Address Resolution in Mobile Ad Hoc Networks using Adaptive Routing

Thomas Finke, Juergen Schroeder
Department of Electronics and Information Technology
Heilbronn University, Germany
{thomas.finke,juergen.schroeder}@hs-heilbronn.de

Sebastian Schellenberg, Markus Hager, Jochen Seitz
Communication Networks Group
Ilmenau University of Technology, Germany
{sebastian.schellenberg,markus.hager,jochen.seitz}@tu-ilmenau.de

***Abstract*—In disaster scenarios, communication systems usually consist of heterogeneous nodes and damaged infrastructure. Communication is important for rescue teams and victims as well but a serious problem because normal network systems like wired or mobile radio Internet could be unreliable or simply not available. To deal with these problems, much effort has been spent to mobile ad hoc networks (MANETs) and their specialties. Those networks are usually unsteady and highly mobile. Therefore, classical network services like resolution of names to their regarding local IP addresses is a big challenge. In this paper, we present our new approach for consistent and efficient name resolution using adaptive routing techniques. With this approach, routing and name resolution can be combined to decrease the latency of the lookup and to reduce the caused traffic.**

***Index Terms*—MANET; name resolution; adaptive routing.**

I. Introduction

A lot of applications in networks, nowadays, use names to address communication partners because it is easier for human users to remember names instead of numbers. For example, if someone wants to access a website, the web browser gets a Uniform Resource Locator (URL) and has to resolve this name to an IP-address. The mapping between such names (e.g., hostnames) and numbers (network addresses) in the Internet is done via the well known Domain Name System (DNS) [1] with centralized DNS servers.

A big problem in mobile ad hoc networks (MANETs) is the possibly high mobility and also the unreliability of nodes. For example, nodes can fail or simply go out of range. DNS is designed for infrastructure networks with centralized servers and is therefore not suitable for highly dynamic MANETs. If the centralized DNS server would fail, the whole MANET would be unable to resolve names. Therefore, mechanisms to prevent single points of failure must be developed.

MANETs have limited bandwidth and energy resources. Therefore, the communication protocols should only sparingly use a node's resources. To achieve this, we piggyback name resolution messages in routing packets to avoid additional traffic.

As the network topology can change in a wide range in highly dynamic MANETs, because of changing node speed or power resources for example, it is not very efficient to use only one routing protocol for all network constellations. Hence, we use an adaptive routing framework in our system to switch between different routing protocols during runtime and to achieve best performance in multiple network constellations by selecting the optimal routing protocol to a given network scenario. In our system, we use one reactive and one proactive routing protocol, however it is possible to integrate more than two routing protocols in our adaptive routing framework.

Our name resolution in reactive routing scenarios is inspired by Engelstad et al. [2] (cf. II-B). In proactive routing scenarios, we use our new proactive name resolution approach as discussed later in Section III-B, which allows hostname to address resolution and route finding without any latency for the sending application if the system has already learned the topology.

A challenging problem in MANETs are the rapidly changing network addresses of nodes in scenarios with high mobility, due to nodes that continuously enter, leave, or change the network. The founders of the Internet did not consider that there could be so many Internet using devices with high mobility, e.g., smartphones or laptops. That is why the Internet Protocol (IP) address is a locator and an identifier at the same time. If a node changes its location, it gets a new IP address, even if the identification remains the same.

The idea to solve this problem is to introduce a logical addressing scheme on top of IP to split the locator and identifier functionality. This splitting is well known in literature (e.g., [3]). However, mapping an identifier to a locater requires a working name resolution mechanism optimized for MANETs.

This paper shows our decentralized, fully distributed approach for name resolution using adaptive routing techniques. In Section II, we discuss related work regarding adaptive routing and existing name resolution approaches followed by our motivation for a new approach. Section III shows details about our name resolution system and about the used message types. We also show how our approach is transparent to the application and how names can be resolved over external networks.

