Universitat Politècnica de Catalunya
Department of Signal Theory and Communication
Radio Communication Group

Łukasz Budzisz

Stream Control Transmission Protocol (SCTP), a proposal for seamless handover management at the transport layer in heterogeneous wireless networks.

Barcelona, 2009

This dissertation investigates and evaluates the idea of handling mobility at the transport layer, using mobile Stream Control Transmission Protocol (mSCTP) as an example of a handover transport layer protocol.

To this end, (the first part of) this thesis provides the reader with a necessary background for IP mobility-related aspects, surveying detailedly the most popular of the existing solutions. Provided overview includes Mobile IP (MIP) and its most important derivatives to represent the network-layer-based schemes, as well as Session Initiation Protocol (SIP) as an example of an application-layer approach. The details of the most important transport layer solutions are given on continuation, along with the motivation for the development of such mobility management schemes. Among presented transport-layer approaches, the one based on the mSCTP is chosen as a representative for the analysis performed in this dissertation. This choice is additionally motivated by two interesting features that SCTP protocol introduces, and that are interesting in the context of handover applications: multihoming and multistreaming (to some extent).

(Still in the introductory part) a detailed state-of-the-art of the SCTP protocol is provided, stressing its signaling background and original scope of use that did not consider mobility related application. The described transition from the signaling to a general purpose transport protocol illustrates the dynamics of the development of this relatively recent proposal, and explains why SCTP is currently one of the most interesting innovative transport protocols.

The core of this dissertation outlines major mobility-related considerations in the context of future heterogeneous wireless networks, identifying all important handover scenarios, and specifying the most representative one to conduct the proposed analysis. Several transport-layer handover schemes based on SCTP are analyzed in the selected scenario. First of the discussed schemes, provided also as a reference model for evaluations presented in the following sections of this work, reuses the standard SCTP failover, a mechanism originally devised to increase protocol robustness.

Next, the details of transport-layer loadsharing are explained, to facilitate the introduction of the mSCTP-CMT-PF handover scheme, an essential improvement for transport layer mobility suggested by this work. The devised proposal incorporates one of the most popular loadsharing schemes provided for SCTP, the Concurrent Multipath Transfer (CMT), that originally does not target wireless networks. Evaluation exposes the main challenges of such a design, pointing out the most important constraints limiting its scope of application.

Finally, a quantitative comparison of all identified mSCTP-based handover schemes and two of the most representative network-layer solutions is given in a series of analysis that involves mobility models of different grade of complexity.

Universitat Politècnica de Catalunya
Department of Signal Theory and Communication
Radio Communication Group

PhD Thesis

Stream Control Transmission Protocol (SCTP), a proposal for seamless handover management at the transport layer in heterogeneous wireless networks.

by

Łukasz Budzisz

Barcelona, 2009

Contents

List of Figures

List of Tables

Acknowledgements

I have to first thank my advisor, Dr. Ramon Ferrús, for being an exceptional advisor. His ability to ask the right questions and his incredible patience and attention to detail have never ceased to surprise me. He was always able to make time for me, even if I just walked into his office without any notice, and he was about to leave. I would like to thank Prof. Fernando Casadevall for guiding me throughout the entire process, and for very useful, broader view on my entire research. Over the years, as I have gained a deeper appreciation of an advisors responsibilities, my respect for Prof. Casadevall has only grown, thus I have a lot to thank him for.

I have had the good fortune of interacting and working with some really wonderful people in the Radio Communication Group (GCR) and in the Signal Theory and Communication Department (TSC), and I thank them all. Jakub Majkowski, Dr. Ferran Adelantado, Dr. Juan Sanchez, Dr. Lorenza Giupponi, Dr. Mirosław Klinkowski, Dr. Davide Careglio, Dr. Salvatore Spadaro, Eduard Escalona, Dr. Jad Nasreddine, Vuk Marojević, Francisco Bernardo, Nemanja Vučević, Jose Salazar, Hiram Galeana, Miguel López Benítez have all made my years at UPC a very enjoyable time. I would like to especially thank Jakub Majkowski, for convincing me to start working towards the PhD.

Next, I would like to thank Prof. Anna Brunstrom of Karlstad University for making possible my research visit to work on failover related part of this dissertation. I really appreciate the valuable comments of Prof. Anna Brunstrom and Dr. Johan Garcia, which helped me better understand the SCTP failover mechanism that provided the basic benchmark for this work. Especially, I would like to thank Dr. Johan Garcia, for the time devoted to prepare an user-friendly taxonomy of SCTP-related research. I believe that the time spent on this work, our long discussions and intense brainstorming, helped me develop as a researcher.

I would like to thank all people that were involved in the Newcom project (Network of Excellence, ref. FP6-IST-507325), from which my work on SCTP protocol started, and especially I would like to thank Dr. Giulio Galante, for discussing the ideas that form now parts of this dissertation.

I would like to thank Prof. Paul Amer of University of Delaware for helping to organize my research visit to Protocol Engineering Lab (PEL) to work on the CMT realted issues. During my visit I meet and worked with people in the PEL labs and the Computer and Information Science (CIS) department, whom I would like to thank: Preethi Natarajan, Ilknur Aydin, Jon Leighton, Nasif Ekiz, Ertugrul Yilmaz, and visiting research fellows: Dr. Janardhan Iyengar, Dr. Armando Caro, and Randall Stewart, for valuable comments on my work.

The most important part of my non-technical support is Cristina, whom I would like to thank for an incredible patience, especially for putting up with my irregular working habits and including my consistently predictable, "I'm sorry I haven't left yet", each evening. I am also grateful to her for showing genuine interest in my field.

Last but not least, I am deeply thankful to my family for putting up with me, despite the long distance that separated us. I could not ask for more than to have them.

Abstract

Next generation mobile data networks are expected to achieve a high degree of inter-networking so that the mobile users can truly experience seamless access to their services, irrespective of the radio technology being used. In such scenarios, IP networking is becoming the keystone capable to turn this vision into a reality. Hence, mobility management solutions for IP networks are expected to provide seamless mobility across multiple radio access options. Earlier works on the mobility management problem discussed various solutions, mainly in network and application layer of the ISO/OSI protocol stack. More recently, transport layer handover schemes emerged, and are currently receiving a notable attention in the research community, as they seem to match very well the basic paradigm of the IP networking, where intelligence is moved towards the edges of the network. Therefore, this dissertation investigates and evaluates the idea of handling mobility at the transport layer, using mobile Stream Control Transmission Protocol (mSCTP) as an example of a handover transport layer protocol.

To this end, (the first part of) this thesis provides the reader with a necessary background for IP mobility-related aspects, surveying detailedly the most popular of the existing solutions. Provided overview includes Mobile IP (MIP) and its most important derivatives to represent the network-layer-based schemes, as well as Session Initiation Protocol (SIP) as an example of an application-layer approach. The details of the most important transport layer solutions are given on continuation, along with the motivation for the development of such mobility management schemes. Among presented transport-layer approaches, the one based on the mSCTP is chosen as a representative for the analysis performed in this dissertation. This choice is additionally motivated by two interesting features that SCTP protocol introduces, and that are interesting in the context of handover applications: multihoming and multistreaming (to some extent).

(Still in the introductory part) a detailed state-of-the-art of the SCTP protocol is provided, stressing its signaling background and original scope of use that did not consider mobility related application. The described transition from the signaling to a general purpose transport protocol illustrates the dynamics of the development of this relatively recent proposal, and explains why SCTP is currently one of the most interesting innovative transport protocols.

The core of this dissertation outlines major mobility-related considerations in the context of future heterogeneous wireless networks, identifying all important handover scenarios, and specifying the most representative one to conduct the proposed analysis. Several transport-layer handover schemes based on SCTP are analyzed in the selected scenario. First of the discussed schemes, provided also as a reference model for evaluations presented in the following sections of this work, reuses the standard SCTP failover, a mechanism originally devised to increase protocol robustness.

Next, the details of transport-layer loadsharing are explained, to facilitate the introduction of the mSCTP-CMT-PF handover scheme, an essential improvement for transport layer mobility suggested by this work. The devised proposal incorporates one of the most popular loadsharing schemes provided for SCTP, the Concurrent Multipath Transfer (CMT), that originally does not target wireless networks. Evaluation exposes the main challenges of such a design, pointing out the most important constraints limiting its scope of application.

Finally, a quantitative comparison of all identified mSCTP-based handover schemes and two of the most representative network-layer solutions is given in a series of analysis that involves mobility models of different grade of complexity.

Apart from the analysis of the mobility management aspects, this dissertation reports also on the state-of-the-art in SCTP modeling, very important in the context of further protocol development.

Chapter 1

Introduction

Recently, the rapid evolution and successful deployment of various emerging wireless technologies, e.g., IEEE 802.11 a/b/g, WiMax, etc., has pushed into a strong demand to integrate numerous wireless local area networks (WLANs) with the existing cellular network infrastructure. The typical example involves the integration of WLAN with Global System for Mobile Communications/General Packet Radio Service (GSM/GPRS), third-generation (3G) Universal Mobile Telecommunications System (UMTS), or cdma2000 networks. The inter-working of such heterogeneous, packet-based radio access networks (RANs), also referred to as *next generation* or *beyond 3G (B3G)* mobile data networks, poses many technical challenges [Hui and Yeung, 2003], with *mobility management* that can guarantee service continuity and IP connectivity provision for wireless multi-mode mobile terminals like cellular phones, personal digital assistants (PDAs), and notebook computers, being one of the most important [Yabusaki et al., 2005]. Such understood mobility management requires the deployment of inter-system solutions that can keep users and service providers as much aside as possible from the complexity of inter-networking RANs. In this context, the development of mobility-management solutions over the Internet Protocol (IP) is a key aspect to achieve seamless mobility between heterogeneous wireless access networks, where all services are meant to be IP-based, in scenarios as the one presented in Fig. 1.1.

Earlier works on the mobility management problem in heterogeneous networks [Akyildiz et al., 2004; Eddy, 2004] discussed solutions in various layers of the International Organization for Standardization Open Systems Interconnection Basic Reference Model (ISO/OSI model) [Zimmermann, 1980] of the protocol stack where the mobility can be handled: application, transport, network and data-link layer, respectively. In terms of challenges present in heterogeneous networks, transport layer seems a feasible candidate to host seamless mobility management. Nevertheless, the vast majority of the new mobility-related proposals seem to follow most of the existing schemes and stick to mobility handled at the network layer. Is there really a need and possibility to change this trend?

1.1 Problem statement

In this work the idea of handling mobility management at the transport layer is surveyed to check whether it can offer a viable solution for implementing seamless handover in heterogeneous wireless access networks. Since the mobile Stream Control Transmission Protocol (mSCTP) is at the core of most relevant transport-layer mobility schemes being currently studied, the key scenarios where the protocol can effectively leverage one of its new features, *multihoming*, to enhance handover support are identified. Moreover, to give the reader a complete overview of the mSCTP's application area, the presented dissertation will examine the situations where the use of mSCTP-based schemes is not possible, or incurs some limitations. Hence, the main goal of this thesis is to provide insights on development of a mSCTP-based mobility management scheme. In particular this work addresses the most important challenges of such a design: open issues related to both path management and path-transition optimization process. In a basic approach used as a benchmark for all presented

designs, SCTP failover mechanism, originally supplied to improve the robustness of SCTP protocol, is reused to trigger the handover. In an effort to provide more effective path management scheme, support from the link layer is considered, leading to a broad scope of available handover policies. To draw the reader the range of possible improvements, the theoretically worst and best cases are analyzed. Proposed path-transition optimization incorporates concurrent multipath transfer (CMT), a loadsharing, scheme that in its initial design was not aimed for wireless scenarios. This novel idea, introduced and developed in this work, composes also a future research direction proposed for mobility-related research community. Finally, all mSCTP-based schemes presented in this dissertation are related to the most common existing mobility solutions, based on the Mobile IP (MIP) [Perkins, 2002] and its derivatives.

The structure of the dissertation is illustrated in Fig. 1.2. First, in Chapter 2, all basic definitions and naming convention related to mobility management are given, followed by a detailed survey of existing mobility management solutions for IP-based networks. Presented review analyzes different approaches for mobility management in link, network, transport and application layer, accordingly. Chapter 3 provides the reader with the insights of the SCTP protocol, its new features, state-of-the-art, and relates that to the two most common transport layer protocols nowadays, namely Transport Control Protocol (TCP) [Postel, 1981] and User Datagram Protocol (UDP) [Postel, 1980]. One of the two new features of SCTP, multihoming, constitutes the essence of this dissertation in terms of extending its original scope of use, limited only to increasing protocol robustness, to handover scenarios, as well as providing loadsharing at the transport-layer. Thanks to Dynamic Address Reconfiguration (DAR) protocol extension described further in Chapter 3, SCTP multihoming can be applied to mobility scenarios; such an upgraded configuration is called mSCTP. Consequences of applying mSCTP to handover scenarios are explained in the analysis provided in Section 3.3, which aims to sketch the possible application scope for such a scheme. Next, in Chapter 4, a detailed design of a basic handover scheme based on mSCTP is specified. Failover mechanism provided originally to increase protocol's robustness is reused to address handover triggering. Detailed evaluation of such a failover-based scheme serves to establish a benchmark for more advanced solutions proposed in this work. Chapter 5 suggests the idea of introducing the CMT loadsharing scheme into mobility schemes based on mSCTP (mSCTP-CMT) in order to smooth the transition process, as well as increase the overall throughput of a proposed handover solution. The mSCTP-CMT

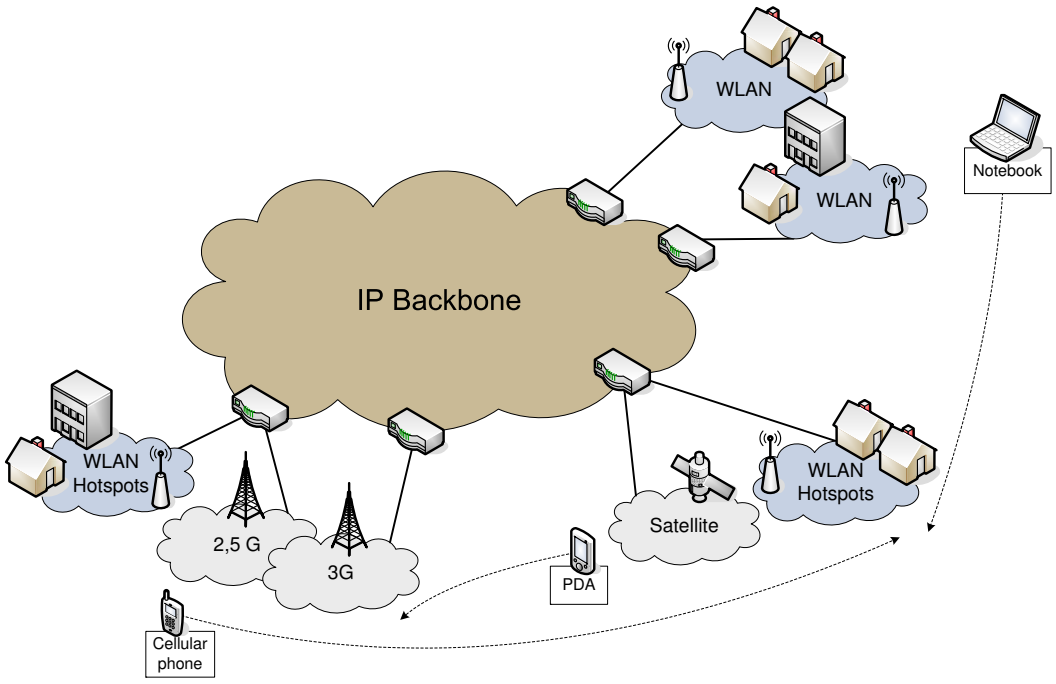


Figure 1.1: Heterogeneous RANs scenario.

