Towards a New Communication Paradigm for Mobile Ad Hoc Networks

Vincent Lenders, Martin May, and Bernhard Plattner
Swiss Federal Institute of Technology (ETHZ)
Zurich, Switzerland
Email: {lenders, may, plattner}@tik.ee.ethz.ch

Abstract— **In mobile ad hoc networks, we envision a network where mobile users obtain services from close-by instances. The architecture of today's Internet was designed for fixed users that obtain services from stationary servers and is not well suited for such scenarios. The reason is that (i) the architecture combines identity and location in an IP address and thus forces mobile elements to change their identity when moving over subnet boundaries; and that (ii) the layered architecture implies a separation of service discovery/selection and routing, which is inflexible and also leads to protocol overhead. In this paper, we revise the existing Internet architecture and propose a novel architecture that is better suited for mobile ad hoc networks. There, clients bind to location-independent service identifiers and send packets that are routed to any instance of the desired service in proximity. The routing mechanism is based on the concept of (electrical) fields with which packets are forwarded towards a region with a high density of service nodes. As a result, this architecture increases the probability of successful packet delivery and leads to a robust routing substrate even in very unstable network conditions.**

I. Introduction

In Mobile Ad hoc NETworks (MANETs) [1], one can think of many services that are provided not only by single nodes, but by multiple instances in the network. Typical services include for example Internet access points, location-based information for tourists or shoppers, timetables for passengers, taxis requested by pedestrians, game servers for mobile players, data sinks in sensor networks, etc. A lot of research efforts [2], [3], [4], [5], [6], [7] have been made to adapt the Internet architecture to such scenarios. We however argue that today's Internet architecture, and in particular the Internet protocol (IP) [8], is not well suited to efficiently implement such applications. This is mainly because the Internet architecture was designed for data transfers (for example downloading a file from a server) between fixed clients and servers.

We illustrate the limitations of the existing Internet architecture for the MANET scenario depicted in Figure I. We assume that mobile users (symbolized with notebooks) may obtain the same service from three different service nodes. If a user is close enough (in direct wireless transmission range) to a node hosting the service, it directly accesses the service. Otherwise, the client accesses the service over a series of hops via intermediate mobile nodes. With the today's Internet architecture, several steps are necessary before the mobile user accesses a service with his notebook:

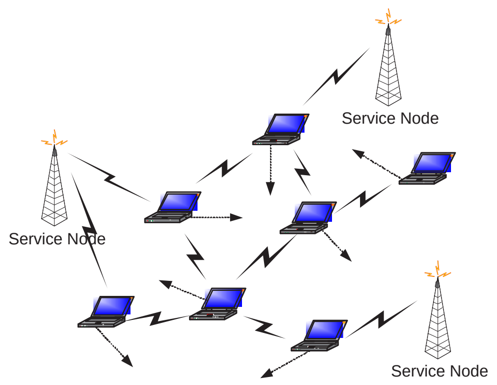

Fig. 1. Mobile Ad Hoc Networking with Multiple Service Nodes

1) A valid IP address must be assigned to all network elements (service nodes and notebooks).
2) A user has to discover and locate the available service nodes and corresponding IP addresses by any kind of lookup mechanism (e.g., by flooding the network).
3) The user selects one service instance and binds himself to it for future invocation.
4) If not previously established by a proactive routing protocol (e.g., DSDV [4], OLSR [5]), a route is setup from the IP address of the mobile user to the IP address of the selected service node by a reactive routing protocol (e.g., AODV [3], DSR[2]).

Using the Internet architecture for such mobile scenarios leads to multiple problems: First, network elements and services are identified with their IP addresses. However, IP addresses are structured and have an inherent topological meaning. Therefore, nodes are forced to change their identity when moving. Furthermore, the services lookup mechanism is handled by a higher layer protocol on top of the routing layer. This inherently produces more control overhead traffic and also forces the client to initially lookup the network address of the service instance to be use and to bind to it. When the client or the service node move away, the client sticks to the chosen service, even if an alternative and closer service instance becomes available. The described shortcomings of the Internet architecture were our main drivers for designing a new architecture.

