

Introduction and Comparison of SCTP, TCP-MH, DCCP protocols

Olga Antonova
Helsinki University of Technology
olga@cc.hut.fi

Abstract

TCP/IP protocols were designed at 1970s-1980s and during that time computers were single-homed. Nowadays there is more and more need for mobility, therefore the new protocols are needed to support mobility and multihoming. In the work I have investigated how mobility and multihoming can be implemented in transport layer and shown main advantages of that. Stream Control Transmission Protocol (SCTP), TCP Multi-Home Options (TCP-MH) and Datagram Congestion Control Protocol (DCCP) protocols were introduced and compared with emphasis on mobility and multihoming features. As protocols are new, some further investigations and development are needed.

KEYWORDS: SCTP, TCP-MH, DCCP, multihoming, mobility.

1 Introduction

In telecommunication area the term *mobility* is used to denote the phenomenon where an entity moves while keeping its communication context active [1]. As IP address of the host changes there should be possibility of sending new IP address to the other parts of communication. Also communication should not be interrupted. Another requirement is that IP address should be sent securely in order to protect against unauthorised access and denial-of-service attacks. Mobility can be implemented in different layers of protocols stack. For example Mobile IP works in network layer. For the transport layer there are few protocols supporting mobility - SCTP, TCP-MH, DCCP. The main benefits of choosing transport layer for providing mobility is that network will stay untouched and still allow roaming between networks [10]. In Mobile IP a special device is needed to maintain states and location of mobile host. Usually router is used for that purpose, but it should maintain a table with hosts' locations and states. Also all data is sent to home agent and home agent redirects packets to mobile node. This causes extra traffic and complexities. If mobility is implemented in transport layer, location management is done at the ends and no any special devices are needed.

Multihoming is an ability for a single endpoint to support multiple IP addresses [2]. A Multi-homed host can be reached by either address. There can be two reasons of that - host has several network interfaces or several IP addresses assigned to one interface. Fig. 1 shows two multihomed hosts: host A has A1 and A2 addresses, host B has B1, B2 and B3 addresses. The main benefits of multihoming are :

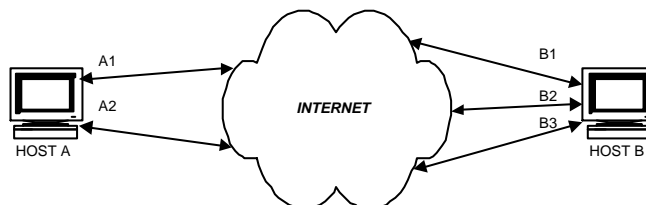


Figure 1: Multihoming.

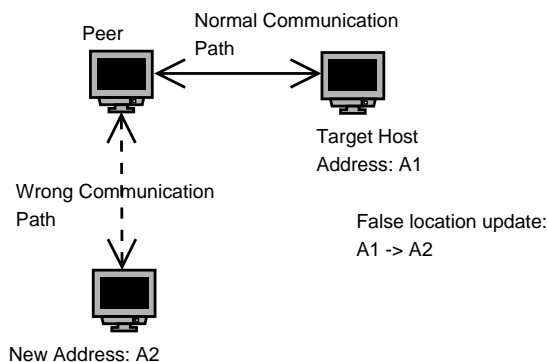


Figure 2: Stealing attack.

- when one communication path is not available, the other one can be used, that means “tolerance against physical network failures” [10]
- load sharing - network traffic can be distributed between different communication paths.

2 Security issues in Mobility and Multihoming

There are two main security problems in Mobility and Multihoming - address stealing and address flooding [1]. Address stealing is an attack when attacker simulates a situation when endpoint thinks that attacker's address is new address of its peer (please refer to Fig. 2). The address stealing attack is available when endpoint is not able to “verify that sender of the update was earlier at the target address” [1].

The flooding attack is an attack when attacker sends location update to several peers. The new address is victim's address. Peers start to send unwanted traffic to victim (please refer to Fig. 3). This happens when endpoint is not able to

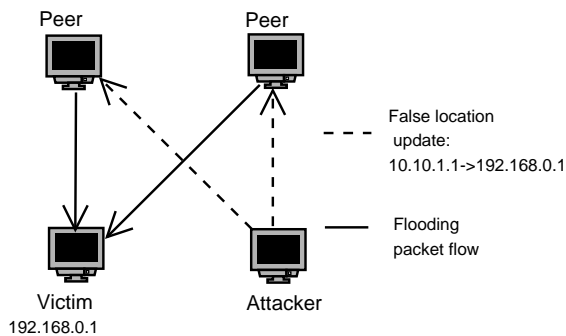


Figure 3: Flooding attack.

verify that the host at new address is the host that sent location update.

