appeared in Proc. of 5th Intl. Workshop The Internet Challenge: Technology and Applications, Berlin, Germany, October 2002

M-SCTP: DESIGN AND PROTOTYPICAL IMPLEMENTATION OF AN END-TO-END MOBILITY CONCEPT

Wei Xing, Holger Karl, Adam Wolisz*
Technical University Berlin, Telecommunication Networks Group,
Einsteinufer 25, 10587 Berlin, Germany
xing|karl|wolisz@ee.tu-berlin.de

Harald Müller
Siemens AG, Information and Communication Networks
München, Germany

Abstract The problem of mobility in IP networks has traditionally been solved at the network layer. We present an alternative solution that solves it at the transport layer in an end-to-end fashion, leveraging the ability of a modern transport layer protocol (SCTP) to use multiple IP addresses per association. This is achieved by dynamically modifying this set of IP addresses with new IP addresses that are assigned to the mobile node as it moves around the network We show that this is a feasible approach, discuss the performance ramifications of underlying protocol mechanisms and identify those that are deterrent to handoff performance.

Keywords: SCTP, mobility, IP, transport-layer, handoff, performance

1. Introduction

The proliferation of laptops, hand-held computers, cellular phones, and other mobile computing platforms connected to the Internet has triggered much research on mobility support in IP networks. Modern mobile terminals are likely to be equipped with wireless communication devices allowing them to constantly reach the Internet and participate in it as normal end systems would. In the near future, a mobile terminal will be able to select among several concur-

*The authors would like to thank their colleagues from the TKN Group, especially Xiaoming Fu and Hongbong Kim. Part of this work was supported by a research contract with Siemens AG, Munich, Germany.

rently available, different wireless technologies the one which best supports the user's current communication needs and offers the best cost tradeoffs or even to use multiple access points to the fixed network infrastructure in parallel.

Such a mobile terminal, however, represents an ill fit with the traditional assumptions upon which the IP protocols is based: a classic end system does not move and has only a single point of attachment to the network. For such an end system, a single handle is sufficient to represent both the identity of a terminal as well as its location within the network; the IP address is this very handle. Such a permanent handle is not appropriate in mobile networks. On one hand, an identifier is necessary to distinguish among different terminals, on the other hand, information about the current location within a network has to be provided to ensure that packets destined to a certain terminal can still be routed towards this terminal. The fundamental mobility problem in IP-based networks is therefore the separation of identity and location.

Current approaches, e.g., Mobile IP [2], solve this problem at the network layer. While general, these approaches have some unappealing characteristics; limited performance and additional complexity for the network architecture are perhaps the two most serious shortcomings. Adding complexity to the network runs counter to some of the basic design principles of the Internet, in particular, the end-to-end principle [4]: anything that can be done in the end system should be done there. And in fact, supporting mobility is, strictly speaking, an end system issue. The question hence arises whether it would be possible and beneficial to attempt to support mobility in IP networks at higher layers than the network layer. The lowest end-to-end layer, in the Internet protocol stack, is the transport layer. As the transport layer is considerably affected by mobility—e.g., it has to be able to quickly adapt its flow and congestion control parameters to the new network situations during and after handovers—it is a natural candidate for mobility support. Attempting to support mobility in the transport layer might enable it to improve its parameter adaptation, it could also make the entire networking architecture simpler by working without additional entities within the network.

Implementing such a concept requires changes to existing transport layers. In previous work, TCP has been modified to support such an end-to-end mobility concept. While TCP is indeed the most often used transport protocol in the Internet, it might not be the perfect platform to experiment with unconventional ways of supporting mobility. In particular, when considering the potential that a mobile terminal could be in contact with multiple access points at the same time, other protocols might offer a simpler starting point. A good candidate is the Stream Control Transmission Protocol (SCTP): an SCTP "association" (essentially, a connection) can use multiple addresses simultaneously. While this property was not originally intended to support mobility (the rationale is to support highly available servers), it presents an excellent platform on which to experiment with new mobility-support mechanisms. In addition, many of its basic mechanisms such as flow and congestion control are very similar to TCP. Therefore, we decided to use SCTP as a starting point and introduce mobility support for it. This paper presents our architecture as well as a prototype implementation along with preliminary performance results.