The contribution of this paper is the design of a new architecture for mobile ad hoc networks with the following properties regarding naming/addressing and routing

Naming and Addressing:

- Whenever possible, clients communicate with services by binding themselves to service types instead of specific service instances.
- Global addresses are not required. It is sufficient for a client to communicate with one service among all instances of the same type.
- We do not identify communication end-points with location-dependent IP addresses but with location independent unique identifiers. These identifiers are created by the network elements themselves when required.

Routing:

- We address service discovery and routing simultaneously to reduce control overhead.
- We do not aim for routing optimality (e.g., determination of the shortest path). Moreover, we consider the probability of successful packet delivery the most important property in an unstable environment.

The rest of the paper is organized as follows. In the next section, we describe our architecture. We then discuss direct implications of our architecture in Section III, discuss related work in Section IV, and conclude in Section V.

II. Architecture

We now present our architecture for communication in mobile ad hoc networks. We first introduce the naming and addressing model of our architecture. Then, we describe how to route packets using a field-based routing approach [9], [10], [11]. Finally, we show how different applications communicate with our architecture by the means of three different communication patterns.

A. Naming and Addressing

A novelty of our architecture compared to traditional IP networking is that we use location-independent identifier spaces for addressing and naming. We introduce two of these name spaces, *service type identifiers* (STID) and *unique identifiers* (UID). STIDs are used for identifying services classes of the same service type, whereas UIDs uniquely identify entities in a network including both service instances and clients.

A STID is a service description. Thus, different service instances have the same STID if they offer the same service. Every service instance must have one STID. Our architecture does not impose any restrictions on the semantics and structure of STIDs. In principle, flat binary strings can be used as well as more sophisticated structures such as hierarchical, human-readable names (e.g., as described in [12], [13], [14]). We show later that clients access services using STIDs and must therefore know the STID of a service for communicating with it. The mechanism to lookup STIDs is outside the scope of this paper. However, to avoid extensive lookups by clients, we propose to use a set of a priori known STIDs, defined for the most common services types when implementing our architecture. Such a concept is for example used in the Bluetooth system [15] where well-known profiles are defined for common services.

The second type of identifiers are UIDs. UIDs uniquely identify clients and service instances. UIDs are similar to IP addresses in the sense that packets are routed according to UIDs to the owner of the UID. However, UIDs are used quite differently. First, a UID is not assigned to a network interface but rather to a client or a service process. Another main difference is that UIDs are generated locally, on the fly, only when required. Typically, a client creates a UID just before contacting a service to allow reply packets from the service to be received. If a client does not require a response from the service, a UID is not needed. An example of the latter is a pedestrian (the client) requesting a taxi (the service) that simply sends a request containing its current position. Since the taxi might not need to send back a message to the client (the taxi just drives to the client), the taxi does not need a UID. Service instances only require a UID in addition to their STID if it is necessary that a client binds to a specific service instance. For example, when the service is a printer, it is necessary that all packets of the printing job arrive at the same printer. For connection-less services, where the specific instance which provides the service is not relevant, the STID of a service is sufficient and a UID is not necessary at services.

Just as STIDs, our architecture does not impose any restrictions on the semantics or structure of UIDs. The only restriction we put on UIDs is that the UID has to be unique in the reach/range of the network. Since UIDs are generated locally on demand, we need a mechanism to guarantee global uniqueness and avoid duplicate UIDs. Instead of using complex coordination mechanisms, we propose to use statistical mechanisms which consist of choosing random values from a large identifier space (> 128 bits). This does not guarantee absolute uniqueness, but duplicate identifiers are so rare that they are not of practical relevance.

Note again that UIDs are only used to identify clients and service instances. Network nodes which only relay packets for others in the MANET do not need a UID. Note also that, since only clients and services have network identifiers, the remaining nodes are not directly addressable within the network. This is totally different to the original IP architecture where all network nodes and routers have to be addressable.

We will see later in this section that the UID and STID name spaces are sufficient for the most common communication types between client and service.

