15th ICCRTS
"The Evolution of C2"

Title:
**Mobile SCIP:**
**Managing terminal mobility in heterogeneous networks with the SIP Handover Extension**

Paper ID: 045

Author:
Elin Sundby BOYSEN

Topics: Networks and Networking, Experimentation and Analysis, C2 Architectures and Technologies

Point of contact:
Elin Sundby Boysen

Affiliation:
Norwegian Defence Research Establishment (FFI)
University Graduate Center, Kjeller (UNIK)
University of Oslo

Address:
University Graduate Center Kjeller (UNIK)
PO.BOX 70
N-2027 Kjeller, Norway

Telephone:
(+47) 918 18 004 / (+47) 64 84 47 60

E-mail:
esb@unik.no

# Mobile SCIP: Managing terminal mobility in heterogeneous networks with the SIP Handover Extension

Elin Sundby Boysen
Norwegian Defence Research Establishment (FFI)
UNIK - University Graduate Center Kjeller
Kjeller, Norway
Email: esb@unik.no

## Abstract

*Advances in Network-centric Warfare require that speech and data must be shared among all tactical levels and across different network domains while the communication still is secured. SCIP (Secure Communications Interoperability Protocol) is an application-layer communication protocol designed to ensure secure end-to-end communication. When SCIP is implemented on mobile terminals operating in a heterogeneous environment, handover delays can have severe implications on the quality of service experienced by the end users. In this paper we discuss the implications of handover delays on a SCIP session. We propose the SIP Handover Extension as a means to avoid handover delays by providing seamless handover for SCIP sessions in heterogeneous networks. Implementation and testing show that a SIP user agent supporting the SIP Handover Extension can perform a seamless handover without interfering with the core functions of the SCIP operations.*

*Keywords: SIP; Mobility; Seamless handover; SCIP; IP*

## 1. Introduction

The main goal of Network-centric Warfare is to improve situational awareness by increased information sharing. To accomplish this, both speech and data must be shared among all tactical levels and across different network domains. This leads to a requirement to establish and maintain end-to-end secure communication even in dynamic environments. This requirement is shared with first responders from different departments and agencies, and diplomatic security officers that need uninterrupted secure communication even in mobile heterogeneous networks.

The Secure Communications Interoperability Protocol (SCIP) is a communication protocol designed to ensure secure end-to-end communication. As it is an application-layer protocol, it is independent of the underlying physical link. Until recently, SCIP was implemented only on circuit-switched networks (PSTN, ISDN).

The commercial Voice over IP (VoIP) technology has been evolving at a rapid pace due to factors like cost savings and ease of implementation and maintenance. This evolution has also influenced tactical networks that have followed the trends in the commercial networks: shifting from circuit switched towards packet switched networks and a significant increase in data traffic. With this, the need for an implementation standard for SCIP over IP has emerged.

Before a SCIP session can be established, an underlying data channel must be established between the end points. In IP networks, the Session Initiation Protocol (SIP) [1] is one of the protocols that can be used to find the end points and establish the underlying data channel.

As the need for secure communication extend out of the offices and into the urban environments, secure, *mobile* communication over IP networks must also be supported. Mobility for real-time communication over IP is still subject to much research. One of the main problems today is that the handover delay when switching from one network to another is too long. This is problematic for real-time communication that has stringent requirements to acceptable handover delays. When using IP networks to transport secure SCIP sessions, these handover delays can become even more problematic due to the Sync Management frames that are sent periodically during the SCIP session. If one of these is lost, the delay experienced by the user can be further increased. The proposed SIP Handover Extension will however be able to support

