DHT-based Localized Service Discovery in Wireless Mesh Networks

Hanno Wirtz, Tobias Heer, Martin Serror, Klaus Wehrle
Chair of Communication and Distributed Systems, RWTH Aachen University
{wirtz, heer, wehrle}@comsys.rwth-aachen.de, martin.serror@rwth-aachen.de

***Abstract*—Wireless mesh networks (WMNs) provide high-bandwidth wireless network access to mobile clients in extensible, robust multi-hop networks. WMNs support distributed service provision and data storage, catering to the advanced capabilities of current mobile devices. Services and data discovery using undirected broadcast or multicast messages, as in traditional discovery protocols, significantly harms network performance due to interference and collisions. In contrast, distributed hash tables (DHTs) offer consistent mapping of service and data identifiers to the providing devices and therefore allow a directed unicast discovery and access. However, traditional DHTs place identifiers at arbitrary distant devices in the network, resulting in frequent use of long multi-hop routing paths. Such multi-hop transmissions suffer from performance loss at each hop and also degrade the overall network performance. We propose DLSD, a DHT-based localized index structure that establishes a hierarchy of locally bounded address spaces ranging from a few nearby devices to the whole network. Iterating through this hierarchy bottom-up allows devices to find the most local provider of the requested item, thereby minimizing multi-hop transmissions while ensuring global reachability. Through this reduction of routing hops, we maintain high transmission performance and minimize interference in the network. We evaluate the feasibility of our approach and show that it significantly reduces routing overhead and outperforms traditional service discovery and DHT approaches.**

I. Introduction

Wireless mesh networks (WMNs) provide network connectivity to mobile clients in large geographic areas by forwarding packets exclusively over wireless links in a mesh topology. Distributed over the area covered by the WMN, wireless mesh routers and clients may access and offer services or communicate with each other. However, discovering services and communicating via multi-hop links inevitably reduces the network performance because of intra-link and inter-link interference [1]–[3]. Hence, frequent multi-hop forwarding leads to a significant service degradation. For example, a client may stream a multimedia file with a throughput of several MBit/s via a one-hop link but will experience a trickling few KBit/s over multiple hops [1]. Using broadcasts can further degrade the performance because their delivery involves transmissions to every device in the network [4]. However, traditional service discovery protocols require multi-hop multicasts or broadcasts to locate services in the network as they do not use location information [5], [6]. We thus identify two distinct factors that reduce the network performance: first, interference caused by broadcasts in service discovery and second, frequent communication via long multi-hop transmissions to reach services.

To alleviate these factors, discovery and use of services benefit from mechanisms that facilitate the use of nearby services and do not require broadcasts or network-wide multicasts. Furthermore, in many cases *localized* storage and retrieval of keys, i.e., confined to a certain area, is preferable over a network-wide approach. If redundant copies of data items or redundant services are available in the network, finding the closest copy suffices and should have preference for performance reasons. Also, an information may only be of interest for devices in the close vicinity resulting in high overhead when using network-wide lookups. Hence, in addition to ensuring global accessibility of keys, efficient provision of services and data benefits from local communication between devices that are close, i.e., separated by only a small number of hops.

Distributed hash tables (DHTs) are lookup structures that provide a *put()*/*get()* interface to store and retrieve information, for example to publish and discover services or devices in a network. In a DHT, clients and routers store *(key, value)* pairs in which data items, services or the location of mobile clients map to keys. In previous work we proposed Mesh-DHT [7], a custom WMN overlay for a DHT-based look-up structure. Mesh-DHT reflects the locality of mesh routers in the overlay construction and forms the basis for a consistent, distributed storage and retrieval mechanism that is provided exclusively by the mesh routers. The goal of Mesh-DHT is to provide an overlay routing mechanism that aligns the physical and overlay network topology for efficient routing between mesh routers in the look-up structure. However, Mesh-DHT can not represent the locality of keys. Hence, directly using the overlay as a network-wide DHT to store and retrieve keys may result in frequent use of long multi-hop routes between distant routers.

In this paper, we propose DLSD (DHT-based Localized Service Discovery), a hierarchy of localized DHT address spaces that enable localized provision and discovery of services and data. Building on the locality-oriented design of Mesh-DHT, DLSD establishes a global DHT address space and partitions it into different scopes with different degrees of locality that are hierarchically organized in *levels*. At different levels, the distance at which data items are visible differs, ranging from locally available data items to network-wide visibility. Hence, by choosing an appropriate level, a lookup can be restricted to only consider locally available information. Likewise, the visibility of data items can be limited to mesh routers in the vicinity by publishing the respective keys on appropriate levels. Specifically, we make the following contributions:

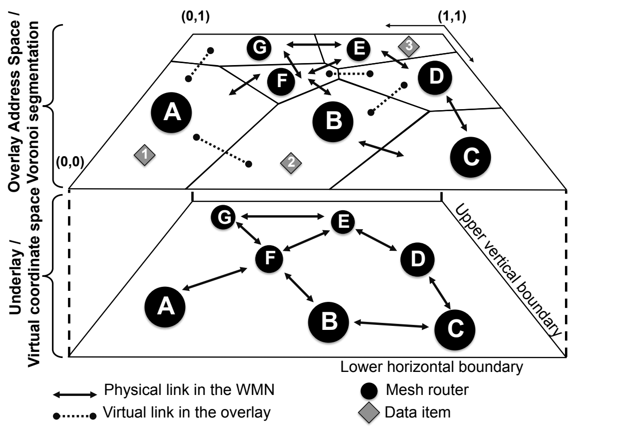

Overlay Address Space / Voronoi segmentation
Underlay / Virtual coordinate space
(0,1)
(1,1)
(0,0)
Upper vertical boundary
Lower horizontal boundary
G
E
F
D
A
B
C
1
2
3
Physical link in the WMN
Virtual link in the overlay
Mesh router
Data item

Fig. 1. Schematic representation of underlay and overlay in Mesh-DHT. Single-hop overlay links map to underlay links of different length. The Voronoi partition of the overlay address space provides a basis for a global DHT scope (level 0) in the network.

i) We enable local service discovery over small hop counts in WMNs by establishing local, hierarchic DHT address spaces (scopes).
ii) We enable devices to control the scope of their store and retrieve operations and allow for a fine-grained trade-off between visibility and overhead.
iii) We provide a mechanism that establishes additional and removes unnecessary local scopes based on the current network topology, router density, and use of the DHT to control the overhead of the hierarchy.
iv) We analyze the performance of the system and show its feasibility in large-scale simulation as well as in a real-world testbed consisting of 45 mesh routers.

The remainder of this paper is structured as follows. In Section II we briefly introduce the design of Mesh-DHT as the basis for the hierarchy of local scopes in DLSD. We introduce our approach to hierarchical local service discovery in Section III and discuss its applications in Section IV. We evaluate the characteristics of our approach and compare it against service discovery protocols and other DHT approaches in Section V. Section VI discusses related work in service discovery in WMNs and Section VII concludes the paper.