3 Why old transport protocols are not suitable?

There are well known and widely used transport protocols (TCP and UDP). Why new protocols should be used? The main issue here is that old protocols do not support multihoming and mobility. But also there are some other problems.

UDP is unreliable connectionless transport protocol, so ordered delivery, loss recovery and duplicate detection are not supported by this protocol. Also another problem is that firewalls and NAT's do not always pass UDP traffic.

TCP is reliable connection-oriented transport protocol that supports congestion control - so it is a very complex protocol and because of that too "heavy" for some applications (TCP headers are too long). As TCP provides ordered delivery, it can cause delays in delivery (head-of-line blocking - refer [16] for more information). Also in mobile world the main problem is security. TCP and UDP are not secure enough.

4 Stream Control Transmission Protocol

SCTP is reliable session-oriented transport protocol. SCTP supports multi-streaming and multi-homing, provides congestion control. SCTP is more secure than TCP, it is resistant to denial-of-service attacks. SCTP provides partially-ordered data delivery (data is sequenced within one stream) (refer to [3]).

4.1 Multihoming

To setup SCTP connection, INIT chunks are sent, which contain different IP addresses of a host if it is multihomed. In INIT-ACK chunk communication peer sends all its IP addresses. Endpoint could have one IP address, then source address of the INIT chunk is considered as IP address of the peer. "Transmission path" set is formed based on the information in INIT and INIT-ACK chunks. "Transmission path"

- is path from SCTP instance to one of the IP address of the peer.

To monitor all transmission paths, host sends HEARTBEAT chunks over all paths that are not currently used in data communication. HEARTBEAT chunk should be acknowledged by HEARTBEAT-ACK chunk. There is a counter that counts unacknowledged HEARTBEAT chunks. When it reaches the certain predefined number, destination address is considered to be unreachable. Another way of detecting unreachable destination is to count unacknowledged data chunks. For data packets transfer one transmission path is selected to be primary. If it fails, alternative transmission path will be used to continue communication.

Although SCTP supports multihoming, it does not support load sharing. It can only select another communication path if one is not available. There is some work done in this direction (refer to [5]). Authors offer solution where primary path is not used throughout the lifetime of the connection, but it can be chosen periodically based on the congestion situation in the network.

4.2 Mobility

There is a SCTP extension for dynamic addition of IP address [4]. With this extension it is possible to add dynamically new peer IP addresses and create new transmission paths. So when a host is moving, it receives a new IP address and a new transmission path is created, therefore data transfer will not be interrupted. Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration Internet-draft [4] defines two new chunks that are needed to support mobility. These chunks are:

- ASCONF - Address Configuration Change Chunk
- ASCONF-ACK - Address Configuration Acknowledgment Chunk

Also six new parameters types are defined for adding and deleting new IP addresses, setting up primary address and some others. The most important types are:

- 0xC004 - Set Primary Address
- 0xC001 - Add IP Address
- 0xC002 - Delete IP Address

To add new IP address, the peer should send ASCONF chunk of 9xC001 type, which should be acknowledged by ASCONF-ACK chunk. This solution provides mobility of only one endpoint.

4.3 Other Features

Multistreaming

SCTP has several streams within a connection and messages are sequenced in the stream independent from other streams, so if message of one stream is lost, it does not affect to the delivery in the other streams. Consequently it is reducing the risk of blocking what happens with TCP as it has only one stream and in-order delivery feature.

Congestion Control

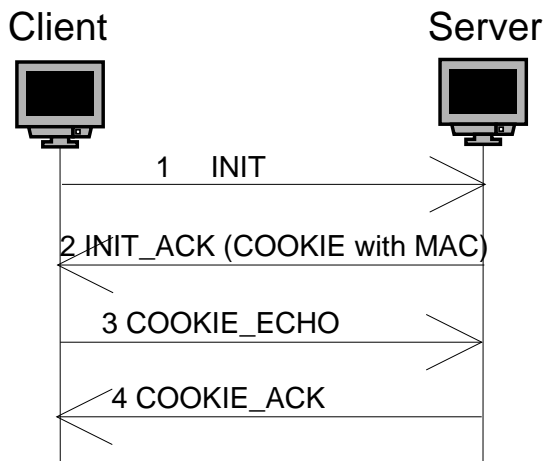


Figure 4: Four-way handshake for connection establishing in SCTP.

