M. Stiemerling and M. Brunner

A Peer-to-Peer SIP System based on Service-Aware Transport Overlays

Martin Stiemerling is senior researcher at the Network Laboratories of NEC Europe Ltd. in Heidelberg, Germany. Currently he is also a Ph.D. student with the Computer Networks Group at the University of Göttingen. He received his Dipl. Ing. (FH) in Electrical Engineering from the University of Applied Sciences (FH Köln) in 2000. His areas of interest are Internet architecture, Internet signalling protocols, network management, and overlay/peer-to-peer systems. Additionally, he is active in several areas of the IETF and IRTF as working document editor, working group chair, and co-author of several RFCs. He is currently TPC member of IEEE IPOM 2007 workshop.

Dr. *Marcus Brunner* is chief researcher at the Network Laboratories of NEC Europe Ltd. in Heidelberg, Germany. He received his Ph.D. from the Swiss Federal Institute of Technology (ETH Zurich), while working in the Computer Engineering and Networks Laboratory (TIK) of the Electrical Engineering Department. He got his M.S. in Computer Science from ETH Zurich in 1994. Aside from the involvement in different national and international projects, his primary research interests include network architectures, programmability in networks, network and service management. He is actively participating in different standards organizations such as the Internet Engineering Task Force (IETF), the Distributed Management Task Force (DMTF), ETSI TISPAN, and 3GPP. Additionally, he is an active member of the network management research community with being in the Organization and Technical Program Committees of major network management conferences such as NOMS, IM, DSOM, Policy Workshop, IPOM. Currently, he is TPC co-chair of NOMS '08. He is in the editorial board of the IEEE Transactions on Network and Service Management and the Journal of Network and Systems Management. Also in the networking area he is in the TPCs of IEEE Globecom, ICC, LCN. He is currently IEEE Globecom '08 symposium co-chair on Next Generation Networking and Internet Protocols.

ABSTRACT

The Session Initiation Protocol (SIP) traditionally establishes and manages its sessions with centralized servers, which have been challenged by issues with TCP/IP networks right from its beginning, for instance, issues with NAT traversal or network congestions. On the other hand, there are peer-to-peer and overlay techniques that potentially can make SIP working better under various network conditions. Overlay networks can provide a good abstraction from the real network, thus hiding the presence of NATs. Some proposals on peer-to-peer SIP have been recently presented. While these proposals attempted to solve a subset of challenges faced by traditional SIP, such as removing the centralized SIP entities, there is to the best of our knowledge no single solution offering a comprehensive view of the peer-to-peer SIP architecture and operation. This paper presents a novel approach for a peer-to-peer SIP system, using overlay techniques for signalling and media transport in the Internet. This system is based on the Ambient Networks Service-Aware Transport Overlay (SATO) system. The proposed system replaces the traditional SIP proxy/registrar function with a distributed lookup mechanism, adding overlay functionality to the SIP signalling and to the RTP traffic. Moreover, different from previous proposals, our approach deliberately places media/packet relays into the SIP/RTP paths, which allows an efficient session management and media communication.

1 INTRODUCTION AND MOTIVATION

Voice over IP (VoIP) services based on the Session Initiation Protocol (SIP) are well known today. Deployment of these SIP-services is unfortunately either limited to rather closed network environments or assumes that the network itself and the network elements (routers, firewalls, Network Address Translators (NATs)) are operating just fine. SIP-based telephony experiences many problems, as soon as the network is not operating within certain limits. Typically, SIP is experiencing severe problems with badly implemented or wrongly configured NATs or firewalls. The protocol itself has been extended in many respects to cope with this challenging environment, for example, the introduction of the ICE protocol [3] to assist in NAT traversal for media transport (the term media refers to any media transfer signalled with SIP, e.g., voice and video over RTP.) The attempt of all extensions and add-ons has helped in fixing some SIP issues but also increased the complexity with respect to implementation and operation.