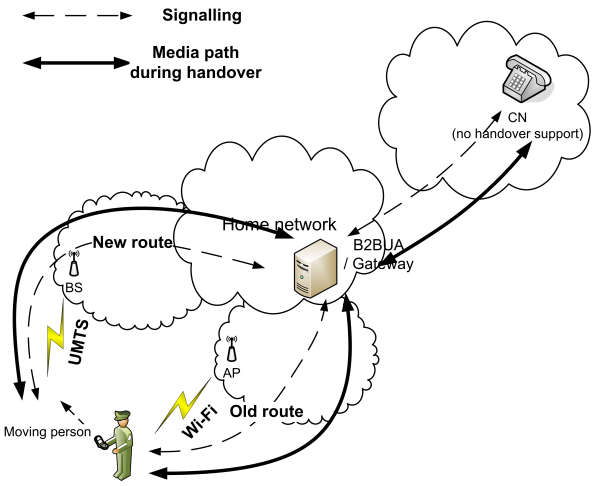


Figure 1. Scenario A: A person can move in-session from one access point to another

the seamless handover that is needed by SCIP-over-IP implementations for mobile terminals.

In the next sections we first present two scenarios where an end user or part of the core network is mobile. Then we give a brief introduction to SCIP, Mixed Excitation Linear Predictive (MELP) speech coding, and SIP before commenting on some of the known problems of handover in heterogeneous networks. A SCIP session using secure MELP encoding is analysed in light of the known handover problems. Finally, our SIP Handover Extension as suggested in [2] is presented and discussed in relation to SCIP.

## 2. Mobility in user scenarios

Different sections or units in a network can be mobile. Figures 1 and 2 show two scenarios where a handover from one network type to another, a so-called vertical handover, needs to be performed while the end users are engaged in an ongoing session. In both scenarios, packet loss or long delays during the handover will degrade the quality of service perceived by the end users.

In Scenario A in Figure 1 it is the end node that is mobile. This would be the case when a SCIP module is implemented on a mobile terminal that can connect to different network types such as WiFi and UMTS. In this scenario, a user is on the move while engaged in a SCIP session. When the terminal enters the coverage of a new network to which the user has access, it can choose to perform a handover to that new network, given that the pre-set requirements to initiate handover are met. For instance when moving out from WiFi coverage and switching to UMTS.

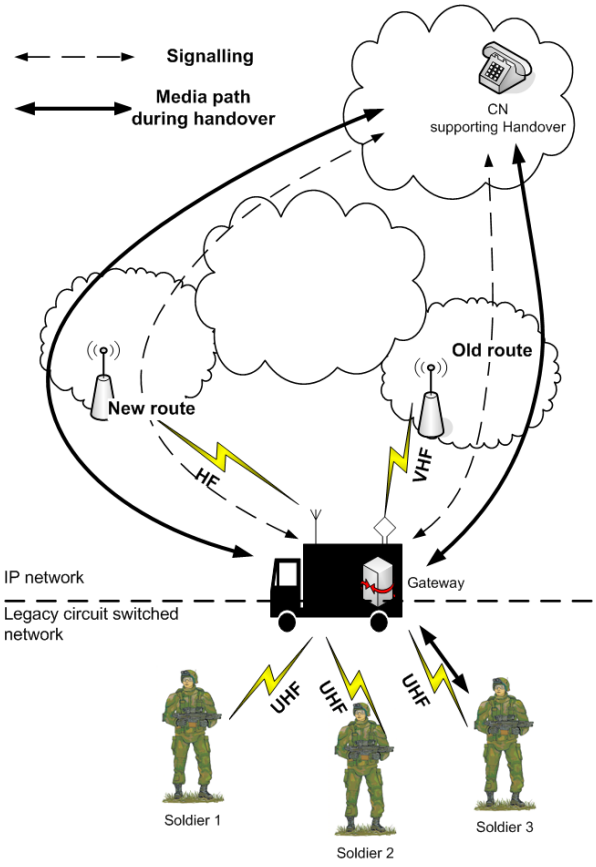


Figure 2. Scenario B: A vehicle acting as an access point for soldiers can change point of connection

In Scenario B shown in Figure 2, the end nodes (soldiers in this case) are mobile, but they use only one point of access – the vehicle– and no handover is needed in that section of the network. The communication vehicle, however, can communicate with its core network through different radio links and may need to switch between them as it moves around or if link properties change.

In both scenarios the sessions will be secured end-to-end using SCIP between the end user (moving person or soldier) and the corresponding node (CN).

