the amount of nodes in the system to the *lookup latency*. We ran 10 simulations where the size of the network was scaled as compared to the settings in Table II. The number of nodes in the simulated system for *tier 2* range from 10 to 100, in increments of 10 nodes. The other tiers were scaled proportionally with regards to *tier 2*. The results of this experiment are shown in Figure 6.

The database *lookup latency* is given (red curve) including *tier 1* (green) and *tier 0* (blue). The *lookup latency* increases rapidly for smaller network sizes but stabilize after network sizes of magnitude 2 or 3 larger. As a result the *lookup latency* converges for larger networks size to ≈ 0.93 s, this indicates scalability. This is not remarkable as DHTs are know to scale well, as compared to unstructured P2P technologies.

The difference in lookup times for *tier 1* is smaller then *tier 0* as a result of the link delays. Recall, that *tier 0* spans the whole world whereas *tier 1* spans across a country. The total *lookup latency* is larger then *tier 1* because there is a probability that a lookup spans across multiple tiers, which affects the total lookup latency.

Real-life scenarios will highly likely pertain more user/nodes. According to these graphs the performance will not likely be affected greatly. We were unable to simulate larger networks to verify this as our simulation hardware limited our capabilities.

2) The query rate of the database: Next we analyze the effect of the query rate of nodes on the performance of the

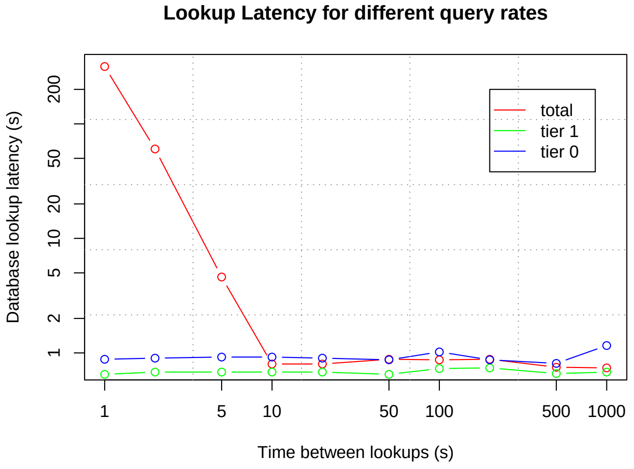

Fig. 7. The effect of the query rate on the lookup performance of the database. Low query rates yield lower lookup latencies. For high lookup rates, the database performance is largely affected and might be even unusable.

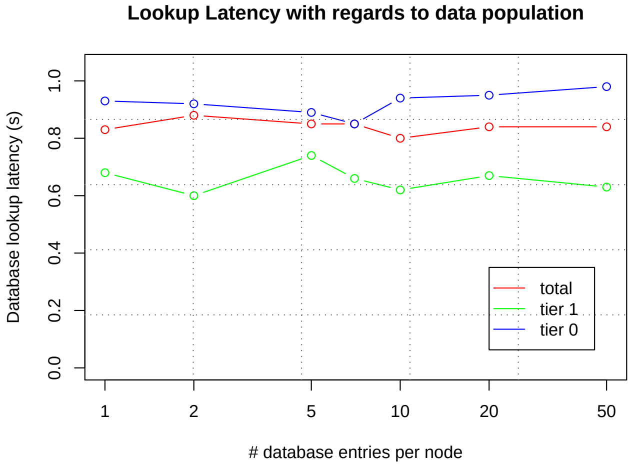

Fig. 8. The increasing amount of stored entries in the database are depicted in relation to the *lookup latency* of the system. The amount of database entries does not seem to affect the performance of the database.

system with regards to the lookup latency. We decreases the query rate from 1/s to 0.001/s in logarithmic steps. The results are given in Figure 7. Note that both axes are logarithmically scaled. The abscissa is the average amount of time between lookups in the database. The query rate is equal to the inverse of the abscissa.

For a lookup rate of 0.1 and lower, or a lookup every 10 s and higher, we get similar behavior as compared to the results in Figure 6. However, for lookups lower then 0.1 we see that the lookup rate increases logarithmically, for query rate 1 the *lookup latency* is even 316.81 s. This is due the overload of the network, e.g., larger queueing times then normal, as requests to the database are very frequent. This is an indication that the system is vulnerable for Denial of Service (DOS) attacks by flooding the database, in our case a Distributed Denial of Service (DDOS) attack. In reality, under normal conditions, an extreme high query rate will not occur by all nodes at the same time. Isolated nodes could show bursts in query rates from time to time, when the mobile users are in critical situations. These burst, however, will not affect the overall performance of the database if they happen occasional and sporadically.