A second issue is that terminals implementing SIP are not able to react to network changes, other than shutting down the media sessions completely. These changes include decreasing available bandwidth in the end-to-end network path and terminal unreachability due to routing issues, also known as host behind the NAT. SIP is apparently not well suited for operating in non-optimal network environments or conditions, which are standard situations in today's Internet and in the future.

Peer-to-peer and overlay techniques offer a means to improve SIP under almost all network condition, as for instance, overlay

networks can provide a good abstraction from the real network as they hide the presence of NATs, etc, plus they add the ability to react to changes in the network environment without impacting the SIP level. There are already some proposals on peer-to-peer SIP out, but typically they just aim at solving a subset of SIP's challenges, such as removing the centralized SIP entities but far from providing a complete solution. In this paper, we advocate a novel while practical approach for a peer-to-peer SIP system, using overlay techniques for signalling and media transport in the Internet. This system is based on the Ambient Networks Service-Aware Transport Overlay (SATO) system. The proposed system replaces the traditional SIP proxy/registrar function with a distributed lookup mechanism, adding overlay functionality to the SIP signalling and to the RTP traffic, takes care about media/packet relay insertion into the SIP/RTP paths. Still with all proposed add-ons, it is just required to change some small parts of SIP within the overlay system and to not change any SIP outside of the overlay system itself.

The related work in the area of peer-to-peer SIP (P2PSIP) is described in Section 2 while Section 3 briefly introduces the Service-Aware Transport (SATO) system to build the foundation for the proposed system. The peer-to-peer SIP architecture and operations is detailed in Section 4, where Section 5 is describing our P2PSIP implementation. The paper concludes with an outlook to future work in Section 6.

2 RELATED WORK

The SIP standard [1] itself actually supports peer-to-peer operations, as it is not mandatory to use any centralized entity. This peer-to-peer approach works as long as each calling partner knows the real username and IP address or host name of the party to be called. Effectively this does not work in the real world, as many hosts do not have fixed IP addresses or host name, etc, thus requiring a SIP proxy or at least a registrar to be deployed in the network. Furthermore, there are some workarounds to make SIP more robust to the Internet environment, for instance, ICE [3], but they do not solve the general issue of SIP being bound to a specific transport protocol, such as UDP, and the resulting implications. These implications includes all issue of UDP (and partially also for TCP) with respect to security (i.e., the transport is unprotected to eavesdropping, changing of content, etc, NAT/firewall traversal (still requires a lot of technical effort to cope with them), etc.

The term peer-to-peer SIP is used quite differently by many people. Among a variety of definitions so far, one meaning is that the protocol operates between peers to establish a direct connection (as the aforementioned SIP usage), another meaning is using the SIP protocol itself as a control protocol for a peer-to-peer system, whereas the actual usage of the system is not necessarily SIP but anything else, for example, file sharing. The SOSIMPLE proposal [1] uses SIP as the control protocol for a peer-to-peer system but not primarily, for session setup. The IETF P2PSIP working group [4] is also working on a P2PSIP solution, which is targeting on evolving SIP towards more peer-to-peer, i.e., removing the centralized proxies in favour of a DHT approach, but also considering using SIP to control the DHT and the overlay (SIP as control protocol). It is worth mentioning the large amount of Internet drafts submitted to the P2PSIP working group. The later approaches are requiring changes to the SIP protocol, adding another plethora of options. Our P2PSIP approach is using SIP as session setup protocol within the overlays system, but not to control the overlay.

There are also other peer-to-peer Voice over IP (VoIP) systems, such as Skype [6] or even some variations of SIP, such as Gizmo [5]. Skype can be seen as the ultimate driver behind some peer-to-peer SIP developments, as it has shown how good VoIP can work and how a successfully deployed VoIP peer-to-peer system looks like [7].

3 SATO SYSTEM OVERVIEW

The adaptive peer-to-peer system presented in this paper, is based on initial development carried out in the framework of the EU FP6 IST Ambient Networks research project [8].

3.1 Basic concepts of SATO

The purpose of SATO as part of the Ambient Networks architecture [13] is to provide a flexible and customisable transport service to the application layer by using overlay networks on top of any type of transport or network layer connectivity.