II. Review: Mesh-DHT, a locality-based overlay

Mesh-DHT establishes a consistent overlay in WMNs that reflects the physical locality of mesh routers in the virtual overlay locality. Fig. 1 shows a schematic scenario that highlights the design characteristics. By aligning the underlay and overlay topology, we aim to avoid indirections when routing between points in the overlay address space.

Mesh routers assign themselves *virtual coordinates* based on the coordinates of their physical neighbors. The distance between the coordinates of routers then reflects their relation, i.e., whether they are one-hop, two-hop or multi-hop neighbors. Given a coordinate distribution among routers, a rectangular bounding box around these coordinates constrains the coordinate address space. Fig. 1 shows the set of mesh routers, the physical links between them and the coordinate address space that constitute the *underlay* in Mesh-DHT.

Mesh routers make use of the topology information reflected in virtual coordinates to construct and segment the overlay address space in close alignment to the underlay. They establish a *Voronoi partition* [8] of the coordinate address space based on the current coordinate distribution. Routers calculate the partition in a fully distributed fashion, i.e., each router determines only its Voronoi cell using the coordinates of adjacent routers in the partition. As shown in Fig. 1, the overlay address space and virtual neighbor relations in the partition constitute the *overlay* in Mesh-DHT.

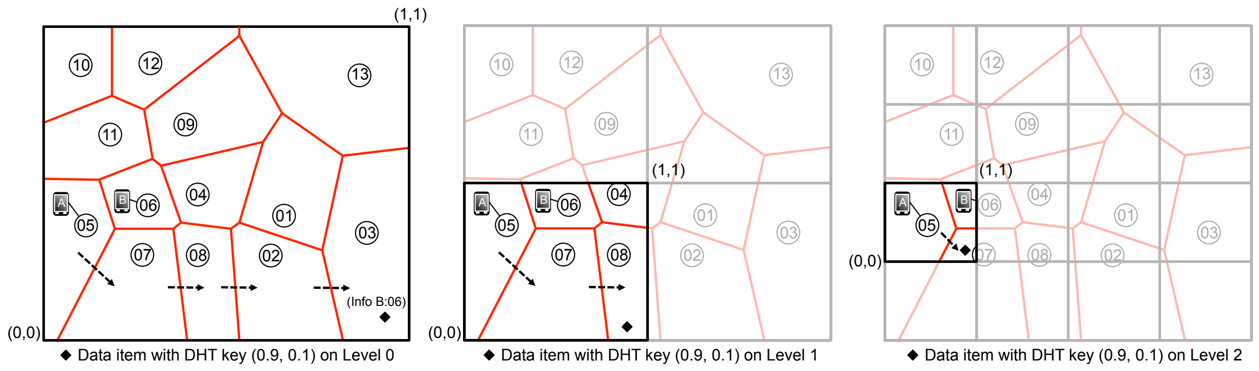
Neighbor relations in the overlay serve as virtual links that map to (multi-hop) links between mesh routers in the underlay. The set of virtual links, together with a Euclidean distance metric between coordinates in the overlay, then allows routing between arbitrary overlay coordinates. In [7], we show that Mesh-DHT achieves a close alignment in routing between underlay and overlay and provides a consistent overlay that caters to the requirements of WMNs. Mesh-DHT thereby reduces the routing stretch between points in the overlay, i.e., the number of underlay hops required in overlay routing.

However, for large networks, lookups of data items become slow and have high overhead because they need to traverse long multi-hop paths to reach the mesh router that manages the corresponding key – even if the actual source and the requester of the key are close. This paper proposes DLSD, in which we provide localized DHT-based service and data discovery by reducing the actual distance to overlay keys. We thereby reduce triangular routing in WMNs and further minimize the need for long multi-hop transmissions that harm network performance.

III. Design

A DHT that builds on the overlay partition of Mesh-DHT distributes the responsibility for storing and managing data items among all mesh routers in a WMN. Fig. 1 shows the mapping of the abstract $(0,1) \times (0,1)$ DHT address space onto the overlay partition. In the DHT, each data item maps to a key by hashing its name or identifier using a hash function H (e.g., SHA-1 or MurmurHash). Keys then map to a topological location in the overlay routing topology. The Voronoi partition distributes the responsibility for these locations among the mesh routers, each router thus stores a small subset of the data items available in the network. Due to the pseudorandom output distribution of H, data items are mapped to keys at arbitrary locations in the DHT address space. As such, a device storing a data item in the DHT, e.g., its current IP address, has no control over the location of the mesh router that stores the item. Hence, even information that is stored and requested by nearby devices may be managed by routers in far areas of the mesh network. Fig. 2(a) illustrates this problem. Although both clients A and B are associated to nearby mesh routers, the look-up of the IP address of client B by client A must be routed to the far router 03 along 4 hops in the network.

In Mesh-DHT, the size of the address space encloses the whole network and correlates with the underlay WMN geometry and network size. Hence, the network size and accordingly


(1,1)
(0,0)
10
12
13
11
09
04
01
03
05
06
07
08
02
A
B
(Info B:06)
◆ Data item with DHT key (0.9, 0.1) on Level 0
◆ Data item with DHT key (0.9, 0.1) on Level 1
◆ Data item with DHT key (0.9, 0.1) on Level 2

(a) Level 0 (global), key managed by router 03. (b) Level 1, key managed by router 08. (c) Level 2, key managed by router 07.

Fig. 2. Example hierarchy of address space levels in DLSD. Clients A and B are associated to adjacent mesh routers. Client B publishes its IP address in the DHT at key $H(\text{"Client B"}) = (0.9, 0.1)$ in all available address space levels. Client A retrieves this information, the key look-up originating at router 05 spans 4 hops (level 0), 2 hops (level 1) and 1 hop (level 2), respectively.

the size of the address space are directly correlated with the negative impact of frequent long multi-hop transmissions that are required to retrieve a data item. The core idea of DLSD is to establish smaller local address spaces that only span parts of the WMN to reduce the hop count of look-ups.

DLSD thus leverages the locality of devices and data in the WMN to provide local, hierarchical DHT address spaces that allow the storage and retrieval of keys in confined areas. We refer to each of these address spaces as scopes, which are organized in *levels* in the DHT (cf. Fig. 2), with level 0 as the global address space and smaller, more local subspaces on higher levels. Scopes on lower levels thus enclose higher numbers of mesh routers whereas scopes on higher levels enclose iteratively less routers. Devices can choose to store keys in local and global scopes by indicating a level, depending on the preference for the respective data item. For example, mesh routers that provide Internet gateway functionality may chose to advertise this functionality *only* in a local scope because distant clients are expected to select a closer gateway to preserve transmission performance.