The remainder of this paper is structured as follows: Section 2 gives an overview of the most relevant related work. Section 3 introduces necessary SCTP basics, Section 4 describes our proposed architecture to enable SCTP to support mobile terminals. A prototypical implementation of this architecture is introduced in Section 5 along with some preliminary performance evaluation results. Finally, Section 6 contains our conclusions and options for future work.

2. Related Work

Several architectures have been proposed to provide IP mobility support; Mobile IP (MIP) [2] is probably most widely known. Mobile IP uses a couple of addresses to manage user's movements. Each time the mobile host (MH) connects to the so-called foreign network, it obtains a temporary address called Care-of-Address (CoA) from a mobile agent in the local network called the foreign agent (FA). The MH must inform its home agent (HA) of this new address by the registration process. The home agent then assumes the MH's permanent IP address. Once a packet destined to the MH arrives at the home agent, it is tunneled towards the MH using the MH's temporary CoA.

The advantage of MIP is a fully-transparent mobility support, which is general and sufficient for many mobile applications. But MIP also has some disadvantages. The main problems are handover latency and opaqueness for the transport layer. While this is in line with general information hiding principles, it also limits optimization potential.

An end-to-end architecture using TCP [5] is proposed for IP mobility, based on dynamic DNS updates. Whenever the mobile host moves to a new network, it obtains a new IP address and updates the DNS mapping for its host name—the separation of identity and location is here achieved using names and addresses, not two different addresses as it is done in Mobile IP. Merely changing the mobile terminal's address, however, would disrupt any ongoing TCP connection whenever a handover occurs. To overcome this problem, a new option, called Migrate TCP, was added and it is used to "migrate" an existing connection from the mobile terminal's old to the new IP address. In a TCP connection migration process, the MH activates a previously-established TCP connection from a new address by sending a special Migrate SYN packet that contains a token identifying this very connection. The correspondent host (CH) will then re-synchronize the connection with the MN using the new address. The time of a TCP connection migration process is called TCP migration delay in this paper. The main advantage of this approach is that it has no need for a third party (home agent) for smooth handoff. However, we believe that such architecture incurs similar handoff delays to those experienced in MIP, or even worse due to DNS update delays and migration delays. The main shortcoming of this TCP extension is its inability to use concurrently available access points: it is this shortcoming that we are especially interested in in overcoming. A similar approach has been proposed in [3], yet this work does not present a prototype implementation or performance results.

In conclusion, Table 1 shows that none of these approaches can fulfill all requirements: no third party, ability to use multiple access points and no modifications for intermediate routers. A mobility extension to SCTP is intended to satisfy all these three requirements.

	Third party requirement for handoff	Concurrent usage of access points	Modification for intermediate router
Mobile IP	Yes	No	Yes
Migrate TCP	No	No	No
SCTP mobility	No	Yes	No

Table 1. Comparison of different IP mobility approaches

3. SCTP Overview

Recently, a new transport layer protocol, ”Stream Control Transmission Protocol” (SCTP) [6], has been proposed running on top of IP. It encompasses many basic functionalities of TCP, and adds a number of interesting protocol mechanisms. One core feature of SCTP is multi-homing, which enables a single SCTP endpoint to support multiple IP addresses *within a single association*.

This feature is a conceptually simple and powerful framework for IP mobility support at the transport layer as it already separates the identity of an end system from the current address to which packets are sent; it is a one-to-many correspondence as opposed to the one-to-one correspondence used by traditional transport layer protocols.

However, the multi-homing mechanism's purpose is to increase association reliability in wired networks. Therefore, the IP addresses of all involved end system are fixed and known in advance. SCTP can thus rely on all communicating peers to learn about all the IP addresses before the association is completely established, and these IP addresses must not be changed (either added or deleted) during the session in the classical SCTP. This does not work in a mobile environment as a mobile host does not have a fixed, previously known IP address: as it moves around, its local address constantly changes to reflect its new position in the IP routing hierarchy. To leverage SCTP's multi-homing mechanism, this set of IP addresses must be made dynamic.

4. The Mobile SCTP Architecture

The main idea of our approach is to let a mobile host have (at least) two IP addresses in the existing association during handoff if two access points are available simultaneously.