– *Service-aware Transport Overlay*: Service-aware Adaptive Transport Overlays (SATOs) enable the flexible configuration of virtual networks consisting of SATO Overlay Nodes (SON) on top of the underlying basic network connectivity. A SATO implements an overlay instance per service, i.e., for each requested service a new SATO is created.
– *Dynamic Inclusion of Network Elements*: The SATO concept allows the transparent inclusion of network-side data processing elements (SATO-Ports) in the end-to-end transport path (between a client and a server or peer-to-peer). These SATO-Ports can provide value-added functions such as overlay routing, smart caching, media adaptation, rate adaptation, synchronisation, filtering, metering, congestion control, etc.
– *On Demand Overlay Set Up and Tear Down*: SATOs are established on demand, based on particular requirements, and terminated when the service is not requested anymore.
– *Overlay Adaptation:* SATOs can adapt as a consequence of a SON joining or leaving the virtual network or due to changes in the network environment or conditions. This introduces the notion of "adaptive overlays" that dynamically re-configure in order to optimise the service delivery.

This SATO is self-contained and does logically not interfere with any other existing or future SATOs. In reality SATOs also compete for resources on nodes and in the network, i.e., no new SATO can be created if all bandwidth is already consumed.

3.2 Elements of SATO

The core of the SATO system are the SATO Overlay Nodes (SON) that participate in the SATO system by providing their network and computing resources to other nodes. A SON can be any kind of Internet device, starting from a mobile phone to fixed PC in an office. A SON can be part of no, one or multiple different SATO instances, providing different services to different applications.

Within a SON there are different building blocks, as shown in Fig. 1. The *Overlay Management* (OM) is in charge of operating the SATO system, i.e., taking request from applications for setup, maintaining existing SATOs and tearing down of, and is the central contact point on each SON. Applications can re-
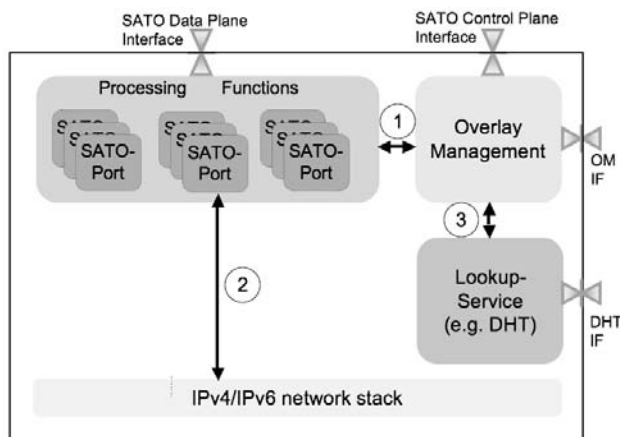
Fig. 1   *SATO Overlay Node Design*

quest a SATO service through the SATO control plane interface (see also Section 3.1). The OM interface (OM IF) is used by the OM to communicate with other OMs on other nodes. The OM of the SATO system is aware of the service requested by the application and about the network conditions in which the service is operated (the so-called service logic).

Each SON is part of the SATO *lookup service*, which is typically a distributed hash table (DHT), but not limited to, e.g., a centralised lookup service approach could also be chosen for a deployment of SATO. The *lookup service* and the OM communicate with each other using the interface 3. This *lookup service* is used to store information required for operating the SATO system and information required for running the applications. The OM is primarily using the *lookup service* to store information about available processing functions, such as packet relays, and their location in the network, i.e., the IP addresses.

The *processing functions* block that hosts the SATO-Ports provides the real power of the SATO system. A SATO-Port provides an arbitrary processing functionality, for example, MPEG transcoders, voice-codec transcoders, or peer-to-peer SIP proxies. SATO-Ports have one external interface to the application, which is the SATO data plane interface in Fig. 1. The interface 2, called SATO network interface, is connecting the SATO-Ports to the network stack of the SON and thus to other SATO-Ports. The control of the SATO-Port by the OM is provided by interface 1. The SATO system is able to include *SATO-Ports* on the fly into existing SATO instances, or when setting up a new SATO. Whenever a *processing function* is needed (this is decided by the OM when constructing or adapting the overlay) a search in the lookup service is performed for the type of SATO-Port required.