B. Routing

Our architecture has to support two different routing mechanisms to route packets based on STIDs and UIDs. Since multiple services have the same STID and since a packet must be delivered to any one of those instances, STID-based routing is a form of *anycast*[1] routing. On the other hand, UIDs always

[1]We use the term anycast routing in this paper as a delivery mechanism where a packet is delivered to exactly one member from a group of members.

belong to a single entity and UID-based routing is thus a form of *unicast* routing. Unicast routing can be viewed as anycast routing with a group size of one, and therefore we propose to apply anycast routing to both cases.

As mentioned before, our design goal is not to perform optimal routing (for example keeping the number of hops as low as possible) but increase the probability of successful packet delivery and thus increase the robustness. In contrast to routing in stable networks, such as for example the Internet, this issue is much more important in MANETs because packets are dropped more frequently due to network dynamics. We propose the use of field-based routing as introduced in [10], [11]. Field-based routing is an anycast routing mechanism designed to increase the robustness in unstable networks such as MANETs. Field-based routing achieves this goal by routing packets in the direction of the highest service density instead of routing to the closest entity as done with traditional IP anycast routing [16]. The intuition behind is that packets are routed towards the location with the most services and hence with high delivery probability; packets are still delivered even if some services move away or disappear. Furthermore, field-based routing allows to differentiate services based on the capacity of the service instances and perform dynamic load balancing if necessary.

In the following, we give an overview of field-based routing as originally introduced in [10], [11], and then discuss how to use it in our context (field-based routing was introduced in previous work as a service discovery mechanism and not for communication between clients and services).

1) Field-based Routing: Field-based routing is inspired from the physical behavior of a probe charges in an electrostatic field. Communication end-points (in this case, service instances or clients) are modeled as positive point charges and data packets as negative test charges which are forwarded in the direction of the field's flux line (steepest gradient) to point charge(s).

A field is expressed by potential values at each node in the network. The potential value of each node is obtained as follows: Consider a charge $Q_j > 0$ at node j and an arbitrary node n. Then, we define the function $dist(j,n)$ as the distance between j and n. This is defined as the minimum number of nodes a message has to pass for propagating from j to n. Note that other metrics are also possible if desired such as for example, the time required for a message to propagate from j to n. The potential value at node n resulting from this charge is defined as

$$\varphi_j(n) = \frac{Q_j}{dist(j,n)^k} \tag{1}$$

where k is a constant > 0. This type of function decreases with increasing distance from node j (and has a very large or "infinite" value at node j itself). The exponent k defines how quickly the potential decreases and is set to $k = 1$ to stick to the physical analogy[2].

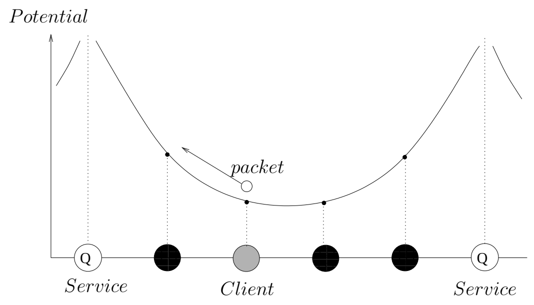

Fig. 2. Potential distribution with two charges. A data packet is forwarded to the left service instance along the steepest ascent of the potential.

Now consider N charges $Q_j > 0, j = 1..N$ belonging to the same field. The potential at an arbitrary node n is defined as

$$\varphi(n) = \sum_{j=1}^{N} \varphi_j(n) \tag{2}$$

Therefore, the potential value of a node is the linear superposition of the potential terms φ_j from all charges. Notice that this superposition of the single potentials is the fundamental mechanism which finally leads to a density-based routing scheme we are aiming at.

In physics, a negative test charge diffuses in an electric field along the steepest ascent of the potential. In analogy, we route data packets towards the steepest ascent in our potential fields. Consider a simple example in Figure 2 with the potential distribution as pictured. A data packet from the client node is forwarded along the steepest ascent to the left service instance. We achieve this by evaluating the potential of the field at each hop. For this, each node maintains a list of all direct neighbors (nodes in direct transmission range) including potential values and link layer (MAC) addresses.