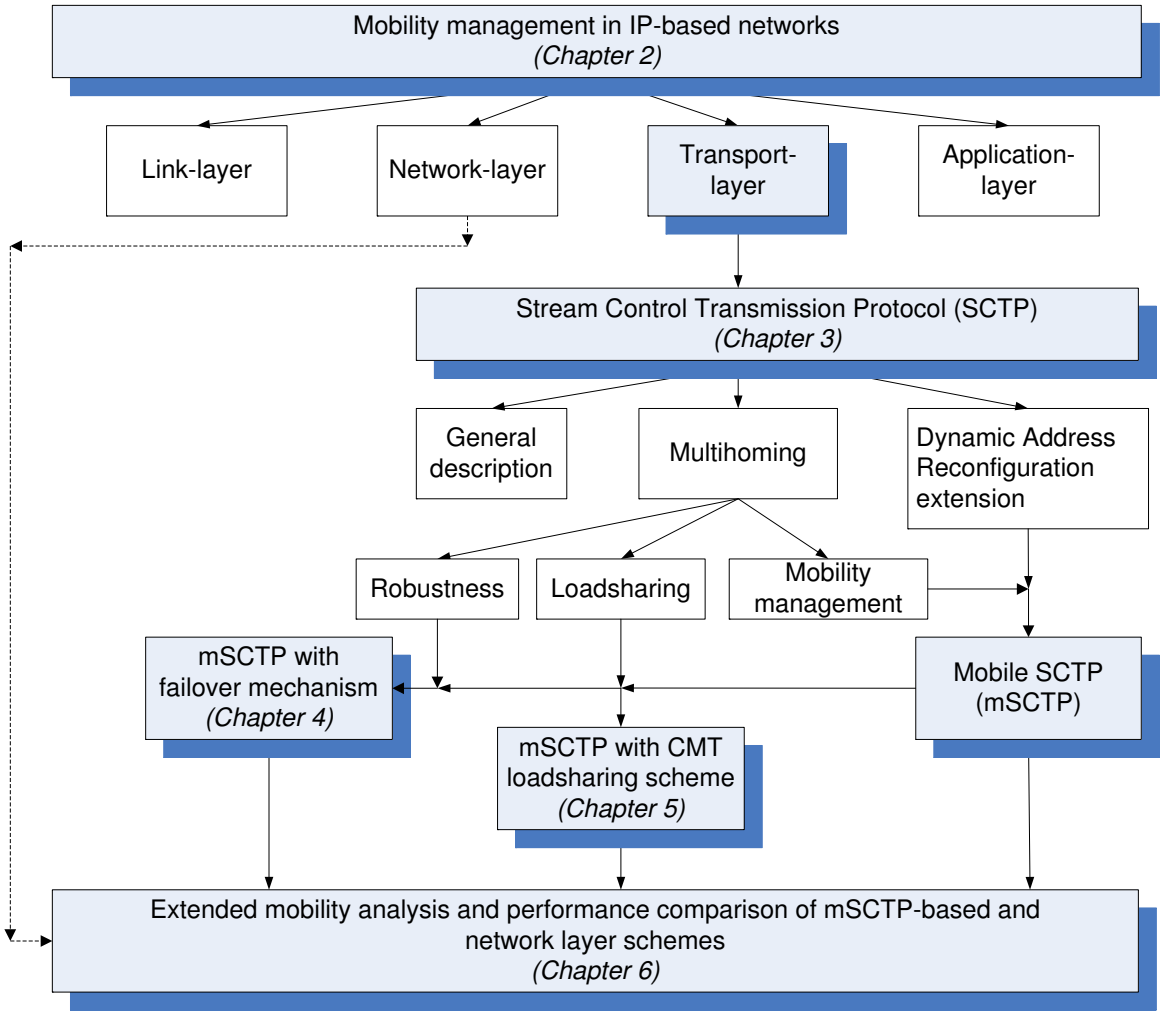


Figure 1.2: Dissertation scheme.

scheme is compared to a failover-based scheme as well as mSCTP with the handover mechanism triggered by the information from the link-layer. Further analysis, presented in Chapter 6, compares all mentioned schemes with network-layer-based solutions in a more complex scenario that captures additional mobility aspects. Final conclusions, as well as possible future work are drawn in Chapter 7. Finally, Appendix A provides the reader with more information on the ns-2 network simulator [NS-2] models used to evaluate the proposed handover schemes.

1.2 Motivation

The most common network layer scheme, e.g. Mobile IP, has several drawbacks, i.e., additional infrastructure requirements, or significant signaling overhead that endorse looking for a new proposal to handle mobility in heterogeneous network scenarios. Other mobility schemes managed at the network layer can only diminish some of these drawbacks, e.g. decrease handover latency or reduce necessary infrastructure modifications, and therefore it is worth checking solutions in higher layers of the ISO/OSI protocol stack. Analyzing the requirements for seamless mobility in heterogeneous wireless scenarios, transport-layer-based schemes seem to be the closest to the desired solution. The key idea beneath the transport-layer proposals is to handle mobility on an end-to-end basis, while keeping the underlying network infrastructure unchanged. Additionally, mobility handled at the transport layer enjoys several advantages, such as inherent route optimization (triangular routes

never occur), or the possibility of smooth handovers if the mobile node has multiple interfaces, to mention the most important characteristics. Author believes that this topic has still not been given enough attention in the research community, and therefore proposes in this work to analyze one of the existing transport-layer mobility proposals in more details.

It is also essential to point out the main inconvenience of some of the transport-layer mobility proposals is indirectly caused by the dominant role of well-established transport-layer protocols, like TCP and UDP that share nearly all Internet traffic nowadays. Therefore, transport-layer schemes not deriving from TCP require significant modifications of pre-existing protocol stacks. Yet, there are several interesting proposals of new, innovative transport-layer protocols, brought recently by the Internet Engineering Task Force (IETF) Transport Area (TSVWG) working group [IETF TSVWG] that still lack broader application, and thus attract the attention of the research community. One of them is SCTP, which although originally not aimed to deal with transport layer mobility, can be seen as a promising alternative to face the requirements of mobility management in heterogeneous wireless network scenarios that TCP is not able to deal with. SCTP, being a fairly new protocol, has still to become better recognized and wider-spread transport protocol, and this particular application may help to achieve that.

Last but not least, when introducing a new protocol, a relatively big effort should be spent not only on its initial design and implementation, but also on making the proposal available for the research community to evaluate, and perform further development. SCTP has already made its way to most of the systems stacks, traffic analyzers and simulation tools (e.g., ns-2 [SCTP-ns2] or Qualnet [SCTP-Qualnet]), making itself widely-available for further evaluation. In this dissertation, the ns-2 simulator [NS-2] will be used as the main tool to conduct the performance analysis when evaluating proposed SCTP-based mobility solutions.

Chapter 2

Mobility management

Mobility management is a fundamental piece of a B3G mobile data network architecture. In this context, an open challenge is the design of solutions that can take full advantage of different IP-based technologies to support the desired mobility of multi-mode terminals, and at the same time provide the necessary Quality of Service (QoS) guarantees. The task becomes even more challenging, taking into account the fact that the suite of TCP/IP protocols was proposed under the assumption that most of the devices are stationary, thus not particularly with the mobility aspects in mind. So far numerous solutions addressing mobility in IP networks have been presented, but the main question still remains open: what layer is the most appropriate for handling the mobility?

2.1 Definitions

Before going into details of the existing mobility solutions, there is a need to provide a set of basic definitions and mobility related terminology. Naming convention used in this dissertation follows the IETF naming convention defined in [Manner and Kojo, 2004].

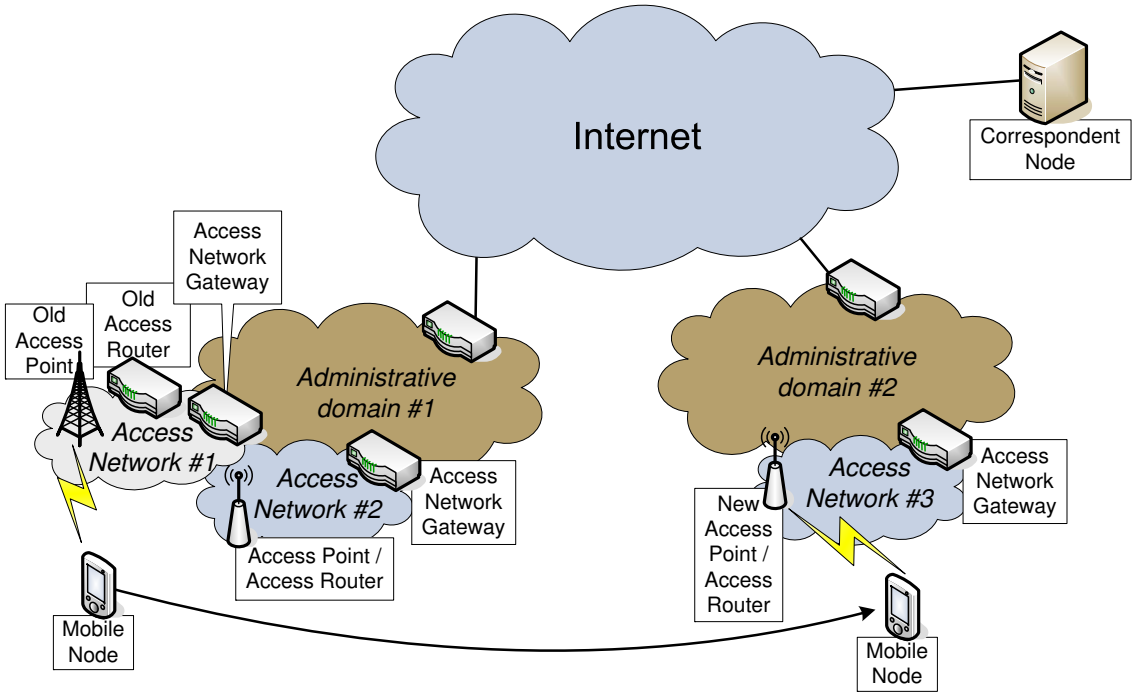


Figure 2.1: Mobility scenario.

Basic terms. First, the following basic terms should be explained and clarified (Fig. 2.1):

- *Point of attachment (PoA)*, the network side endpoint of a layer two link that includes a mobile node as the other endpoint.
- *Correspondent Node (CN)*, is an IP node with which a mobile node is communicating. It is assumed in this work that the CN is a fixed node, i.e., it is not changing its PoA and its IP address.
- *Mobile node (MN)*, is an IP node capable of changing its PoA. A MN may have one or more wireless interfaces, and it either sends and receives packets (so called mobile host (MH)) or just forwards the traffic as in case of mobile Router (MR). In this dissertation the term MN will be used to denote a multi-mode wireless terminal, e.g., a cellular phone, PDA, notebook, etc., however its functionality it is just that of the MH.
- *Access Point (AP)*, sometimes called base station (BS) or access point transceiver depending on technology, is a layer-two device offering wireless link connection to MNs.
- *Access router (AR)*, refers to an IP router that reside on the edge of an access network offering IP connectivity to MNs, and acting as default router for the MNs it is currently serving. Usually, each AR is connected to one or more APs. In case of WLAN networks it is common to find devices with co-located AP and AR functionalities, referred to as *wireless routers*. In this work, for simplicity reasons in each of the access networks considered, an AP will be co-located with its corresponding AR.
- *Old Access Router (OAR)*, is an AR that offered connectivity to the MN before a handover.
- *New Access Router (NAR)*, is an AR that offers connectivity to the MN after a handover.
- *Access Network Gateway (ANG)*, is a router that separates an access network from other IP networks. In a small AN, an ANG may also offer the services of an AR (be the same physical node).
- *Access Network (AN)*, is an IP network which includes one or more access network routers (ANGs, ARs, and optionally other internal access network routers).
- *Administrative Domain (AD)*, defines a collection of networks under the same administrative control and grouped together for administrative purposes.

Mobility management principles. The current addressing scheme of the Internet is a consequence of the design decisions made at the early days, when the Internet was merely just a static network with all the hosts connected through one specific interface. At the time that *address-oriented approach* was developed some issues were considered invariant: (1) addresses were thought to be stationary (non-mobile), (2) an address received was the one sent (no tunneling mechanisms), (3) source and destination were reversible, and (4) all hosts knew the address to which the packets should be sent to, to reach the wanted host. Since then, Internet underwent many revolutionary changes, with one of the most important being the introduction of wireless interfaces. The need to support the increasing number of wireless hosts has led to a problem that the address-oriented architecture is unable to deal with, *mobility*. Changes proposed to accommodate mobility are pushing the Internet to a *host-oriented approach* which separates the concept of the address and the unique device identifier. The addresses, being location-dependent, make the previously constant name-to-address binding change over time as the host address changes with the host mobility. Traditionally, the following aspects of mobility can be distinguished:

- *Terminal mobility* is the ability of a MN to move between IP subnets within an AD or between different ADs, while continuing to be reachable for incoming requests and maintaining sessions across subnet changes.
- *Personal mobility* describes the ability of addressing a user that can be located at several terminals.

- *Session mobility* refers to maintaining (and transferring) a session when a user moves between terminals.
- *Service mobility* can be defined as the ability of users to maintain access to their services even when moving and changing terminals or service providers.

Hereafter the main focus is put on terminal mobility, since it is the foundation of the analysis addressed in this work, and consequently the word mobility used in this dissertation will refer exclusively to terminal mobility. Management of terminal mobility includes two fundamental operations: location and handover management. According to Riegel and Tuexen [2007], *handover management* deals with all the necessary operations to change a MN's PoA to the IP network, while maintaining the communication with the CN. On the other hand, *location management* focuses on keeping track of a MN's current IP address, and providing this address to any entity needing to communicate with the MN, while being transparent to its peers. Additionally, mobility management poses several performance and deployment challenges. The most important performance indicators are handover latency (time between the reception of the last packet in the old network and arrival of the first packet in the new network), packet loss, signaling overhead and throughput. Meanwhile, the deployment requirements for mobility management focus on application transparency (minimum changes possible to the current applications and services), and simplicity of integration with the existing infrastructure (changes, if any, should be as simple as possible, and adding third-party devices should be avoided).

Terminal mobility of the MN can be addressed in different ways, depending on the scope, performance characteristics, control modes of handover techniques, etc. When classifying mobility as a function of its scope, the following categories can be named [Giaretta, 2009]:

- *Local mobility* includes movements within an AN, e.g., intra-AN handover (change of the AR within the same AN), or just a change of the AR's network interface to the MN affecting the routing path of the IP packets. Local mobility may also refer to a movement across different subnets belonging to the same AN.
- *Global mobility* covers movements across different ANs, or even ADs in various geographical regions, as shown in Fig. 2.1.

Handover naming convention. Taxonomies provided so far in the literature present different approaches for classifying various handover types, e.g., [Dutta et al., 2008]. Here, also a sample classification of different handover types is given that takes into account the following aspects (provided list is orthogonal, so that each handover could be classified with one of the features in each group):

- technology of the APs, between which the handover is made: either involving the same technology (*intra-technology or horizontal handover*) or different technologies (*inter-technology or vertical handover*). The difference between horizontal and vertical handovers is not always clear, e.g., a handover from an 802.11b WLAN AP to a 802.11g WLAN AP can be interpreted either way.

- entity that makes the initial handover decision: mobile- and network-initiated handover.

- entity that has the primary control over the handover process: mobile- and network- controlled handover.

- entity that provides information where to handover to: mobile-, network-assisted and unassisted handover.

- which of the ARs initiates the handover: either OAR or MN via OAR (push handover), or NAR or MN via NAR (pull handover).

- whether the handover is expected and some handover-related signaling can be done in advance (planned or proactive handover) or not (unexpected or reactive handover).

- as a function of performance aspects handover can either aim at: minimizing packet losses not dealing with the additional delays in packet forwarding (*smooth handover*), minimizing

handover latency not dealing with losses (*fast handover*), not provoking change in service capability, security, or quality (*seamless handover*).

2.2 Approaches to mobility management

Many proposals aspiring to solve the problem of terminal mobility management in heterogeneous wireless networks providing IP connectivity can be found in the literature. A good survey on the current state of the art for mobility management in next-generation all-IP-based wireless systems can be found in [Akyildiz et al., 2004]. Aiming at not repeating this work here, this dissertation will examine the most important schemes in network, transport and application layer. Also a short reference to the sub-network-layer solutions will be given.

2.2.1 Sub-network layer mobility issues

Handling terminal mobility below the network layer, which in IP networks provides globally usable addresses, poses, especially in heterogeneous environments, several serious challenges that need to be solved. The main limitation is introduced by the fact that the IP address can not be reconfigured from the underlying link layer, and thus the scope of application of sub-network layer mobility solutions is limited to the same subnet, i.e., does not include a change of the IP address. Therefore, proposed solutions deal mainly with the dynamic update of the MAC switching tables. Nevertheless, this type of solution can be applied to the heterogeneous networks, as long as the MN stays within the same subnet. Akyildiz et al. [2004] provide a detailed overview of link-layer mobility solutions in the current all-IP-based wireless systems. Due to the limited scalability, the discussed solutions will not be addressed more in this work.

2.2.2 Network-layer schemes

Network layer, originally proposed to handle global addressing and routing, seems a natural candidate to host the mobility management [Bhagwat et al., 1996]. Maintaining both functions and providing support for mobility can be done twofold, either (1) routers will be required to use host-specific route information, updated as each host moves, or by (2) providing a hierarchical addressing structure with its use limited to the domain of its definition, and extended by dedicated indirection agents forwarding all traffic to a host staying beyond the given domain. The first approach can be easily discarded due to the scalability problems, given the large number of Internet hosts nowadays, whilst the latter option stays feasible and aligned with the current Internet routing structure.