4   THE PEER-TO-PEER SIP ARCHITECTURE

The proposed peer-to-peer SIP system re-uses the SATO network and node architecture and extends it by adding two new types of SATO-Ports and removing some of the control functionality of the OM. The general SATO architecture described in Section 3.2 assumes that the OM is aware of the service provided by a SATO. This has the drawback that the OM on each deployed SATO node must be upgraded for each new introduced service. Furthermore, the OM would be in charge of detecting how a service is currently executed in the SATO, i.e., if all SATO-Ports involved are operating correctly and if the net-

work has enough resources available along the data path. This knowledge is at best obtainable and usable for judgement by a SATO-Port itself. The P2PSIP system removes all service-specific parts of the OM and places them in the respective SATO-Ports. Second, all control over a particular service is bound to a SATO-Port and not anymore to the OM. Fig. 2 shows the P2PSIP SATO node with the two new SATO-Ports, the connected SIP user agents, and the different control and data flows. These new SATO-Ports are the peer-to-peer SIP port (P2PSIP-Port) and the RTP proxy/relay port (RTP-Port).
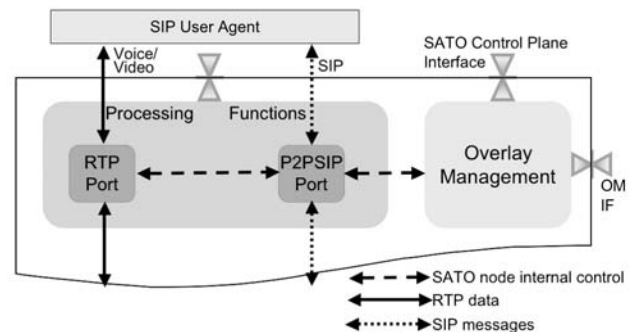

Fig. 2   *P2PSIP SATO Node*

With these extensions, standard SIP can be enhanced to cope with a variety of networking conditions, solving the major issues, such as NAT traversal and resilience to network changes. SIP and the transport of media are decoupled from the real network by using the overlay capabilities of the SATO system. The P2PSIP system creates a new SATO per SIP call, with the requirement for reliable transport, and all SIP signalling messages for this call are forwarded through this SATO. P2PSIP is also creating a new SATO for each media exchange, e.g., one SATO for voice and one for video. Whenever a call is established between two user agents, the P2PSIP-port at the calling side will request the setup of a new SATO from the OM for the transport of the media. Typically for voice and video the service requirement is low delay and jitter, but not necessarily for reliable transport. Both SATOs, i.e., the SIP signalling and the media transport, are controlled and monitored independently, giving the chance to later introduce a self-adaptation functionality to the whole peer-to-peer SIP system. Until now, the changes of the transport of SIP and the media in the overlay are neither requiring any change to the protocols nor any change in the user agents used.

This way, P2PSIP will be not only able to establish direct communication links between calling parties whenever possible, but also offering the chance to include network-side support functions, i.e., the SATO-Ports, such as packet relays, media transcoding, etc. Whenever a SATO detects a change of network conditions, such as congestion or node failures, either by a SATO-Port or by the OM, the SATO-Port in charge can trigger the adaptation of the existing SATO. The adaptation can result in removing or adding SATO-Ports and a reconfiguration of the overlay topology.