The *tier 1* and *tier 0* lookup latencies seem not to be affected. This is understandable as the query rate of these tiers are lower by a factor of β and $\beta\gamma$ respectively compared to *tier 2*.

3) Number of entries in the database: We now investigate the affect of the amount of entries that is stored in the database with regards to the *lookup latency* of the database. During the different simulation runs the users will increasingly store more entries in the database. We then see if the *lookup latency* gets effected by this. Figure 8 presents the output of the simulation batches.

Overall the curves tend not to decrease nor increase in a consistent manner with increasing database entries in the system. This is a good property of the system as this indicates scalability of the hierarchical DHT architectures. In our simulation we assumed a constant distribution with regards to the entries in the database owned by a mobile user. Real life scenarios will probably have a minimum of entries with no or a large maximum limit. The maximum limit will highly likely not affect the performance of the system greatly, unless these values are updated frequently.

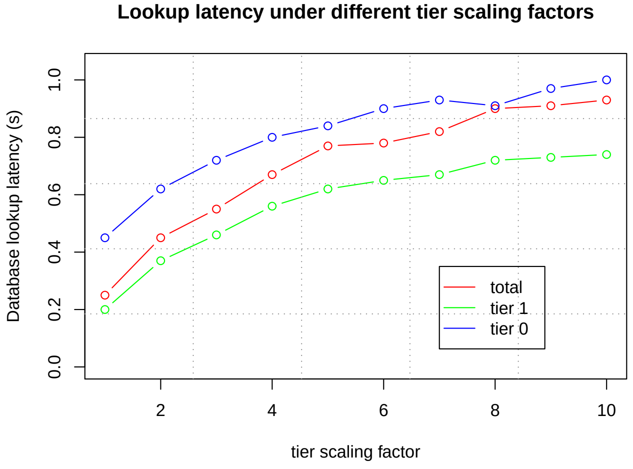

Fig. 9. Influence on the lookup latency under different tier scaling factors. Smaller scales seem to perform better but will yield smaller data hit-rates.

4) Size of the tiers: Previously we assumed that each tier is ten times larger then the tier underneath. This might not necessarily hold in reality. Therefore we analyze the influence of changing the tier size scaling. Before, the ratio was 10, in these simulation runs we will decrease this scaling in steps. The outcomes of this experiment are shown in Figure 9.

We observe that all the curves in Figure 9 tend to stabilize, which is a desirable feature. Smaller scaling factors tends to

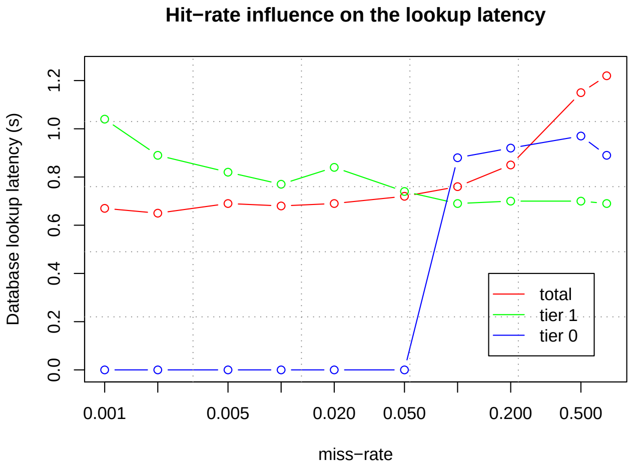

Fig. 10. Effect of changing the hit-rate on the lookup latency of the MIH Information Service. The higher the hit-rate the more efficient the system becomes with regards to lookup latency.

yield lower lookup latencies. This is makes sense as fewer nodes reduce the number of hops during lookup. One might conclude that smaller scalded tiers are beneficial, but remind that the probability that you will find what you are looking for in the database decreases with decreasing scaling factor. Thus the scaling factor enables a trade-off between *lookup latency* and *hit-ratio*.

5) The tier lookup hit-rate: Due to a lack of supportive statistical information we assumed that the data a peer is looking for in that tier is present with 80 % chance, we assumed the Pareto principle previously. We now study the consequence of changing the lookup hit-rate. Our aim is to introduce locality in the database this means data about a close neighbor should be stored virtually close to its location. Not finding the data in a tier indicates: its not available or its stored geographically elsewhere. During these runs we decreased the hit-rate of the tiers. The results are presented in Figure 10.