In that sense the standard proposed by the IETF, Mobile IPv4 (MIPv4, or just MIP) [Perkins, 1997, 2002], is one of the most common approaches to support mobility on the Internet. MIP, illustrated in Fig. 2.2, preserves the IP address originally assigned to the MN in its home network (HN), so called *home address*, as an unique MN's identifier, to ensure application transparency. As long as the MN stays in the HN (1), it is treated as any other fixed node of that network, thus not requiring any kind of mobility support. Whenever the MN moves out of the HN and gains the access to a foreign network (FN) (2), it obtains a *care-of address (CoA)*. The CoA can be acquired either from agent advertisements sent by a *foreign agent (FA)* (a so-called foreign agent CoA; this is the preferred method and all further considerations presented here scope around this option) or by some external assignment mechanism such as DHCP (a co-located CoA; the FA functionality is not needed). The CoA serves to capture the location of the MN in the FN, and such a location update must be communicated by sending a registration request message to a dedicated entity in the HN called *home agent (HA)* (3). The HA maintains an up-to-date list of the mobility bindings (i.e., pairs of MN's home address and its current CoA) and confirms any recently made change with a registration reply message sent to the MN. An important security consideration is that both registration messages (from MN and from HA) must be authenticated to prevent packet hijacking. Since then, the HA intercepts any packet arriving at the HN (5), e.g., using Proxy Address Resolution Protocol (ARP) [Plummer, 1982; Postel, 1984], and forwards it to the MN at its current CoA using *IP tunneling*. The IP encapsulation is removed at the FA (6), which then delivers the packet to the MN (7).

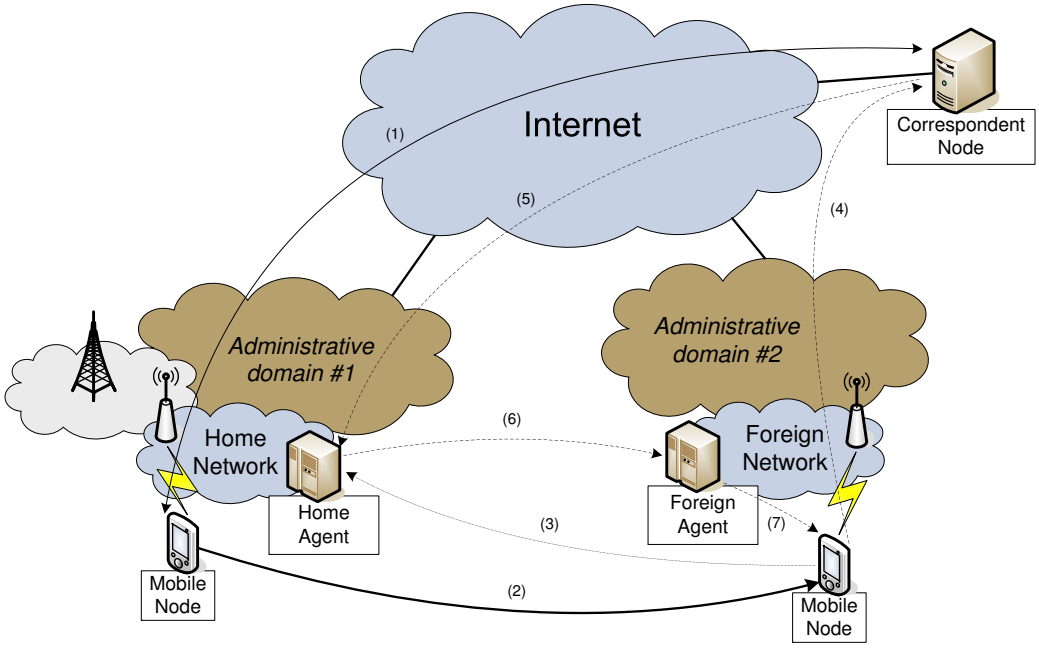


Figure 2.2: Mobile IPv4 architecture and operations.

In the opposite direction, the MN can directly contact the CN (4), using its home address and the FA acting as the default router. This asymmetry is named the *triangular routing*. In case there are additional security constraints posed by the FN, e.g., router's ingress filtering, the packets from the MN to the CN must also be routed via the HA, using so-called *reverse tunneling* [Montenegro, 2001].

Triangular routing can be diminished with the route optimization extension (MIP-RO) [Johnson and Perkins, 2001; Perkins and Johnson, 1998]. MIP-RO requires the MN to register its current binding at the CN. Then, the packets can be routed directly from the CN to the CoA, and the triangular route via the HA is avoided. However, triangular routing is not the only of the MIP's problems, fully discussed by Perkins et al. [2007]. MIP registration process takes a long time, increasing therefore handover latency. While registering in the FN, the MN is not able to send or receive packets until the registration at the HA is completed, what may induce a considerable packet loss. In addition, all agent advertisement and registration messages provoke a significant network overload. MIP scheme is usually classified as a global mobility scheme, and is tailored to follow a MN's movement across different subnets within an AD, or across different subnets belonging to different ADs. Yet, if the MN's CoA is changed frequently, the MIP tunneling mechanism may lead to an unacceptable network overhead, especially in terms of signaling. To address these issues, several schemes have been proposed so far: Hawaii [Ramjee et al., 1999], Cellular IP (CIP) [Campbell et al., 1999], as well extensions to MIP: Hierarchical Mobile IP (HMIP) [Fogelstroem et al., 2007], Low-latency Mobile IP [El Malki, 2007], and Fast Handovers for Mobile IP (FMIP) [Koodli and Perkins, 2007], to mention the most important ones. Additionally, a detailed discussion of different mobility protocols can be found in [Campbell et al., 2002].

Because of various limitations posed by IPv4, especially a limited 32-bit address space, the IETF has proposed a new network layer protocol IPv6 [Deering and Hinden, 1998], to replace the IPv4. Mobility support in IPv6 is provided by Mobile IPv6 (MIPv6) [Johnson et al., 2004] that follows the basic concepts of MIPv4. The main difference is the lack of the FA entity, as the MIPv6 protocol can operate in any location without the support of local routers. Instead of IP encapsulation the IPv6 headers are used to forward the packets to the MN that is away from its HN. Route optimization is included as a part of the MIPv6 protocol [Nikander et al., 2005], not as a separate extension as in case of MIP-RO, and can coexist with routers that perform ingress filtering. To increase protocol robustness the MIPv6 uses the IPv6 Neighbor Discovery Protocol (instead of ARP) what makes the MIPv6 solution independent of any particular link layer.

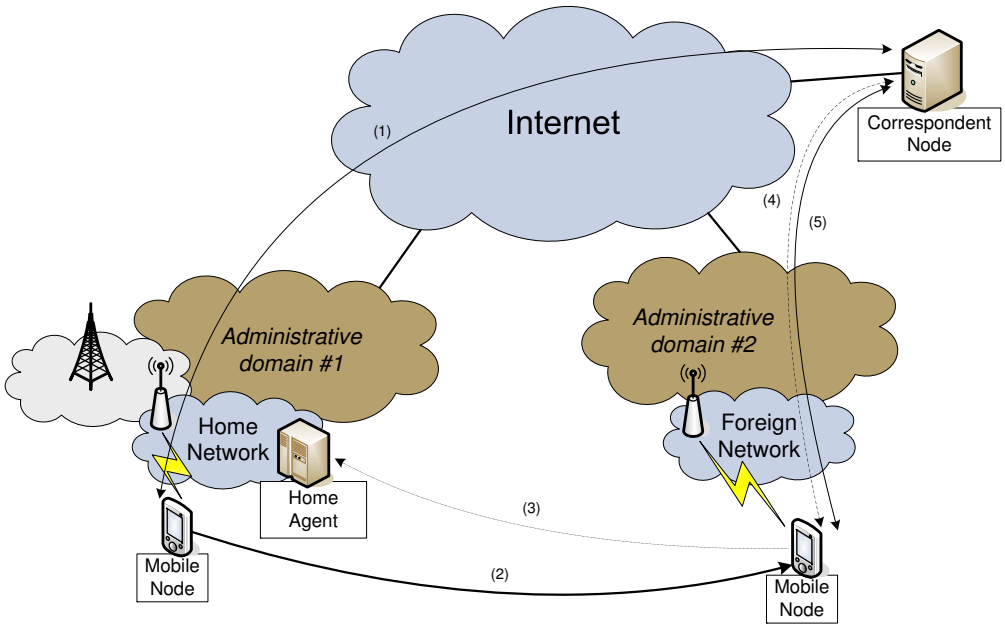


Figure 2.3: Mobile IPv6 architecture and operations.

The functionality of the MIPv6 is presented in Fig. 2.3. Again, no mobility support is needed as long as the MN stays in the HN (1). Once the MN moves out from the HN to a FN (2), it can obtain its CoA via either stateful (as in case of MIP), i.e., DHCP for IPv6 (DHCPv6) [Droms et al., 2003] or stateless [Thomson and Narten, 1998] address auto-configuration procedure. Newly obtained CoA must be registered at HA (3) and CN (4) using *binding update* messages. Both HA and CN must maintain the list of the current MN's bindings. As soon as the MN's bindings are updated, the packets can be routed directly from the CN to the MN's CoA and similarly in the opposite direction, so that the triangular routing is avoided (5). In case the communication between CN and MN is established when the latter is already in the FN, the first packets from the CN are tunneled via the HA to the CoA, like in MIP, until the binding update process is completed.

MIPv6 improves multiple aspects of MIP, such as inherent mobility, security and route optimization, but also preserves its most important disadvantage, having HA as a single point of failure. Moreover, binding update messages used in MIPv6 can provoke an additional overhead in case of the increased MN's mobility that results in often changes of the AN. To reduce the signaling and fasten the handover for movements within the same AD, the Hierarchical Mobile IPv6 (HMIPv6) [Soliman et al., 2005] local mobility scheme have been proposed. HMIPv6 introduces a new entity, Mobile Anchor Point (MAP) that acts as a local HA in a FN and handles MN's local mobility hiding it to the nodes from outside of the FN. Another proposal aiming at reducing handover latency and packet loss during the handover is called Fast Handovers for Mobile IPv6 (FMIPv6) [Koodli, 2008]. FMIPv6 attempts to make the handovers proactive if it is possible to obtain information about the candidates for the NAR from the co-operating ARs before disconnecting from the OAR. A combination of HMIPv6 and FMIPv6 called Fast Handovers for HMIPv6 (F-HMIPv6) [Jung et al., 2004] aims at aggregating the advantages of both schemes, and additionally reduce the signaling overhead. Still, scalability and complexity of proposed solutions are the most important concerns for all three described MIPv6 extensions. More recently, network-based IP mobility solutions where the terminal is not directly involved in managing IP mobility (e.g., Proxy MIPv6 (PMIPv6) [Giaretta, 2009; Kong et al., 2008]) are also being introduced in wireless networks. PIMPv6 provides a solution for local mobility without requiring the MN to participate in any mobility related signaling.

Similar to PIMPv6, another interesting approach to mobility management at the network layer is introduced by Yabusaki et al. [2005]. The proposed solution advocates for the network itself to transparently handle mobility for mobile terminals. Thus, Yabusaki et al. suggest a network-centric

solution to handle IP mobility in analogy with conventional 2G/3G networks, where mobility management has mainly been implemented as *network intelligence*, a concept just opposite to the *end-to-end intelligence* architectural principle of the Internet [Bush and Meyer, 2002; Carpenter, 1996]. In this approach, IP addresses are used separately as host addresses and routing addresses. Thus, a *host address* is semi-permanently assigned to a MN and a *routing address* is temporarily assigned to the MN when datagrams are delivered to it. Datagrams are sent from a CN to a MN with the host address of the MN but then, within the IP mobile network, datagrams are transported using the routing address generated from the host address. All in all, user terminals are unaware of this rerouting management that is handled entirely in the network.

For more information and proposals for handling mobility at the network layer readers can refer to [Campbell et al., 2002; Saha et al., 2004], here Table 2.1 summarizes the most important aspects of the discussed network layer schemes. Values specified for latency and signaling overhead provide a relative comparison among discussed schemes.

2.2.3 Transport-layer schemes

Unlike network-layer schemes such as MIP, which make mobility transparent to upper layers by increasing the burden and responsibility of the Internet infrastructure, transport-layer schemes are based on an *end-to-end approach* to mobility that attempts to keep the Internet infrastructure unchanged by allowing the end-hosts to take care of mobility. This approach is gaining an increasing attention in the recent years, also because transport-layer-based schemes enjoy several advantages such as inherent route optimization (triangular routes never occur), no dependence on the concept of HN or additional infrastructure beyond DHCP and Domain Name System (DNS) [Mockapetris, 1987], and more precisely Dynamic DNS (DDNS) [Vixie et al., 1997], as well as either the possibility of smooth handovers if the MN has multiple interfaces, or the ability to pause transmission during mobility-induced temporary disconnections [Eddy, 2004]. It is also essential to point out that the main inconvenience is caused by the dominant role of well-established transport-layer protocols, like TCP and UDP, which were not targeted for the wireless scenarios. Consequently, most of the proposed transport-layer schemes focuses on improving the performance of the TCP in the wireless networks and providing TCP support for the mobility. In contrast, proposals based on the new innovative transport-layer protocols, like SCTP or Datagram Congestion Control Protocol (DCCP) [Kohler et al., 2006], require significant modifications of the current protocol stacks.

According to Riegel and Tuexen [2007], *transport-layer mobility* is handled by the transport layers of the connection endpoints so that it is transparent to application-layer protocols not using IP addresses in their messages. A *mobility-enabled transport protocol* supports an IP-address change on the underlying network layer, while keeping the end-to-end connection alive. E.g., a possible way to achieve that is as follows: the MN first obtains a new IP address, then tells the CN (using the established transport-layer connection) that it is now reachable by the new IP address, and then handover can be performed. So far, several proposals to handle mobility at the transport layer have been developed. A complete survey and classification of mobility management schemes at the transport layer can be found in [Atiquzzaman and Reaz, 2005]. In particular, the mentioned classification distinguishes between:

- *Connection-migration protocols*, provide a migration scheme for the connection, once the new IP address acquired, and before the old IP address is retained. Connection-migration schemes involve also a notification to the CN about the change. Most typically, during the migration process the connection is stopped or put under wait for the time the migration is performed, in order to reduce packet loss in the presence of long and frequent disconnections throughout the handover process. An example of connection-migration protocol is Freeze-TCP [Goff et al., 2000] that enters the persist mode, on the perceived disconnection or handover event, indicated by a MN with zero window advertisements (ZWAs). Once in the persistent operation the CN sends zero window probes (ZWPs) to check the availability of the MN, and on the reception of a positive response immediately starts sending data.
- *Gateway-based mobility schemes*, as, e.g., the Mobile Socket Service (MSOCKS) scheme [Maltz and Bhagwat, 1998], introduce a dedicated gateway in the network for handling the mobility. Gateway splits the connection between the CN and the MN, allowing the latter to change its

Table 2.1: Summary of network-layer mobility proposals.

Feature	MIP	HMIP	FMIP	CIP	Hawaii	MIPv6	HMIPv6	FMIPv6	F-HMIPv6
Mobility Management scope	Global	Local	Local Global	Local	Local	Global	Local	Local Global	Local Global
Location management	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes
Route optimization	Only with MIP-RO extension	No	No	No	No	Yes	Yes	Yes	Yes
Signaling overhead	High	Medium	High	Very Low	Low	High	Medium	High	Medium
Latency	High	High	Low	Low	Low	High	High	Low	Low

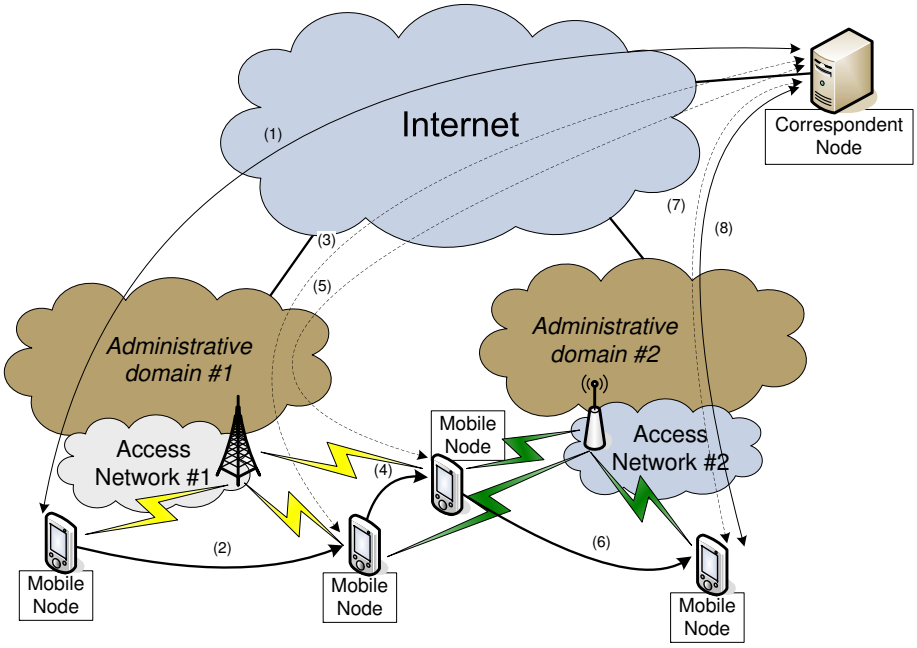


Figure 2.4: mSCTP architecture and operations.

connection with the gateway when performing the handover. These solutions, however, do not deal with location management, bring in a single point of failure and may decrease fault tolerance.