II. Related Work

A. Adaptive Routing

In MANETs, the choice of the right routing protocol is important. We can divide the routing mechanism into two basic groups, the proactive and the reactive protocols. Proactive

 ISBN: 978-1-61208-231-8

routing protocols calculate the routes before they are needed. Therefore, the nodes exchange their routing information continuously. This is performed periodically or with every change in the topology. The proactive approach provides lower delays if a node wants to establish a connection. The drawback is the higher traffic caused by this strategy. One example is the well known Optimized Link State Routing (OLSR) protocol [4]. In systems with high mobility, this approach is not advisable.

In reactive or on-demand routing protocols, nodes only ask for routes when they need them. Therefore, the nodes do not have full knowledge about the whole network. This approach decreases the cost of synchronization but increases the delay for establishing a connection. An example is the Ad-hoc On-demand Distance Vector routing protocol (AODV) [5]. This type of routing algorithm is used in scenarios with high mobility and rapidly changing topologies.

In disaster scenarios it is difficult to assess which routing protocol is the best. Due to changing topologies and scenarios, the preferred routing protocol could change over time. If the positions and the connections of the nodes are stable, a proactive approach is best. If the nodes begin to move frequently, the reactive routing is better [2]. To provide the best protocol with respect to the current situation, adaptive routing allows to switch between different protocols. In our system, we use adaptive routing techniques to optimize routing and thus the efficiency of name resolution messages, which are piggybacked on routing packets.

Related work in that field has been done by Nanda et al. [6]. In this work, the nodes can switch between different routing protocols whereas these protocols do not have to be modified. The drawback of this work is that all nodes in the MANET have to use the same routing protocol at the same time and that the routing table entries have to be copied each time the protocol is changed. Another work was done by Hoebeke et al. [7], where each node can choose the best routing protocol for its requirements. Also multiple routing protocols are possible at the same time and a node does not have to support all used protocols. However, the drawback is that the used routing protocols have to be modified before they can be used in this framework. A further alternative is the hybrid Zone Routing Protocol [8] (ZRP), which provides a reactive and a proactive routing zone for each node. The nodes are able to modify the radius of their zones and, therefore, the ZRP also changes between different routing protocols to a certain degree.

B. Name Resolution Mechanism

In wired communication systems or scenarios with a robust infrastructure, name resolution is usually done by the well known Domain Name System [1]. One task in our system is to map the host names of the nodes to their current local network address or addresses if they are multi homed. It is obvious to use DNS for that, too [9]. But, in MANETs, the use of a centralized entity is difficult, because single points of failure could crash the whole system. There are three main approaches to adapt the DNS approach to MANETs. The first is to use centralized, but modified DNS [10]; second option is using multicast based approaches [11]; the third way is using routing techniques [2].

The Zeroconf Working Group has proposed a multicast based protocol for name resolution and service discovery in networks without conventional DNS servers [11]. Multicast DNS (mDNS) uses a multicast group with a well known multicast address for name resolution. Every node that wants to know the network address, which corresponds to a given name sends a request to that multicast address and the corresponding node answers the request with its network address. The drawbacks of mDNS are that an extra protocol for name resolution is needed and that a lot of additional traffic is produced because of flooding name requests to multicast addresses.

One idea of getting away from DNS is the routing-based approach proposed by Engelstad et al. [2], where the idea is to see the name resolution as a similar problem of finding a route. Instead of finding the local network address for a hostname and then in a second step finding a route, the approach asks for both at the same time (cf. Figures 1 and 2).

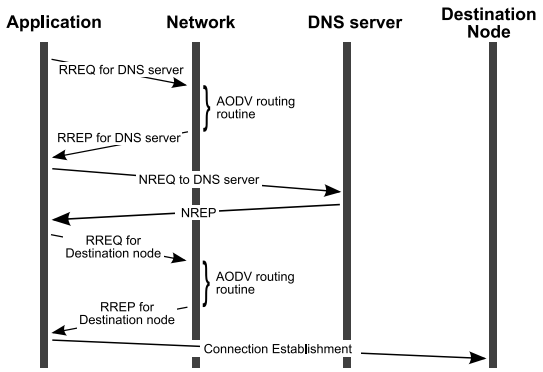

Fig. 1. Message sequence chart of connection establishment with reactive routing protocol