Then, when a node receives a packet, it forwards the packet to the link layer address of the neighbor with the highest potential value among its neighbors. In case different neighbors have the same maximum potential value, the next hop is chosen arbitrarily.

Following the steepest gradient guarantees that loops cannot occur even when the network is dynamic since in a loop it is not possible that the potential value increases at each hop. Also the field-based routing approach is specifically resilient to link failures and topology changes as packets always reach their destinations (a point charge) as long as each hop knows at least one neighbor with a larger potential value than itself.

In addition, our scheme easily integrates load balancing by mapping the current service capacity to the charge Q. Services which advertise themselves with a large Q establish a high potential distribution in their neighborhood and consequently

[2]In analogy to physics, the electrical potential at position $\vec{r}$ which relates to a point charge Q_j located at $\vec{r_j}$ is $\varphi_j(\vec{r}) = \frac{1}{4\pi\varepsilon}\frac{Q_j}{|\vec{r}-\vec{r_j}|}$. Note that this function is continuous whereas in our definition, the potential function has discrete values at the network nodes.

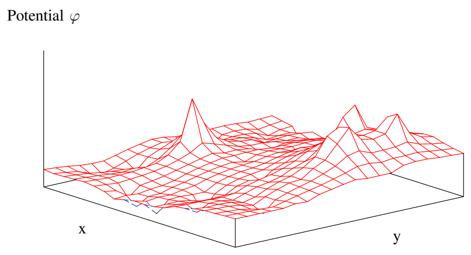

Fig. 3. Typical STID Routing field with 4 service instances (at the peaks).

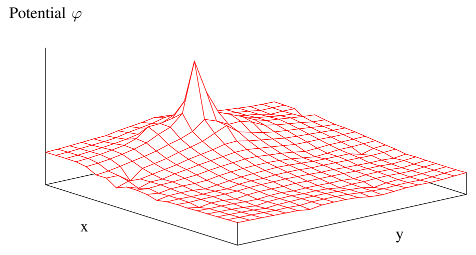

Fig. 4. Typical UID routing field.

attract more packets than services which advertise themselves with a small Q. During high load, the service charge is reduced to attract less requests and as result, requests are equally distributed among all services instances of the same type.

2) Applying Field-based Routing to our Architecture: We use field-based routing to route packets based on STIDs and UIDs. A separate field is established per STID and per UID. Therefore, a series of fields co-exist on the network simultaneously. Note that fields are soft state and it is necessary to constantly update them when topology changes occur.

We obtain the routing fields of STIDs by assigning a charge Q to all service instances of the same STID. A typical STID routing-field is pictured in Figure 3. Peaks are visible at the location of each service instance. By assigning a charge at each service instance which is proportional to the current load of the service, we obtain a field which reflects the momentary availability of the neighboring service instances. Since STIDs fields are used to initiate the communication between clients and services, we propose to establish them pro-actively with a mechanism as for example the one described in [10], [11]. However, it is in principle also possible to establish such fields on demand only when needed (when a client has to send a packet to a specific STID).

UID fields are obtained by assigning a charge to the owner of the UID (a service instance or a client). Unlike STID fields, the absolute charge intensity of UIDs does not have an inherent meaning. Only the relative descent between neighbors is relevant. A typical UID field with a single peak at the owner of the UID is pictured in Figure 4. Note that since UIDs are unique by design, all packets addressed to a UID will reach the owner. In fact, packets in single charge fields are always routed on the shortest path according the metric used to calculate the distance function $dist()$ in Equation (1). This is because the potential is inverse proportional to the distance and packets are forwarded along the steepest ascent at each hop. As UIDs are transient identifiers which are created on demand, we establish UID fields on demand only when needed.