- *Handover protocols*, address only handover management issues, without considering location management. Protocols such as mSCTP or Mobile Multimedia Streaming Protocol (MMSP) [Matsuoka et al., 2003] usually offer a soft-handover solution, and aim at reducing handover-induced packet loss, providing scalability and fault tolerance.
- *Complete-mobility schemes*, also called *mobility managers*, like Migrate [Snoeren and Balakrishnan, 2000] and Seamless IP diversity-based Generalized Mobility Architecture (SIGMA) [Fu et al., 2005], include both handover and location management. Both solutions use DNS for location-management purposes, supporting either hard (Migrate) or soft handover (SIGMA). Such protocols only require modifications at the transport layer, thus leaving the existing network infrastructure unchanged.

Fig. 2.4 illustrates the basic operation of a handover process handled by mSCTP, as an example of transport-layer mobility scheme. When establishing an mSCTP association between MN and CN, both nodes exchange first the lists of the IP addresses valid for the communication (1). Only one of the source-destination pairs is selected to send the data to (a so-called primary path), whereas all remaining pairs serve only for backup purposes. As long as the MN stays in the area where the initially defined IP addresses are available there are no mobility-related concerns. If the MN moves to an area where a new IP address (an address that was not included in the initial list) can be obtained from a network (2), as soon as the new IP address becomes available it has to be communicated to the CN using specific control messages (address configuration chunks) (3). The security concern here is that the address manipulation creates an opportunity for hijacking attacks. To prevent hijacking attacks mSCTP specification recommends using IPsec or Transport Layer Security (TLS) [Stewart, Tuexen, and Camarillo, 2007]. Once the new IP address has been added to the association, the MN may decide whether still use the old IP address to send the data to, or in an appropriate moment (e.g., basing its decision on the signal measurement information obtained from the link-layer) (4) switch the primary path to the new IP address (5). After the address change, the transmission can continue uninterrupted on the new IP address (a smooth or even seamless handover). Leaving the old AN (6) the unnecessary IP address(es) can be removed (7), as further transmission goes on (8). Still, an open issue of the presented mobility scheme is lack

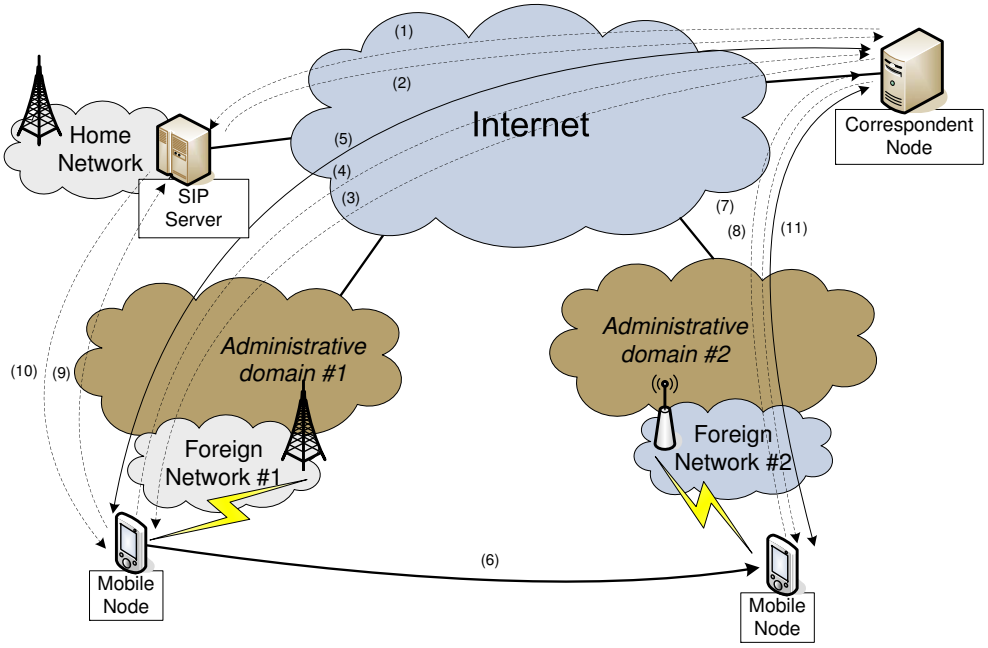


Figure 2.5: SIP architecture and operations.

of any support for the location management. As the location management can not be handled at the end hosts, a possible solution is either to reuse the MIP, as shown in [Koh and Xie, 2004; Noonan et al., 2002] or to use the DDNS.

2.2.4 Application-layer schemes

Handling mobility at the application layer has also received a lot of attention, since a solution that is almost independent of the underlying wireless or wired access technologies and network-layer elements can be envisaged. In this context, the most important proposal is Session Initiation Protocol (SIP) [Rosenberg et al., 2002] that although initially developed by the IETF as an application multimedia signaling protocol, can be also used for mobility management [Kim et al., 2004; Schulzrinne and Wedlund, 2000; Shacham et al., 2007; Wedlund and Schulzrinne, 1999]. SIP architecture introduces the following entities: user agent (UA), proxy server (PS), redirect server (RS) and registrar server (RGS). The UA identifies SIP messages coming from the user, and tracks SIP messages following the user actions, acting as both client and server. The PS relays SIP messages, the RS receives SIP messages and identifies the current location of the MN, whereas the RGS accepts *register* requests from the UA who has logged onto the network, and then places that information into the location service. The PS, RS and RGS functions are usually placed together in one entity called SIP Server (SS).

Fig. 2.5 explains the details of a mobility management scheme based on SIP. Firstly, a MN registers to the SS to provide reachability information. In order to initiate a SIP session with the MN, the CN consults the current MN's location with the SS (1,2) and then sends a *session invitation* message to the MN (3). Then, the regular SIP procedure to establish the session follows (4), refer [Rosenberg et al., 2002] for details. Once the session is established the data can be sent (5). When a MN during an active session moves into a different network (6), it first receives a new network address via DHCP, and then sends a new session invitation to the CN (7) with an updated session description containing the same call-id (session identifier) and the new IP address. After receiving new session invitation at the CN, subsequent data packets are forwarded to the MN (8) using either this new address (in case the underlying protocol is UDP), or using IP encapsulation (for SIP sessions over TCP) [Vakil et al., 2001]. Then, the MN updates its location information at the SS by sending a *register* message (9, 10), so the session can be correctly redirected (11). SIP mobility scheme is

Table 2.2: Comparison of different mobility management approaches.

Feature	Network layer	Transport layer	Application layer
Location management support	Included	Requires external location manager	Included
Route optimization	Binding update necessary	Not required	Not provided
Network support	Required	Not required	Not required
Seamless transition	Transport layer must deal with losses and path changes	Included	Not included
Security	Included	Included	Included
Required infrastructure changes	– hosts – specialized routers (HA, FA)	– hosts	– hosts – SIP servers

characterized by significant handover latency, due to signaling, and overhead caused by the IP encapsulation. Moreover, SIP by itself does not guarantee the maintenance of established TCP sessions or UDP port bindings when moving, so further extensions such as S-SIP [Zhang et al., 2007] are needed to provide seamless handover capabilities.

2.3 Conclusions

The most important approaches for handling mobility management presented in this chapter are summarized in Table 2.2. As discussed in some detail here, it has been noted that currently there is not a single mobility solution approach able to cover in a satisfactory manner all possible mobility management aspects. Network layer solutions require significant modifications to the existing infrastructure (providing single points of failure with the specialized routers they introduce, despite of the numerous efforts to introduce robustness in this matter, e.g., back-up HA, multihoming in MIP [Huang et al., 2008], etc.), as well as considerable support from the network (significant signaling overhead). In contrast, transport layer solutions do not provide location management service and to do so, are dependent on other layers protocols, such as DDNS or MIP. Meanwhile, application layer proposals are aiming at specific type of applications, e.g. SIP for real-time traffic and multimedia, and in this sense seem quite limited. All presented mobility management schemes can deal to some extent with the security issues, however not all the proposals have developed their security considerations with enough detail, e.g., the use of IPsec to prevent hijacking attacks with mSCTP has been recommended without specifying a detailed procedure.

A comprehensive discussion on the pros and cons of handling mobility management at different stack layers can be also found in [Eddy, 2004]. Eddy concludes that transport-layer mobility schemes best fit the requirements of today's IP-based services, and that there should be more inter-layer communication to avoid conflicts and inefficiencies. Following this conclusion this work will scope on the transport layer mobility, and in particular on mSCTP, as an example of a mobility-enabled transport protocol, and also because of its new, interesting feature, multihoming.

Chapter 3

SCTP for transport-layer mobility

The Stream Control Transmission Protocol (SCTP) was first announced in October 2000 in the, now an obsolete, RFC 2960 [Stewart et al., 2000]. Publication of the RFC 2960 concluded over two-year long IETF Signaling Transport (SIGTRAN) working group [IETF SIGTRAN] project on a new reliable protocol for transporting packet-based Public Switched Telephone Network (PSTN) signaling over IP networks. However, all began even before SIGTRAN group was formed late in 1998. A (protocol) proposal, called Multi-network Datagram Transmission Protocol (MDTP) [Stewart and Xie, 1998], developed independently from the IETF to overcome TCP weaknesses, had been submitted to the IETF editors in August 1998. Few months later when SIGTRAN group formed, MDTP was the only working implementation that met most of the requirements for its future-projected Common Transport Protocol (CTP), described in [Ong et al., 1999]. Namely, solution to head-of-line (HoL) blocking problem, multihoming feature and performance comparable to TCP made the MDTP a top pick for the CTP first draft. In nearly two years from that point CTP underwent many deep revisions, and so did the protocol name (CTP, CSTP, SCTP), before finally evolving to the SCTP in its RFC 2960 shape.

Already at the specification stage authors envisaged that SCTP capabilities would let extend its scope of use to a general transport protocol[1]. Indeed, the following years saw a growing range of possible applications of SCTP in many works discussing both signaling, and more general purposes [Coene, 2002; Stewart et al., 2004]. From 2001, the maintenance of the protocol has been tracked by the IETF Transport Area (TSVWG) working group [IETF TSVWG]. Since then, the original (RFC 2960) protocol specification was slightly modified (checksum change [Stone et al., 2002]), and updated with suggested implementer's fixes (specification errata and issues [Stewart et al., 2006]). Both updates are included in the current protocol specification RFC 4960 [Stewart, 2007], released in September 2007 that will be further referred in this work as *standard SCTP*.

3.1 SCTP overview

Standard SCTP provides a reliable, full-duplex connection with flow and congestion control algorithms that are derived from TCP, thus following the same Additive-Increase Multiplicative-Decrease (AIMD) behavior. An SCTP connection is called association, and is established using a four-way handshake (instead of a three-way handshake as in TCP) in order to improve protocol security and make it resistant to blind denial of service (DoS) attacks (such as flooding and masquerade). SCTP offers message abstraction to the application, in contrast to the byte stream abstraction provided by TCP, in order to better suit the communication pattern of signaling applications. What made SCTP a subject of considerable interest however, are two new features it introduces: multihoming and multistreaming.

[1] Originally SCTP was developed as signaling telephony transport protocol and that was reflected in the first MDTP drafts. However, already in fourth MDTP draft dating to April 1999, this limitation was removed from the specification.

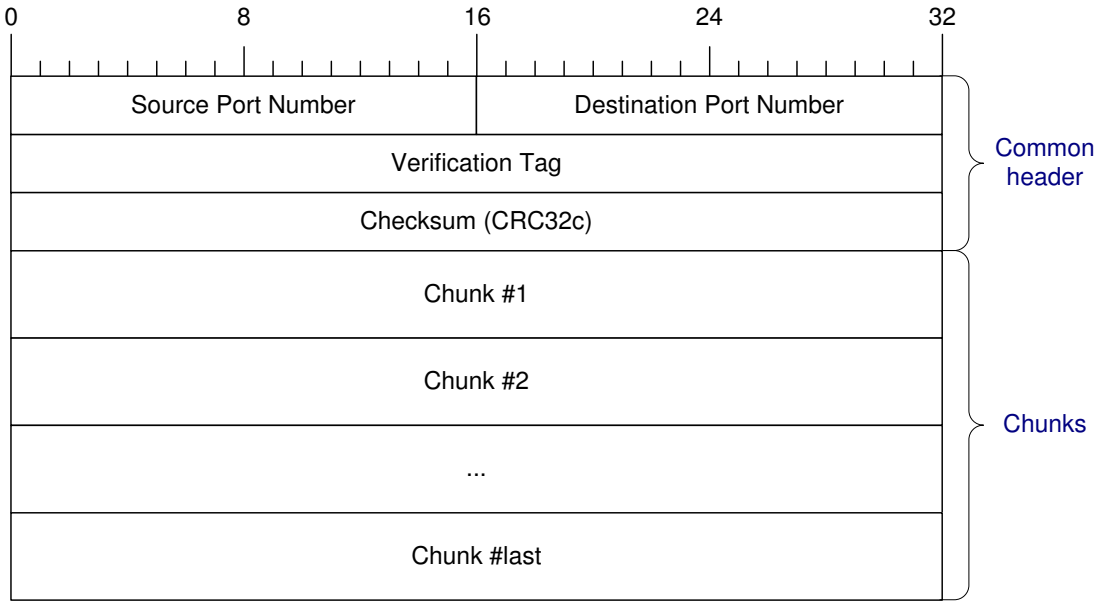


Figure 3.1: SCTP PDU structure.

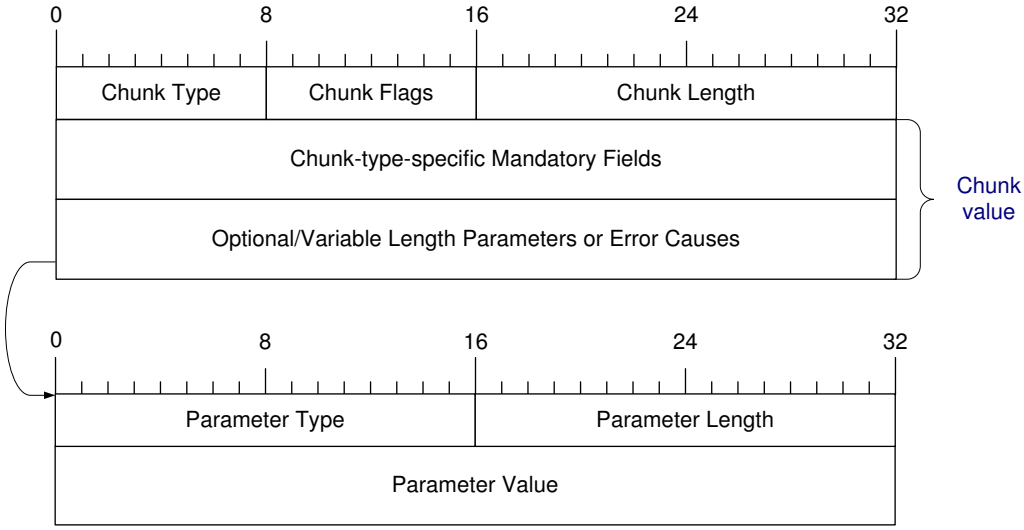


Figure 3.2: Chunk details.