Congestion happens when a router receives much more packets than it can forward and a big packet queue appears. *Congestion control* is process of detecting of congestion events and preventing sending packets to congested region, decreasing sending rate or taking other actions helping to resolve congestion. SCTP uses the same congestion control mechanism as TCP - rate-adaptive window-based congestion control scheme with small differences (refer to [17, 18] for more information).

4.4 Security

SCTP has four-way handshake for connection establishing (please refer to Fig. 4).

1. Client (or active side) sends INIT chunk to Server
2. Server replies with INIT-ACK chunk containing COOKIE with a Message Authentication Code (MAC). Cookie also contains transmission control block, cookie generation time, cookie expiration time.
3. Client sends back a copy of COOKIE in COOKIE-ECHO chunk.
4. Server calculates new MAC based on transmission control block from COOKIE-ECHO and compares it with MAC that it sent to client earlier and sends COOKIE-ACK chunk.

Protection against the flooding attack (leading to DoS) and the address stealing attack is done by sending MAC in INIT-ACK chunk and verifying it in COOKIE-ECHO. Addition and Deletion of IP Addresses using SCTP Dynamic Address Reconfiguration provide more opportunities for connection hijacking. Authors do not offer any mechanisms in SCTP to protect against it and offer to use IP Authentication Header [14, 4].

5 TCP Multi-Home Options

There is an Internet-draft [11] proposing another way of solving multihoming problem.

TCP-MH maintains access-lines associated with different addresses in one TCP session and if one access-line goes down, it can switch to other access-line.

If host is able to use MH options it should send MH-permitted option during connection establishing (in SYN packet). If other side accepts it and after connection is established, the peers can start using MH options. Hosts should exchange their IP addresses using MH-Add-IPv4 or MH-Add-IPv6 options. After receiving MH-Add option endpoint should register new transmission paths based on the address mentioned in option. There are MH-Delete-IPv4 and MH-Delete-IPv6 options for deleting address, although due to security reasons endpoint should not delete path right away after receiving MH-Delete option. Fig. 5 illustrates the packets flow between endpoints.

1. Client (or active side) sends a SYN packet containing MH-Permitted option.
2. Server replies with a SYN-ACK packet containing MH-Permitted option.
3. Client sends ACK packet, now connection is established in client side.
4. Server receives ACK packet, now connection is established in server side also.
5. Now Client sends DATA with MH-Add-IPv4 option included. Option contains another IP address of client.
6. Server accepts MH-Add-IPv4 option and sends MH-Ack option.
7. Client wants to add another IP address (IPv6) and sends DATA with MH-Add-IPv6 option included.
8. Server accepts MH-Add-IPv6 option and sends MH-Ack option.
9. After that peers communicate in normal mode.
10. If communication is lost, server switches to the other address.

TCP-MH does not support all forms of mobility, authors were concerned only with situation when host is non-mobile and has several IP addresses (multi-homed).

5.1 Security

TCP-MH does not provide any additional protection against connection hijacking, man-in-the-middle and other types of attacks. Regarding hijacking connection authors made some investigation and claim that “some measures are taken in this specification so as not make TCP more vulnerable than ever” [11]. Regarding man-in-the-middle attack authors haven’t done any improvement or degradation. Regarding flooding attack TCP-MH options also does not add any improvement or degradation. For address stealing (or redirection attack) authors offer to use Return Routability Test [12].