C. Communication Patterns

We now show how clients and services communicate using STIDs and UIDs in our architecture. For this, we differentiate three basic communication patterns (see Figure 5 for an

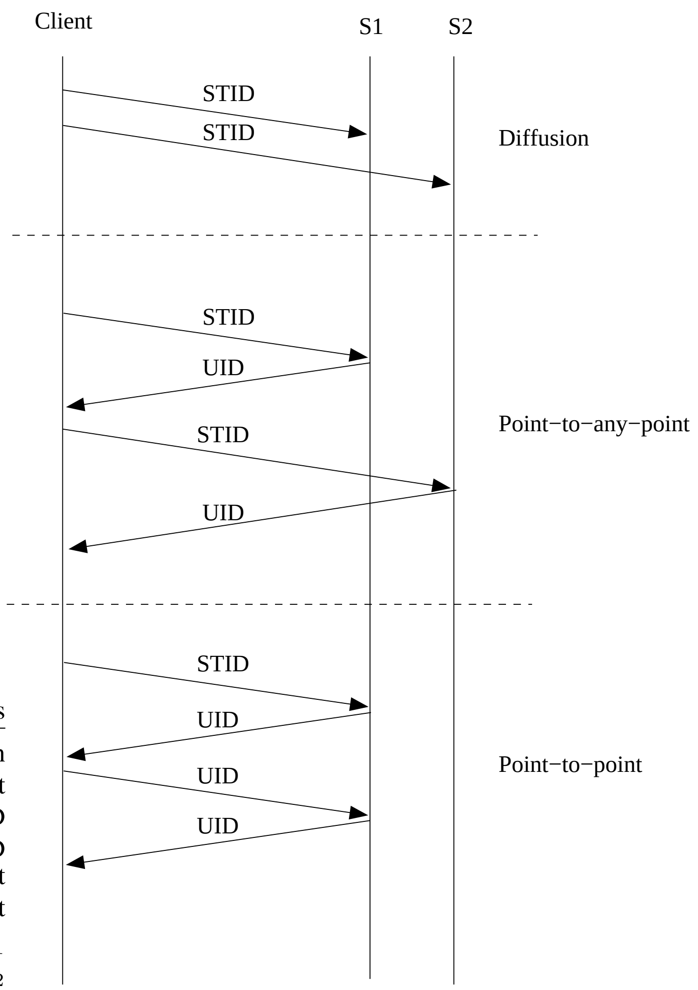

Fig. 5. Overview of the three communication patterns: In this example, a client sends two consecutive data packets. The two service instances S_1 and S_2 have the same STID. With the diffusion pattern, the client does not bind to a specific service instance, and it is possible that each packet arrives at a different service instance (as shown in this example). The diffusion pattern does not allow reply packets to the client. The point-to-any-point pattern is similar to the diffusion pattern but reply packets to the client are possible. With the point-to-point pattern the client binds to the receiver of the first STID packet (service S_1) and both packets are sent to the same service instance.

overview). The reason why we differentiate between these patterns is that each pattern requires a different handling at the protocol and routing layer.

All three communication patterns start with a client sending a data packet including the STID of the desired service type. This data packet is then routed to *any* service instance of this type. The three communication patterns differ in the remaining part of the communication after the initial setup.

We call the first and simplest communication pattern *diffusion*. In this communication pattern, a client sends *one* or multiple independent data packets to any service. Such a pattern is typical for sensor applications where sensor nodes (clients) send sensed data to any sink (the service). With this pattern, sending the packet with the desired STID is sufficient and UIDs at the client or at the service are not require. Note that in this case, a client might send more than one packet by repeating the diffusion pattern, however, there is no guarantee that consecutive packets reach the same service instance.

The second pattern is a little more sophisticated. We call it the *point-to-any-point* communication pattern. In this pattern, a client sends one data packet to any service instance, and the service replies one packet back to the client. Typical applications showing such a communication pattern include query-reply applications or reliable sensor applications where a receiver needs to acknowledge received packets. With this pattern, in order for the service to be able to reply to the sender, it is necessary that the client has a UID and that the service knows this UID. Thus, when communicating using this pattern, the client first creates a UID and appends it into its (first) packet to the service. Since the service instance only replies with one packet, it is not necessary to establish the UID field of the client throughout the network which would cause a large overhead. We assume that the network topology will change during this short communication period and that the reverse path from the client to the service instance will still be available when the service sends its reply packet. Therefore, it is sufficient to establish the UID field at the nodes which are on the forward path to the service.