The segmentation in levels does not affect the topology or routing logic of the DHT. Hence, the established neighbor relations and routes stay identical for each scope on each level. However, for each scope, the mapping between a data item's key and its assigned position changes. Data items that are present in multiple scopes are thus duplicated and managed by multiple mesh routers.

In the remainder of this section, we first introduce our design of establishing local DHT sub-address spaces given a Mesh-DHT overlay. We then detail how storage and retrieval operations are supported in DLSD. To manage the degree of redundancy, introduced by the additional scopes, we propose a mechanism to dynamically establish levels of local address spaces based on the usage and density of the current DHT.

A. Establishing Local DHT Address Spaces

Fig. 2(a) shows the DHT address space on a Voronoi partition of the 2D overlay address space in an example network. The DHT address space ranges from $(0, 0)$ and to $(1, 1)$ and stretches over the physical expanse of the whole mesh network.

Algorithm 1 Address space level mapping

```
 1: function MAP_TO_LEVEL($Router(x, y), Key(a, b), Level\ l$)
 2:     $c = (\text{CALCULATE\_ORIGIN}(x, l) + a \cdot 1/2^l)$
 3:     $d = (\text{CALCULATE\_ORIGIN}(y, l) + b \cdot 1/2^l)$
 4:     return $Key(c, d)$
 5: function CALCULATE_ORIGIN($Coordinate\ z, Level\ l$)
 6:     $z_b = 0$
 7:     for $(i = 1; i \leq l; i++)$ do
 8:         if $z > 1/2^i$ then
 9:             $z = z - 1/2^i$
10:             $z_b = z_b + 1/2^i$
11:     return $Coordinate\ (z_b)$
```

In practice, the outermost routers of the physical mesh network are very likely to be located at the edges of the DHT address space. In DLSD, we use this view as the global scope where all routers in the network can publish and lookup information.

DLSD introduces local scopes by iteratively segmenting the global address space in a region quadtree [9]. The depth of the quadtree determines the degree of the segmentation and the expanse of the local address spaces. On level 0, the address space is not segmented and maps to the overlay address space of Mesh-DHT. On level 1, each scope is a sub-address space that covers a fourth of the level 0 address space. Fig. 2(b) shows four sub-address spaces with $x, y \in (0, 0) - (0.5, 0.5), (0, 0.5) - (0.5, 1)$, etc. The local scope of router 05 is highlighted. Devices can publish information in local scopes equivalent to publishing in the global address space (level 0). A value maps to the same key (a, b) in every sub-address space (e.g., level 1) as on level 0. However, due to the segmentation of the address space and the resulting different boundaries of scopes, this key is managed by a different mesh router in different scopes. Fig. 2 shows where the same key is managed in local scopes on successive segmentation levels.

Establishing multiple levels of address spaces does not affect DHT routing because coordinates in local scopes can easily be mapped to level-0 coordinates, where they are routed by Mesh-DHT. Mapping a coordinate of level l to the according coordinate in level 0 depends on the location of the calculating router and the level. To perform the mapping, a router must determine the local address space boundaries, i.e., the scope, of the level in which it intends to publish or look up a key.

Using Algorithm 1, devices may calculate this mapping locally and autonomously. For both axes of the 2D sub-address space, the algorithm maps the origin $(0, 0)$ of the level l address space into the level-0 address space (lines 5 - 11). It then shifts the coordinates by the position of the origin and scales the coordinates of the sub-address space to the coordinates of the level-0 address space (lines 2 and 3). In the example in Fig. 2, for router 05, the coordinate $(0.9, 0.1)$ on level 1 maps to the coordinate $(0.45, 0.05)$ on level 0. Hence, the level 1 request for $(0.9, 0.1)$ only has to be routed to the closer router 08 instead of the more distant router 03 on level 0.

Note that in Fig. 2(b), routers 04, 05, 06 and 08 are responsible for coordinates outside of the highlighted quadrant in which their coordinates lie. However, by mapping these level-1 coordinates to level-0 coordinates, the actual segmentation is not visible to the routers that manage these overlapping regions. Hence, such occurrences require no special handling.

B. Storage and Retrieval of Values

Mesh routers store keys and values along with the designated address space level, extending the traditional $(key, value)$ pair in the *put()* primitive to a $(key, value, level)$ tuple. Hence, in difference to the use of quadtrees in tree structures, store operations in DLSD occur bottom-up, with routers selecting the specific level or levels in which a key is stored, e.g., level 3 and 2 or only level 2. Upon reception of a $(key, value, level)$ tuple, a router checks whether the respective coordinate belongs to its Voronoi region and, if positive, stores the tuple. In this calculation, it uses the coordinates of the requesting router to determine the correct origin of the sub-address space. Given the overlapping responsibilities of single routers in the segmentation, routers may store tuples that map to a coordinate outside the router's quadrant on the indicated level, e.g., router 06 in Fig. 2(b).

To retrieve a value, devices indicate the level on which a key should be retrieved, extending the $get(key)$ primitive to a $get(key, level)$ primitive. This enables an iterative search for keys that starts on a local level and subsequently proceeds to higher levels. Additionally, devices may simultaneously issue requests for a key on multiple levels and subsequently choose the most local response. Different to searches in quadtrees that originate at the root node and proceed in a top-down binary search, this enables local look-ups in bottom-up fashion.

Storage of data items on multiple levels establishes a degree of replication and redundancy in the network. However, a data item may be managed by the same router on several levels on which it is stored. Therefore, storing a data item on multiple levels is insufficient to ensure its availability in case of node failures. To achieve reliable replication and redundancy, using multiple hash functions to derive different key mappings are appropriate analog to traditional DHT approaches [10].

C. Dynamic Creation and Deletion of Levels

WMNs are extendable per design through the addition of mesh routers and typically possess dense as well as sparsely populated areas. A fixed or pre-configured depth of address space levels thus does neither consistently fit the network topology nor the DHT usage in local areas. We employ a dynamic mechanism to determine the need for additional levels based on the usage of the DHT. While we argue that at least one local level, as in Fig. 2(b), is sensible in most cases, the creation of more local scopes can be done dynamically and locally. For example, router 13 is already the only router in its level-1 scope. Further segmenting this address space would create empty level-2 scopes. Selecting the appropriate depth of levels for a certain region of the WMN allows a tradeoff between reduced routing overhead in smaller scopes and increased storage overhead for maintaining redundant copies of data items in multiple scopes. We employ the following procedure to create and delete local scopes:

Routers decide on the need for additional levels based on the usage of the DHT. They monitor the overhead of lookups and storage operations on the current scopes by observing the source coordinate of the routers that store or request data of keys managed by this router. Moreover, the managing mesh router observes the WMN hops that were necessary to forward a $put()$ or $get()$ operation. It returns these statistics to the origin of the operation in a STAT data container even if no data item is found in the scope.