## 3. Background

In this section we give an introduction to the different protocols, codecs and main challenges that will be present in a mobile SCIP-over-IP system.

### 3.1. SCIP

SCIP, formerly known as FNBDT (Future Narrow-Band Digital Terminal) was developed by US Government and national industry. It has later been adopted by NATO as a common protocol for secure voice interoperability. SCIP security is based on the Public Key Infrastructure/Key Management Infrastructure framework for cryptographic key exchange and allows different nations or groups to have their own cryptographic algorithms. Thus the same equipment can support both national sovereignty and interoperability in multi-national operations. SCIP can also support the use of traditional key management using symmetric keys. Use of Pre Place Keys (PPK) will be necessary for multipoint communication.

The main SCIP document defining the signalling and messages formats is the specification SCIP-210 [3]. Currently, different cryptographic specifications for US national use and NATO use have been defined. There is an ongoing work to restructure these documents into one common core document, accompanied by a set of documents targeting specific operational environments.

An application that employs the SCIP protocol will first ensure that the initial network connection is established. Then control is passed to a SCIP module that handles its own session setup signalling. During the initiation of a SCIP session, a set of messages are exchanged between the terminals. SCIP has six different modes of operation: Secure voice, secure data, enhanced secure data, clear MELP voice, native clear voice and secure electronic rekey [3]. Here, we will focus on the secure voice using MELP. In this mode, the Capabilities Exchange message is followed by the exchange of optional Extended Keysets List Messages, Parameters/Certificate Messages, and Cryptosync Messages.

### 3.2. Secure MELP

Many different codecs can be used with SCIP, but all implementations should as a minimum support the military standard MELP speech codec operating at 2400 bit/s for low-rate voice communication. Two variants of MELP are defined, MELP Blank and Burst, and MELP without Blank. In the prior, every 24th data frame is a replaced by a Sync Management frame as depicted in Figure 3. In MELP without Blank, the Sync Management frame is inserted prior to the first MELP frame. Thus, a so-called superframe consists of 24 frames in MELP Blank and Burst, and 25 frames in MELP Burst without Blank. The latter requires more than the minimum channel capacity requirement of

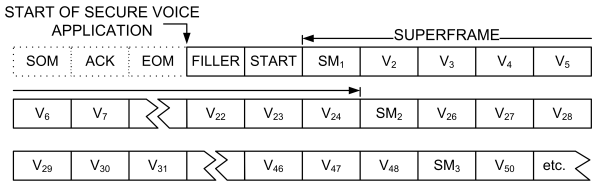


Figure 3. Secure MELP Blank and Burst

2400 bit/s. The current SCIP specifications also define the processing of secure voice using the G.729d codec at 6400 bits/s.

### 3.3. SCIP over IP

The transition from circuit switched to packet switched networks is not done over night, and because of the enormous investments that are laid down in PSTN/ISDN networks these will continue to exist for a long time. In networks where the end points are in PSTN and one (or multiple sections) of the transport network is IP network, it is necessary to make sure that the signalling used in PSTN remains unchanged when transported in the IP networks. To solve this, gateways are located at the cross-section between the IP and ISDN/PSTN networks. The ITU V.150.1 standard [4] describes how these should operate. The initial work on SCIP over IP has been focused on making it possible for a SCIP-over-IP end point to communicate with a SCIP-over-ISDN/PSTN end point via V150.1 gateways, resulting in SCIP over V.150.1. A reliable transport protocol, SPRT (Simple Packet Relay Transport) also defined in the V.150.1 standard is used to carry SCIP signalling and SCIP data. SPRT runs over UDP/IP. Still, advantages of using RTP as a transport protocol has become evident and different vendors have already made several implementations of SCIP over RTP. These products currently use undocumented and non-standardized approaches. However, the first standard to specify SCIP-over-RTP operations was recently published [5].

Both SCIP over V.150.1 and SCIP over RTP require that certain capabilities are negotiated before the SCIP signalling begins. This is done via the selected VoIP signalling protocol, for instance SIP/SDP or H.323.