The last and most complex communication pattern is called *point-to-point*. With this pattern a client sends an arbitrary number of packets to the same service instance and the service back to the client. Applications with such a communication pattern include all applications where the client binds itself to a specific service instance such as for example the printer service. To achieve this pattern, both the client and the service instance need a separate UID. The mutual exchange of UIDs works as follows: The client creates a UID and appends it to the first packet sent to the STID of the service instance. In addition, the client requests the service to generate a UID. When the service instance receives this packet, it in turn generates a UID and appends it to the reply packet for the client. Note that, if the service instance already has a UID (because it created one earlier to communicate with another client), it is not necessary to generate a new UID, but the same UID is used to communicate with both clients. After these two packets, both end-points know their mutual UID and communicate with each other using the corresponding UIDs. Again, it is not necessary to establish the UID fields throughout network. It is sufficient to establish a field in the neighborhood between the client and service. The field range should be adapted to the dynamics of the network. For nearly static networks, it suffices to establish the field on the path from the client to the service (for example on the path where the first STID packet traversed). Whereas for highly dynamic networks, the field must be established in a larger neighborhood to allow sender, receiver, and intermediate nodes to move without interrupting the connection.

III. Discussion

In this section we discuss the implications and benefits of our design choices.

A. IP Addresses vs. UIDs

IP addresses are used to indicate both network location and client/service identity. This approach has several advantages. For example it is possible to aggregate IP address ranges into domains for scalability purposes. However, routing with mobile entities is more difficult. We propose to identify communication end-points with location-independent random identifiers (UIDs). Thus, communication end-points do not need to change their identity when moving.

B. Combining Service Discovery and Routing

Combining service discovery and routing has two major advantages in our architecture. First, it allows to design a single protocol for distributing routing and service information state, thus reducing control overhead. Second, it allows for clients binding to service type identifiers (STIDs) for connectionless services without binding themselves to a specific service instance. Such a scheme is much more appropriate for mobile communication where a client always communicates with the optimal service instance as he moves and as the network conditions change.

C. Routing Optimality vs. Robustness

Traditional IP routing mechanisms such as link state or distance vector routing have the objective to minimize routing costs. A cost is assigned to each link and the optimal route is the path with the minimal sum of the link costs. The objective of our field-based routing approach is quite different. Instead of calculating a path which strictly minimizes the routing cost, we route packets in the direction of the highest service concentration. The motivation behind this design choice is to increase the probability of successful packet delivery. The reason why this increases the probability of delivery is straightforward: We have to consider that nodes may be moving and disappear at any time. It is statistically more probable that a packet reaches a service when packets are routed towards a cloud of redundant service instances instead of routing towards a single service instance which is a potential "single point of failure".

D. Service Discrimination and Load Balancing

The choice of using field-based routing for accessing a service makes it easy to accommodate to the current load of close-by services. The strength of the charge Q assigned to each service of a STID field is used as an indicator for the load or capacity of the service. A service with high capacity/load uses a large value of Q, and thus generates an elevated potential distribution in its proximity. As a consequence, more packets are attracted towards this service instance. Balancing client load is done by varying the charge Q in accordance to the current load of a service. A service overloaded by client requests gradually reduces Q. As a result, the potential around the overloaded service decreases and packets are routed to alternative service instances. As soon as the load is tolerable, the original potential is reestablished by increasing Q.

E. Scalability

STID routing fields are maintained pro-actively by flooding packets from the services. Extensive flooding is always a major scalability concern. However, only STID routing-fields must be established pro-actively. Therefore, the number of pro-active flooding is proportional to the number of services instances. We expect the number of service instances to be significantly smaller than the overall number of nodes in the network. UID routing fields are less critical since they only have to be maintained for the duration of active communication and are established in a limited scope around the client and the service.

To further improve scalability, we propose to limit the distribution of routing field information if the potential value drops below a certain threshold value. As a consequence, certain services will remain invisible for distant clients. This issue is straightforward since in most application scenarios, the utility of a service is directly related to its proximity.