If the origin of a $put()$ operation deems the distance to its stored information too large, it can re-insert the data item indicating a higher level. Similarly, if the origin of a $get()$ operation observes frequent multi-hop forwarding it can request its neighbors to insert their data items at higher levels. This request contains the origin's address (x, y) and the proposed new level l. By using the coordinates (x, y) and l, each neighbor can calculate if it manages parts of the proposed new scope. If the neighbor is part of the new scope, it forwards the request, containing (x, y) and l to its neighbors. The process continues until all routers that belong to the scope space have taken notice. Through these requests, the concerned routers are encouraged to publish local information at the new level. Similarly, origins of $put()$ and $get()$ operations that receive STAT containers with very low WMN hop counts may indicate lower levels. With this mechanism, levels are dynamically created or deleted based on the density, use, and topology of the WMN.

Conceptually, there is no need for a network-wide synchronization of the depth of levels. Devices simply insert or look up keys indicating a level that they deem appropriate and, with respect to this level, identify the managing router using Algorithm 1. Independent from the view of other routers, routers simply store data items along with the designated level and responds to requests for the respective key. However, harmonizing the use of levels in WMN areas is beneficial to *a)* avoid insert operations on levels with one or few routers and *b)* avoid lookups in empty levels without data items.

IV. Applications of DLSD

So far, we described the benefits of DLSD for abstract $put()$ and $get()$ operations. However, these generic primitives, in combination with local address spaces, directly enable many important service discovery and name resolution functions in WMNs. In this section we briefly highlight how DLSD can improve the performance of three mechanisms: a) a fully distributed DHCP-like IP address management, b) DNS functionality and c) support for mobile devices. In d) we show how DLSD can support the provision of composed services as envisioned by i3 [11].

Host Configuration: There are two options to provide a DHCP-like service using DLSD. First, the DHT may globally store all assigned addresses distributed over the mesh routers, thereby removing the need for a central DHCP instance. A new mesh router or client in the network can select any IP address A and queries the key $H(A)$. If no entry for that key exists, the address is free and the router can use it without address conflicts. If the IP addresses in the WMN are topologically aligned (close mesh routers have similar addresses), redundant copies of the IP address information in higher levels can speed up the collision detection. A router could simply chose an IP address similar to its physical neighbors and query at a higher level to detect collisions early.

Second, the mesh router at a certain key (e.g., $H("DHCP")$) provides DHCP functionality. Using the scopes defined by DLSD, the nearest of multiple local DHCP servers in the network can be determined efficiently. These local servers can manage shares of the IP address range, which are handed to them by servers on lower levels. This way, DLSD provides a simple and consistent way finding the DHCP service in the network and allows distribution of DHCP server functionality over multiple devices without costly additional protocols or consensus mechanisms.

Name Resolution: WMNs require name resolution, e.g., via DNS, to address devices and services. A DHT can provide such name resolution service. A lookup (e.g., "http://Toms-Notebook" or "Internet-Gateway") can be mapped to a DHT key at which the physical address (e.g., IP address or other routing information) of the device is stored. DLSD allows to restrict the visibility of information in the network to offer local services. Hence, clients can search for close devices or gateways to establish efficient communication links across few hops.

Mobility Support: Mobile clients in a WMN may change their point of network attachment (i.e., the mesh router they use to access the network). Supporting these clients therefore requires mobility support and name resolution. DLSD can provide such support in two ways. First, as proposed in i3 [11], a DHT node at a certain key can serve as an indirection point with a time-to-live (TTL). A client can simply send a packet toward the key $H(\text{“Toms-Notebook”}) = (a, b)$ from where it is forwarded to the actual device. While the reference point (a, b) never changes, the mobile device updates the forwarding information at (a, b) when it associates to another mesh router. Using DLSD, routers can select indirection points on successive scopes, avoiding the negative effect of triangular routing to and from the indirection point between nearby devices.

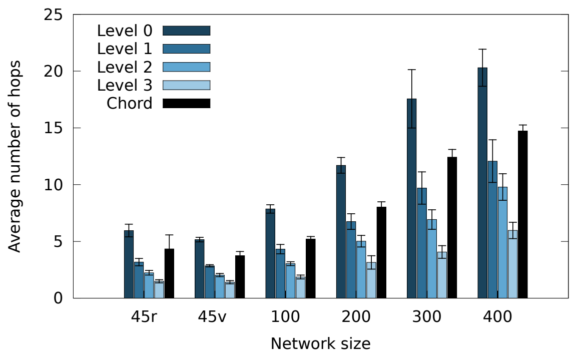
A second option for supporting mobile clients is to use the DHT as a register in which the routing information to reach a client is stored (see DNS approach above). Whenever the mobile device moves, it updates this information and becomes reachable again. By publishing this information in successive scopes, a device enables local reachability over short hop counts while ensuring global reachability in the network. Once the device associates to a router outside of the current scope, the information in the current scope becomes invalid requiring devices in the scope to re-query the information on a lower address space level. To speed up the detection of a mobility event, the router that manages the routing information may proactively notify all interested communication partners, i.e., devices that requested the routing information in a given timespan or a set of devices that subscribed to this information.

Service Composition: In accordance to the distributed design of WMNs, services may not be autonomous but may build on other services. An example for such a composed service is the composition of a media streaming service, in a format that not all clients support, and another service that transcodes the stream into another format. Clients then request the composed service consisting of the stream provider and the transcoder. The composition of services requires a redirection of the request and the transmitted payload across the single services. As these services may be located at different mesh routers or clients, redirections benefit from short hop counts between services to preserve transmission performance such as throughput. Composing services in local scopes thus enables bandwidth-intensive services that could not be supported over longer routing paths.

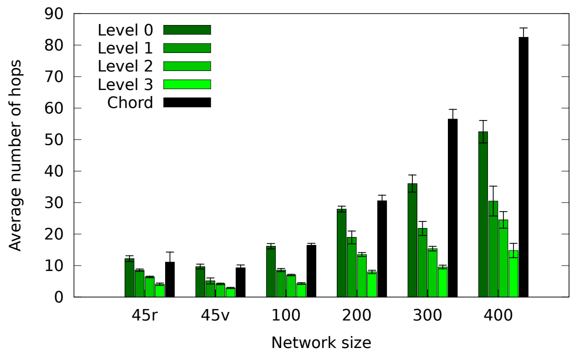
V. Evaluation

In this section, we evaluate the performance and applicability of DLSD in wireless mesh networks. We first evaluate the characteristics of DLSD in comparison to existing protocols and solutions and then evaluate the feasibility and performance of supporting localized service provision. For our evaluation, we make use of the wireless mesh network at our university, the UMIC-Mesh [12], [13]. The network consists of both a physical testbed of 45 ALIX 2c2 and ALIX 3c2 802.11b/g wireless routers (*rmesh*) and a virtual testbed of Xen-driven emulated routers (*vmesh*).