The traditional SIP registrar function is replaced by a DHT approach. The DHT is distributing the SIP contact information, i.e., the SIP URI to IP address/host name mapping over a larger set of SATO nodes. It should be noted, that the current approach will rely on existing DHT concepts and protocols. Using a DHT as distributed SIP registrar requires a change to the SIP
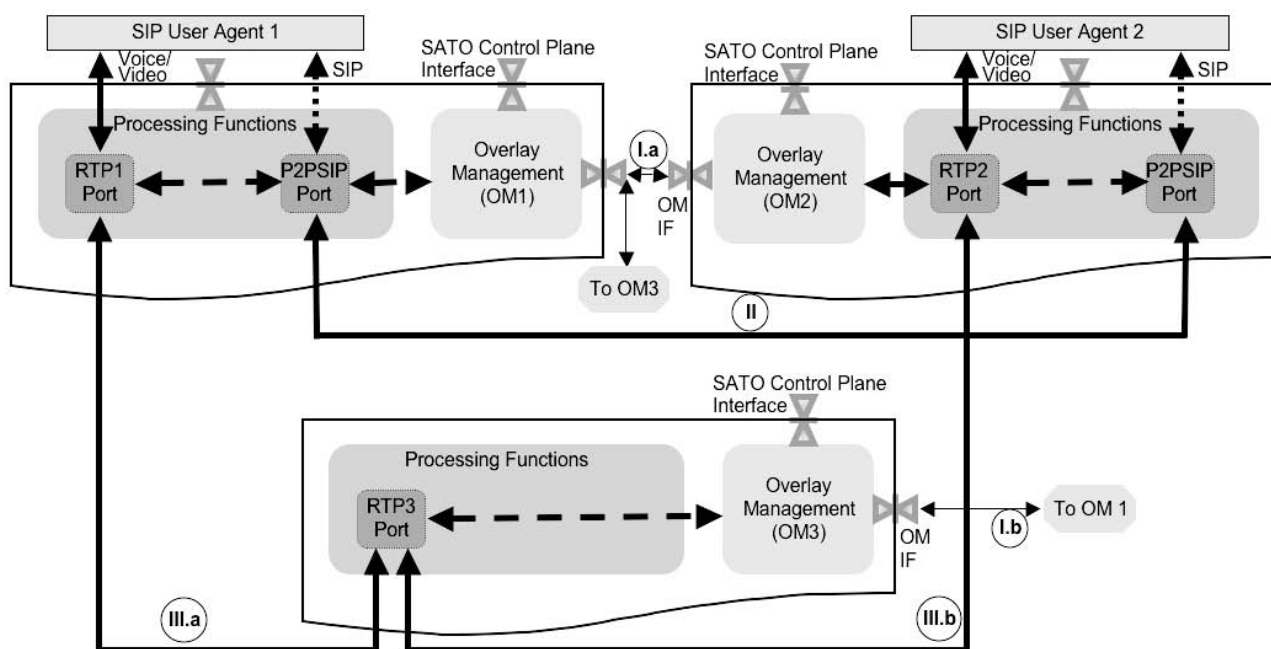
Fig. 3   P2PSIP call setup with all flows

message routing within the peer-to-peer SIP (P2PSIP) system, as this concept does not follow the standard SIP routing approach of using the DNS name system and SIP proxies (see Section 5.1).

Fig. 3 shows the schematic connection between the different overlay nodes and the SATO-Ports for a three party overlay. The overlay nodes, directly connected to the respective user agents 1 and 2, are nodes that are hosting SIP user agents. The SIP user agents talk directly to the overlay nodes. The OM of the calling node (here OM1) then connects to all involved overlay nodes via the OM IF (I.a and I.b). The OM IF is used to instruct other overlay nodes to start or stop SATO-Ports and to obtain status information about the nodes and also possibly about the environment of a node, for example, the actual network throughput on a given link. The arrow depicted with II is the SIP signalling SATO und the arrows depicted with III.a and III.b are showing the RTP SATO. The RTP SATO uses the RTP3 port for processing media streams on the fly in the network, e.g., transcoding or just relaying.

5   PEER-TO-PEER SIP IMPLEMENTATION

This section describes our implementation of the peer-to-peer SIP architecture as proposed in Section 4. The SATO P2PSIP system implementation is built in such a way that traditional SIP user agents (UAs) can participate without any modification. SIP user agents use their standard way of transporting SIP signalling and RTP media, for instance, UDP for both. However, UAs need to point for their SIP registrar and outbound proxy [9] configuration to their overlay peer (see Fig. 2 where the UA is using the P2PSIP-Port as registrar and outbound proxy). This ensures that all outgoing SIP signalling messages will be forwarded to the respective P2PSIP overlay peer.

