Wireless Ad Hoc VoIP

Patrick Stuedi and Gustavo Alonso

Department of Computer Science
Swiss Federal Institute of Technology (ETH Zurich)
8092, Zurich, Switzerland
Contact student: stuedip@inf.ethz.ch

ABSTRACT

VoIP is one of the traditional application scenarios for Mobile Ad Hoc Networks (MANETs) in settings such as emergency response. Ideally, VoIP would be transparent to the network type such that users would not have to worry whether their machine is currently part of a MANET or attached to the Internet. However, setting up a VoIP session using the Session Initiation Protocol (SIP) is difficult in MANETs since SIP relies on centralized components. In this paper, we show that the goal of transparent VoIP, for both isolated and Internet connected MANETs, can be achieved using SIPHoc, a SIP middleware for MANETs. We further discuss the deployment on laptops and iPAQ handhelds, and we look at various SIP interoperability issues.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design - Wireless communication

General Terms

Design, Experimentation

Keywords

VoIP, SIP, Ad Hoc Networks

1. INTRODUCTION

Everywhere in the world people have a basic need to communicate, whether this is with neighbors down the road or friends and family on the other side of the country. VoIP is the upcoming technology for voice communication using the Internet with IP packet transmissions as the base carrier. VoIP has several advantages that come hand in hand with its coupling to the Internet. It is cost free when both participants are directly attached to the Internet, and typically cheaper than the traditional telephone system even if only one participant is attached to the Internet. Moreover, VoIP allows to easily combine telephony with other services known from the Internet, such as video, chat, file sharing, etc.. The tight coupling with the Internet, however, also poses problems. First, Internet access may not always be available. Second, Internet access, e.g. wireless, may not be cost free. Mobile Ad Hoc Networks (MANETs) are one way to cope with these issues. MANETs do not require any pre-configured infrastructure, such as the Internet, and communication within MANETs is free. These properties make MANETs attractive for VoIP and lead to many interesting application scenarios. For instance, VoIP over MANETs could be used to help developing countries improve their communication. Many developing nations consist of large expanses of desert or other terrain without reliable power sources. In these areas, setting up base stations for mobile phone communication is either unworkable or would involve large investments of money and time. Another potential application scenario is in densely populated areas like big cities or on a university campus. Here, VoIP over a MANET would provide users with a free communication system. MANETs are further envisioned as playing a significant role in emergency response situations in which the network infrastructure might temporarily be broken [14] [11] [8].

VoIP could work in many different devices - ranging from cheap mobile phones used in the university campus environment, to the handheld organisers commonly run on a company network. At a fringe cost, any handheld device - an organiser, gamepad or camera, for example - can be transformed into a wireless phone and text communicator simply by adding a small piece of software. Ideally, the use of VoIP would be transparent to the network type such that users would not have to worry whether their device was currently part of a MANET or attached to the Internet. If that were the case, a wireless interface and a VoIP application would be enough to support all the application scenarios mentioned above. However, VoIP typically uses the Session Initiation Protocol (SIP) to set up a session between the two parties of the call [16]. As SIP was designed for the Internet and relies on centralized components, setting up a VoIP session using SIP is difficult in MANETs. Thus, additional mechanisms are required to cope with the decentralized nature of MANETs and to support calls to and from the Internet whenever nodes in the MANET have Internet connection. In this paper, we are going to show that the goal of transparent VoIP for both isolated and Internet connected MANETs can be achieved using SIPHoc [17], a SIP middleware for MANETs. We also discuss the deployment on laptops and iPAQ handhelds, and we look at various SIP