The abscissa displays the *miss-rate*, which is the complement of the hit-rate. We observe that the total average *lookup latency* increase as the hit-rate decreases. This is easily explained as the results of multi-tier lookups when an entry is not found. Lookups in multiple tiers creates extra latency. Interesting to observe is also that as the hit-rate up till 99.5 % is high, *tier 0* doesn't even have to perform any lookups, whereas *tier 1* seems to have some work to in all scenarios.

VI. CONCLUSION

In this paper we presented a decentralized approach to the the Information Service of the Media Independent Handover (MIH) framework. We proposed an hierarchical approach based on Distributed Hash Tables (DHTs). We discussed the performance analytically and presented a set of simulation results. We have shown that the hierarchical Information system is close to scalable with regards to the network size and the amount of entries stored in the system. The system is however prone to Denial of Service (DOS) attacks by flooding, this must be addressed in some manner.

Future work includes the study whether there are other types of DHTs that might perform better then the DHT, CHORD, that we used in this study. Some security issues must be addressed. Also, the effect of the entry updates in the database should be studied to get a better picture of the performance of the system. The simulated values of the *lookup latency* should also be tested for normality. Deviance of the normal assumptions might yield a bias in the estimates of the *lookup latency*.

ACKNOWLEDGEMENTS

We would like to the thank the Institute of Electrical and Electronics Engineers (IEEE) for accepting our proposal and funding our research. We also want to express our greatest gratitude to our supervisor David Erman for his guidance and enthusiasm.

REFERENCES

[1] IEEE802.21, "Tutorial," IEEE 802.21 session nr 15, 2006.
[2] IEEE, "Draft ieee standard for local and metropolian area networks: Media indipendant handover," *P802.21/D8.0*, December 2007.
[3] InterDigital, "Media independent handover," white paper, 2009.
[4] E.Piri and K. Pentikousis, "Ieee 802.21: Media-independent handover services," *The Internet Protocol Journal*, vol. 12, no. 2, pp. 7–27, 2009.
[5] L. Garces-Erice, E. W. Biersack, K. W. Ross, P. A. Felber, and G. Urvoy-Keller, "Hierarchical p2p systems," in *Proceedings of ACM/IFIP International Conference on Parallel and Distributed Computing (Euro-Par)*, ser. LNCS, vol. 2790. Klagenfurt, Austria: Verlang, August 2003, pp. 1230–1239.
[6] P. Ganesan, K. Gummadi, and Garcia-Molina, "Canon in g major: Designing dht's with hierarchical structure," in *24th International Conference on Distributed Computing Systems*, Tokyo, Japan, March 2004.
[7] M. J. Freedman and D. Maziéres, "Sloppy hashing and self-organizing clusters," in *IPTPS*, 2003, pp. 45–55.
[8] K. Sethom, H. Afifi, and G. Pujolle, "A distributed architecture for location management in next generation networks," in *International Conference on Wireless Networks, Communications and Mobile Computing*, June 2005.
[9] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys and Tutorials*, vol. 7, pp. 72–93, 2005.
[10] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *SIGCOMM Comput. Commun. Rev.*, vol. 31, pp. 149–160, August 2001. [Online]. Available: http://doi.acm.org/10.1145/964723.383071
[11] Z. Mao and C. Douligeris, "A distributed database architecture for global roaming in next-generation mobile networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 1, pp. 146–160, 2004.
[12] A. Popescu, D. Ilie, D. Erman, M. Fiedler, A. Popescu, and K. D. Vogeleer, "An application layer architecture for seamless roaming," in *6th International Conference on Wireless On-demand Network Systems and Services*, February 2009.
[13] D. Ilie, "Optimization algorithms with applications to unicast qos routing in overlay networks," Research Report No.2007:09, Blekinge Institute of Technology, September 2007.
[14] I. Baumgart, B. Heep, and S. Krause, "OverSim: A Flexible Overlay Network Simulation Framework," in *Proceedings of 10th IEEE Global Internet Symposium (GI '07) in conjunction with IEEE INFOCOM 2007, Anchorage, AK, USA*, May 2007, pp. 79–84.
[15] A. Varga, "The omnet++ discrete event simulation system," *Proceedings of the European Simulation Multiconference (ESM'2001)*, June 2001.