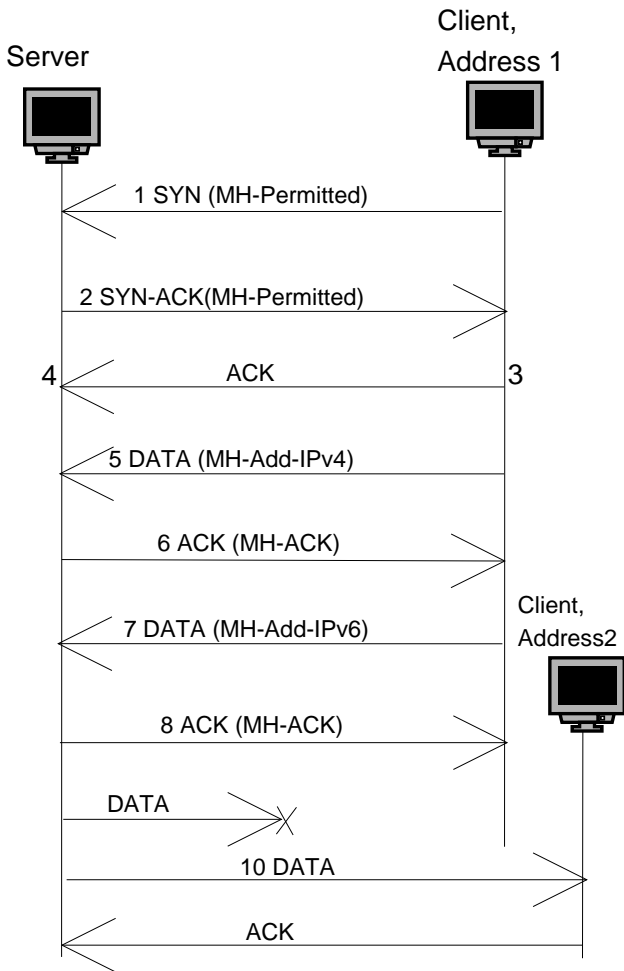


Figure 5: Packets flow with TCP-MH options

6 Datagram Congestion Control Protocol

New DCCP Internet draft was published on February 16 2004. DCCP protocol is one more transport protocol that should be used by applications that need flow-based semantics of TCP, but don't need "in-order delivery and reliability semantics" [13] and also for those that don't need multi-streaming feature of SCTP. Among others DCCP has the following features:

- unreliable flow of datagrams, with acknowledgements
- reliable handshake for connection setup and teardown
- reliable negotiation of options [13].

6.1 Mobility

DCCP provides primitive support for multihoming and mobility via a mechanism for transferring a connection endpoint from one address to another [13]. DCCP supports mobility of only one endpoint, the other one should remain stationary. Before the moving endpoint must notify other peer about it (using Mobility Capable Feature) and once it gets new IP

address it must send DCCP-Move packet containing its new address to stationary peer. Upon the receiving DCCP-Move packet stationary endpoint changes its connection state and starts using new address of moving peer. The diagram on Fig. 6 illustrates the packets flow between endpoints:

1. When communication starts Mobility Capable feature and Mobility ID have zero value in both endpoints.
2. Then mobile endpoint (B on diagram) sends "Change R" option with value "1" for Mobility Capable Feature. In DCCP "Change R" means that endpoint B wants to negotiate some feature (in our case Mobility Capable Feature) for remote endpoint A. There is also "Change L" option for the situation when endpoint wants to negotiate feature for itself.
3. Endpoint A confirms feature value by sending "Conform R" option.
4. After that endpoint A sends a value for Mobility ID feature, that will be used by endpoints to identify connection. The value of Mobility ID feature is selected randomly for security reasons, also new value should be chosen after each move of mobile endpoint (can be also done more frequently). Zero value cannot be used in DCCP-Move packets, such packets should be discarded.
5. Endpoint B confirms value of Mobility ID feature by sending "Conform L" option.
6. After mobile endpoint (endpoint B on diagram) has moved or changed port, it sends DCCP-Move packet containing Mobility ID value that was chosen for connection identification.
7. Endpoint B should send DCCP-Move packets until it gets DCCP-Sync packet.
8. Endpoint A gets new address and port of B from the received DCCP-Move packet. Then Endpoint A sends DCCP-Sync message containing new value for Mobility ID feature and conforming B's move.

Stationary endpoint may refuse move by sending DCCP-Reset option. The move can be refused because of for example address policy. If move is refused, the old address of B cannot be used, the address should be communicated again by DCCP-Move messages.

6.2 Security

The DCCP mobility mechanism, like DCCP in general, does not provide cryptographic security guarantees [13]. To perform address stealing attack attacker should know valid Mobility ID number. So attacker should although spoof network traffic or guess Mobility ID number. In previous protocol draft the length of Mobility ID number was 64 bits, in new version it is proposed to increase it to 128 bits. This will reduce the probability of guessing Mobility ID value.