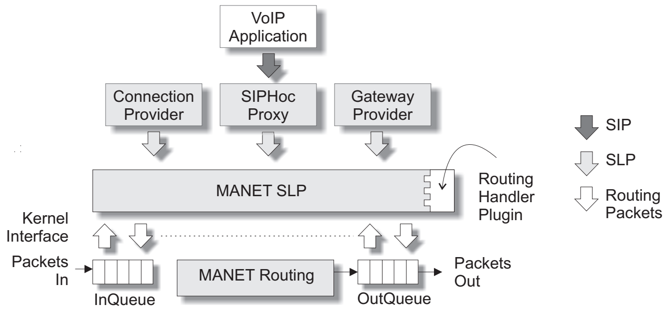


Figure 1: Components used to provide wireless ad hoc VoIP

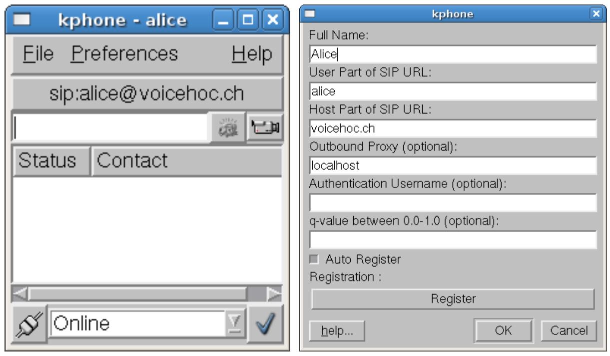


Figure 2: An example VoIP application and how to configure it for use with SIPHoc

interoperability issues.

Our paper is structured into five sections: Section 2 describes the components we use to provide VoIP in MANETs. In Section 3 we talk about how to actually run ad hoc wireless VoIP. The deployment of our system on different platforms/devices is discussed in Section 4. Related work is discussed in Section 5. Section 6 concludes the paper.

2. SYSTEM OVERVIEW

The system is built on top of SIPHoc [17], a SIP middleware for MANETs. The architecture is shown in Figure 1. It is based on five components running as independent operating system processes within a node in the MANET.

- A *VoIP application*. This can be any SIP compatible VoIP client like, e.g., Kphone [1], Twinkle [5], Ekiga [3], etc..
- A *SIPHoc Proxy* with a standard SIP interface but implementing MANET-specific functionality. Each *SIPHoc Proxy* serves as an outbound SIP proxy for the local VoIP application.
- A *MANET SLP* layer providing a regular SLP (Service Location Protocol) interface but implementing efficient and decentralized service lookup functionality. MANET SLP works by piggybacking service information onto routing messages. This is done by capturing routing messages (using the *libipq* library under linux) and extending them with service information. To assure generality, the routing specific functionality is encapsulated within a *routing handler*. The routing handler is a software module that receives raw routing packets as input and generates altered packets that include the piggybacked service information.
- A *Gateway Provider* that, if a node has Internet connectivity, makes this information available to other nodes by publishing an SLP *gateway* service. It also starts a layer two tunnel server ready to accept connections.
- A *Connection Provider* that manages connections of the node to the Internet when there is a gateway in the MANET. It periodically checks whether it can find an *gateway* service (using MANET SLP) and open a layer two tunnel connection to the node offering the tunnel server. Since the gateway node will directly forward all the traffic it receives on the tunnel interface to the Internet, any node with a tunnel connection is automatically attached to the Internet as well.