3.1.1 Protocol basics

The SCTP protocol data unit (*SCTP-PDU*), also called simply *SCTP packet*, consist of a *SCTP common header*, and one or more *chunks*, as shown in Fig. 3.1 [Stewart, 2007]. SCTP common header contains: 16-bit source and destination port number fields (to identify the association which this SCTP-PDU belongs to and demultiplex it to the correct receiving application, respectively), 32-bit verification tag field (that serves to validate the sender), and a 32-bit checksum field (the CRC32c checksum value is calculated for the entire SCTP-PDU, with all zeros in the checksum field). A chunk carries either control or user data, and has Type-Length-Value (TLV) structure as shown in Fig. 3.2 [Stewart, 2007]. Chunk includes a 8-bit chunk type field that identifies the chunk type (refer Table 3.1 for a full list of currently defined chunk types, extending the standard list given in [Stewart, 2007]; the chunk ids are specified in a way that the highest bit defines SCTP-PDU processing action and the second highest bit handles error reporting on an unrecognized chunk type), a 8-bit chunk flag field (chunk-specific sets, if not specified all the bits are set to zeros), a 16-bit chunk length field

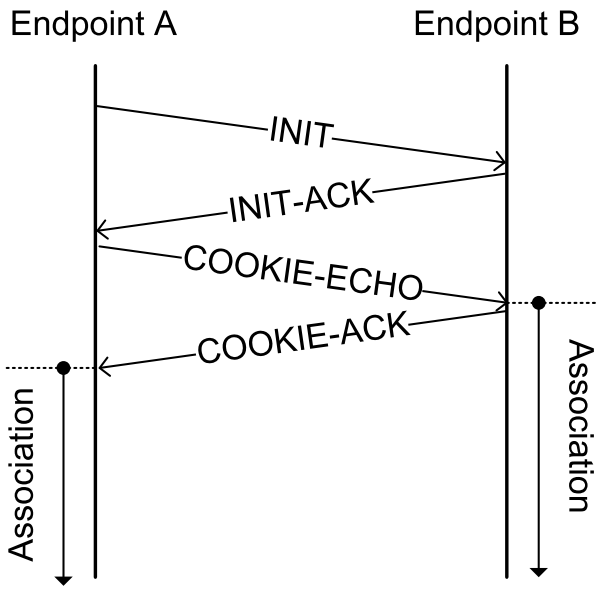
Table 3.1: List of chunk types (if not stated otherwise, chunks are defined within the standard SCTP).

ID Value	Chunk type
0x00	Payload data (DATA)
0x01	Initiation (INIT)
0x02	Initiation Acknowledgment (INIT ACK)
0x03	Selective Acknowledgment (SACK)
0x04	Heartbeat Request (HB)
0x05	Heartbeat Acknowledgment (HB ACK)
0x06	Abort (ABORT)
0x07	Shutdown (SHUTDOWN)
0x08	Shutdown Acknowledgment (SHUTDOWN ACK)
0x09	Operation Error (ERROR)
0x0A	State Cookie (COOKIE)
0x0B	Cookie Acknowledgment (COOKIE ACK)
0x0C	reserved for Explicit Congestion Notification (ECNE)
0x0D	reserved for Congestion Window Reduced (CWR)
0x0E	Shutdown Complete (SHUTDOWN COMPLETE)
0x0F	Authentication (AUTH) [a]
0x10	DDP Segment Chunk (DDP-SC) [b]
0x11	DDP Stream Session Control (DDP-SSC) [b]
0x80	Address Configuration Acknowledgment (ASCONF ACK) [c]
0x84	Padding (PADDING) [d]
0xC0	Forward Transmission Sequence Number (FORWARD TSN) [e]
0xC1	Address Configuration Change (ASCONF) [c]
0x3F, 0x7F 0xBF, 0xFF	reserved for IETF-defined chunk extensions
rest	future chunk definitions

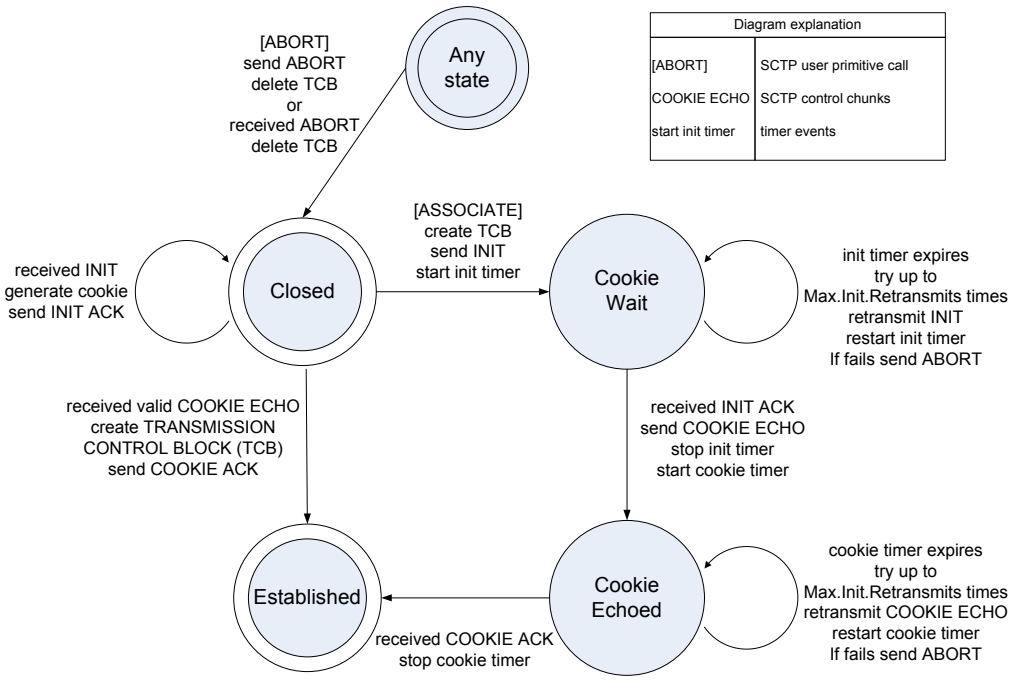
[a] covered by authentication specification [Tuexen et al., 2007].
[b] defined with Direct Data Placement (DDP)
[c] added by the DAR extension [Stewart et al., 2007].
[d] provided with the padding chunk specification
[e] added by the PR-SCTP extension [Stewart et al., 2004].

(count in Bytes, without chunk padding), and chunk value field of a variable length that carries the chunk-specific transfered information within mandatory and optional parameter fields (the latter also have the TLV format). The length of a chunk must be a multiple of 4 Bytes, and is padded with zeros to do so if necessary (no more than 3 Bytes of padding is added at the sender). All the chunks but INIT, INIT ACK and SHUTDOWN can be bundled into one SCTP-PDU, unless the size of Maximum Transmission Unit (MTU) is not exceeded.

Standard SCTP is a connection-oriented protocol working on the top of the connection-less network. The term *association* is used to name the relationship between two SCTP endpoints, distinguishing it from a TCP connection as a more complex structure that can span over multiple IP addresses at each endpoint. An SCTP association is set up in a four-way handshake as shown in Fig. 3.3 [Stewart, 2007]. This is one of the major differences to the TCP, which with its three-way handshake is prone to blind DoS attacks such as SYN flooding. To prevent that SCTP introduces so called cookie mechanism that adds the additional leg to the association setup. Another difference to the TCP can be seen during the release of the association, which is simpler in SCTP, and involves

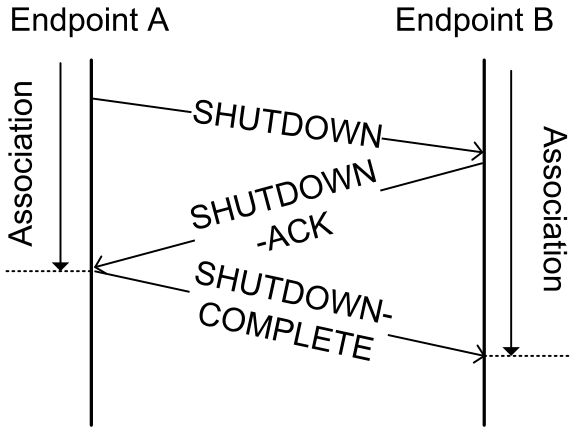


(a)

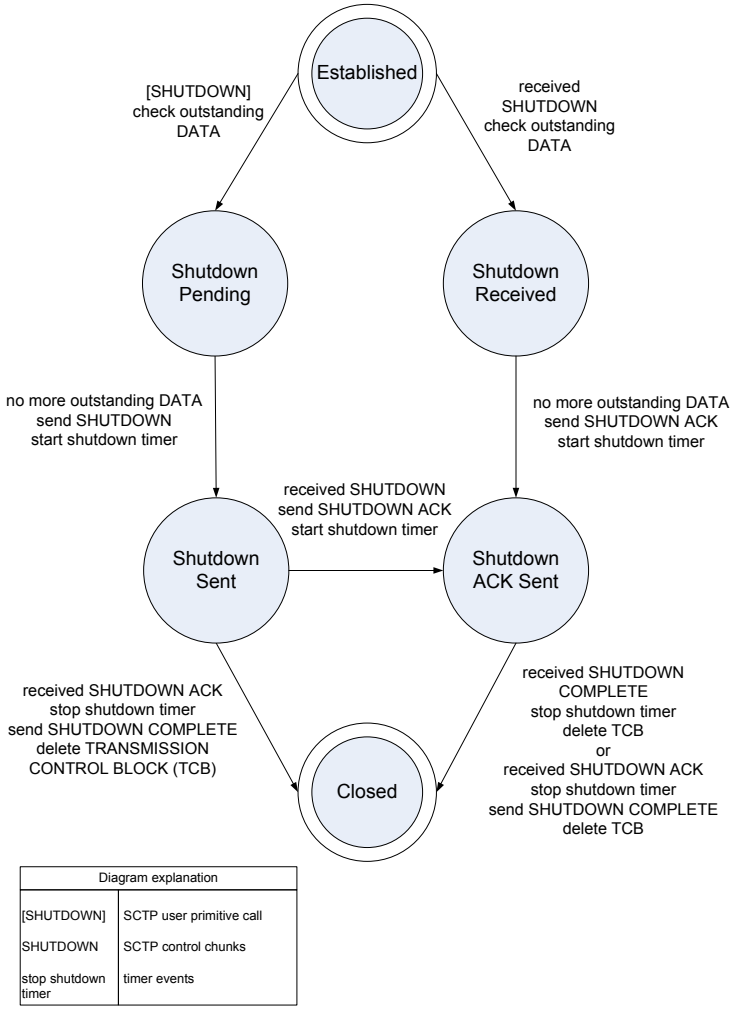


(b)

Figure 3.3: SCTP association setup: (a) chunk flow; and, (b) finite state machine.



(a)



(b)

Figure 3.4: SCTP association release: (a) chunk flow; and, (b) finite state machine.

only a three-way handshake, as shown in Fig. 3.4 [Stewart, 2007]. As a consequence in contrast to TCP, SCTP does not allow the half closed connections.

SCTP flow- and congestion-control algorithms are essentially the same as in TCP. SCTP provides flow control service to prevent the sender from overflowing the receiver's buffer. Identically as in TCP, the SCTP sender maintains an association variable called advertised receiver window size (a_rwnd) to keep track on the space that is currently available in the receiver buffer. Initial a_rwnd credit is announced in the INIT and INIT ACK chunks, and is updated in each SACK. According to the flow control algorithm the sender can not send any new data if the receiver indicates it has no space to buffer that data (a_rwnd is 0). The only exception from this rule is that the sender regardlessly of the a_rwnd value can have one DATA chunk in flight to provide information about rwnd changes that have been missed due to the SACK being previously lost. In case the a_rwnd is zero this is called a *zero window probe*.

Congestion control in contrast, prevents the sender from overwhelming the network (following the so called *non-greedy* approach of TCP). The basic idea is to drastically reduce the sending rate at the event of loss (the assumption here is that the loss is caused by a congestion in the network. This assumption is hold true for the fixed IP networks, but it is not in wireless links where packet losses are not necessarily associated to link congestion). So the SCTP sender additively increases its sending rate when there is no congestion event, and once the congestion is detected, i.e., a loss occurs, the SCTP sender multiplicatively decreases its sending rate. This approach, called *AIMD behavior*, can be achieved using multiple algorithms that SCTP adopted from TCP (TCP congestion control state of the art as for late 1990s is described in [Paxon et al., 1999]). One of the few modifications is that SCTP adds one more control variable to regulate its sending rate. Apart from congestion control window (cwnd), and slow-start threshold (ssthresh) known from TCP, there is also a variable (partial_bytes_acked) that keeps track on all data acknowledged, not necessarily in sequence, from the last cwnd increase during the congestion avoidance phase. The four main congestion control algorithms used in SCTP will be shortly described here. Any further details can be found in [Stewart, 2007].

- *Slow start*, used to probe the network in order to determine the available capacity, either when data transmission starts or when recovering from the retransmission timeout (RTO). During the slow start phase, upon reception of each SACK that advances the counter of the last DATA chunk Transmission Sequence Number (TSN) received in an unbroken sequence, a so-called Cumulative TSN ACK Point (CumTSN), the cwnd value is increased by the lesser of two: number of total bytes acknowledged by that SACK and the MTU size. The result is an exponential growth of the cwnd, doubling its value every round-trip time (RTT). The slow start is started with the initial value of the cwnd (`cwnd.init`) on the transmission start, and with one MTU when recovering from the RTO expiration. Once the cwnd exceeds the ssthresh value (initial ssthresh value on the transmission start, or half of the value it had before the RTO expiration) the SCTP quits the slow start algorithm and starts the congestion avoidance.
- *Congestion avoidance*, an algorithm linearly increasing the cwnd value by one MTU every RTT. There are several ways to accomplish that. The one described by the standard SCTP specification increases the cwnd by one MTU after receiving a SACK that advances the Cum TSN, if the partial_bytes_acked is at least equal to the cwnd value and the sender has at least cwnd of data on flight. The congestion avoidance continues until the network starts losing packets what is interpreted as a congestion, and ignites RTO expiration. An RTO expiration brings back the slow start algorithm, setting the ssthresh to half of the cwnd value in the moment the loss was detected, and dropping the cnwd to one MTU, accordingly.
- *Fast retransmission*, an algorithm that limits the number of RTO expirations by retransmitting a missing SCTP PDU after receiving 3 duplicate SACKs (three consecutive reports indicating a miss of the same TSN), and before the corresponding RTO expires. Upon such a loss detection from SACK the cwnd is cut in half and the sshtresh is set to the new cwnd value. Chunks once marked for fast retransmission can not be subject to a consecutive fast retransmission.
- *Fast recovery*, an algorithm that once the fast retransmission is launched prevents any changes to the cwnd and ssthresh (i.e., further cuts if there is more data that can fit in one SCTP PDU to be retransmitted) until all data marked for fast retransmission is retransmitted.

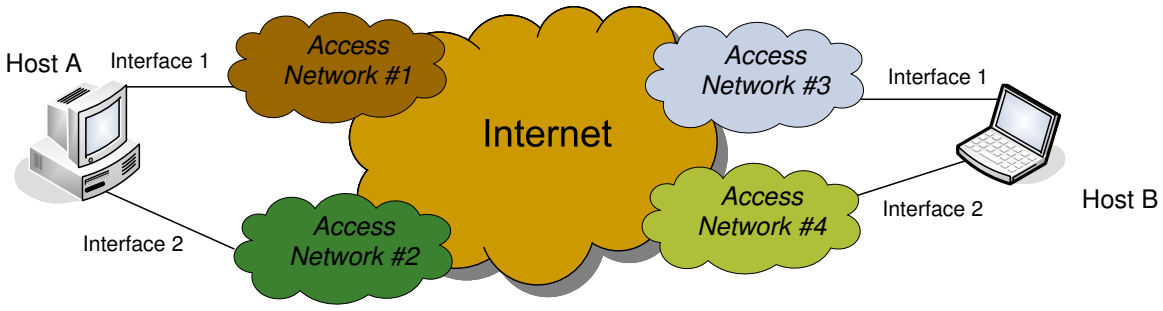


Figure 3.5: SCTP multihoming.

3.1.2 New protocol features

One of the new features provided by the standard SCTP is *transport-layer multihoming*. Multihoming (Fig. 3.5) binds multiple source-destination IP addresses for a single association between two SCTP endpoints. These IP addresses are exchanged and verified during the association setup, and each destination transport address is considered as a different path towards the corresponding endpoint. An important remark regarding the *path concept* in the SCTP must be made here. SCTP specification [Stewart, 2007] defines the path as: *the route taken by the SCTP packets sent by one SCTP endpoint to a specific destination transport address of its peer SCTP endpoint*. This definition is a consensus achieved at the early stage of the protocol specification development [Stewart et al., 2000], and was specifically not changed since then. The main argument in favor of the current path definition is that source based routing is not widely deployed over the Internet, so the SCTP implementation does not need to control the source address on which packets are sent to a given destination. One could argue that in case there are multiple local interfaces and multiple remote addresses, the number of possible paths should be simply a combination of all possible source-destination IP address pairs. Such an approach would be more robust in case of failures affecting both endpoints (if an alternative path through an intermediate router exists), and also would provide additional benefits for loadsharing applications. Nevertheless, at this stage there are several procedures that are handled as *per destination address*[2], e.g., path verification, that would have to be adjusted to such a modified path definition. Also the necessary changes would have to affect congestion control as well as error counting in its current shape.