### 3.4. SIP overview

SIP is an application-layer protocol designed to establish, modify and terminate calls. When used by SCIP applications, SIP is used to find the corresponding end point and to carry SDP (Session Description

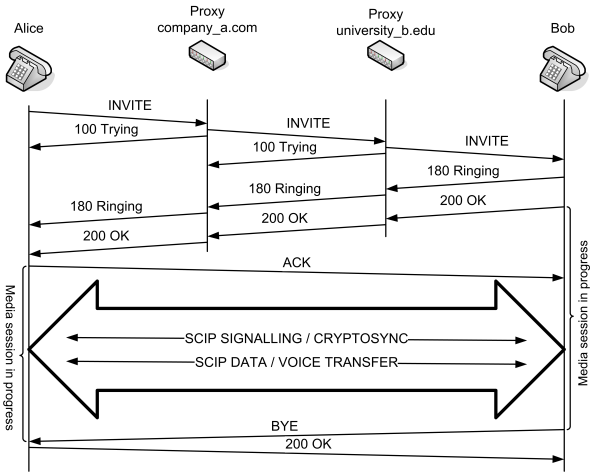


Figure 4. Example of call setup and termination. When the initiation of the SIP call has succeeded, the SCIP session is initiated

Protocol) messages used to negotiate the properties of the SIP session.

The six basic methods in SIP are INVITE, ACK, REGISTER, OPTIONS, BYE and CANCEL. A message consists of the method name and a set of header fields. It does not contain information about the call itself, but the SIP message body can carry an SDP message describing the call. Through the SDP messages the end points agree on the parameters for the call, for instance that it is a SCIP session and which ports to use. Then, other protocols, such as RTP, are used to transfer the media packets. These packets can take a different network route than the SIP signalling packets.

When Alice wants to communicate with Bob, the SIP user agent on her phone sends an INVITE message to Bob `sip:bob@university_b.edu` as depicted in Figure 4. The request is first handled by a proxy in Alice's domain. The stateful proxy will perform a DNS lookup and forward the request to Bob or to a new proxy one step closer to Bob as in Figure 4. It also sends the provisional response 100 Trying back to the previous node. When Bob accepts the call, his user agent replies with a 200OK message, and the transport of media packets can begin. These packets can be ordinary voice or video traffic using RTP, or as in Figure 4: SCIP over V.150.1 or SCIP over RTP.

### 3.5. Mobility in heterogeneous networks

In this paper we focus on handover delays in heterogeneous networks. In a heterogeneous network, a user can be within the coverage of many different network types simultaneously. For instance, a mobile phone can have internet connection through UMTS and WLAN. In this example, the UMTS base station is owned by a commercial mobile operator and the WLAN router can be in the user's own home with an internet provider independent of the mobile operator. In this case there is no central mobility management in the heterogeneous network, and the different access points do not share any information about the mobile node's location, connection preferences or link quality. This means that a terminal operating in a heterogeneous environment must have the capability to initiate and perform a handover itself. The mobile SCIP terminal holding a SIP user agent will in the rest of this paper be referred to as a mobile node (MN).

Terminal mobility allows the user to move around with the terminal and the terminal roams between different IP subnets. In the simplest solution for terminal mobility with SIP, the user agent sends a new INVITE message, a re-INVITE to its corresponding node (CN) informing about its new IP address. The problem with this solution is that the handover delay is too long. By handover delay we mean the time it takes for the MN to

1) discover that it has lost connection to its previous access point (AP)
2) find a new access point to which it has access, and establish a new data link
3) inform the CN about the new address

It is stated that maximum handover delays should ideally be less than 100ms, not more than 200ms to prevent degraded user experience, loss of media sync or session disruption [6]. However, Nakajima et al. [7] measured handover delays of more than 30 000ms when performing WLAN to WLAN handover using IPv6 in their testbed. The long delay is mostly due to the Duplicate Address Detection (DAD) of IPv6. Still, when omitting the DAD, the media delay was still in the range of 450ms. Wu et al. have modelled handover between WLAN and WWAN such as UMTS [8]. Here, a 128kbps channel in the UMTS network gives a handover delay of approximately 1500ms due to channel loss. These studies, supported by others [9], [10], show that the handover delay is too long even for ordinary VoIP. In the next section we will discuss the implications of the handover delays on SCIP sessions.