In order to provide IP mobility, the packets sent to a mobile host need to be forwarded to the new IP address in the new location visited by the MH without disrupting the current session. The basic idea is to exploit the case that the coverage areas of an old access point, which the mobile node currently uses, and the new access point, to which it will perform a handoff, are overlapping. Then, the mobile node can obtain an IP address from the new access point and use this new address to prepare the actual handoff process, i.e., to modify the set of IP addresses that describe a particular SCTP association in the correspondent host. This process is illustrated in Figure 1.

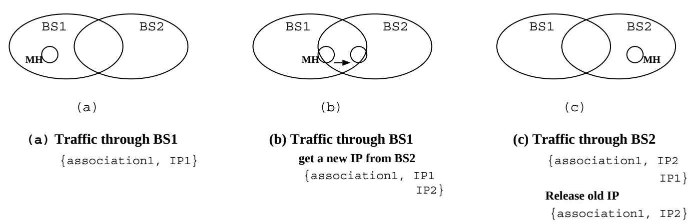

Figure 1. Mobile SCTP Mechanism

In more detail: A mobile host (MH) is in the coverage of BS1, as shown in Figure1 (a). In this case, the traffic is through BS1 and SCTP association1 has only one IP address (IP1), which it got from BS1. When the MH enters the area that is covered by both BS1 and BS2 (Figure1 (b)), it obtains a new IP address (IP2) from BS2, typically via Dynamic Host Configuration Protocol (DHCP) [1] or similar mechanisms. This new IP address will be immediately added into the current active association (association1). During this overlap time, it would even be possible to use both transmission paths (via both BS1 and BS2). After the MH enters the coverage of BS2 (Figure 1 (c)), traffic is through BS2 and here IP2 must be used in association1 because IP1 is no longer usable. A similar concept has been presented in [7].

An evident problem with such an approach is a movement of a mobile host that continuously results in its switching back and forth between both base stations. As long as the IP addresses from these base stations are still valid (i.e., their DHCP lease has not expired), it would be possible to reuse them and safe quite some signaling overhead. This requires the base stations to identify themselves to the mobile node so that the mobile node can recognize whether it is re-connecting to a base station that it has just been using; in addition, the mobile node needs to store base station identifiers and local IP addresses in order to be able to easily reuse these addresses.

5. Mobile SCTP Prototype

5.1 Implementation

We have implemented M-SCTP under Linux based on a user-space SCTP implementation (available via `http://www.sctp.de`) Specifically, we implemented a module to support dynamically adding a new IP address into an existing SCTP association. An additional module is responsible for detecting the availability of new base stations; this module serves as a link-layer trigger mechanism for M-SCTP (cp. Figure 2). If a new base station is found, i.e., a new interface with a new IP address is available, a user-defined signal will be sent to the SCTP application. The ADD IP process will then be started to add the new IP address into current SCTP instance (Mobile Host). After the new IP address is successfully put into the SCTP instance, an ADD_IP chunk will be immediately generated and sent to the remote SCTP peer. When the SCTP peer get the ADD_IP chunk, an addip callback function will be triggered to start the addition of the peer's new IP address from the ADD_IP chunk into the current association. Once completed, an ADDIP-ACK chunk is generated, acknowledging the availability of the new IP address at the correspondent host. After the MH receives ADDIP-ACK chunk from the remote SCTP peer the add-ip process has finished; the new IP address can be used according to the SCTP path management mechanism which are responsible for detecting unavailable IP addresses and switching over to secondary IP addresses. In summary, this allows the handover process to be solved using existing multi-homing mechanisms.

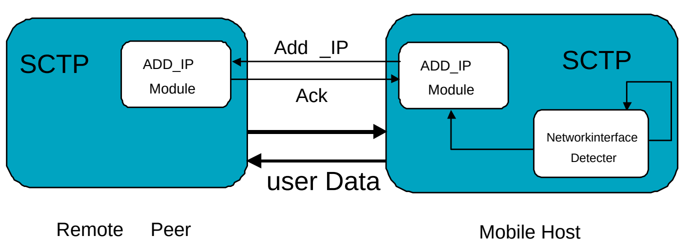

Figure 2. M-SCTP Prototype Implementation

5.2 Experiments

In order to evaluate our prototype implementation, we set up a small testbed, containing a mobile host and a remote correspondent host, both running our prototype M-SCTP implementation. The two nodes where connected via a WAN emulator node in order to introduce some delay, and the mobile node is connected to this WAN emulator (NIST Net) via two network interfaces which can individually be turned on or off, emulating the obtaining or loosing of contact with a base station. The WAN emulator was set to provide a delay of 80 ms and 10 MBit/s bandwidth. The mobile node continuously sends packets to the correspondent node. This setup is outlined in Figure 3.