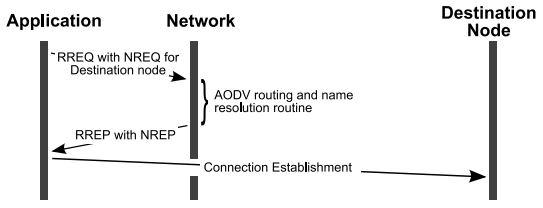

Fig. 2. Message sequence chart of connection establishment with reactive routing protocol and name resolution extension

Engelstad uses the reactive AODV routing protocol and sends the name resolution requests (NREQ) and replies (NREP) piggybacked with the route request (RREQ) and reply (RREP) messages [12]. Because of the combined name resolution and route finding approach, the packet overhead and the time between the need of an application to send data and the time where the first data packets can be transmitted is reduced. Nevertheless, this approach is limited to reactive routing approaches and can therefore not be used efficiently in

 ISBN: 978-1-61208-231-8

adaptive routing frameworks without a counterpart for proactive routing. In our system, we adapt Engelstadt's approach with some changes and introduce a proactive method. Details are shown in the next section.

III. Our Name Resolution over Adaptive Routing Approach

A. Adaptive Routing

As mentioned above, a specific routing protocol performs only well in one special network scenario. To cope with highly dynamic MANETs, our adaptive routing system switches between the two routing protocols AODV and OLSR. Therefore, a monitoring agent is used, which gathers information about the network state. This information is used by a decision maker, which selects the current routing protocol by operating selector switches, which control the routing packet flow. For coming to a decision, the module has access to the routing mode information telling what protocol should be used. Every node distributes its routing protocol decision inside an additional packet header throughout the network. Each implemented routing protocol uses its own independent routing table to store routes. The data packet forwarding module accesses the information inside the routing tables through a routing table wrapper, which looks inside all routing tables. The overall memory consumption of the routing tables is held low as a result of the dynamic memory allocation of the routing protocols. If a route's lifetime expires, the route is deleted and the memory is freed.

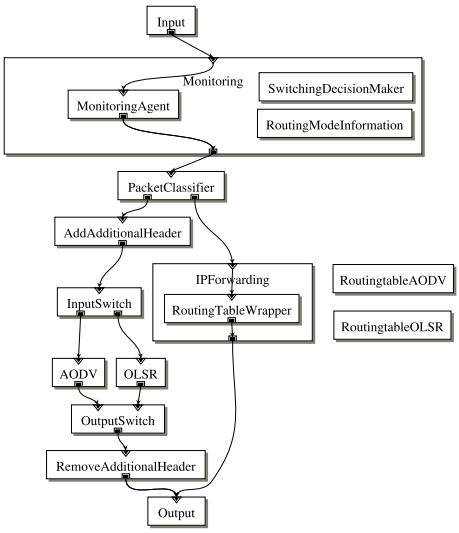

Fig. 3. Block Diagram of the Adaptive Routing Framework

We can switch to the reactive routing protocol when nodes are highly mobile and there are long communication sessions or we can switch to the proactive one when there are only some sporadic connections. Of course, we are not limited to implement only two routing protocols, our system is also able to support many different strategies as well. In this way, our routing performs well in a couple of network scenarios and outperforms prior routing approaches, which can only perform well in one special scenario.

B. Name Resolution via Proactive Routing

To avoid centralized DNS servers for resolving hostnames to network addresses, we use a decentralized routing-based approach, where each node is part of the name resolution system. For increasing the network performance during proactive operation, in terms of reducing the routing overhead and the latency for name resolution and route finding, we combine the name resolution mechanism with the routing protocols. Each node stores additional information about the mapping between hostnames and network addresses.

This subsection describes our proactive approach to resolve names to addresses. The names are usually human readable names to make it easier for users to identify nodes. If an application wants to resolve a name, it has to know this name. If the name of the desired destination is unknown, the node cannot trigger a name resolution process. Rather, it could bypass name resolution and directly use a node's address or it could try to find a node by service discovery, but this is not part of our work.

If an application, like a web browser, wants to contact another node in our proactive routing mode, the route to the destination is available immediately without any latency for route finding if the network address of the destination node is known. However, the problem is that the route look-up cannot be done until the hostname of the destination has been translated into a network address. To avoid the latency for name resolution we use a proactive name resolution approach (cf. Figure 4).