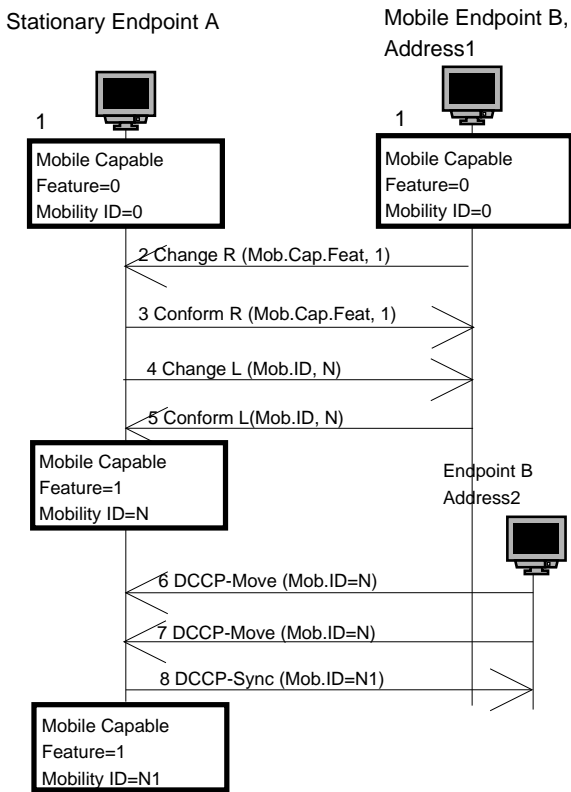


Figure 6: Packets flow in DCCP

7 Comparison of Protocols

In this section I will compare the protocols described above. They are compared not only regarding mobility and multihoming but also regarding some other features (e.g. reliability, order of delivery, security and others). The other features should be considered as for example some applications (e.g. streaming media) require unordered data delivery, for some application congestion control is quite important and so on.

7.1 Mobility and Multihoming

As said above, TCP-MH authors describe the usage of TCP-MH options only for multihoming, not mobility. Other two protocols support mobility of only one endpoint.

mSCTP (SCTP with ADDIP extension) is targeted for the client-server services, in which the mobile client initiates an SCTP session with the fixed server. For supporting the peer-to-peer services, in which a session is terminated at the mobile host, the mSCTP must be used along with an additional location management scheme such as Mobile IP ([7]), Session Initiation Protocol (SIP), Reliable Server Pooling (RSerPool) ([15]) or Dynamic DNS (DDNS) [6]. Mobile IP supports mobility of both endpoints. Mobile IP operates in network layer using IP-in-IP tunnels and the protocols described above operate in transport layer. Sec. 1 describes benefits of implementation mobility in transport layer.

Address Changing

Does multihoming change address automatically when a communication link goes down or can it be done only manually? Lets see how it is implemented in SCTP, TCP-MH and

DCCP.

In SCTP if one link goes down, a new address will be selected automatically. SCTP host monitors all transmission paths by sending HEARTBEAT chunk. When counter counting unacknowledged data chunks reaches predefined number, alternative communication path will be selected automatically.

In TCP-MH changing of address is done in the same way - if data acknowledgement does not arrive, data is retransmitted. After several retransmissions (exact number should be predefined), endpoint automatically switches to the other address.

In DCCP automatic switching to the other path is not available as server gets new address of the endpoint only after move is completed and DCCP-Move packet is sent. Then server starts to send packets to new address and it does not have list of other client's addresses, even if movement is refused the using of old address should be communicated using DCCP-Move packet.

7.2 Middleboxes

Some hosts do not have public IP addresses, they have addresses that are unique only inside private network and NATs (Network Address Translators) are used for routing datagrams to and from such hosts. The problem appears when such endpoint adds its addresses into message body (e.g. into INIT chunk in SCTP). As addresses are valid only inside private network, no any endpoints outside of that network can use those addresses (routers do not understand where to send such datagrams). Let's see how described protocols manage with this issues and what other problems related to middleboxes can appear.

For SCTP L. Coene in his work [8] offers to choose between the following solutions to solve the problem described above: one option is to use single-homed session and no any IP addresses should be included into INIT and INIT-ACK chunks, then IP address of the message will be used as endpoint's address. In this case multihoming feature of SCTP is not used. For multihoming the NAT must have a public IP address for each represented internal IP address. The host can preconfigure an IP address that the NAT can substitute, or, the NAT can have internal Application Layer Gateway (ALG) which will intelligently translate the IP addresses in the INIT and INIT ACK chunks [8]. The second option is to use DNS to resolve the internal address. The hostname should be put into INIT and INIT-ACK chunk and DNS should resolve it before association is setup (refer to [9] for more details).