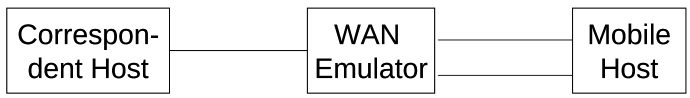

Figure 3. Network topology used for M-SCTP experiments

5.3 Result and analysis

The first result of the prototype implementation is that it is indeed possible to support handoff based on SCTP's multi-homing mechanisms. As a first performance evaluation, we were interested in the resulting interruption time. Figure 4 shows a sequence number trace over time. At point $t = 8$, the old network interface is turned off, and it takes about one second until new packets arrive. This long delay is due to the fact that our prototype is still using the original SCTP path management mechanisms. These mechanisms are tuned for fixed-network error handling, not for supporting mobility. Nevertheless, replacing these mechanisms with better solutions based on link-layer information is clearly possible and one objective of current work.

6. Conclusion and Future Work

Solving the IP mobility problem in an end-to-end fashion is possible. A transport layer protocol that provides multi-homing-like mechanisms as SCTP does is a natural candidate for supporting such a mobility approach. Additionally, leveraging multi-homing opens the possibility to exploit simultaneous connectivity to multiple access points in a flexible and straightforward fashion.

The main reason for prototype's somewhat long interruption time is mostly due to its reliance on SCTP's path management mechanisms: these must be replaced in order to realize the full potential of our architectural approach. This is the focus of current work and, in particular, the integration with link-layer triggering should provide considerable improvements over the standard case.

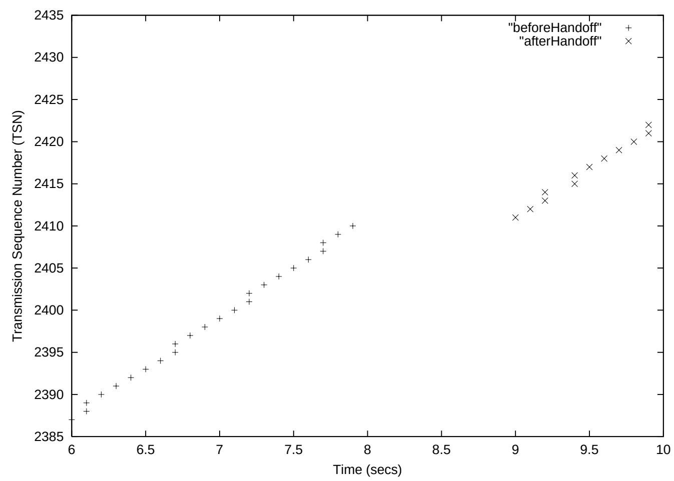

Figure 4. An SCTP association sequence trace during handoff

When to actually start the registration of a new IP address with the correspondent hostis a further issue.

Finally, the behavior of M-SCTP in the case that both end points are mobile hosts is not clear when both are moving simultaneously. The immediately obvious solution for this problem is to use dynamic DNS mechanisms, but as this is a rather heavy-weight solution, there should be some potential for additional improvements here.

References

[1] R. Droms. Dynamic host configuration protocol. RFC 1531, October 1993.

[2] C. E. Perkins. IP Mobility Support. RFC2002, October 1996.

[3] M. Riegel and M. Tuexen. Mobile SCTP. Internet draft draft-riegel-tuexen-mobile-sctp-00.txt (work in progress), February 2002.

[4] J. H. Saltzer, D. P. Reed, and D. D. Clark. End-to-end arguments in system design. *ACM Trans. on Computer Systems*, 2(4):277–288, 1984.

[5] A. C. Snoeren and H. Balakrishnan. An end-to-end approach to host mobility. In *Proc. 6th ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MobiCom)*, Boston, MA, August 2000.

[6] R. Stewart. Stream Control Transmission Protocol. RFC2960, November 2000.

[7] R. Stewart and Q. Xie. SCTP extensions for Dynamic Reconfiguration of IP Addresses. Internet draft draft-ietf-tsvwg-addip-sctp-04.txt (work in progress), January 2002.