During the association setup, one of the available paths is selected as the *primary* path, used for transporting all new data chunks during normal data transmission, whereas all remaining paths, called *alternate* paths, serve only for retransmissions. Alternate paths are often referred in the literature as *backup paths*, especially in the robustness context of multihoming. Multihoming, in the case of IP networks, means multiple IP addresses, and typically (but not necessarily) implies multiple link-layer interfaces.

Multihoming was originally designed for environments requiring high application availability and reliability, such as the delivery of Signaling System No. 7 (SS7) messages. Despite the evolution of SCTP towards a general transport protocol, this design principle has been kept. Consequently, the scope of use for multihoming defined within RFC 4960 [Stewart, 2007] is only for handling single retransmissions and performing primary path failover in case of a permanent link failure. Any other applications, e.g., transport-layer handover or loadsharing over multiple network paths, are not supported within the standard SCTP specification, and instead should be covered by dedicated protocol extensions. Regardless of this limitation, SCTP multihoming seems a promising protocol feature that may easily be leveraged to provide support for both mentioned applications.

When considering the use of multihoming in transport-layer handover context, it is very important to keep in mind that standard SCTP has no mechanisms to allow dynamic changes to the set of IP addresses specified for an active association (at the association setup). Thus, in a mobile network scenario, if an association has already been established for a given IP address and a new

[2]Note however that the path supervision process (DATA and corresponding SACK chunks) is handled per source-destination IP address pair

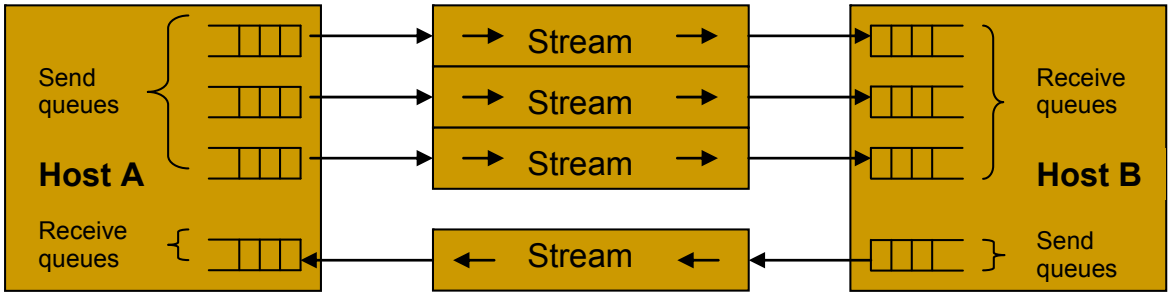


Figure 3.6: SCTP multistreaming.

PoA with a different IP address becomes available, there is no way to include it in the association and switch the primary path over to the new network connection. An extension to standard SCTP that facilitates dynamic address manipulation, and hence enables handover support using SCTP multihoming is described in Section 3.1.3. Another important consideration about handover application of multihoming is that with standard SCTP only a single path is used for data transmission (i.e., the primary path) while all other available paths can handle retransmissions only. Then, the decision of changing the primary path relies mainly on the failover mechanism, what is inspected in more detail in Chapter 4.

The biggest challenge for loadsharing application of multihoming is that simultaneous data transfer over multiple paths can provoke packet reordering at the receiver, what would deteriorate SCTP performance, since congestion control algorithms in standard SCTP are derived from TCP, and hence do not work well when reordering is common [Iyengar et al., 2006]. Thus, to accomplish loadsharing the SCTP send-buffer management and congestion control must be updated to take into account the problems of sending data over multiple paths using a single sequence-number space, and the consequences of sender-introduced reordering. So far, there is no commonly defined extension that facilitates loadsharing for SCTP, existing proposals are analyzed in Chapter 5.

Multistreaming, illustrated on Fig. 3.6, is the second of the newly introduced SCTP features. Multistreaming allows the establishment of associations with multiple streams. Streams are unidirectional data flows within a single association. The number of requested streams is declared during the association setup and the streams are valid during the entire association lifetime. Each stream is distinguished with the Stream Identifier field included in each chunk, so that chunks from different streams can be bundled inside one SCTP PDU. To preserve order within a stream the Stream Sequence Number (SSN) is used. Consequently, TCP's HoL blocking problem stalling entire TCP connection is reduced to the affected SCTP stream only, as data received in order within a stream (handled by SSN) but not within the entire association (counted using TSN) can be delivered to the application. Among the most important applications of multistreaming are priority stream scheduling, preferential treatment, and reducing the latency of streaming multimedia in high-loss environments. Last but not least, multistreaming, jointly with the partially reliable extension to SCTP (PR-SCTP) [Stewart et al., 2004] can be used to support real-time applications.

3.1.3 Protocol extensions

The *Dynamic Address Reconfiguration* (DAR) SCTP extension [Stewart et al., 2007], although originally defined to help with IPv6 renumbering and hot-pluggable cards by the IETF SIGTRAN working group [IETF SIGTRAN], can be easily leveraged to make SCTP a mobility-enabled transport protocol. It should be emphasized that this extension is seen as a mobility enabling feature, but not as a mobility solution by itself [Koh and Xie, 2005]. The DAR extension allows SCTP to dynamically add or delete IP addresses, and request the primary-path change during an active SCTP association, by means of two new chunk types: *ASCONF and ASCONF-ACK* (check the Table 3.1), and six new parameters: *Add IP Address, Delete IP Address, Set Primary Address, Error Cause Indication, Success Indication, Adaptation Layer Indication*. Modifying the IP address(es) of the association increases the risk of association hijacking [Stewart, Tuexen, and Camarillo, 2007] and therefore the ASCONF chunk

must be sent in an authenticated way (an AUTH chunk is bundled *before* the ASCONF chunk), as described in [Tuexen et al., 2007]. Standard SCTP enhanced with the DAR extension is also referred to as *mobile SCTP* (*mSCTP*) [Koh and Xie, 2005; Riegel and Tuexen, 2007].

With mSCTP, the IP address(es) may be announced to the endpoints during association initialization and changed whenever it is needed during the association lifetime. When adding (deleting) an IP address to (from) an association, the new address *is not* considered fully added (deleted) until the ASCONF-ACK message is received. An addition or deletion of an IP address may be combined with changing of the primary path. However, only addresses already belonging to the association can be set as the primary, otherwise the Set Primary Address request is discarded. Before any DATA can be sent to the newly added destination, the new destination must undergo a path verification procedure. The path verification procedure is done by means of HB chunk that may be piggybacked on the ASCONF-ACK response to a new destination. Once the HB-ACK response is received from the remote peer, the destination is considered as confirmed and available for normal data transfer. mSCTP preserves the same congestion control rules as standard SCTP, and logically, a lot of research performed recently on SCTP could be useful for mSCTP development.

Another extension to standard SCTP, PR-SCTP, offering a non-duplicate, in-order data delivery service with controlled loss (standard SCTP provides all these with no-loss data delivery), will not be discussed here, as it is not directly related to the subject of the dissertation. Readers interested in more detail should refer to RFC 3758 [Stewart et al., 2004].

3.1.4 Summary

The array of the features offered by SCTP that were discussed through Section 3.1 is summarized in Table 3.2, first provided in [Stewart and Amer, 2007]. An interesting comparison relates SCTP capabilities to two transport-layer workhorses of the current TCP/IP stack: TCP and UDP, raising the question whether SCTP can be a viable substitute to any of these protocols.

3.2 SCTP state of the art in research

In order to complement presented protocol overview and resume many emerging ideas that have been brought so far in the literature, the SCTP-related research will be revised here by the means of specially dedicated taxonomy, first proposed in [Budzisz et al., 2008]. Before proceeding with the taxonomy results, it must be stated that especially the new protocol features have attracted attention of the researchers from diverse fields, being one of the main reasons for creating this classification.

3.2.1 Taxonomy

The purpose of a taxonomy is, in general terms, to provide a classification into an ordered system. How the system should be constructed is an open question and must be tailored to each specific instance. Ideally, the categorization should be complete and non-overlapping. The taxonomy proposed by Budzisz et al. [2008] for SCTP is constructed using four orthogonal dimensions with a number of non-overlapping categories in each dimension, as shown in Fig. 3.7. Every single classified object, namely a research article related to SCTP, can be projected to one or more categories within each dimension.

The main design goal and, unsurprisingly, the hardest problem to tackle at the early stage of defining the classification categories, was to minimize the possible overlap between them, in order to reduce as much as possible the ambiguity of what category some research aspect may relate to. As classification work progressed, the initial set of proposed categories was refined (merge/split operations) to create a final taxonomy that has a high degree of orthogonality and is versatile enough to evaluate practically all SCTP-related research. This iterative mode of taxonomy development also resulted in a reduced complexity for the final taxonomy that will be now presented in detail.

Table 3.2: Comparison of transport-layer protocols

Service / Feature	SCTP	TCP	UDP
Connection-oriented	Yes	Yes	No
Half-closed connections	No	Yes	N/a
Protection against blind DoS attacks	Yes	No	N/a
Dynamic address manipulation	Optional [a]	No	N/a
Reliable data transfer	Yes	Yes	No
Partial-reliable data transfer	Optional [b]	No	No
Preservation of application message boundaries	Yes	No	Yes
Application PDU fragmentation/bundling	Yes	Yes	No
Ordered data delivery	Yes [c]	Yes	No
Unordered data delivery	Yes	No	Yes
Full-duplex data transmission	Yes	Yes	Yes
Flow and congestion control	Yes	Yes	No
Selective acknowledgments	Yes	Optional	No
Path max. transmission unit discovery	Yes	Yes	No
Explicit congestion notification support	Yes	Yes	No
Multistreaming	Yes	No	No
Multihoming	Yes	No	No

[a] covered with DAR extension.
[b] covered with PR-SCTP extension.
[c] the data within a stream is delivered in order.

Dimension 1: Protocol feature examined

The first dimension used in this taxonomy classifies the research into different categories defined upon the protocol feature or functionality examined. As pointed out in Section 3.1, compared to TCP, SCTP provides new, interesting functionality. Much of SCTP research obviously targets this new functionality and examines it from different viewpoints. Besides that, SCTP has a number of features, identical or similar to TCP, that have spurred some research too. In case of multihoming, the feature that attracts almost half of the classified articles, it was considered appropriate to distinguish three separate subcategories, depending on the aim in which this feature is used. The categories in dimension 1 are the following:

1. **Multihoming**. The multihoming feature was originally designed for enhancing end-to-end robustness by using transport layer failover to an alternate path when the primary path fails. Later research has explored other uses of multihoming, and to classify all these uses three subcategories are introduced:
   (a) **MH-Robust**. This is the original use considered for multihoming, the end-to-end robustness by using failover to the alternate path.
   (b) **MH-Handover**. The multihoming functionality of SCTP can also be used as a building block to provide transport layer mobility management solutions.
   (c) **MH-Loadsharing**. The multihoming feature may be used to concurrently transfer data over multiple paths in a load-balancing fashion. This creates both a potential for im-

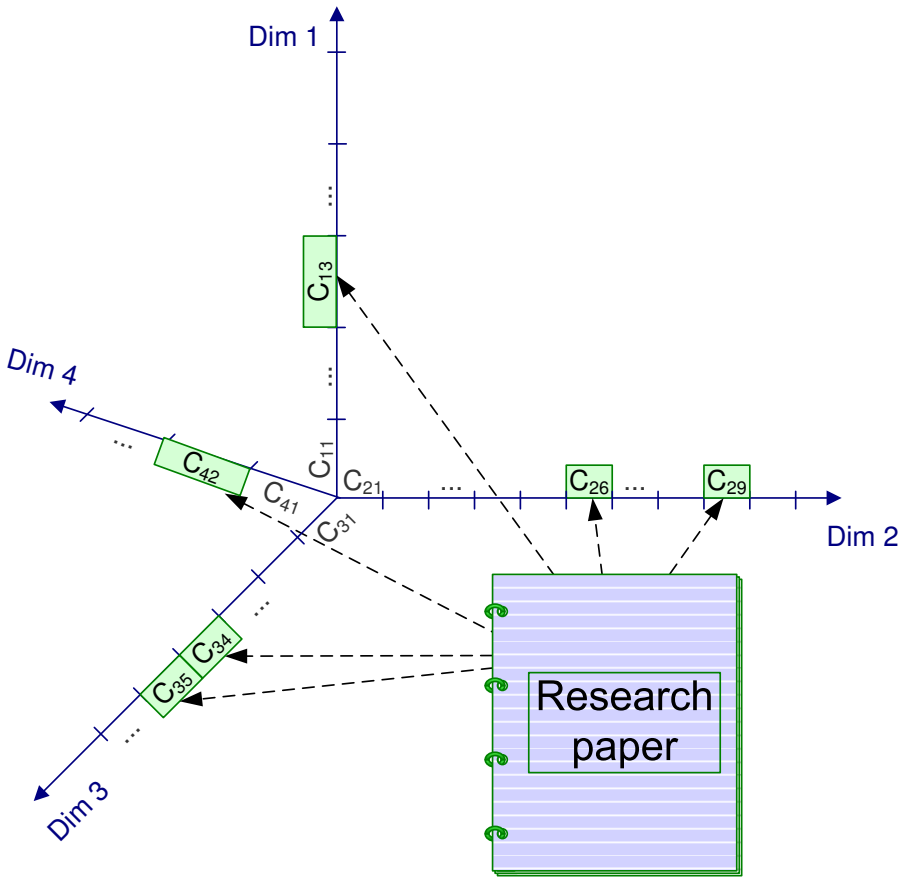


Figure 3.7: Graphical illustration of proposed taxonomy.

proved end-to-end performance and a number of complicating issues that need to be addressed.

2. **DAR**. The extension to SCTP for a dynamic address manipulation during an active SCTP association is an essential building block for handling mobility at the transport layer for either single-, or multi-homed nodes.
3. **Multistreaming**. The capability of SCTP that allows a single association to have multiple logically separate streams. This functionality is new relative to TCP and one major advantage is that it can reduce the HoL blocking which can occur with TCP since there is no ordering requirement between the streams.
4. **PR-SCTP**. The partially reliable extension to SCTP offers more flexibility with regards to the reliability of the transport service. The ability to provide partial reliability opens up new possibilities in how to handle the reliability-versus-latency trade-offs at various layers in the protocol stack.
5. **SH-Congestion**. Congestion control is a central issue for any transport protocol that is to be deployed on the Internet. SCTP's congestion control for single-homed associations is to some extent similar to TCP's, but also has some differences. There is an abundance of literature on TCP congestion control, and the impact of SCTP's congestion control nuances on wired and wireless networks is clearly a relevant topic.
6. **Security**. SCTP provides some new security enhancements, and the multihoming capabilities of SCTP may potentially lead to an increased confidentiality since multihoming allows splitting data over separate physical paths.

7. **Survey**. This specific category captures all publication that provide a general protocol overview, rather than an analysis of a particular protocol feature.
8. **Other**. Some additional and diverse aspects of SCTP that are not covered by any of the above categories (checksum usage, ...) are classified here.

Dimension 2: Application area

This dimension focuses on the application area that the research to be classified relates to. The application area influences both the traffic data pattern (more precisely, the transfer size distribution as well as the data generation process, i.e., if data is always available or not, and also the consumption rate for streaming) and the required performance targets. The categories in dimension 2 are the following:

1. **Signaling**. Since SCTP was originally designed for transporting SS7 signaling, the performance in this application domain is important. SCTP can also be used to transfer other kinds of signaling traffic such as SIP.
2. **Multimedia**. SCTP, and especially together with the PR-SCTP extension that provides partial reliability, can be used to transfer multimedia data. The multistreaming capabilities of SCTP maps very well to multimedia traffic having multiple media streams. This application area has its own set of challenges and is therefore handled in a separate category.
3. **Web**. Web transfer is a large application area for TCP, and may be so also for SCTP. SCTP's multistreaming ability is one of the factors that may impact the transport layer performance of SCTP in this application area.
4. **Bulk**. Applications such as File Transfer Protocol (FTP) are commonly used when examining transport layer protocols, and these are also relevant for SCTP. Examination of single-homed SCTP for bulk transfer provides insights into the steady-state protocol performance and allows comparison to similar TCP results. Bulk traffic is characterized by being large enough to have the transfer time decided by the steady-state behavior and to be greedy, i.e., always have data to send if there are resources available to transmit it.
5. **MPI**. The Message Passing Interface (MPI) application area covers the use of SCTP in local and wide area cluster and grid environments. MPI is nowadays the dominant model used in high-performance computing, and SCTP can be seen as a promising option of IP-based transport support for MPI.
6. **Other-applications**. This category groups few other specified (but diverse) applications, not captured by any of the above categories, such as for example Data Acquisition Systems (DAQ).
7. **Unspecified**. This category is used when no application area has been defined in the classified research, e.g., when the research focus on conceptual discussions of some aspect.