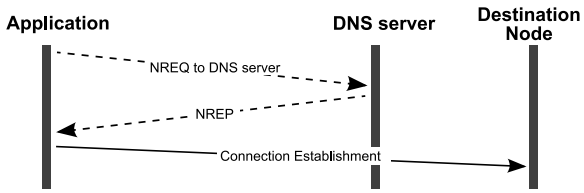

Fig. 4. Message sequence chart of connection establishment using a proactive routing protocol with and without (including the dashed lines) our extension

In proactive routing protocols, messages are periodically exchanged between the nodes. Therefore, the messages have to be small in terms of provoked traffic. For naming over proactive routing, the hostnames and network addresses have to be exchanged between the nodes, which are part of the network. A Fully-Qualified Domain Name (FQDN) [13] should have a maximum length of 255 characters containing a-z (case-insensitive), 0-9 and -. The FQDN consists of different labels with a length of 1 to 63 characters, each separated by a dot. A transmission with FQDNs directly stored inside the routing packets would let the packets grow very large and would lead to an exhaustive usage of bandwidth.

To avoid this, we calculate an MD5 hash key, which has a length of 128 bits for each hostname. This hash key is used inside the routing packets instead of the original hostname. If a

 ISBN: 978-1-61208-231-8

node searches for a specific hostname, it hashes this hostname and looks it up in the Hostname-Address-Mapping (HAM) (cf. Section III-C) table. In this way, each transmitted 'hostname' has a length of 128 bits, which is the same as an IPv6 address, instead of a maximum of $255 * 8\,\text{bits}\ (1\,\text{character}) = 2040\,\text{bits}$ (each character should be represented by 8 bits [13]). If a node wants to advertise its hostname in the proactive network, it decides whether to use the real hostname or the hash key, depending on which value is shorter or otherwise preferred by the node. In this way, each transmitted hostname has a maximum length of 128 bits, except after a COLERR message (cf. Section III-D2).

To detect hash collisions between different hostnames, each node has to check its database for identical hash keys. If a node detects a hash collision, it has to send a special Collision Error message (COLERR) inside the next OLSR routing packet throughout the network with a Time-To-Live (TTL) set to the maximum value of 255 to reach every node. Each node that receives a COLERR has to delete the corresponding hash key to address mapping inside its database and is only allowed to forward the COLERR message once to restrict flooding and to prevent exhaustive bandwidth usage. When the nodes whose hostnames corresponds to the collided hash keys will receive the COLERR, they have to send out a Name Advertisement (NADV) message containing the hostname to network address allocation. To avoid further hash key collisions these messages do not use hash keys to represent the hostname, rather they contain the hostnames in a not encrypted readable form.

As pointed out above, the usage of readable hostnames inside proactive routing messages should be avoided. For saving bandwidth we encode the single characters of the hostnames in a way that frequently used characters are represented by a shorter bit string and rarely used characters are representated by a longer bit string (c.f. 8-bit UCS Transformation Format (UTF-8) [14], which uses a similar idea). It should be noted, that processing the hostname to save bandwidth results in a higher load of a node's processor and therefore in energy usage.

If there are hostnames available that are mapped to multiple nodes, e.g., for load-balancing or multihoming, the other nodes could detect this behavior as a hash key collision. To avoid this, a hostname with multiple corresponding network addresses has to be marked in Name Advertisement (NADV) messages to notify other nodes about this circumstance. Therefore, if a node is a member of such a composition with one hostname and multiple network addresses, it has to set a special flag in its NADV messages.

If a node wants to connect to another node outside the local MANET, the node first tries to find a mapping in its HAM table. If there is no matching entry, the node forwards the request inside a conventional DNS request to one or more of the gateway nodes, which are connected to other networks.

To minimize additional packet overhead for name resolution we use the fact that mappings between hostnames and network addresses change rarely compared to the changes in routes between the nodes. Based on this behavior, the nodes exchange routing messages more often than NADV messages.