We use the *physical testbed* to validate our implementation and to measure real-life usage scenarios. In the *virtual testbed*, a controllable number of emulated mesh routers, running Linux in a configurable topology, establish a virtual mesh network without networking artifacts such as link dynamics or packet loss. Each topology defines the physical neighbor relations between routers. In our current setup, the virtual testbed supports the emulation of 400 router instances, hence we can emulate large WMNs to evaluate the scalability of



(a) Overlay hops per level compared to Chord.



(b) Underlay hops per level compared to Chord.

Fig. 3. Average number of overlay and underlay hops required to reach a trigger on each level in networks of increasing size. Routers request 121 artificially inserted keys located at 0.1 intervals in both dimensions between $(0.0, 0.0)$ and $(1.0, 1.0)$ on the respective address space level. The number of Chord overlay hops to retrieve 121 equidistantly distributed keys in the Chord identifier ring in the same networks are shown for comparison.

DLSD for large-scale deployments. In figures, physical testbed results are labeled *45r*, results from a 45 router virtual testbed are labeled *45v*. All unlabeled results are virtual testbed results.

We perform all evaluation measurements in the virtual testbed in 10 different random virtual network topologies for each network size to avoid measuring any similarity effects specific to an underlay network topology. In each topology, routers are randomly connected with a node degree between 5 and 10, in correspondence with the node degrees observed in our physical testbed [13] and in [1]. In the physical testbed, the network topology is fixed because it is constrained by the location of the routers. We thus repeat all measurements 20 times with different startup sequences of the mesh routers. We provide evaluation results for both testbeds wherever appropriate to underline the applicability of results gathered from the virtual testbed.

A. Routing Hops in Address Space Levels

The central goal of DLSD is to foster local communication in topology-oriented DHTs to reduce the negative impact of long multi-hop transmissions. To this end, DLSD establishes local DHT address spaces, in which devices may store and access keys at locations that require fewer hops to reach.

To evaluate the locality of keys at each address space level, we measure the number of hops it takes to query keys at each level. To allow for comparison of the DHT performance in DLSD, we provide results of equivalent measurements in Chord [10] (i3 Berkeley implementation) as arguably the most popular and widespread traditional DHT approach. In order to achieve an evaluation that is independent from the placement of single keys, we artificially insert keys in DLSD at 0.1 intervals between $(0.0, 0.0)$ and $(1.0, 1.0)$ at each address space level, i.e., at $(0.0, 0.0)$, $(0.0, 0.1)$, $(0.1, 0.1)$, $(0.1, 0.2)$ and so forth, resulting in 121 keys. We then measure the hops required for every device that is present at a level to query each of these 121 keys. In Chord, each router queries 121 keys that are equidistantly distributed over the Chord identifier ring.

Fig. 3(a) shows the average *overlay* hop counts for each level in networks of different sizes in DLSD and in Chord. The results allow two main observations. First, we observe a significant decrease in the required hop count with each additional level in DLSD. This reflects our design goal of gradually reducing the applied search range from a network-wide scope to local scopes. By choosing an appropriate level, clients are able to control the visibility of their data items and to control the routing overhead of accessing information. The positive effect of the hierarchic sub-address spaces is also visible in Fig. 3(b), which shows the average *underlay* hop counts for the overlay look-ups detailed in Fig. 3(a). Comparing both graphs also shows the practical differences in the overlay design of Mesh-DHT/DLSD and Chord. While Chord logarithmically distributes its virtual neighbors across the identifier space, the overlay in Mesh-DHT/DLSD reflects the topology of the network and mesh routers. Hence, even for higher overlay hop-counts at level 0 than Chord, the actual number of underlay hops is smaller. As the figure shows, the reduction in overlay hops in Mesh-DHT/DLSD maps to a significantly reduced overhead in the WMN underlay.

Note that we do not directly compare DLSD against traditional service discovery protocols like SSDP [5] or SLP [6]. This is because for these protocols, each lookup requires a multicast or broadcast to all mesh routers. Hence, the hop count is at least equal to the number of all nodes within the network. While broadcasts can be restricted to only address close services (e.g., by setting a low TTL field), the number of queried nodes is a lower bound for the number of messages. In contrast, DLSD uses a publish-subscribe approach that only requires to send or retrieve information to or from the manager of a key, resulting in a single multi-hop unicast transmission. However, DLSD requires a higher initial overhead to establish the DHT overlay structure. We evaluated this overhead in [7]. However, these conceptual differences make a precise comparison of the hop counts for both approaches (DHT and broadcast/multicast) questionable.

B. Routing Stretch

A single overlay hop in the DLSD and in Chord *overlays* typically does not map to a single hop in the network *underlay*. To provide a general statement on the underlay routing effort in DLSD as a complement to Section V-A, we evaluate the routing stretch, i.e., the average required number of underlay

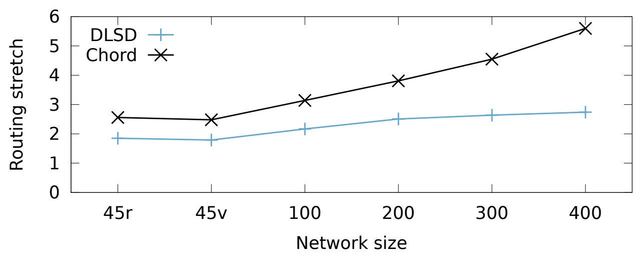


Fig. 4. Routing stretch in DLSD and Chord in networks of different size.

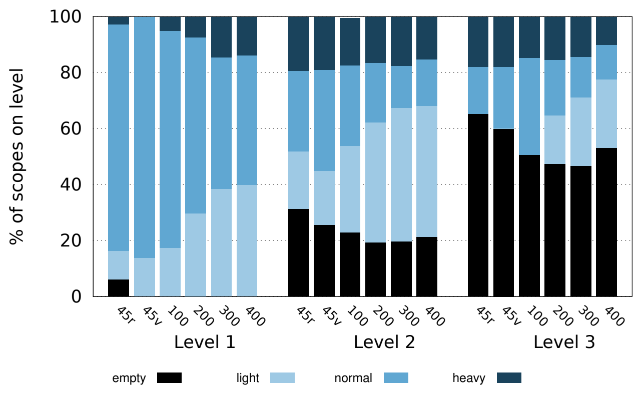


Fig. 5. Average number of mesh routers in local scopes. Results are grouped with regard to their population, ranging from empty to heavily populated.

hops per overlay hop, in Chord [10] and DLSD. We use the same network topologies and sizes as in the previous section and establish a level-0 DLSD overlay and a Chord overlay in each network. We then measure the routing stretch as the average number of underlay hops required to reach each overlay neighbor of every mesh router. Fig. 4 shows the measurement results for the different network sizes.