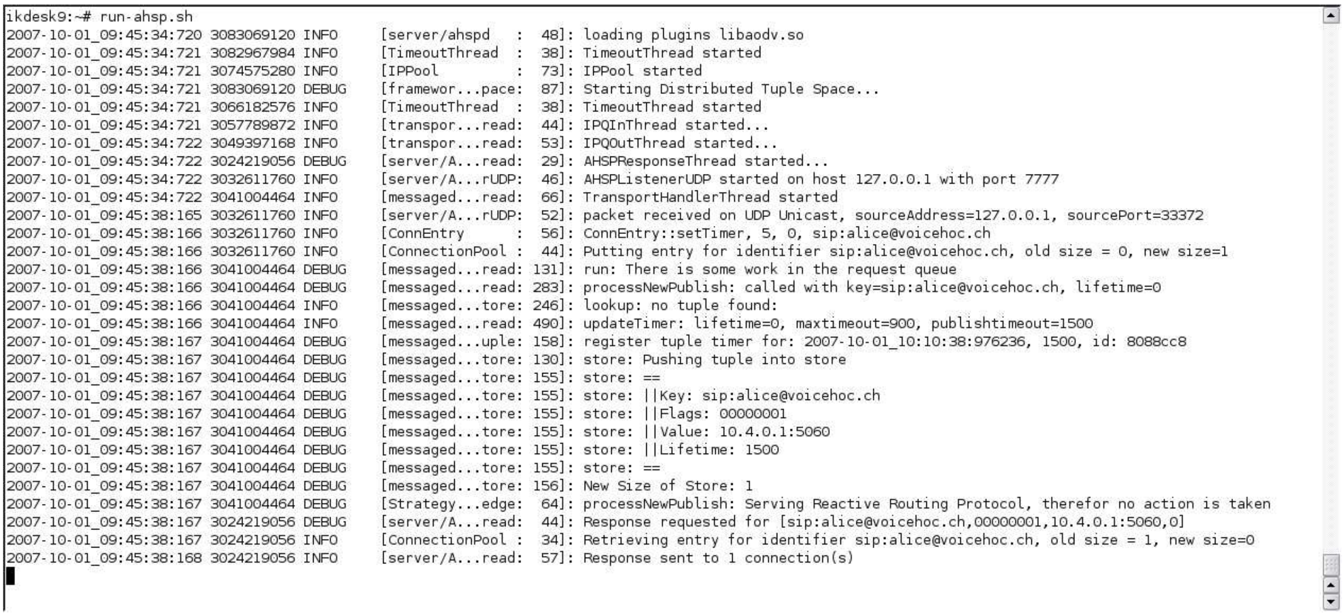
A complete description of each of the components can be found in [17]. Rather than going into more details of the architecture, we want to show in the next section how the system actually provides wireless ad hoc VoIP.

3. RUNNING OUT-OF-THE-BOX VOIP APPLICATIONS

3.1 Making calls within the MANET

The best way show how the different components of the system interact with each other is by means of an example. One goal of this project was to allow out-of-the box VoIP applications to run seamlessly in MANETs. Let's therefore see what it takes to run Kphone in a MANET without any centralized server by using SIPHoc. Typically, VoIP applications require a SIP configuration for your SIP user account. Imagine that your SIP provider is *voicehoc.ch* and your username is *Alice*. In that case, what you would do is configure your Kphone application similar to what is shown in Figure 2. The only difference to the traditional configuration for use in the Internet is that an outbound proxy is specified. By specifying the outbound-proxy to be *localhost*, we make sure that all the SIP traffic is routed through the SIPHoc proxy running locally.