In DCCP there is no such problem with NATs as in SCTP because endpoint's address is not included into packet data, and server receives new endpoint's address from the source address of DCCP-Move packet sent from new location. But there are other things that should be considered in networks with middleboxes (firewalls, NATs and others). DCCP developers list them in [13]:

- First of all there is a Service Code field in DCCP-Request packet which tells to what protocol or application connection is established. It used instead of port numbers and "helps middleboxes identify the protocol

used on a given connection” [13]. If endpoint tries to connect to unexpected service, middlebox can send DCCP-Reset packet with Reset Code 9 (“Bad Service Code”) and close connection.

- The other thing is that Source and Destination port numbers are located in the same places in the packet as in TCP and UDP, probably middleboxes can use this feature to make implementation simpler.
- Middleboxes should not change packet’s sequence numbers as DCCP-Move mobility mechanism can stop working.

In TCP-MH the same problem can happen as in SCTP as IP addresses are included into MH-Add/Delete packets. The TCP-MH authors say that: “Though NAT/NAPT traversal feature is not included in the present TCP-MH Options specifications, this can be solved by enhancing return routability mechanism”. So more investigation of this issue is needed.

7.3 Other Features

Reliability

SCTP and TCP-MH are reliable protocols. DCCP provides unreliable flow of datagrams, with acknowledgements [13].

Order of Delivery

SCTP provides in-order delivery within one stream. Unordered delivery is also supported (refer to [3] part 6.6). TCP-MH has in-order delivery. DCCP provides unordered delivery.

Streaming

As mentioned in Sec. 4.3 SCTP has multiple streams and each stream consists of messages. TCP-MH has single byte stream. DCCP is packet stream protocol.

Congestion Control

SCTP and TCP-MH provide almost the same congestion control mechanism as SCTP took over it from TCP with some slight changes. DCCP offers strong congestion control mechanism allowing to choose between different congestion control forms.

Connection Setup

SCTP uses 4-way handshake for connection setup (refer to Sec. . 4.4). TCP-MH and DCCP use 3-way handshake for connection setup.

Security

All mentioned protocols provide method for protecting against flooding attack. In SCTP it is done using State Cookie that contains MAC code (integrity check mechanism based on cryptographic hash functions using a secret key [3]). 4-way handshake is done to verify the client. It is not defined what exact algorithms should be used for producing MAC, it should be chosen during implementation. Standard also does not define how often the secret keys should be changed, just that “SHOULD be changed reasonably frequently” [3]. As I said above in mSCTP addition and deletion of new addresses brings additional risks and authors do not provide any mechanisms for defence.

In TCP-MH return routability test is used. When a Server gets a new address from a Client, it sends Conform message

to the new address of the Client. After getting conformation acknowledgement, the Server sends acknowledgement of new address addition to the Client’s old address. So if Add-Address request was sent by attacker, Client can deny addition of new address. But it is not clear what happens if connection to Client’s old address is lost.

In DCCP Mobility ID is used. Mobility ID is 128-bit value, that is sent in DCCP packet. Mobility ID is not encrypted and DCCP also does not have any requirements for encrypting packets, authors just recommend to use IPSec or some other protocol for providing end-to-end security. Mobility ID should be changed after each move of mobile endpoint (can be done also more frequent)

In TCP there is SYNcookies algorithm [SYN-COOKIE] that helps to avoid blind denial of service attack. It is not mandatory for TCP, but in SCTP it is mandatory (COOKIE and COOKIE-ECHO chunks.)

Simplicity of Implementation

TCP-MH is quite easy to implement as it is based on existing TCP protocol, but it is not very secure (especially without SYN-COOKIE algorithm) and does not describe supporting of mobile hosts. SCTP and DCCP are new protocols. SCTP it is already proposed standard, others two are only Internet Drafts.