Due to its locality-oriented design, DLSD consistently establishes overlay neighbor relations with a better routing stretch than Chord. By reflecting the locality of mesh routers, DLSD strives to incorporate a high fraction of local physical neighbors into the overlay construction at each router. In Chord, overlay neighbors are chosen randomly, there is thus no guarantee of a specific routing stretch. With regard to these differences in the overlay design, Chord establishes a reasonable stretch between 2.5 and 6, whereas the stretch in DLSD is below 3 for all network sizes. As the measurements for large network topologies, i.e., 200 to 400 routers, show, DLSD maintains a bounded routing stretch whereas the routing stretch in Chord grows with the network size due to longer routing paths between overlay neighbors.

C. Population of Scopes

The usefulness of local address space levels depends on the number of mesh routers in the respective address space. With respect to the non-uniform distribution of routers in the physical network and the resulting non-uniform distribution of routers in the DHT address space, highly local address spaces may be empty, i.e., their boundaries do not enclose a router. Other address spaces, in densely populated regions of the network, may enclose a high number of routers. To show the need for a dynamic mechanism to create local scopes, as presented in Section III-C, we evaluate the number of routers that are present in the address spaces with regard to the network size.

Each new level segments the current address space in four new sub-address-spaces, consisting of fewer routers and spanning fewer hops than the lower level. For example, in a network of 400 uniformly distributed mesh routers, we expect 1/4 (= 100) of the routers in each level-1 address space, 1/16 (= 25) of the routers in each level-2 address space and 1/64 (= 6.25) of the routers in each level-3 address space. However, the non-uniform distributions of routers in the *rmesh* and the *vmesh* inevitably leads to dense and sparse regions. Fig. 5 shows the fraction of address spaces on a given level that contain a given number of routers with regard to the network size. We sort address spaces in four groups. *Heavy* address spaces contain more than twice the expected number of nodes in a uniform node distribution. *Light* address spaces contain less than 1/2 of the expected routers. *Normal* address spaces are between these extremes. The graph also shows the *empty* address spaces that contain no routers on this level. These empty spaces would, in real-life networks, neither be created nor be used because no devices request or publish information within their scopes.

Fig. 5 confirms the intuitive assumption that the fraction of sparsely populated address spaces (*empty* and *light*) increases with the level depth. While the population of level-1 address spaces is distributed between sparse and dense, the majority of level-3 address spaces is sparsely populated or empty. Such light address spaces are inefficient because only few nodes publish or lookup information in them, reducing the probability of a successful lookup.

Note that in this evaluation, we established address spaces based on a globally set level value, regardless of their population or usefulness. This leads to a massive amount of underpopulated and to a small number of crowded areas. The results show the requirement for an adaptive mechanism, as described in Section III-C, to create and delete local address spaces on demand to avoid inefficient regions. Our adaptive approach prevents the creation of empty, light, and heavy address spaces by design. For example, 50 % of the level-2 address spaces in Fig. 5 would not be created while almost 20 % of the level-3 address spaces would be segmented in four level-4 address spaces.

D. Key Re-Distribution under Network Expansion

In analogy to the transfer of keys, induced by node joins in traditional DHT approaches, the topology-orientation of DLSD requires stored keys to be transmitted to neighboring routers in case the network topology changes. This is because the responsibility of routers for keys correlates with the current network topology (cf. Fig. 2(a)).

To evaluate the overhead of adapting the distribution of keys on multiple address space levels, we measure the number of overlay hops that keys move because of an expansion of the network. As in Section V-A, we artificially insert DHT keys at intervals of 0.1 on all levels in each network size. Fig. 6 shows

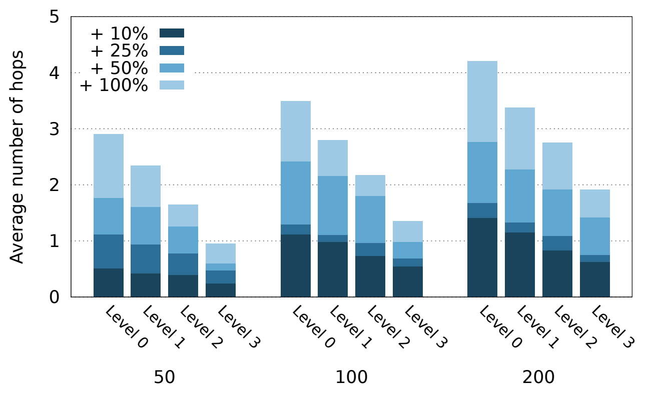


Fig. 6. Average number of hops that keys, distributed at 0.1 intervals at the respective address space levels, move under network expansion. Bars show absolute values, i.e., a +100 % expansion of a network of 50 mesh routers induces an average key movement of 2.99 hops on level-0 address spaces (+50 %: 1.86 hops, +25 %: 1.12 hops, +10 %: 0.51 hops).

the absolute average number of hops that a key moves when a network of 50, 100 and 200 routers is expanded by 10 %, 25 %, 50 % and 100 %, respectively. To expand a network, we add routers with random neighbor relations to the network. We pre-establish local address space levels 0 to 3 to observe the movement of keys with regard to the level hierarchy. As the overall overlay address space expands when mesh routers join, all local address space levels expand accordingly with respect to the current level-0 address space boundaries. To measure the movement of keys, we measure the overlay hop distance between the router that originally stored the respective key and the router that stores the key at the end of the network expansion.

Our evaluation again shows a reflection of locality in the respective address spaces. Keys in local address spaces, i.e., on level 2 and 3, move over a significantly lower number of hops compared to keys in more global address spaces, i.e., on level 0 and 1. However, even when doubling the number of mesh routers in the network, keys on lower levels 0 and 1 move only along a very confined area of 3 to 5 overlay hops. In addition, we observe that on all levels, the movement of keys is limited and not proportional to the the number of added mesh routers, i.e., an expansion of 100 % does not induce twice the movement of a 50 % expansion, highlighting the scalability of DLSD. This is due to the fact that DLSD maintains the inherent locality of address space levels during network expansion. Keys thus do not move through the whole of the network but stay close to their original placement. The placement in local address spaces on higher levels and the movement over less than two hops on these levels thereby maintains a reachability of keys over short hop distances. Furthermore, while the expansion of larger networks induces more movement, this is due to the larger number of routers being added to the network and their subsequent assumption of responsibility for keys. Key movements, especially over short hop counts in higher levels, thus only induce little overhead and allow a look-up of and access to localized keys even in case of network dynamics.

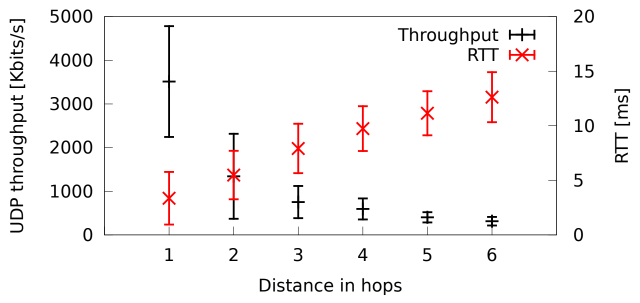


Fig. 7. Average UDP throughput and round trip time between all node pairs grouped by hop distance in the 45 router UMIC-Mesh network [12], [13].