The following actions refer to the steps in Figure 3. VoIP applications – once they are started – register with the SIP proxy that is either pre-configured or can be deduced from the domain part of the SIP URI [16]. Since we configured the outbound proxy of our VoIP application, it registers with the local SIPHoc proxy (step 1). The SIPHoc proxy will then advertise itself with MANET SLP (step 2). Other users similarly register with their local SIPHoc proxy (steps 3 and 4). Figure 4 shows the state of the MANET SLP process after the SIPHoc proxy has advertised its own SIP endpoint address (10.4.0.1:5060 in our case) as the responsible contact address for the given user (Alice). Since MANET SLP is a distributed SLP service designed for ad hoc networks, this information is available to all nodes in the network. Since information between different MANET SLP processes is exchanged via routing message piggybacking, we have to load the right plugin for the routing protocol we are using (second line in Figure 4).



```
ikdesk9:~# run-ahsp.sh
2007-10-01_09:45:34:720 3083069120 INFO     [server/ahspd   :  48]: loading plugins libaodv.so
2007-10-01_09:45:34:721 3082967984 INFO     [TimeoutThread  :  38]: TimeoutThread started
2007-10-01_09:45:34:721 3074575280 INFO     [IPPool         :  73]: IPPool started
2007-10-01_09:45:34:721 3083069120 DEBUG    [framewor...pace:  87]: Starting Distributed Tuple Space...
2007-10-01_09:45:34:721 3066182576 INFO     [TimeoutThread  :  38]: TimeoutThread started
2007-10-01_09:45:34:721 3057789872 INFO     [transpor...read:  44]: IPQInThread started...
2007-10-01_09:45:34:722 3049397168 INFO     [transpor...read:  53]: IPQOutThread started...
2007-10-01_09:45:34:722 3024219056 DEBUG    [server/A...read:  29]: AHSPResponseThread started...
2007-10-01_09:45:34:722 3032611760 INFO     [server/A...rUDP:  46]: AHSPListenerUDP started on host 127.0.0.1 with port 7777
2007-10-01_09:45:34:722 3041004464 INFO     [messaged...read:  66]: TransportHandlerThread started
2007-10-01_09:45:38:165 3032611760 INFO     [server/A...rUDP:  52]: packet received on UDP Unicast, sourceAddress=127.0.0.1, sourcePort=33372
2007-10-01_09:45:38:166 3032611760 INFO     [ConnEntry      :  56]: ConnEntry::setTimer, 5, 0, sip:alice@voicehoc.ch
2007-10-01_09:45:38:166 3032611760 INFO     [ConnectionPool :  44]: Putting entry for identifier sip:alice@voicehoc.ch, old size = 0, new size=1
2007-10-01_09:45:38:166 3041004464 DEBUG    [messaged...read: 131]: run: There is some work in the request queue
2007-10-01_09:45:38:166 3041004464 DEBUG    [messaged...read: 283]: processNewPublish: called with key=sip:alice@voicehoc.ch, lifetime=0
2007-10-01_09:45:38:166 3041004464 INFO     [messaged...tore: 246]: lookup: no tuple found:
2007-10-01_09:45:38:166 3041004464 INFO     [messaged...read: 490]: updateTimer: lifetime=0, maxtimeout=900, publishtimeout=1500
2007-10-01_09:45:38:167 3041004464 DEBUG    [messaged...uple: 158]: register tuple timer for: 2007-10-01_10:10:38:976236, 1500, id: 8088cc8
2007-10-01_09:45:38:167 3041004464 DEBUG    [messaged...tore: 130]: store: Pushing tuple into store
2007-10-01_09:45:38:167 3041004464 DEBUG    [messaged...tore: 155]: store: ==
2007-10-01_09:45:38:167 3041004464 DEBUG    [messaged...tore: 155]: store: ||Key: sip:alice@voicehoc.ch
2007-10-01_09:45:38:167 3041004464 DEBUG    [messaged...tore: 155]: store: ||Flags: 00000001
2007-10-01_09:45:38:167 3041004464 DEBUG    [messaged...tore: 155]: store: ||Value: 10.4.0.1:5060
2007-10-01_09:45:38:167 3041004464 DEBUG    [messaged...tore: 155]: store: ||Lifetime: 1500
2007-10-01_09:45:38:167 3041004464 DEBUG    [messaged...tore: 155]: store: ==
2007-10-01_09:45:38:167 3041004464 DEBUG    [messaged...tore: 156]: New Size of Store: 1
2007-10-01_09:45:38:167 3041004464 DEBUG    [Strategy...edge:  64]: processNewPublish: Serving Reactive Routing Protocol, therefor no action is taken
2007-10-01_09:45:38:167 3024219056 DEBUG    [server/A...read:  44]: Response requested for [sip:alice@voicehoc.ch,00000001,10.4.0.1:5060,0]
2007-10-01_09:45:38:167 3024219056 INFO     [ConnectionPool :  34]: Retrieving entry for identifier sip:alice@voicehoc.ch, old size = 1, new size=0
2007-10-01_09:45:38:168 3024219056 INFO     [server/A...read:  57]: Response sent to 1 connection(s)
```