7.4 Conclusion

In the work I presented how mobility and multihoming could be provided using SCTP, TCP-MH and DCCP protocols and tried to compare them. SCTP is the most investigated protocol among those three, for DCCP and TCP-MH more investigations could be done. For example considering using TCP-MH for mobile hosts, maintaining mobility of both endpoints in DCCP and TCP-MH, using TCP-MH in networks with middleboxes. With the current state SCTP is the most powerful as it provides multihoming and mobility, also with using MobileIP, SIP, RSerPool or DDNS mobility of both endpoints can be managed. As streaming, order of delivery, congestion control mechanisms and others features are different for SCTP, DCCP and TCP-MH, each of them should be considered while selecting protocol for some specific implementation.

References

- [1] Pekka Nikander, Jukka Ylitalo, Jorma Wall. Integrating Security, Mobility, and Multi-homing in a HIP way. Network and Distributed System Security Symposium, 2003.
- [2] L. Ong, J. Yoakum. An Introduction to the Stream Control Transmission Protocol (SCTP). RFC3286, IETF Network Working Group, May 2002.
- [3] Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytin, M. Kalla, L. Zhang, V. Paxson. Stream Control Transmission Protocol. RFC 2960, IETF Network Working Group, October 2000.
- [4] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen, I. Rytina, M. Belinchon, P. Conrad. Stream Control Transmis-

- sion Protocol (SCTP) Dynamic Address Reconfiguration. Work in progress (IETF Internet-Draft draft-ietf-tsvwg-addip-sctp-08.txt), September 2003.
- [5] J. Kumar, L. Coene. Multihomed Loadsharing. Work in progress (IETF Internet-Draft draft-coene-multi-share-00.txt), October 2002.
- [6] Seok J. Koh, , Mee Jeong Lee, Maximilian Riegel, Mary Li Ma, Michael Tuexen. Mobile SCTP for Transport Layer Mobility. Work in progress (IETF Internet-Draft draft-sjkoh-sctp-mobility-03.txt), October 2002.
- [7] Seok J. Koh, , Qiaobing Xie. mSCTP with Mobile IP for Transport Layer Mobility . Work in progress (IETF Internet-Draft draft-sjkoh-mobile-sctp-mobileip-03.txt), February 2004.
- [8] L. Coene. Stream Control Transmission Protocol Applicability Statement. RFC 3257, IETF Network Working Group, April 2002.
- [9] P. Srisuresh, G. Tsirtsis, P. Akkiraju, A. Hefferman. DNS extensions to Network Address Translators (DNS-ALG). RFC 2694, IETF Network Working Group, September 1999.
- [10] Christopher Ross. Mobility in the Transport Layer - Mobility Without Touching the Network. FACE One-Day Workshop, Aalborg University, Denmark, December 9, 2002 http://cpk.auc.dk/FACE/documents/FACE_Workshop_Ross.pdf
- [11] Arifumi Matsumoto, Masahiro Kozuka, Kenji Fujikawa, Yasuo Okabe. TCP Multi-Home Options. Work in progress (IETF Internet-Draft draft-arifumi-tcp-mh-00.txt), October 2003.
- [12] Arifumi Matsumoto, Masahiro Kozuka, Kenji Fujikawa, Yasuo Okabe. TCP Multi-Home Options. <http://spa.jssst.or.jp/WIT/2003/papers/matsumoto.pdf>
- [13] Eddie Kohler, Mark Handley, Sally Floyd, Jitendra Padhye. Datagram Congestion Control Protocol (DCCP). Work in progress (IETF Internet-Draft draft-ietf-dccp-spec-06.txt), February 2004.
- [14] Kent, S. and R. Atkinson. IP Authentication Header. RFC 2402, IETF Network Working Group, November 1998.
- [15] M. Tuexen, Q. Xie, R. Stewart, M. Shore, L. Ong, J. Loughney, M. Stillman. Architecture for Reliable Server Pooling. Work in progress (IETF Internet-Draft draft-ietf-rserpool-arch-07.txt), October 2003.
- [16] Pravin Bhagwat, Partha Bhattacharya, Arvind Krishma, Satish K. Tripathi. Using channel state dependent packet scheduling to improve TCP throughput over wireless LANs. Wireless Networks, Volume 3, Issue 1 (March 1997), Pages: 91 - 102, ISSN:1022-0038
- [17] M. Allman, V. Paxson, W. Stevens. TCP Congestion Control. RFC 3390, IETF Network Working Group, April 1999.
- [18] M. Allman, S. Floyd, C. Partridge. Increasing TCP's Initial Window. RFC 3390, IETF Network Working Group, October 2002.