Dimension 3: Network environment

This dimension covers the network environment that is considered in the research to be classified. This dimension is divided into two domains (wireless/wired) with several categories. The categories in dimension 3 are the following:

Wireless domain

1. **WLAN**. The WLAN environment is characterized by relatively high bandwidth and a reliable link layer that to a large extent shields the upper layers from physical layer problems such as bit-errors and frame losses, but can show significant end-to-end delay variation. When WLANs are used in a MANET setting the research article falls under the next category.
2. **MANET/VANET**. The Mobile Ad-hoc Network (MANET) environment has a number of defining characteristics, such as relatively low bandwidth, large variation in end-to-end delays and losses directly caused by congestion or temporary route unavailability. This category also includes vehicular networks (VANET), since they are similar in nature and there are too few articles in the latter group to form a separate category.

3. **Cellular**. This category, which spans from GSM to 3G high speed packet access (e.g., 3GPP HSDPA), offers medium to low bandwidths and considerable variation in end-to-end delays. In these types of networks, a reliable link layer can ensure that there will be no wireless losses, but only congestion losses.
4. **Heterogeneous**. This category refers to the case where coexisting heterogeneous wireless networks are considered in the research, e.g., vertical handover scenarios.
5. **Space**. Space (also referred as satellite) networks typically have long round-trip times which affects the transport layer behavior.

There are also cases where the wireless environments used are not sufficiently detailed to be placed in one of the above categories:

6) **Wireless-General**. This category catches research where a wireless environment is used to motivate the existence of, e.g., bit-errors, but no further description of any specific technology is provided.
7) **Wireless-Unspecified**. This category reflects the case where wireless environment is not explicitly defined in the publication, but it can be inferred from other information that a wireless setting is present. This is specially the case for some research concerning handover.

Wired domain

8) **Managed**. This category captures network environments where a high degree of control exists over the network for the entire end-to-end path. This allows appropriate dimensioning, traffic engineering and QoS mechanisms to provide the desired network characteristics. Operator-owned IP-based signaling networks are a typical example of managed networks, and an original design target for SCTP.
9) **QoS**. This network environment refers to cases where some Quality of Service (QoS) enhancing mechanisms are employed, which leads to the possibility of providing a better service than pure best effort.
10) **Best effort**. The best effort network environment provides only best effort end-to-end packet transfer service (e.g., best effort Internet) without any assumptions about, or restrictions on, delay bounds, loss rates, etc.

Dimension 4: Study approach

The fourth dimension considers the method used to obtain the results. Different approaches have different benefits and drawbacks, and the results are strengthened, if multiple approaches are used. The following are the categories:

1. **Conceptual**. The conceptual description approach describes and reasons about ideas, mechanisms and functionalities in a general way without providing quantitative data to analyze the performance.
2. **Analytical**. The analytical modeling tries to describe the essential behavior of an entity (such as a protocol) with a mathematical expression that given some input parameters provides some metric of interest. When a suitable expression has been derived, it can then easily be used to predict the performance of the entity under a range of conditions. However, in order to create a tractable formula the expression must often be simplified which introduces inaccuracies and highlights the need for model verification.
3. **Simulation**. The simulation approach also uses abstract representations of the entity under study, but in this case the abstractions are much more detailed and can include all relevant protocol functionality. Also with simulations, there is a need to verify that the abstraction used in the simulation is correct and representative. Simulation allows a large parameter space to be explored and can provide considerable detail in the output.
4. **Emulation**. In contrast to analytical modeling and simulation, the emulation approach uses actual protocol implementations running on a real hardware. The emulation approach naturally captures the behavior of a protocol *implementation*, and also allows for factors such as possible interaction effects with the operating system, device drivers and communications hardware. Here, it is the behavior of the end-to-end connectivity that is abstracted to some degree by the employed emulator.

5. **Live**. The live network approach entails performing experiments on a running communications network. This naturally includes all aspects, both at the endpoints and at the network level. However, live experiments are hard to control and repeat. Getting access to live networks to the extent necessary may also be problematic in some instances.

3.2.2 SCTP research analysis

Classification methodology

In the presented study only works that could have had a considerable audience in research community are included, to be precise, journal or conference articles that were publicly available, and written in English (some research work in other languages was also identified but not included in overall results, as possibly not having that big impact on the research community). The information about research works was collected having in mind general availability as a principal rule. As a source of information the most common databases were taken into account, and in particular:

- The IEEE Xplore database [IEEEXplore],
- The ACM Digital Library [ACM],
- The BibFinder database [BibFinder],
- The Engineering Village database [EngVillage],
- CiteSeer.IST [CiteSeer],
- Google^{TM}Scholar [Google Scholar],
- The ISI Web of Knowledge [ISI].

Once the information about the articles was collected, the articles underwent a detailed classification process, to allocate them in the appropriate category within each dimension. In case the article covers more than one category within a dimension, the article is counted equally in each of the categories that reflect the contents of the given article (i.e., the article is considered to fit or not each specific category without any kind of weights for articles that fall into multiple categories). The entire classification is available on-line [SCTP Survey].

An important consideration is also the time frame for the collected research. RFC 2960 marked the initial year when the SCTP research took off (i.e., year 2000), whereas on the other end, to secure the trustworthy metrics, the end of 2007 was set up as the cut-off date. It should be noted that the numbers for the articles published in 2007 is less reliable, since there can be quite some lag between paper publication and the time it appears in the examined databases. Fig. 3.8 shows the percentage distribution of all years analyzed. Out of 279 total articles collected, merely 22% were published in the first four years since the protocol specification was announced in 2000. The largest number of articles so far (66) has been published in 2006, however the differences between any of the last four years are relatively small. Year 2007 brings the ambiguity to the interpretation of the most recent trend. Yet it is not clear, whether the SCTP research enters a steady state fluctuating around 50 articles per year, or there is still not enough data available from the year 2007 to reflect a steady growth.

SCTP research profile

As mentioned in Section 3.2.1, an article can be projected to more than one category within each dimension. To quantify the extent of this multiple assignment we define α, as a ratio of the number of all category assignments to the total number of articles published, calculated for each dimension. Thus, α is influenced by both versatility of the analyzed research and the difficulty of creating a non-overlapping classification. This is especially true for dimension 1 which has the highest value of α (1.34).

Fig. 3.9 presents an overview of the investigated research within each of the taxonomy dimensions. Dimension 1, protocol feature examined (Fig. 3.9a), clearly shows the domination of two major new SCTP features in SCTP-related research. Multihoming alone stands for 46% of all hits within the dimension, and adding multistreaming brings it up already to more than 59% of all hits. The DAR extension provides also a considerable contribution (nearly 15% of all hits). However, most of the articles treats DAR in conjunction with the multihoming feature (MH-Handover

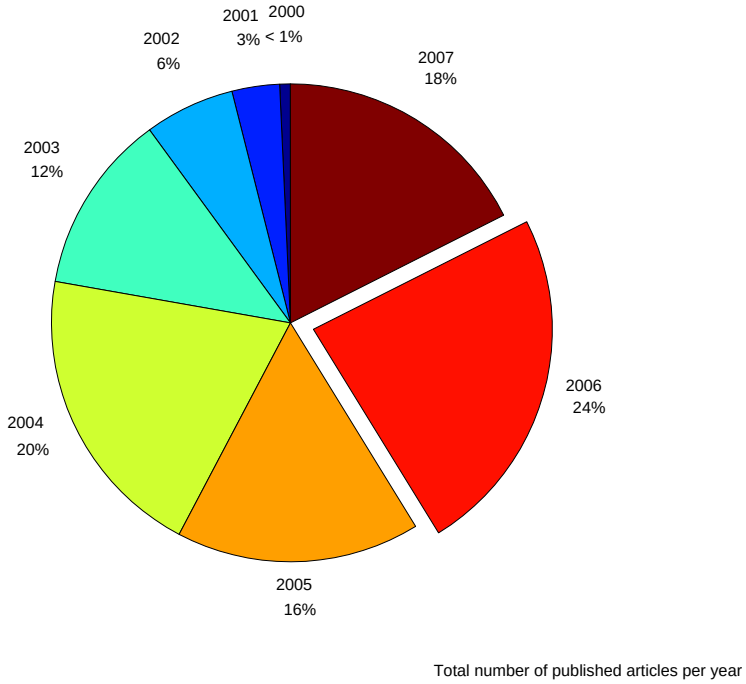


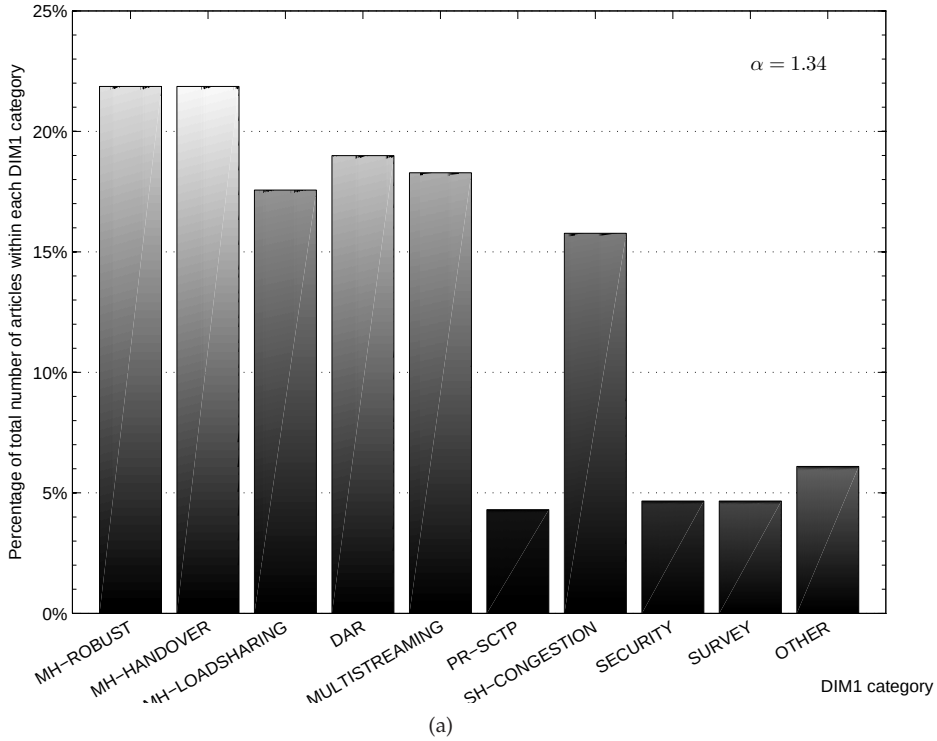
Figure 3.8: Annual distribution of all published articles.

category). Out of the remaining protocol features, the most explored is single-homed congestion, a topic that has already been well studied for the SCTP ancestor TCP. Relatively little attention in the research community has been spent on security issues and the PR-SCTP extension, both falling below 5% of all hits.

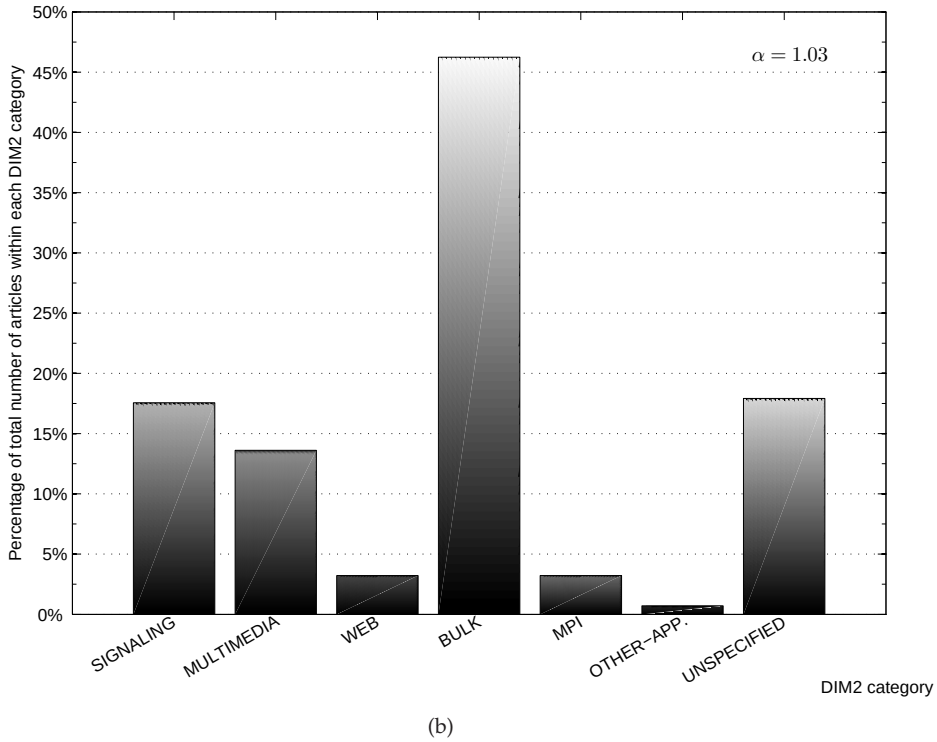
Dimension 2, application area (Fig. 3.9b), in contrast is rather highly polarized. With a very low α (1.03), almost all articles had been assigned to only one category. Most of the research, nearly 50% of all hits and articles in this case, use the bulk transfer application model. Almost one third of all these articles classified as bulk transfer does not have any explicit specification about the application model that was used, but had implicit indications to the bulk model. The original application area of SCTP, signaling transport yields slightly more than 17% of all articles. The same number of articles have an unspecified application model, a quite common case especially within the research devoted to transport-layer mobility. The multimedia category, reaching almost 15%, seems a promising research direction for SCTP. The combined total for the rest of the categories is less than 5%, with web transfer and MPI being the biggest contributors.

Dimension 3, network environment (Fig. 3.9c), has again a very low α (1.03), and a quite clear classification. Wireless related research provides slightly more than a half of all classified articles (about 53%). In contrast the biggest single category belongs to the wired domain, best effort counts for about one third of all the research. The remaining two wired categories have some contribution too, but both below 10%. Heterogeneous networks (18%) are the most commonly analyzed wireless network environment, again mostly because of the transport-layer handover research. Also general wireless environments catch a considerable number of articles (14%), most of them being conceptual models, or general handover schemes without having any particular network clearly specified. WLAN and MANET/VANET networks have attracted less attention so far, slightly over 5% each.