Figure 4: The MANET SLP process after the SIPHoc proxy has advertised its contact address

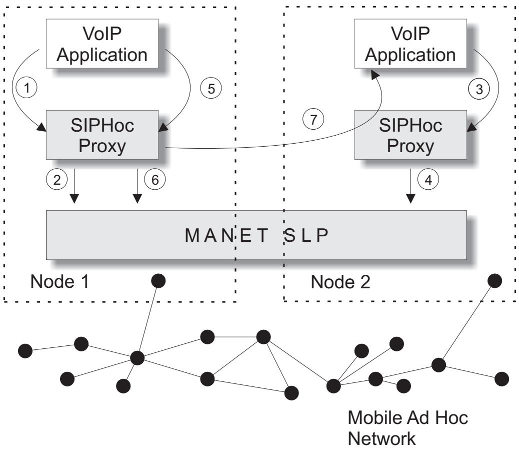


Figure 3: How a call between two user in an ad hoc network is established

After the two parties of the potential phone conversation have registered with their local SIP proxies, calls between users in the MANET become possible. Again, SIP call setup requests (SIP INVITE message) from the VoIP application will be routed through the local SIPHoc proxy (step 5). The SIPHoc proxy, in order to determine the target address where to send the message to, consults the MANET SLP service (step 6). Figure 5 shows how MANET SLP retrieves the endpoint address information for a given user (Bob), using routing message piggybacking. The Figure displays a snapshot of a linux packet analyzer tool [6]. The picture is taken at the moment the tool has captured the routing packet augmented with SIP address information (currently, our system supports two routing protocols, AODV [15] and OLSR [9]). Now that MANET SLP knows about the SIP endpoint address for the requested user (Bob), it is able forward the request accordingly (step 7). Once the call setup request arrives the SIPHoc proxy running on the node with the target VoIP application, the request is simply forwarded (step 8), which makes the VoIP application ring.

3.2 Phone calls to/from the Internet

SIPHoc allows setting up SIP sessions to and from the Internet by using MANET nodes connected to the Internet as gateways [17]. From a VoIP perspective this means that users can use their official SIP phone number (SIP address) transparently for phone calls within the MANET and for calls to the Internet as soon as one node in the MANET is connected to the Internet. Should the MANET be temporarily connected to the Internet, also VoIP calls from the Internet to user in the MANET become possible. We have tested this feature with three different SIP providers, *siphoc.ch*, *netvoip.ch* and *polyphone.ethz.ch*. Typically, SIP providers have their SIP proxy running on the domain they assign the SIP addresses from. If that is the case (as for *siphoc.ch* and *netvoip.ch*), one can make phone calls to and from the Internet without a problem. However, a problem occurs if the SIP provider requires a special outbound proxy to be set in the VoIP configuration (as for *polyphone.ethz.ch*). Since the outbound-proxy field is overwritten to point to the localhost, there is no way for the SIPHoc proxy to deduce the correct next hop SIP address. This is an open issue which we plan to address in the near future.

4. DEPLOYMENT

Wireless ad hoc VoIP is a pervasive application. One could imagine to use such an application on various different devices ranging from laptops, to handheld organisers, up to cheap mobile phones. Currently, we have two different deployments: Laptops (Debian, i386) and iPAQ handhelds (Familiar Linux, ARM/XSCALE) (see Figure 6). Having