C. Hostname-Address-Mapping table

For storage of hostname to address mappings we use our new introduced Hostname-Address-Mapping (HAM) table. Each node has one HAM table that is independent from the routing protocol. Each entry in the table represents the mapping between one hostname and one network address (cf. Figure 5). If one hostname corresponds to multiple network addresses (e.g., load-balancing) or if one network address corresponds to multiple hostnames (e.g., virtual hosts), the HAM table contains multiple entries to store all mappings. If a node recognizes multiple destination network addresses for a looked-up hostname, it selects the network address with the 'best' route (e.g., in terms of hop count, bandwidth, available power) to the destination. Figure 6 shows an example for a HAM table entry with a hashed hostname mapped to an IPv4 address.

The HAM table in our proactive routing mode also provides the possibility for a simple reverse lookup from network addresses to hostnames.

Field	**Value**
Hostname Type	Type of the hostname: 0=hash key, 1=hostname
Hostname Length	Length in octets (bytes)
Hostname Value	The hostnames value either in readable form or as hash key
Address Type	Type of the network address: 0=IPv4, 1=IPv6
Address Length	Length of the network address in bytes
Address Value	The node address as IPv4 or IPv6
Flag Multihomed	This flag signalizes if this hostname corresponds to multiple network addresses
Flag Gateway	This flag signalizes if this network address belongs to a gateway node

Fig. 5. Structure of one entry in the HAM table

Field	**Value**
Hostname Type	0
Hostname Length	16
Hostname Value	e3198adf5c74a66165a458045960d51e
Address Type	0
Address Length	4
Address Value	10.1.1.3
Flag Multihomed	0
Flag Gateway	0

Fig. 6. Example of an entry in the HAM table

 ISBN: 978-1-61208-231-8

D. The Packet Types for Proactive Name Resolution

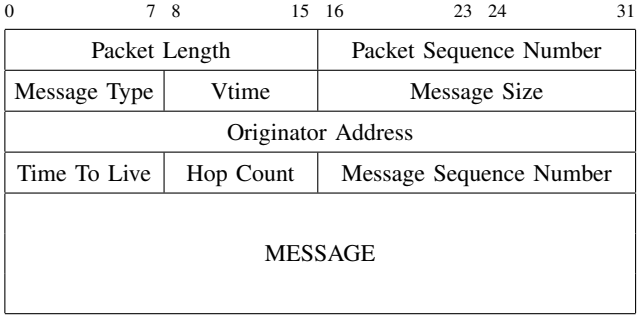

Fig. 7. Structure of OLSR messages [4]

In the reactive operation mode, our adaptive routing will use AODV as routing protocol and a name resolution mechanism similar to the approach of Engelstad et al. [2]. We do not only take the NREQ and NREP packets piggybacked with the routing messages but really integrate them inside the algorithm. So, there are two new AODV messages with included name requests and responses.

In the proactive mode, our approach will use OLSR for routing and two additional OLSR messages for name resolution. Of course, our adaptive routing framework supports also other routing protocols, but for simulation and demonstration we use AODV and OLSR.

The OLSR protocol transmits routing packages containing different messages efficiently between the nodes via Multipoint Relays (MPRs) [4]. The common structure of such routing messages (cf. Figure 7) contains a standardized header and the message body. To advertise hostname information, our proactive name resolution approach uses two additional message types, which are identified by 'Message Type' 128 for Name Advertisement (NADV) messages and 129 for Collision Error (COLERR) messages. The payload of these two introduced messages is transmitted inside the message body of the OLSR packets. The standardized header part of these messages is untouched to keep compatibility to nodes that do not support our new name resolution. The body structure of the new message types is shown in Figure 8.

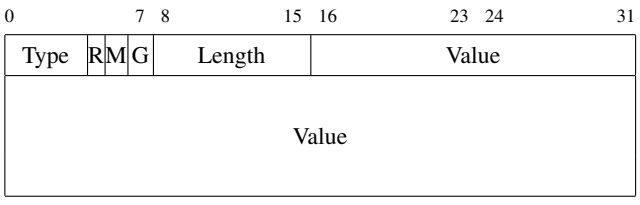

Fig. 8. Body structure of NADV messages

1) Name Advertisement Message: For hostname advertising the nodes use a Type-Length-Value (TLV) message structure (cf. Figure 8). The type field denotes the type of the hostname (0 = hash key, 1 = hostname). The 'R' bit is reserved for an extension in the future. The 'M' flag is set to a value of '1' to signalize that the following hash key or hostname is part of a multihomed system with one hostname and multiple corresponding network addresses (e.g., a load-balanced system). The 'G' flag is set to signalize a gateway node with at least one additional network interface to other networks. If this flag is set, the node can be used from other nodes as gateway to connect to networks outside the local MANET. The length field shows how many octets (bytes) are used for the following hostname. The length field has always a value of 16 if the type field is set to zero and hence signalizes an MD5 hashed hostname.