The two new two new required SATO-Ports, one for the SIP signalling and one for the RTP media handling, are described in the next sections.

5.1   P2PSIP SATO-Ports

5.1.1   Peer-to-Peer SIP Port

The Peer-to-Peer SIP Port (P2PSIP-Port) is a standard-based SIP proxy with registrar functionality. All the message parsing and processing is kept as defined by the SIP standard [1]. In standard SIP the user lookup and message routing is based on the user and domain name of the SIP URI. The P2PSIP concept extends this to a peer-to-peer lookup service by allowing user lookup without SIP Registrars/Redirect Servers, i.e., by using the *lookup service* provided by the SATO architecture. The P2PSIP port is acting as a registrar and instructs the OM to store the SIP contact information, i.e., user name and the IP address of the SATO-Port, in the *lookup service*. When a user is called, the P2PSIP port queries the lookup service for the user and if the user is registered the destination P2PSIP port is contacted (the destination P2PSIP port is the one where the user has been registered).

The P2PSIP port is providing a standard SIP interface to the user agents, which allows reusing existing SIP user agent applications. The SATO-Ports itself are using the SATO network interface to exchange data with each other.

5.1.2   RTP Port

The RTP port can take different roles, depending on the required functionality, i.e., either it is solely a packet relay or a voice-codec transcoder. It also has two interfaces, one RTP/UDP-based media interface to the user agent and the SATO network interface.

5.2   Calling Peer-to-Peer

The scenario described here assumes the participation of three overlay nodes, i.e., the overlay node serving the caller (user

agent 1), the overlay node serving the callee (user agent 2), and an overlay node providing a packet relay (same configuration as Fig. 3). Both user agents are registered at their respective P2PSIP port and the user agent contact information is stored in the *lookup service*. The packet relay is registered in the *lookup service* as RTP port beforehand.

Placing a call from UA1 to UA2 (see Fig. 4) follows the possible call flows in [1] from the UA to the P2PSIP port. The P2PSIP port at the overlay peer receiving the INVITE call request uses the SIP URI for the DHT lookup (not shown in Fig. 4). If the user is registered in the DHT, i.e., registered at some overlay peer, the IP address of the serving overlay peer is obtained. This IP address is transferred from the P2PSIP port 1 to the local overlay management OM1. The OM1 sets up a new SATO for the SIP signalling. OM1 uses the OM interface (OM IF in Fig. 2 and Fig. 3) to contact OM2 on the destination overlay node where UA2 is registered. OM1 and OM2 negotiate the possible transport connection between OM1 and the remote P2PSIP port 2 (shown as "Connect and negotiate …"). This negotiation takes the service requirements, such as security, reliable transport, etc, into account. While the transport connection for the SIP signalling is negotiated, it is detected that the network path between OM1 and OM2 is slightly congested (by observing the network behaviour between both). The overlay management at OM1 decides to later route the voice media through a different part of the network by dynamically adding RTP port 3. The RTP port 3 is found by querying the lookup service for existing RTP ports.

The calling P2PSIP port 1 is notified, once the transport connection between both sides is set-up. P2PSIP port 1 adds a *record-route* header to the SIP message and forwards it to P2PSIP port 2 via this transport connection, which is TCP/TLS in the example. Subsequent SIP messages are transferred via this transport connection. In parallel P2PSIP port 1 requests the setup of RTP port 1 and port 3, while P2PSIP port 2 is requesting locally the setup of RTP port 3. The P2PSIP port 2 forwards the INVITE message to UA2. When UA2 is accepting a call, a 200 Ok is sent back to P2PSIP port 2. OM2 instructs OM1 ("Connect RTP") to create a new SATO, once the 200 Ok is received. OM1 creates this new SATO for the media by setting up transport connections between RTP1, RTP3, and RTP2 (in this order).
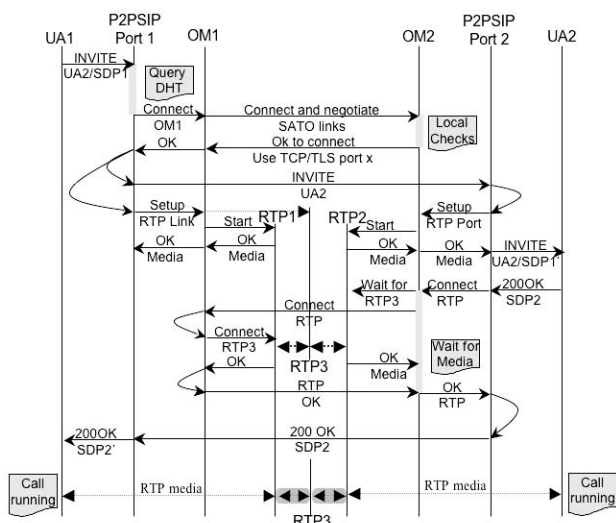