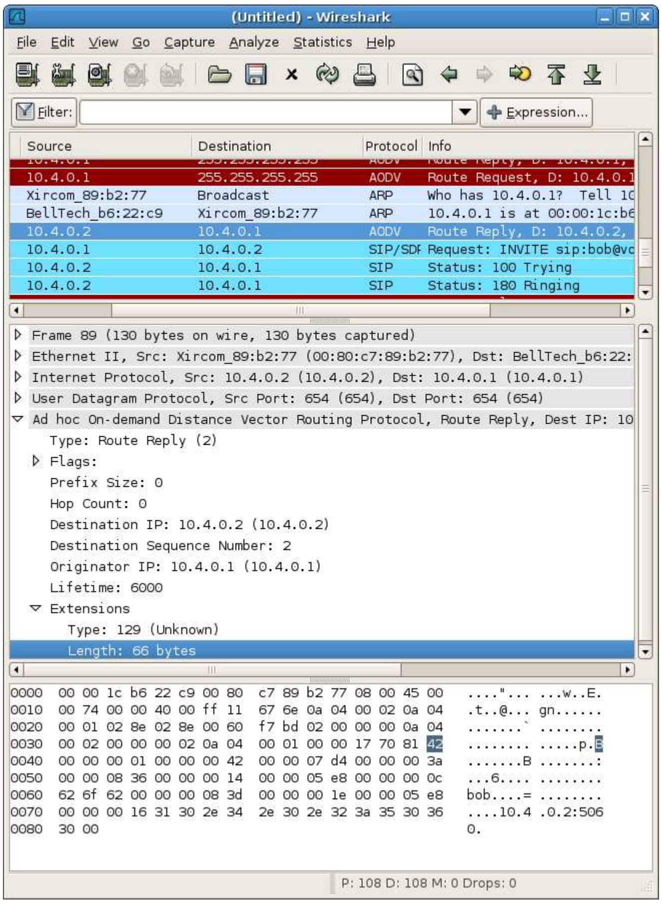


Figure 5: Snapshot of a packet analyzer showing an AODV route reply with encapsulated SIP contact information.

wireless ad hoc VoIP on iPAQ handhelds emphasizes the pervasiveness of the application. Handheld organisers have become a standard equipment in many people's daily life. They are lightweight and easy to carry, and often they are equipped with WiFi, microphone and loudspeakers. Turning such devices into a VoIP telephone, that runs without the need of a pre-existing infrastructure, may be particularly useful in many of the situations mentioned at the beginning of this paper. As a first prototype, we have built a deployment for the iPAQ/h5000 under Familiar Linux. For this reason, we have implemented a C version of the SIPHoc proxy, rather using the old Java implementation described in [17]. By doing so, we were able to come up with a system that has a footprint of 1.2M. The system includes four services (SIPHoc proxy, Gateway Provider, Connection Provider and MANET SLP) and about 20 shared libraries used by these services. This fits well into the flash memory of the iPAQ, which is 32M (from which the operating system consumes 25M). Additionally, the VoIP application consumes 1M. While on the laptops, various VoIP applications can be used (Kphone, Twinkle, Linphone), there is not such a big choice on the iPAQ. We have used Minisip [2], which is a small SIP based VoIP application which uses the GTK toolkit for the graphical user interface.

Figure 6: Turning iPAQ handhelds into VoIP phones

Currently, we run our system on a testbed of about 10 laptops and a bunch of handhelds. Some of the devices are separated by firewalls to enforce multihop communication. As a next step, we plan to explore the scalability of the system as the number of nodes grows.

5. RELATED WORK

Although VoIP is a promising application for MANETs – apart from [17] – we are not aware of any complete, standard compliant implementation of an infrastructure supporting VoIP in such networks. The swedish company terranet has announced a phone which will be able to work in an ad hoc environment [4]. The main challenge to provide VoIP in MANETs is to have a mechanism for SIP session establishment that works without centralized components. One of the earliest attempts to adapt SIP to MANETs is based on a pro-active mapping of all SIP clients in the MANETs using a HELLO method [13]. This leads to inefficient utilization of resources if the mappings remain unused, and introduces incompatibilities with the registration procedure of SIP. An alternative approach to adapt SIP to MANETs is to enforce a hierarchical topology with special gateways [10]. Such a solution requires a concrete topology and is not applicable to purely infrastructure-less networks.