F. Security

A major security threat of IP networks are Denial of Service (DoS) attacks. In a DoS attack, one or more malicious hosts send a large amount of unwanted traffic to a victim, unable to process all incoming traffic. As a side-effect, the victim is no longer able to handle its desired traffic. Our architecture mitigates this kind of threat. First, the routing fields are limited in scope. UID fields are established only in a restricted range between clients and services, and STID fields allow a sender to send packets only to one service (the one on the steepest gradient of the field). Thus, an attacker would have to place himself at strategic locations to be able to send unwanted traffic to a selected victim. This is a much higher burden as compared to IP networks, where a host can send packets to any other hosts in the network. Also, unwanted traffic addressed to a specific UID can be mitigated by changing the UID in use, since UIDs are created locally and on the fly. Thus, a victim host is able to creat a new UID and communicates this one to the hosts with which it currently communicates.

IV. Related Work

The limitations of the Internet architecture for communication in mobile and/or ad hoc networks has led to a lot of research in the past years. Here, we present related work proposing new solutions for naming, addressing and routing in these networks.

A. Naming and Addressing

The concept of indirection was proposed to overcome the problem that hosts have to change their identity (IP address) when moving to a different network location. For example, Mobile IP [7] or i3 [17] propose to communicate over middleboxes. These middleboxes are fixed and therefore do not have to change their address. A signalling protocol takes care to inform the middlebox when a mobile host changes its address but this change remains transparent for the other communication end-point which communicates via the address of the middlebox. The concept of middleboxes might work for networks where there exists a fixed infrastructure. However, in mobile ad hoc networks, the inherent lack of infrastructure makes such solutions unsuitable.

Another approach to handle mobility is to add additional layers in the IP communication model. For example, in [18] and [19], the authors propose to use an additional location-independent name space on top of the IP address name space for identification of persistent communication end-points. These persistent names are then resolved to an IP address. These solutions are targeted at the Internet and were designed to keep the installed Internet routing infrastructure, which routes packets based on IP addresses, unchanged. Thus, the communication is still based on an underlying IP routing. We believe that it is more promising to completely revise the IP communication architecture instead of fixing the existing flaws.

B. Routing and Alternative Communication Paradigms

IP routing in mobile ad hoc networks has been researched extensively [2], [3], [4], [5], [6], [20] in the past. However, these works focus only on how to efficiently distribute routing information and react to link or route failures when nodes are moving. These solutions do not propose complete communication solutions to improve the fundamental limitations of IP.

The idea of combining service discovery and routing is not new. Publish/subscribe systems (e.g. [21]) or the Intentional Naming System (INS) [22] have been proposed in this context. However, these systems have mostly been studied in wired networks without taking full advantage of the broadcast nature of wireless radio networks which is the case of field-based routing.

IP anycast [16] has been proposed as means for service discovery [23], [24] . However, IP anycast has severe limitations. In IP anycast, a packet is always delivered to the closest host of an anycast group and unlike our approach does not support important features such as density-based forwarding, service discrimination, and load balancing.

Alternative communication paradigms have been proposed for sensor networks. For example, directed diffusion [25] was proposed to collect data in sensor networks. Directed diffusion is similar to the diffusion communication pattern in this work. In this paper, we go a step further and combine different communication patterns in single architecture.

V. Conclusions

This paper proposes a new architecture for ad hoc communication in mobile networks. The novel architecture is service-centric by naming services and clients instead of network interfaces and considers mobility by design. The name space consists of location-independent identifiers which allows entities to keep their identity while moving. Also, we provide a set of identifiers (STIDs) which allows for clients communicating with services without binding themselves to specific service instances. Thus, the optimal service is always contacted as a client moves. We also revise the routing mechanisms of IP routing to increase the robustness. By forwarding data packets in the direction of the highest service concentration, we increase the probability that packets reach a service when nodes are mobile.

There are still open research problems with regard to our architecture. So far, we have investigated communication between a client and a service. We are currently investigating if it is possible to implement more complex communication schemes (such as for example multicast) with the three basic communication patterns introduced in this paper (see Section II-C).

It is also worth mentioning that we do not consider our architecture as the ultimate solution for ad hoc communication in mobile networks. Moreover, we consider this work as an initial step towards a novel communication paradigm uncoupled from IP. We believe that more research efforts are necessary to better understand the limitations of IP in mobile ad hoc environments and to design applicable network architectures.