The network address of the originator of the NADV message is not stored in the message body, because it is already available in the 'Originator Address' field of an OLSR message header (cf. Figure 7). If a node uses multiple network interfaces and therefore multiple network addresses, the other nodes receive information about such addresses via 'Multiple Interface Declaration' (MID) messages of OLSR [4].

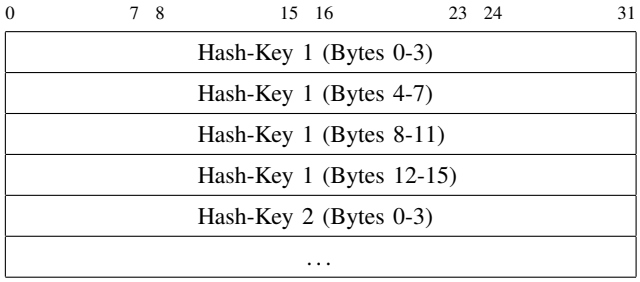

Fig. 9. Body structure of COLERR messages

2) Collision Error Message: If a node detects a hash key collision in its HAM table, it sends out a COLERR message (cf. Figure 9) with a list of all colliding hash keys. The message is broadcast throughout the whole network and all receiving nodes have to analyze such a message and to delete all corresponding hash keys in their HAM tables.

E. Transparency to the Applications

The name resolution mechanism presented in this paper has to be transparent to the applications. Therefore, all DNS requests from the application layer have to be caught, analyzed and if necessary answered by our network layer. The application should not recognize that the name resolution system has changed. Our name resolution design has the advantage that nodes that do not support the new name resolution system are still able to use conventional DNS requests and replies if there is a DNS server available in the network, which can handle the requests. This is especially important for nodes that are unaware of our new name resolution approach. Our network layer will catch such DNS requests and answer them if possible. If answering is impossible, the DNS request will be forwarded to a DNS server. If a node is aware of our new name resolution system, it can either send DNS requests, which are answered by our network layer or it could use an Application Programming Interface (API) to look directly into the HAM table to increase the performance.

 ISBN: 978-1-61208-231-8

F. Name Resolution over External Networks

Our approach is designed for usage in MANETs, where most data transmissions are limited to the MANET itself and therefore they need no special consideration. For transmissions between a node inside the MANET and a node outside the local MANET (e.g., a DNS request to the Internet), we provide name resolution over external networks.

In our system, we assume that multihomed nodes act as gateways to other networks. These networks could be other ad hoc networks or an infrastructure network with access to the Internet. Unmanned Aerial Vehicles (UAVs) with several interfaces could participate in a MANET and also be connected to a base station with Internet access at the same time. DNS requests can be forwarded to the conventional Domain Name Servers in the Internet.

In reactive networks, requests are broadcast through the network till one node replies. If the request reaches a gateway node, this node can ask the network it couples. To avoid network flooding over several MANETs, we introduce a gateway count in every NREQ packet. Each time a gateway forwards the packet, the gateway count is decreased. If the counter reaches zero, the packet will be dropped. If the gateway node has access to the global Internet, the NREQ messages are converted to normal DNS requests.

If proactive routing is used, the nodes periodically exchange topology information and therefore have an up-to-date routing table at each time. This means that each node has full knowledge about the network and furthermore knows all possible gateway nodes in the current subnet and can therefore send well-directed DNS requests to these gateways if the hostname resolution could not be resolved locally. A gateway node has to check, which name resolution system is used in the networks the name request is forwarded to. Then, it has to convert the request according to the corresponding system to achieve an adaption to the used mechanisms.