There are also proposals for fully distributed SIP session initiation [12]. However, similar to [13], the basic SIP mechanism is extended by incorporating REGISTER broadcast messages which makes the approach inefficient and SIP incompatible. To overcome the incompatibility problem, it has been suggested to adopt SLP (Service Location Protocol) to discover the SIP bindings [12]. Unfortunately, SLP in its original form is very inefficient in MANETs [7] due to its heavy use of multicast messages. There are ways how to provide SIP-based VoIP to Internet connected MANETs [8]. This work, however, does not discuss how to support VoIP in MANETs not connected to the Internet. In addition, the work assumes a fixed topology with one node acting as gateway.

6. CONCLUSION

In this paper, we have described a solution for wireless ad hoc VoIP. The system we propose is built on top of SIPHoc, a SIP middleware platform for ad hoc networks. The paper describes how traditional VoIP applications can be run in ad hoc networks without any centralized SIP server. One aspect of our system – which is also highlighted in the paper – is that SIP contact information is propagated in the network using routing layer piggybacking. The system we propose features a small footprint and can easily be deployed on various devices. In the paper, we discuss the deployment on laptops and iPAQ handhelds. As part of future research, we plan to extend our deployment to a larger scale. We think that wireless ad hoc VoIP could be used to both improve communication in regions where network infrastructure is not available, and to provide free voice communication for instance within a university campus.

Acknowledgements

The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

7. REFERENCES

[1] *OpenSource SIP-based VoIP.* http://www.wirlab.net/kphone.

[2] *A SIP-based VoIP client for linux.* http://www.minisip.org.

[3] *Telephony on linux.* http://www.linphone.org/.

[4] *Free local wireless calls, free texting, and long distance VoIP calls.* http://www.terranet.se/, 2007.

[5] *Twinkle: SIP softphone for Linux.* http://www.twinklephone.com/, 2007.

[6] *Wireshark: The World's Most Popular Network Protocol Analyzer.* http://www.wireshark.org, 2007.

[7] H.-G. S. amd Ki-Hyung Kim, W.-D. Jung, and J.-S. Park. Performance of service location protocols in manet based on reactive routing protocols. In *Proceedings of 4th International Conference on Networking (ICN'05)*, 2005.

[8] J.-J. Chen, Y.-L. Cheng, Y.-C. Tseng, and Q. Wu. A push-based voip service for an internet-enabled mobile ad hoc network. In *Proceedings of 3rd IEEE VTS Asia Pacific Wireless Comm. Symposium (APWCS'06)*, 2006.

[9] T. Clausen, P. J. (editors), C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L.Viennot. Optimized link state routing protocol (olsr). RFC 3626, October 2003. Network Working Group.

[10] A. Dutta, R. Jain, K. Wong, J. Burns, K. Young, and H. Schulzrinne. Multilayered mobility management for survivable network. In *Proceedings of MILCOM*, 2001.

[11] J. P. Hubaux, T. Gross, J. Y. L. Boudec, and M. Vetterli. Towards self-organized mobile ad hoc networks: the Terminodes project. volume 31, pages 118–124, Jan. 2001.

[12] S. Leggio, J. Manner, A. Hulkkonen, and K. Raatikainen. Session initiation protocol deployment in ad-hoc networks: a decentralized approach. In *2nd International Workshop on Wireless Ad-hoc Networks (IWWAN)*, 2005.

[13] M. O'Doherty. Pico sip. Internet Draft, February 2001.

[14] J. p. Hubaux, J. y. Le Boudec, S. Giordano, and M. Hamdi. The terminode project: Towards mobile ad-hoc WANs. In *Proceedings of IEEE International Workshop on Mobile Multimedia Communications (MoMuC '99).*

[15] C. Perkins and E. Royer. Ad hoc on-demand distance vector routing. *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, Feb 1999.

[16] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), June 2002. Updated by RFCs 3265, 3853.

[17] P. Stuedi and G. Alonso. Siphoc: Efficient sip middleware for ad hoc networks. In *Proceedings of the 8th ACM/IFIP/USENIX International Middleware Conference*, 2007.