## 4. Analyzing implications of handover delay and packet loss on SCIP sessions

It has already been pointed out that SCIP can be used over a variety of networks with different characteristics. In addition, SCIP can be used by different

groups of people with very different expectations in terms of the quality of the communication. A soldier in a battlefield can be accustomed fading links and periods of radio silence. In contrast, DoD officials, or even the Prime Minister, operating in a more stable environment are used to the quality of service experienced in fixed and mobile PSTN telephones. For these groups, reduced quality of service as a result of too long handover delays will be an issue that should be avoided.

In secure SCIP, a new Traffic Encryption Key (TEK) is negotiated for each new call, and a block cipher operating in counter mode is used for encryption. A state vector is fed the block cipher as input, and the output is XORed with the MELP output. The result is a cipher text that is transmitted over the network. Both the transmitting and the receiving side must initialize and increment their counters in the same manner. Otherwise, the receiver will not be able to decipher the data, and the data block is considered lost. The Sync Management frame that is present in each MELP superframe contains a certain number of bits from the current counter along with a Cyclic Redundancy Check. Depending on how many frames the receiver has lost, it can use this information to regain synchronization. A terminal that has lost synchronization with its counterpart can either wait for three Sync Management frames to have the complete counter used to resync, or to initiate a re-sync sequence. The latter is done by sending an escape sequence followed by a Crypto Sync Message. This procedure will force both terminals out of secure MELP mode and secure communication can only be resumed after both terminals have completed the Crypto Sync procedure by a sending a start-message.

In the 2400 bit/s MELP Blank and Burst, each MELP frame consists of 54 bits which equals 22,5ms of voice. As Each Sync Management frame is transmitted every 24th frame, a new Sync management frame is transmitted every 540ms. To have enough information to get the full state vector and be able to resync, a user must wait for 3 consecutive superframes, 1620ms, before the communication can be resumed.

If we consider the handover delay of 450ms given in [7], we have already stated that this would be too long considering the constraints given by [6]. However, when using SCIP it must also be taken into consideration that the packet loss during the handover delay can result in *additional* sync delays of 540ms or in the worst case 1620ms which is more than ten times the acceptable delay. Handover routines are necessary to reduce or avoid these handover delays.

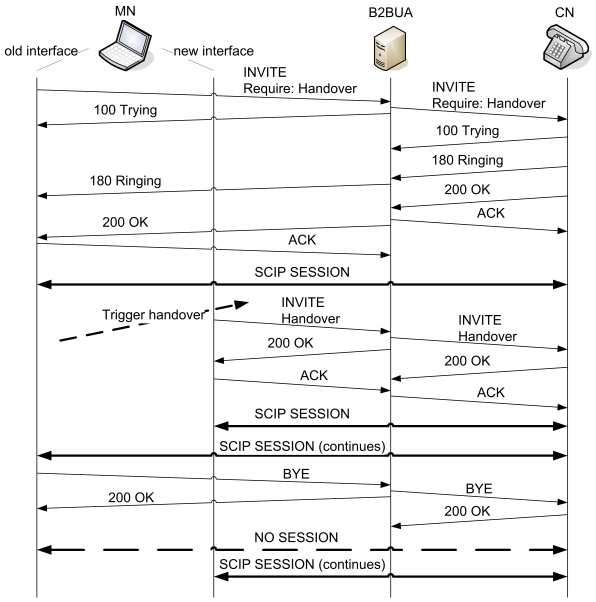


Figure 5. When receiving the INVITE with Handover, the CN sends media to both interfaces. If CN does not support handover, this would be done by the B2BUA