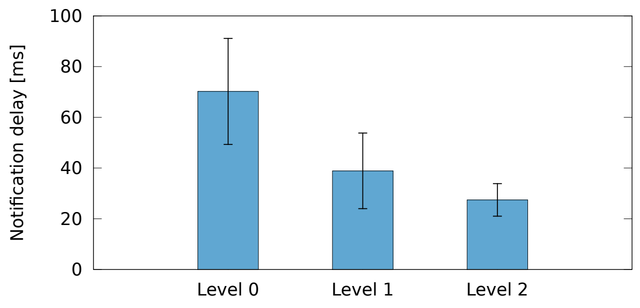


Fig. 8. Delay of notification messages on different levels.

E. Application Support in DLSD

In this section, we evaluate the applicability of DLSD to support services in real-world WMNs such as the UMIC-Mesh. Specifically, we evaluate the impact of using the local scopes provided in DLSD with regard to the two main transmission characteristics, i.e., throughput and latency. Fig. 7 shows the average throughput and round trip time (RTT) between all node pairs in our physical testbed for comparison. We discuss the results with regard to supporting host mobility and service composition as proposed in Section IV.

Host Mobility Support: Satisfactory host mobility support in WMNs depends on the delay between a mobility event and the notification of the communicating hosts. Slow mobility updates prolong the time in which both peers cannot communicate because the packets are routed to outdated locations in the network. In Section IV, we discussed two options of supporting host mobility in DLSD. Common to both approaches is the dependence on updating the routing information, either at the indirection point or at the router that stores the routing information. In this section, we thus evaluate the delay of notification messages with regard to the level on which they are distributed. To acquire a comprehensive result that is independent from specific key placement, we transmit update messages from keys distributed at 0.1 intervals (cf. Section V-A) to all routers that are present on a level. Note that, due to the application scenario, we measure one-way transmissions and not the RTT of notification messages.

Fig. 8 shows the average notification delays obtained in our real-world testbed for levels 0 to 2. While the notification delay never exceeds 100 ms, the impact of the locality of higher levels is significant with level-2 notifications requiring only 40 % of the notification delay on level 0. Host mobility support in DLSD therefore scales with the distance between

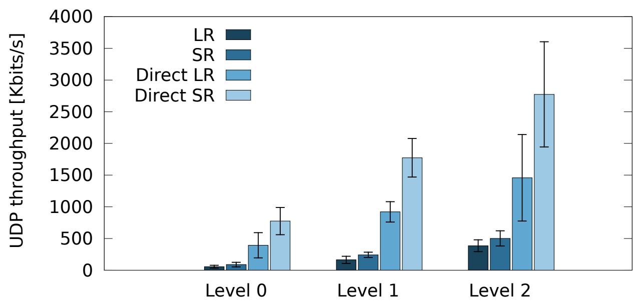


Fig. 9. UDP throughput of composed services on different levels over a short range (SR) and a long range (LR) in the DHT address space. Direct service access (Direct LR & SR) shown for comparison.

the communicating devices. The results support the intuitive requirement that notifications between nearby devices, e.g., devices in level 2, occur quickly and thereby do not disrupt the observed transmission performance (cf. Fig. 7). While network-wide notifications, i.e., on level 0, require more time, they still support a timely update of routing information.

In comparison with Fig. 7, Fig. 8 furthermore shows the impact of DLSD on the message latency in the network. The longer routing paths, due to indirections in the overlay routing as evidenced by the routing stretch shown in Fig. 4, negatively influences routing performance in terms of latency. However, the increased latency only occurs in case of notifications and does not harm regular communication.

Service Composition: The composition of services, as proposed in i3 [11], enables complex services that are provided successively by a sequence of multiple routers or clients. Providing composed services using traditional DHT approaches or service discovery protocols is challenging because each device in the sequence may be very distant in the network, resulting in overall high packet loss and low throughput. In contrast, composing a service in local scopes in DLSD results in small hop counts between each device in the sequence, thereby preserving throughput and latency. When accessing a composed service, the service at each hop handles the payload data and forwards it to the next key in the sequence. The last device then forwards the payload to the requesting device.

In this section, we evaluate the performance of establishing composed services in terms of throughput in our physical testbed. To this end, we artificially establish two composed services of a sequence of four devices each on levels 0 to 2. The keys of the "long range" service (LR) are located at $(0.1, 0.1)$, $(0.9, 0.1)$, $(0.9, 0.9)$ and $(0.1, 0.9)$ and thus cover large distances in the DHT address space. In contrast, the "short range" service (SR) are located at $(0.1, 0.1)$, $(0.3, 0.1)$, $(0.3, 0.3)$ and $(0.1, 0.3)$. In both cases, the router at $(0.1, 0.1)$ serves as the requesting device.

Fig. 9 shows the results of 20 UDP *iperf* throughput measurements of 30 sec with regard to each level. We provide the result for accessing a service directly without any composition (Direct LR & SR) on each level for comparison. In this measurement, we average the achieved throughput from the requesting router towards all of the aforementioned identifiers. Fig. 9 highlights the benefit of providing services in local scopes as throughput on lower levels decreases significantly. Furthermore, service composition on lower levels results in throughput rates that do not allow complex or bandwidth-dependent services due to the increase in the number of underlay hops. While this harms the performance of composed services on higher levels as well, throughput rates of 500 KBit/s allow for music and other audio transmissions, for example. Local address spaces in DLSD thus enable the high-bandwidth provision of single services and composition of complex, distributed services in WMNs.

VI. Related Work

DLSD provides service discovery in wireless mesh networks based on a topological DHT. It therefore relates to approaches in service discovery and DHT approaches that could provide similar services.

A. Service Discovery Protocols

UPnP [5] (based on SSDP) and SLP [6] are examples for service discovery protocols in local area networks. To discover and advertise services, devices broadcast or multicast messages in the network. In a WMN, both network-wide multicast groups and broadcasts significantly harm the network performance. First, frequent broadcast messages severely degrade communication performance in the WMN due to interference and collisions. Second, when using multicasts, to reliably discover services, every mesh router in the WMN has to be part of the multicast group. Because of this, a multicast message still requires sending a message to every router in the WMN. The lack of a consistent mapping of information to a specific router and the resulting need for broadcasts or multicasts make traditional service discovery protocols not suited for multi-hop wireless networks. We refer to [14] for a survey and comparison of service discovery protocols and a discussion on service discovery in wireless multi-hop networks.

Efforts to adapt service discovery protocols to ad-hoc networks [15]–[17] still rely on flooding or multicast techniques. While node mobility in ad-hoc networks may justify these techniques, WMNs show a static backbone topology and thus benefit from a consistent mapping and unicast addressing of content and services. mSLP [18] extends SLP by establishing a mesh topology between directory agents, but does not target multi-hop wireless mesh networks and thus does not alleviate the aforementioned problems of SLP.