Fig. 4   SIP and SATO Flow Diagram for Call Setup

In the process of the call signalling, both P2PSIP ports will replace the media IP address(es) and port(s) by an IP address and port of their local RTP port in the SDP part of the SIP message. The modified SDP1' and SDP2' ensure that media is sent to the local RTP port and not directly to the other end. The RTP port can then transport the media via the overlay, ensuring the requested service. Finally, the call can proceed and when the call ends all transport connections between P2PSIP port 1/P2PSIP port 2 and RTP port 1/RTP port 2 are torn down.

5.3 Implementation Status and Verification Details

The P2PSIP system has been implemented on FreeBSD in C to proof the semantically correctness of the system specification and the working peer-to-peer SIP approach. For implementing the P2PSIP ports we are using an off the shelf SIP stack (the libosip library [15]), with modified peer-to-peer SIP message routing. The RTP port is currently implemented as a packet relay without media processing, enabling easy NAT traversal of the media.

For the SIP user agents we are using well-known SIP phones, such as the SNOM 360 and Eyebeam SIP phones, which provide a good SIP standard compliant implementation. The Bamboo DHT implementation [10] has been used for storing the information about the registered SATO-Ports and SIP users. The overlay manager uses gSOAP [16] to communicate with other overlay managers on other peer nodes. The overlay peer internal communication is using Unix sockets. The design decisions for SOAP, Unix sockets, and Bamboo has been made in favour of a rapid prototyping, i.e., to evaluate the proposed system quite fast.

The system was tested to work properly with off the shelf SIP phones in two combinations. First, the SIP softphones (PC-based) were integrated on the same computer as the P2PSIP system and second, the SIP phones were connected via Ethernet to the P2PSIP system. The P2PSIP system and the SIP phones worked correctly without any problem.

Furthermore, the automatic inclusion of packet relays has been tested, by placing two P2PSIP overlay nodes behind symmetric NATs. The overlay peers used for both SATOs, SIP signalling and media transport, a packet relay that was not located behind any NAT. This enables both nodes to communicate with each other, which has not been possible without the P2PSIP system.

6 CONCLUSION AND FUTURE WORK

We have presented a novel approach for a peer-to-peer SIP system based on the Service-Aware Transport Overlay concept. The network abstraction provided by the P2PSIP system takes away the need from SIP and the media transport to take care of the network, the possible impact of network elements and a changing network environment. Furthermore, the P2PSIP system replaces the traditional centralized SIP registrar with a DHT removing the centralized point of failure. However, not all features of the P2PSIP system haven been implemented yet, as for instance, the possibility for self-adaptation of the P2PSIP system, e.g., the overlay system can autonomously decide to dynamically adapt the SATO, if the network path is getting congested by introducing a packet relay.

An interesting topic still to be explored is the disconnected operation of such peer-to-peer networks: what happens if a part of the peer-to-peer system is disconnected from the Internet, for example, as a result of a natural disaster. Typically, today's peer-to-peer networks will also collapse, as the connection to the DHT or other infrastructure parts is interrupted. A possible solution can be the network based search approach ([11] [12]).