## 5. SCIP and our SIP Handover Extension scheme

Several architectures and implementations have been suggested to overcome the challenges of too long handover delays. Many of these require new network elements that must be present in either access points or in the subnets to achieve seamless handover [11], [12], [13]. An IP telephony provider cannot control all the possible access networks its customers may use to access their service (WLAN, UMTS etc.). Therefore, a solution that can easily be implemented and deployed, and that requires only small changes in core entities and in the MN is desirable. We have previously provided such a solution [14] and a similar one has been suggested by Salsano et al. [15]. These schemes require that a Back-to-back user agent (B2BUA) or a Session Border Controller (SBC) acts as an RTP proxy that forwards the media packets between MN and CN. A B2BUA/SBC acting as an RTP proxy can become a vulnerable hot spot. In addition, the bridging of all the calls can become a bottleneck in terms of scalability. To avoid this, and also make it possible to make use of the SIP property of signalling and media taking different paths, we propose a new SIP extension, the *Handover Extension*[2] to provide seamless handover of SCIP sessions. The main differences between this scheme and the two previously mentioned is that *(i)*

a separate SIP extension is proposed to signal that a handover is required; and *(ii)* if the CN also supports the Handover Extension, media handover can be performed directly from the CN and not by the B2BUA. Thus, in an all-IP network where one can assume that all nodes support the Handover Extension (such as a tactical network), it will be possible to omit the use of a B2BUA as anchor point as all nodes will be able to support the other nodes' mobility. When setting up a session, the MN will send an INVITE message with *Handover* as a parameter in the *Require* header. This is sent to the B2BUA in its home network. The B2BUA forwards the requests to the CN. Two situations can occur:

1) The CN does not support handover and will answer with a 420 Bad extension response. When receiving this, the B2BUA will transmit a new INVITE message without the *Require* header and it will instead take the responsibility for any handover that may occur during the session. In this case all SCIP traffic will be bridged in the B2BUA, like in Scenario A in Figure 1. CN will be ignorant of any handover that might occur during the call.
2) The CN responds with a 200 OK message containing a *Supported* header with *Handover* as one of the listed supported extensions. In this case all SCIP traffic can be sent directly between the end points, or as in Scenario B in Figure 2 from the gateway in the access point to the CN. This call setup is shown in Figure 5.

A new INVITE message is sent over the new network interface when the handover is triggered. The new INVITE message has a new Call id and a new From-tag. It is consequently seen as a new dialog. In this request is also the *Handover* header with the Call id, From-tag and To-tag belonging to the ongoing call. This makes it possible for the receiver to identify which ongoing call is about to change access point. Upon receiving the Handover-INVITE, the B2BUA (in the first case) or the CN (in the second case) will start duplicating the media packets to both the old and the new IP address. When the MN starts to receive media packets over both network interfaces, it will send a BYE message with the original dialog's session parameters, and the media transfer over the old interface will stop. Thus, the new dialog has taken over the session using the new network interface. If needed, it would potentially be possible to set up a new dialog with the *Handover* header referring to the old dialog and put the new dialog directly on hold. In this case the media would still be sent over the old network