OLSR-mDNS [19] realizes service discovery in WMNs by integrating multicast DNS messages in OLSR. Mesh routers store overheard services advertisements for a given timespan while clients trigger scoped flooding of requests via OLSR multi point relays. As OLSR-mDNS lacks a consistent mapping of service identifiers to routers, service outside of the flooding scope can not be found. Furthermore, limiting this scope requires client or routers to have a measure of the appropriate hop count, an assumption that is not met in typical WMNs and especially in larger networks.

B. DHT Approaches

Chord [10] and CAN [20] are examples of Internet-scale DHT approaches. They assume a stable, low-latency communication medium over which DHT nodes are equally well reachable. Random node identifier assignments prevent a representation of node locality and result in random responsibility for keys in the address space. The performance of such DHT approaches suffers in wireless multi-hop scenarios such as WMNs, as evaluated in [21], [22].

Virtual Ring Routing (VRR) [23] establishes a DHT in wireless ad-hoc networks, in which nodes assign themselves random, location-independent identifiers in an identifier ring structure. VRR provides only a global DHT and does not support discovery of services or data items with regard to locality. Random node identifiers furthermore lead to long routing paths and additional overhead in the discovery process.

CrossROAD [24] constructs a DHT in ad-hoc networks using the topology information contained in ad-hoc routing protocol messages. While reducing the construction overhead, node identifiers are still derived pseudo-randomly from hashing the IP address of a node. DHT routing on node identifiers will thus take unpredictable, location-unaware paths.

The Internet Indirection Infrastructure (i3) [11] proposes a rendezvous structure on top of a DHT that supports service stacking and indirection-based mobility support, similar to our approach. i3 does not represent the locality of nodes or keys and thus does not allow a localized service provision or discovery. Our approach shares its design goals with i3 and adapts their realization to the requirements of WMNs.

VII. Conclusion

The performance of wireless mesh networks benefits from short routing paths when clients access services or data in the network. In this paper, we proposed DLSD, a DHT-based look-up structure that enables consistent localized look-ups over short paths based on a hierarchy of scaled DHT address spaces. In contrast to service discovery protocols, DLSD avoids flooding or multicasts in the network and enables a consistent, locally computable mapping of information to the managing mesh router. DLSD differs from traditional DHT approaches as it establishes hierarchical topological sub-address spaces that only enclose a fraction of the routers in the WMN. Look-ups iteratively or in parallel search these sub-address spaces and find the most local available copy of the requested information. In our evaluation we showed that searching in local sub-address spaces significantly reduces the required number of overlay and underlay hops compared to non-hierarchic DHT approaches such as Mesh-DHT and Chord. DLSD thus enables look-ups to only traverse short routing paths to simultaneously speed up the look-up, access the closest source of information or service and to reduce the negative impact of forwarding look-ups on the network. We further detailed the use of DLSD to establish well-known services such as DHCP-like host configuration and DNS-like name resolution and evaluated the performance of DLSD in providing direct and composed services in local and global scopes. We envision the specific implementation of composed services and the implementation and evaluation of reliability and redundancy mechanisms in DLSD as future work.

References

[1] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *Proceedings of the 11th annual international conference on Mobile computing and networking*, ser. MobiCom '05, 2005.

[2] J. Li, C. Blake, D. S. De Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, ser. MobiCom 2001.

[3] R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks," in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '04, 2004.

[4] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, ser. MobiCom '99, 1999.

[5] "Upnp device architecture, white paper." [Online]. Available: http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf

[6] E. Guttman, C. Perkins, J. Veizades, and M. Day, "Service Location Protocol, Version 2," RFC 2608 (Proposed Standard), Internet Engineering Task Force, June 1999, updated by RFC 3224. [Online]. Available: http://www.ietf.org/rfc/rfc2608.txt

[7] H. Wirtz, T. Heer, R. Hummen, and K. Wehrle, "Mesh-DHT: a Locality-Based distributed Look-Up structure for Wireless Mesh Networks," in *IEEE ICC 2012 - Ad-hoc and Sensor Networking Symposium (ICC'12 AHSN)*, Ottawa, Ontario, Canada, Jun. 2012.

[8] F. Aurenhammer, "Voronoi diagrams, a survey of a fundamental geometric data structure," *ACM Comput. Surv.*, 1991.

[9] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational geometry: algorithms and applications.* Springer-Verlag New York, Inc., 1997.

[10] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of the ACM SIGCOMM '01*, San Diego, 2001.

[11] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet indirection infrastructure," *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, pp. 73–86, Aug. 2002.

[12] A. Zimmermann, D. Schaffrath, M. Wenig, A. Hannemann, M. Güneş, and S. A. Makram, "Performance evaluation of a hybrid testbed for wireless mesh networks," in *Proceedings of MASS'07*, 2007.

[13] "Umic-mesh network." [Online]. Available: http://www.umic-mesh.net

[14] E. Meshkova, J. Riihijärvi, M. Petrova, and P. Mähönen, "A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks," *Comput. Netw.*, vol. 52, no. 11, pp. 2097–2128, Aug. 2008.

[15] M. A. El Saoud, T. Kunz, and S. Mahmoud, "Slpmanet: service location protocol for manet," in *Proceedings of IWCMC '06*, 2006.

[16] S. Helal, N. Desai, V. Verma, and C. Lee, "Konark - a service discovery and delivery protocol for ad-hoc networks," in *WCNC 2003*, 2003.

[17] U. Kozat and L. Tassiulas, "Network layer support for service discovery in mobile ad hoc networks," in *INFOCOM 2003.*, vol. 3, 2003.

[18] W. Zhao and H. Schulzrinne, "mslp - mesh-enhanced service location protocol," in *In Proc. of ICCCN 2001.*

[19] M. Krebs, K.-H. Krempels, and M. Kucay, "Service discovery in wireless mesh networks," in *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, 2008.

[20] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proceedings of the ACM SIGCOMM '01 Conference*, San Diego, 2001.

[21] S. Burresi, C. Canali, M. E. Renda, and P. Santi, "Meshchord: A location-aware, cross-layer specialization of chord for wireless mesh networks," in *PerCom*, 2008.

[22] C. Cramer and T. Fuhrmann, "Performance evaluation of chord in mobile ad hoc networks," in *Proceedings of MobiShare '06*, 2006, pp. 48–53.

[23] M. Caesar, M. Castro, E. B. Nightingale, G. O'Shea, and A. Rowstron, "Virtual ring routing: Network routing inspired by dhts," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, 2006.

[24] F. Delmastro, "From pastry to crossroad: Cross-layer ring overlay for ad hoc networks," in *PerCom 2005 Workshops*, march 2005.