IV. Conclusion and Future Work

Our work introduces a framework for efficient name resolution in MANETs based on routing techniques. We used adaptive routing as base of our name resolution to have the best performance in different network scenarios for route finding and name mapping as well.

Because we enhanced the routing mechanism, we did not need an additional protocol for name resolution. This makes our system less complex.

In reactive routing mode, a node can directly search a route to another node's name. Compared to conventional MANETs in reactive operation mode, where a node firstly has to resolve the name and secondly has to find a route, this saves one step and therefore latency.

In our proactive routing mode, each node has all information about names and routes in the local MANET, and therefore, can transmit packets without requesting such information first. Compared to conventional proactive routing modes, this avoids the additional delay for name resolution.

We showed that centralized name resolution approaches cannot cope with the requirements of MANETs. Therefore, our fully-distributed approach eliminated centralized Domain Name Servers and increased the robustness of our MANET against failures.

There is no latency, if the name resolution uses proactive routing and decreased latency, if it uses the reactive mode. We did not need a separate protocol for name resolution by adapting the routing protocols.

As future work, we will extend our addressing scheme and mapping system to a service discovery mechanism. The problem of service discovery is similar to the problem of name resolution. In both cases a (service-)name exists and the system has to resolve this name to an address.

Furthermore, we will consider security aspects to prevent foreign nodes from masquerading other identities.

Acknowledgment

The authors would like to thank the administration and members of the International Graduate School on Mobile Communications (Mobicom) for their support and the German Research Foundation (DFG) for their kind funding.

References

[1] P. Mockapetris, "Domain Names - Concepts and Facilities," RFC 1034 (Standard), Internet Engineering Task Force, Nov. 1987, updated by RFCs 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592, 5936.

[2] P. Engelstad, D. Van Thanh, and T. Jonvik, "Name Resolution in Mobile Ad-Hoc Networks," in *10th International Conference on Telecommunications, 2003. ICT 2003.*, vol. 1, March 2003, pp. 388 – 392 vol.1.

[3] M. Menth, M. Hartmann, and D. Klein, "Global Locator, Local Locator, and Identifier Split (GLI-Split)," Institut für Informatik, Technical Report 470, April 2010.

[4] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626 (Experimental), Internet Engineering Task Force, Oct. 2003.

[5] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561 (Experimental), Internet Engineering Task Force, Jul. 2003.

[6] S. Nanda, Z. Jiang, and D. Kotz, "A Combined Routing Method for Ad Hoc Wireless Networks," Dept. of Computer Science, Dartmouth College, Tech. Rep. TR2009-641, February 2009.

[7] J. Hoebeke, I. Moerman, B. Dhoedt, and P. Demeester, "Adaptive Multimode Routing in Mobile Ad Hoc Networks," in *PWC'04*, 2004, pp. 107–117.

[8] N. Beijar, "Zone Routing Protocol (ZRP)," *Networking Laboratory Helsinki University of Technology Finland*, vol. 9, no. 4, pp. 427–438, 2001.

[9] O. Ponomarev and A. Gurtov, "Using DNS as an Access Protocol for Mapping Identifiers to Locators," in *Proc. of Workshop on Routing in Next Generation*, December 2007.

[10] S. Ahn and Y. Lim, "A Modified Centralized DNS Approach for the Dynamic MANET Environment," in *9th International Symposium on Communications and Information Technology*, Sept. 2009, pp. 1506 – 1510.

[11] S. Cheshire and M. Krochmal, "Multicast DNS (IETF Internet-Draft)," Feb 2011, expires: 18 August 2011.

[12] P. Engelstad, D. Thanh, and G. Egeland, "Name Resolution in On-Demand MANETs and over External IP Networks," in *IEEE International Conference on Communications (ICC '03)*, vol. 2, May 2003, pp. 1024 – 1032.

[13] R. Braden, "Requirements for Internet Hosts - Application and Support," RFC 1123 (Standard), Internet Engineering Task Force, Oct. 1989, updated by RFCs 1349, 2181, 5321, 5966.

[14] F. Yergeau, "UTF-8, a Transformation Format of ISO 10646," RFC3629 (Standard), November 2003.

 ISBN: 978-1-61208-231-8