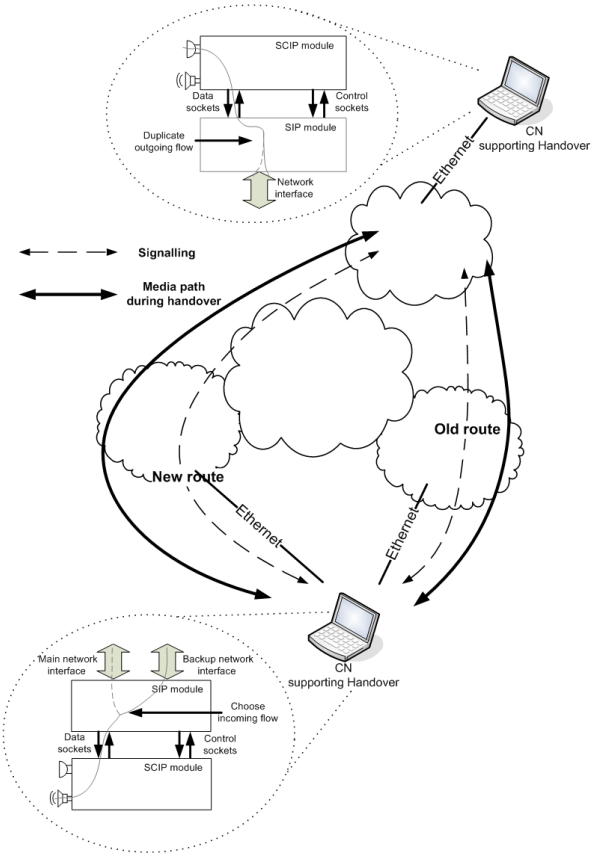


Figure 6. Test setup

interface, and the new dialog will be a passive backup dialog ready to be taken off hold whenever needed.

What triggers the handover and how the new data link is found and established, will be dependant on the implementation and the physical device that is used.

## 6. Implementation and testing

To test the functionality of SCIP over SIP with the Handover Extension, the two have been implemented and tested. An existing open-source Java-based SIP soft phone named Peers [16] was downloaded and modified to support the described Handover Extension. For simplicity, the modified Peers will from here on be called PeersHO.

A SCIP demonstrator developed by Thales Norway was used in the end points to provide SCIP signalling and SCIP data in the form of MELP encoded voice. To adhere to security regulations, the crypto module was removed from the demonstrator, and the tests were thus done only on clear-mode data. The SCIP demonstrator, written in C, includes a layer for setting up native transport connection. This was bypassed so

that all signalling was done using PeersHO. To make a connection between the two modules (SCIP and SIP), each were extended with two sets of sockets. One set for sending control data between the modules, and one set for forwarding the SCIP data from the SCIP module through the SIP module, where it would be wrapped in RTP headers and transmitted into the network. Likewise, data entering from the network to the SIP module would be stripped of the RTP headings and the data forwarded to the SCIP module for processing.

Figure 6 shows the test setup. In this setup, no intermediary nodes were used for bridging or proxy. Two laptops were used, running Microsoft Windows XP 2002 with Intel Pentium Processors, one at 1.6GHz and one at 2.0GHz. A Netgear FA411 PCMCIA Mobile Adapter was used to provide two Ethernet Interfaces to one of the laptops. The PeersHO user agent on each of the two laptops were able to support the SIP Handover Extension both as a client (on the MN) and as server (on the CN).

The tests show that SCIP data is transferred from the MN to the CN. Some extra delay is introduced due to processing in the intersection of the different modules. Considering the scenarios presented in section 2 this would mean that in Scenario A, where the end user is a SCIP/SIP client on a mobile terminal, the SIP Handover Extension can be used to switch between the WiFi and the UMTS network without losing packets during the transition. Or, in the case of Scenario B, the gateway in the vehicle can act as an MN forwarding data from the soldiers. The gateway performs a conversion between the circuit switched domain and the IP domain according to the standards, and on the IP side, the gateway with the SIP Handover extension can duplicate packets to the two routes during a handover.

## 7. Conclusion and future work

In this paper we have discussed challenges that will appear when SCIP over IP is used on mobile terminals in heterogeneous environment. Even though SCIP is supposed to be independent of the underlying layers, it is inevitable that changes in the parameters of the underlying layers will influence the SCIP communication. We have shown that the implications due to handover delays are more critical for SCIP sessions than for an ordinary VoIP call, due to possible loss of Sync Management frames. As SIP is becoming an important protocol for SCIP over IP networks, the challenges of mobility in IP networks must be met, and the choice of handover method will be important. As of yet, there is no standard way of supporting handover in SIP that provides seamless handover without packet loss. The SIP Handover Extension that is presented here is proposed as a means to provide seamless handover in heterogeneous networks. In contrast to other handover schemes it can provide handover support without the use of a centralized unit when all nodes in the network support the extension. It can also be used in a part of a network as described in Scenario B in Figure 2. Implementation tests have shown that the SIP handover Extension can operate without interfering with the core functions of the SCIP operations. Future work includes a more thorough testing of the SCIP-SIP-Handover solution, preferably in a heterogeneous radio environment. Also testing with the sync module in place will be necessary to verify that a fully operating SCIP system will still function properly during a handover using the SIP Handover Extension.