A major advantage of the P2PSIP system is the usage of a DHT to store any type of information. However, using DHTs can cause a performance penalty if the lookup of other nodes or SIP users takes too long. Furthermore, the secure enrolment process of users to the DHT is missing and the security of the DHT and entries in the DHT is also a major concern. These issues are still to be investigated.

7 ACKNOWLEDGMENTS

8 REFERENCES

[1] J. Rosenberg, et al: "SIP: Session Initiation Protocol", RFC3261, June 2002.

[2] D. Bryan; B. Lowekamp; and C. Jennings: "SOSIMPLE: A Serverless, Standards-based, P2P SIP Communication System", Advanced Architectures and Algorithms for Internet DElivery and Applications (AAA-IDEA), June 2005.

[3] J. Rosenberg: "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", Internet draft (work in progress), draft-ietf-mmusic-ice-17, July 2007.

[4] "Peer-to-Peer Session Initiation Protocol ("P2PSIP)", website http://www.ietf.org/html.charters/p2psip-charter.html, September 2007.

[5] Gizmo, website http://gizmoproject.com/, September 2007.

[6] Skype, website http://www.skype.com, September 2007.

[7] Salman Baset and Henning Schulzrinne: "An Analysis of the Skype Peer-to-Peer Internet Telephony", IEEE Infocom 2006.

[8] "IST project 507134 Ambient Networks (AN)", Website http://www.ambient-networks.org, September 2007.

[9] C. Jennings and R. Mahy: "Managing Client Initiated Connections in the Session Initiation Protocol", Internet draft (work in progress), draft-ietf-sip-outbound-10, July 2007.

[10] BAMBOO DHT, website http://bamboo-dht.org/, September 2007.

[11] K. Lim; C. Adam; and R. Stadler: "Decentralizing Network Management", IEEE electronic Transactions on Network and Service Management (eTNSM), Vol 1(2), 2004.

[12] E. Asmare; S. Schmid; and M. Brunner: "Setup and Maintenance of Overlay Networks for Multimedia Services in Mobile Environments", 8th International Conference on Management of Multimedia Networks and Services, Barcelona, Spain., October 2005.

[13] Ambient Networks Deliverable: "System Design of SATO & ASI", website http://www.ambient-networks.org/Files/deliverables/D12-F.1_PU.pdf, September 2007.

[14] M. Kampmann; M. Vorwerk; M. Kleis; S. Schmid; S. Herborn; R. Agüero and Johnny Choque: "A Multimedia Delivery Framework for Ambient Networks", WWRF#15, Wireless World Research Forum Meeting 15, December 2005.

[15] Libosip, website: http://www.gnu.org/software/osip/, September 2007.

[16] gSOAP, website: http://www.cs.fsu.edu/~engelen/soap.html, September 2007.

Aufruf zur Einreichung von Beiträgen

Praxis der Informationsverarbeitung und Kommunikation (PIK)

Rubrik: Übersichtsartikel

Gesucht sind Übersichtsartikel und Erfahrungsberichte aus den Bereichen Rechnernetze, Verteilte Systeme und Informationssysteme. Besonders angesprochen sind Doktoranden und Wissenschaftler, welche die Vorarbeiten zu einem Arbeitsgebiet sorgfältig aufgearbeitet haben und das so erworbene Wissen mit der Leserschaft der PIK teilen möchten. Ebenso sind Mitarbeiter aus Rechenzentren angesprochen, die eine neue Technologie zur Einführung in den Experimental- oder Produktivbetrieb evaluiert haben. Die Artikel sollen dem Leser einen Einstieg in ein neues Gebiet ermöglichen und eine gute erste Übersicht bieten.

Allgemeine Informationen

Vorschläge für Einreichungen und Anfragen in der Rubrik sollten an Martin Mauve (mauve@cs.uni-duesseldorf.de) gerichtet werden. Artikel können sowohl in deutscher als auch in englischer Sprache verfasst werden. Ihr Umfang liegt in der Regel zwischen vier und sechs Druckseiten, dabei entspricht eine Druckseite 6000 Zeichen, einschliesslich Leerzeichen.