Acknowledgments

The authors would like to thank Ulrich Fiedler for his valuable feedback. This work was partly funded by the Swiss National Science Foundation (SNF) under grant 200021-103578.

References

[1] "IETF Mobile Ad-hoc Networking (manet) Working Group," http://www.ietf.org/html.charters/manet-charter.html, 2004.

[2] D. B. Johnson, D. A. Maltz, Y.-C. Hu, and J. G. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," February 2002.

[3] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing," IETF Internet Draft, draft-ietf-manet-aodv-12.txt, November 2002.

[4] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," in *Comp. Commun. Rev.*, October 1994, pp. pp 234–44.

[5] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol," IETF Internet Draft, draft-ietf-manet-olsr-11.txt, July 2003.

[6] R. Ogier, M. Lewis, and F. Templin, "Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)," IETF Internet Draft, draft-ietf-manet-tbrpf-09.txt, June 2003.

[7] C. Perkins, "IP Mobility Support," IETF RFC 2002, October 1996.

[8] J. Postel, "Internet Protocol," IETF RFC 791, September 1981.

[9] A. Basu, A. Lin, and S. Ramanathan, "Routing Using Potentials: A Dynamic Traffic-Aware Routing Algorithm," in *Proceedings of the ACM annual conference of the Special Interest Group on Data Communication (SIGCOMM'03)*, Karlsruhe, Germany, August 2003.

[10] V. Lenders, M. May, and B. Plattner, "Service Discovery in Mobile Ad Hoc Networks: A Field Theoretic Approach," in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Taormina, Italy, June 2005.

[11] ——, "Service Discovery in Mobile Ad Hoc Networks: A Field Theoretic Approach," *Elsevier Journal on Pervasive and Mobile Computing*, vol. 1, no. 3, September 2005.

[12] S. Microsystems, "Jini Architecture Specification," version 2.0, June 2003.

[13] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan, "Service Location Protocol," RFC 2165 (http://www.ietf.org/rfc/rfc2165.txt), June 1997.

[14] Microsoft, "The Universal Plug and Play (UPnP) Forum," http://www.upnp.org, 2003.

[15] *Specification of the Bluetooth System*, 1st ed., February 2001.

[16] C. Partidge, T. Mendez, and W. Milliken, "Host Anycasting Service," IETF RFC 1546, November 1993.

[17] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet Indirection Infrastructure," in *Proceedings of ACM SIGCOMM Conference*, Pittsburgh, PA, USA, August 2002.

[18] R. Moslowitz, P. Nikander, and T. Henderson, "Host Identity Protocol," draft-ietf-hip-base-01, IETF Internet draft, October 2004, work in progress.

[19] H. Balakrishnan, K. Laksminarayanan, S. Ratnasamy, S. Shenker, I. Stoica, and M. Walfish, "A Layered Naming Architecture for the Internet," in *Proceedings of the ACM SIGCOMM*, Portland, Oregon, USA, August 2004, pp. 343–353.

[20] V. Park and S. Corson, *Temporally-Ordered Routing Algorithm (TORA)*, IETF Internet Draft, July 2001.

[21] T. S. Inc., "TIB/Rendezvous Concepts," Palo Alto, CA, Tech. Rep. Release 6.4, October 2000.

[22] W. Adjie-Winoto, E. Schwartz, and J. Lilley, "The Design and Implementation of an Intentional Naming System," in *Proceedings of the 17th Symposium on Operating Systems Principles (SOSP '99)*, Charleston, SC, USA, 1999, pp. 186–201.

[23] T. Hardie, "Distributing Authoritative Name Servers via Shared Unicast Addresses," RFC 3258, April 2002.

[24] D. Kim, D. Meyer, H. Kilmer, and D. Farinacci, "Anycast Rendevous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)," RFC 3446, January 2003.

[25] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *Proceedings of the ACM/IEEE Internation Conference on Mobile Computing and Networking (Mobicom '00)*, Boston, USA, 2000.