## Acknowledgements

The author would like to thank Thales Norway for letting me use their SCIP demonstrator in this experiment. Many thanks also go to Leif Nilsen (Thales Norway and University Graduate Center, Kjeller (UNIK)), and Joakim Flathagen and Torleiv Maseng (both with the Norwegian Defence Research Establishment (FFI) and UNIK) for comments on earlier drafts of this paper.

## References

[1] 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), Jun. 2002, updated by RFCs 3265, 3853, 4320, 4916. [Online]. Available: http://www.ietf.org/rfc/rfc3261.txt

[2] E. S. Boysen and T. Maseng, "Seamless handover in heterogeneous networks using SIP: A proactive handover scheme with the Handover Extension," *International Journal on Advances in Internet Technology*, vol. 2, no. 1, pp. 184 –193, 2009. [Online]. Available: http://www.iariajournals.org/internet_technology/

[3] General Dynamics, "SCIP-210 - SCIP Signaling Plan," December 2007, revision 3.2.

[4] ITU, "Modem-over-IP networks: Procedures for the end-to-end connection of V-series DCEs," January 2003.

[5] General Dynamics, "SCIP-214.2 - Secure Communication Interoperability Protocol (SCIP) over Real-time Transport Protocol (RTP) Revision 1.0," January 2010.

[6] ETSI, "TS 101 329 -2 v2.1.3 (2002-01) Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 3; End-to-end Quality of Service in TIPHON systems; Part 2: Definition of speech Quality of Service (QoS) classes," www.etsi.org, 2002. [Online]. Available: www.etsi.org

[7] N. Nakajima, A. Dutta, S. Das, and H. Schulzrinne, "Handoff delay analysis and measurement for SIP based mobility in IPv6," in *IEEE International Conference on Communications, 2003. ICC '03.*, vol. 2. Convent Station, NJ, USA: IEEE, May 2003, pp. 1085–1089.

[8] W. Wu, N. Banerjee, K. Basu, and S. K. Das, "SIP-based vertical handoff between WWANs and WLANs," *IEEE Wireless Communications*, vol. 12, no. 3, pp. 66–72, June 2005.

[9] H. Fathi, S. S. Chakraborty, and R. Prasad, "On SIP session setup delay for VoIP services over correlated fading channels," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 1, pp. 286–295, January 2006.

[10] C.-H. Yeh, Q. Wu, and Y.-B. Lin, "SIP Terminal Mobility for both IPv4 and IPv6," in *26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS)*. IEEE, July 2006, pp. 53–53.

[11] N. Banerjee, S. K. Das, and A. Acharya, "SIP-based Mobility Architecture for Next Generation Wireless Networks," in *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*. IEEE Computer Society, March 2005, pp. 181–190.

[12] P. Bellavista, A. Corradi, and L. Foschini, "SIP-Based Proactive Handoff Management for Session Continuity in the Wireless Internet," in *26th IEEE International Conference on Distributed Computing Systems Workshops 2006, (ICDCSW06)*. IEEE Computer Society, July 2006, pp. 69–69.

[13] S. Tsiakkouris and I. Wassell, "PROFITIS: Architecture for Location-based Vertical Handovers Supporting Real-Time Applications," in *25th IEEE International Performance, Computing, and Communications Conference, 2006 (IPCCC 2006)*. IEEE, April 2006, pp. 629–634.

[14] E. S. Boysen, H. E. Kjuus, and T. Maseng, "Proactive handover in heterogeneous networks using SIP," in *Proceedings of the Seventh International Conference on Networking 2008 (ICN 2008)*. IEEE, April 2008, pp. 719–724.

[15] S. Salsano, L. Veltri, A. Polidoro, and A. Ordine, "Architecture and testbed implementation of vertical handovers based on SIP session border controllers," *Wireless Personal Communications*, vol. 43, no. 3, pp. 1019–1034, November 2007.

[16] Y. Martineau, "Peers SIP soft phone," open-source java-based SIP soft phone. [Online]. Available: http://peers.sourceforge.net/