Dimension 4, study approach (Fig. 3.9d) with $\alpha = 1.14$, counts for some articles having more than one category assigned, i.e., combining two different study approaches. This is especially the case for the analytical models being verified by either simulations or emulations. More than a half of all SCTP related research has been conducted using simulations, either the ns-2 simulation model [SCTP-ns2] or Qualnet [SCTP-Qualnet], both delivered by University of Delaware. Emulations contribute to about 20% of the articles. Conceptual descriptions and analytical models come

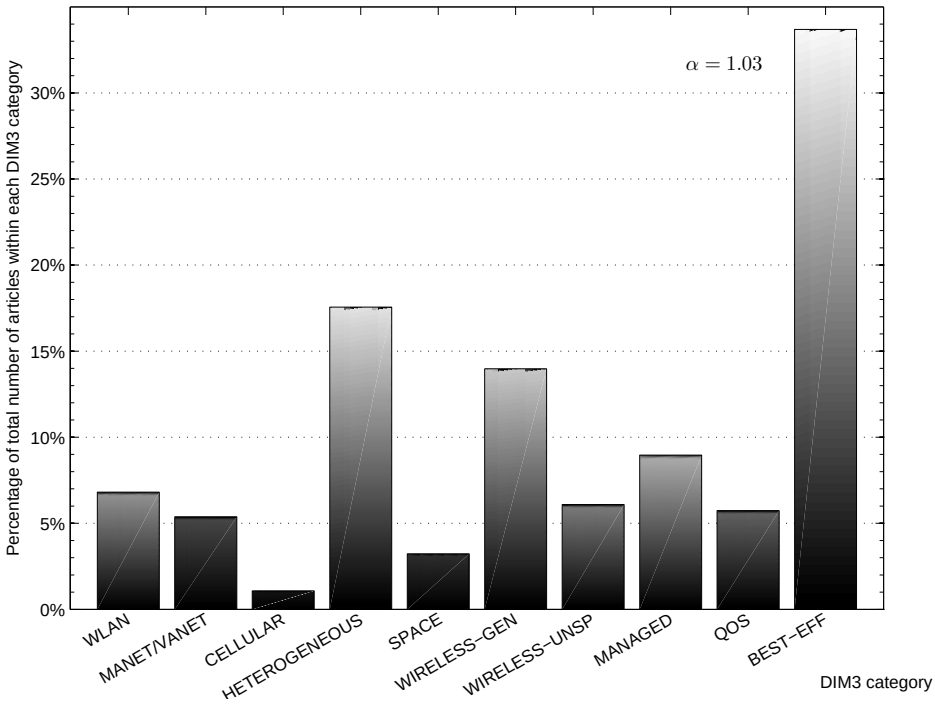


(a)

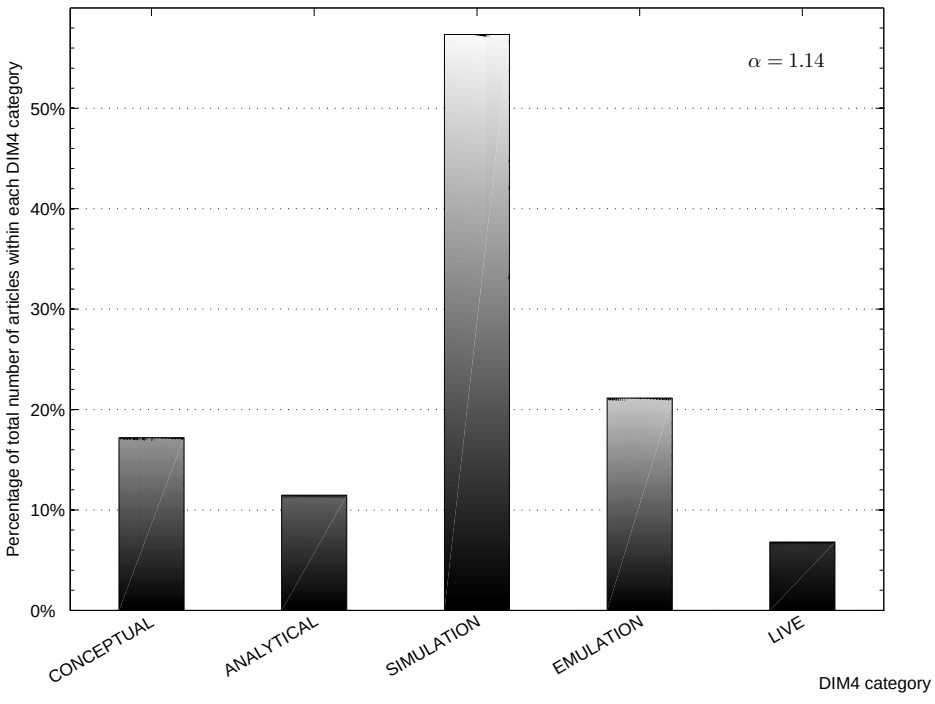


(b)

Figure 3.9: Number of articles within each category for: (a) Dimension 1: Protocol feature examined; (b) Dimension 2: Application area. Wondering why the total is greater than 100%? Documents with multiple categories within a dimension may be counted multiple times.



(c)



(d)

Figure 3.9: Number of articles within each category for: (c) Dimension 3: Network environment; and, (d) Dimension 4: Study approach. Wondering why the total is greater than 100%? Documents with multiple categories within a dimension may be counted multiple times.

close behind, both combined contribute for nearly 30% of all articles, leaving the remaining 3% for live experiments.

Category annual distribution

Fig. 3.10 illustrates the annual distribution of the classified research. The graphics presented for each dimension reflect the contribution of each category for the annual mark, as well as total number of articles published each year, providing a graphical interpretation of the α ratio, previously defined in Section 3.2.2.

In Dimension 1 (Fig. 3.10a), its two leaders contribute to the overall score in a different way over time. The MH-Robust category presents quite stable annual contributions almost from the first year after the protocol specification was released. This initial focus on robustness issues was expected having in mind that this was the original use of the multihoming feature. In contrast, the MH-Handover category represents a new trend in SCTP research, started in the end of 2003. From this moment (over the last five years) an incrementing annual contribution of MH-Handover articles can be observed. More detailed analysis of the handover-related work can be found in Section 3.3.1. A similar trend can be seen for the third application of multihoming, the loadsharing, discussed here in Chapter 5. MH-Loadsharing hardly starts in 2003, 2004 being the first year with a considerable contribution and a steady growth until now. As mentioned before in Section 3.2.2, the DAR extension is strongly related to the MH-Handover feature and therefore follows its characteristic. Multistreaming, the second novel feature of SCTP, similarly to MH-Robust provides a stable contribution over all inspected years. In contrast, two of the analyzed categories face a decreasing trend recently. Single-homed congestion, after providing a considerable bulk of research in the first few years, seemed to lose researchers' interest in 2007. The same happened to survey articles, once SCTP became fairly known protocol (about 2005) this type of publications appear less frequently.

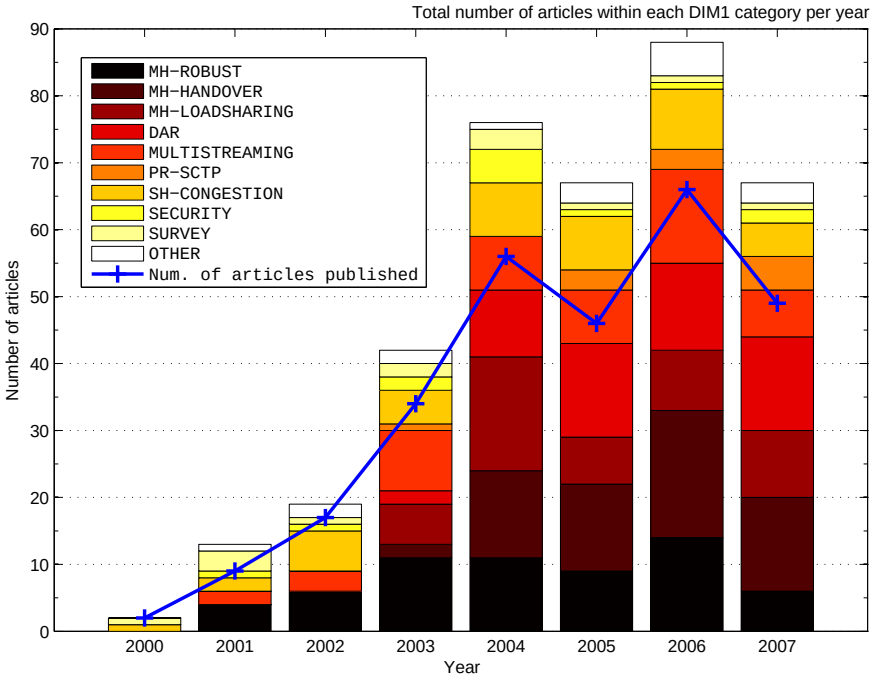
Dimension 2 (Fig. 3.10b) is a quite clear cut with stable contributions from the bulk traffic that dominates this dimension all the years from the beginning. Signaling and multimedia applications also show stable trends over the analyzed years. Sensitivity of the remaining categories firmly prevents to draw the exact tendencies, since with relatively little research done, already one article makes a significant difference. E.g., web applications over SCTP and SCTP use for MPI applications were both first announced in 2003, however a stable contribution in each case dates back to 2005. Experiments with other applications of SCTP appeared in last two of the analyzed years, nevertheless a single paper a year can not be seen as a stable trend.

The dimension 3 time evolution (Fig. 3.10c) is also dominated by one main player, in this case the best-effort category. The main wireless category, heterogeneous networks, dates back to 2003. However, observing its evolution, it can be said that it is one of the driving forces of the SCTP research nowadays. In contrast, a considerable research devoted to satellite environments in the initial years, ended up in 2003. Then, the topic came back again in 2005, but with a considerably lower attention. Another tendency that can be observed is that the wireless research in recent years is less general, the articles become more specific to a particular network solution. Looking at the shares between the wired and wireless fields, we observe the wired domination before year 2003. Now the tendency over the last two years is slightly favorable for wireless networks.

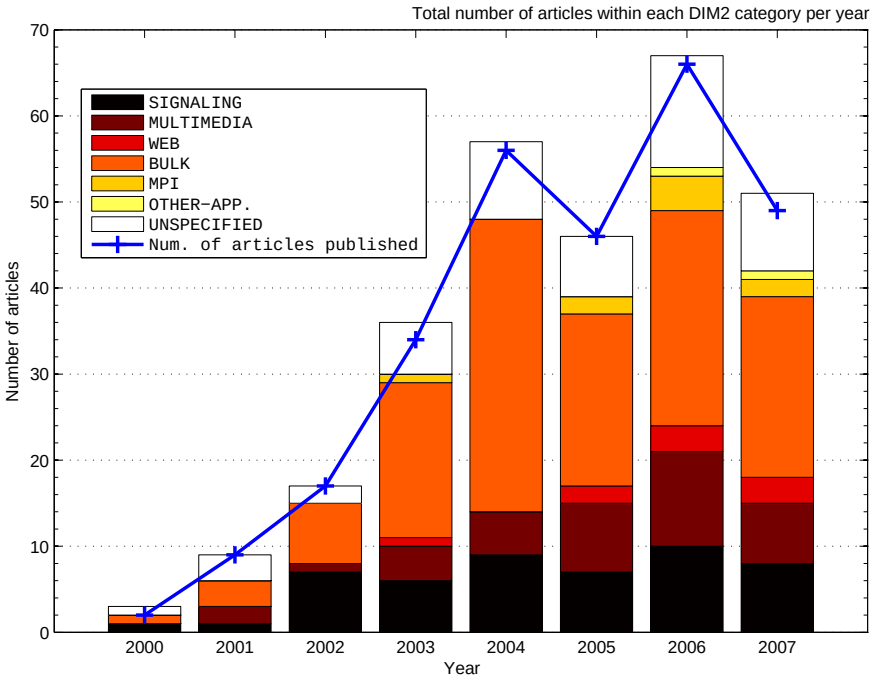
Dimension 4 (Fig. 3.10d) observes a domination of the simulation approach since the publication of the protocol model for the ns-2 simulator in 2002 [SCTP-ns2]. The emulation approach provides less contributions, however dates back to the beginning when the protocol was just introduced. Over the years there is a stable contribution of conceptual and analytical approaches (with peaks dating back to 2004 and 2006, respectively). The novelty is the increase in the number of live experiments over the last two years.

3.3 Mobility implications

As argued in Section 2.2.3, there are important advantages in handling handover at the transport layer. mSCTP can be seen as a viable candidate for providing transport-layer mobility that, unlike a pure network-layer scheme, has the potential to perform smooth handovers, if the mobile node

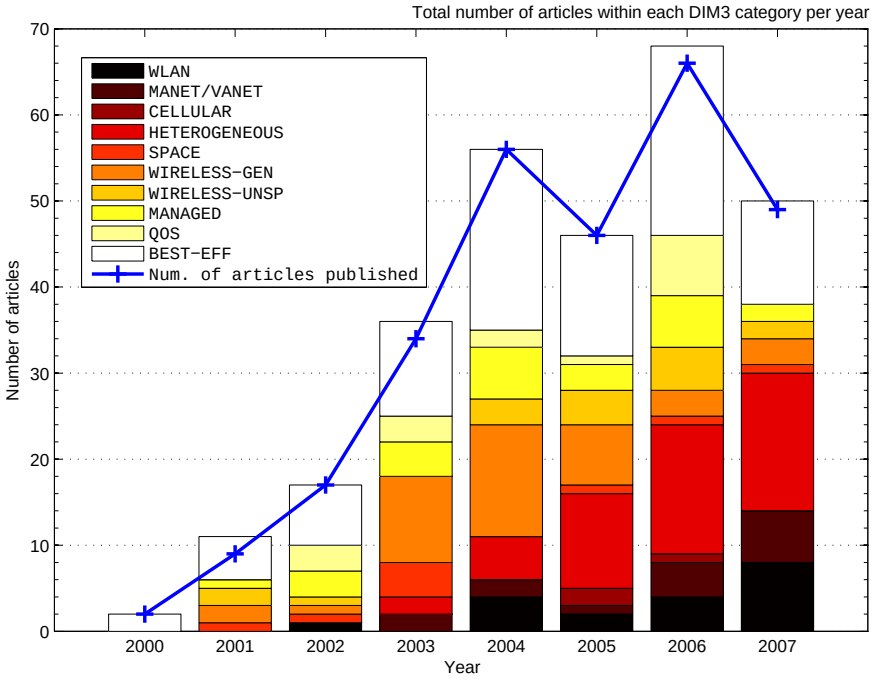


(a)

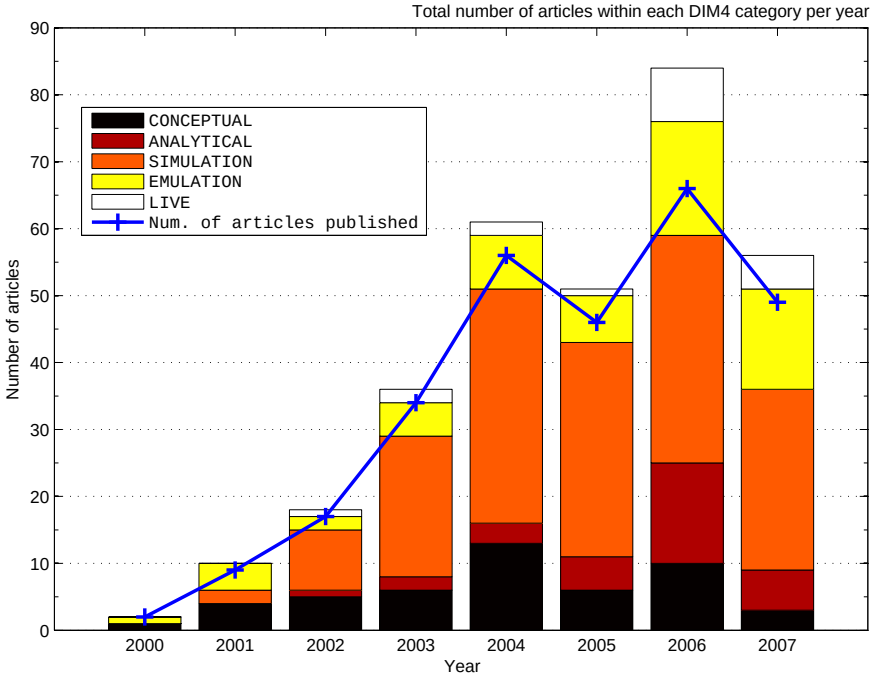


(b)

Figure 3.10: Annual distribution of all articles within each dimension: (a) Dimension 1: Protocol feature examined; (b) Dimension 2: Application area.



(c)



(d)

Figure 3.10: Annual distribution of all articles within each dimension: (c) Dimension 3: Network environment; and, (d) Dimension 4: Study approach.

(MN) is multihomed. Moreover, mobility handled by the transport layer is fully transparent to application-layer protocols that do not include end-host IP addresses in their application data. Nevertheless, despite the importance of the aforementioned advantages, it is vital and highly relevant to identify under which situations/conditions such potential can be really exploited. Therefore, this section will first provide an overview of the related work on transport-layer handover based on mSCTP, and then discuss the applicability of mSCTP in a set of practical heterogeneous radio access networks scenarios, firstly presented in [Budzisz et al., 2008]. In particular, the identification of mSCTP's applicability scenarios is based on the consideration of aspects such as the number of network interfaces on the MN, the number of IP addresses configured for the CN, and the IP address change during the handover process. For each scenario, expected benefits and open issues of the application of mSCTP are also stressed. It is worth to remark here that, even though the discussion focuses on mSCTP, most conclusions drawn here can be extended to any transport-layer handover solution.

3.3.1 Related work

Using SCTP multihoming only for robustness-related purposes has been seen as not taking the full advantage of what this feature can relatively offer. This idea had been discussed already during RFC 2960 specification, however due to time constraints other applications of multihoming were not included into the final document. This status quo has been maintained in the current SCTP specification, nevertheless at this point the considerable benefits of alternative uses of multihoming are rather evident, and are reflected in the analyzed research. Yet the specification stays unchanged, and any non-standard applications of multihoming should be subject to a separate protocol extension. Indeed, this is the case with transport layer mobility. The idea of using the multihoming feature to provide mobility support surged as soon as the work on the DAR extension specification got to an advanced stage. Paradoxically, the original scope of use of the DAR extension, as mentioned in Section 3.1.3, for IPv6 renumbering and hot-pluggable cards, has never gained attention that can be compared to that of handover-related research. Thus, both protocol features, DAR and multihoming, are usually put together when handover support is discussed. The are only a few exceptions to this rule, e.g., mobility support for single-homed nodes based on the DAR feature only, as described in [Honda et al., 2007]. Mobility for a single-homed nodes is analyzed in scenario B in Section 3.3.2.

mSCTP as a candidate for providing transport-layer mobility has still several open points that boosted the research classified as MH-Handover in the taxonomy presented in Section 3.2.1. Apart from articles that introduce the mSCTP-based handover concept, and these that compare the mSCTP proposal to other handover schemes (mainly MIP and SIP), there are several subgroups of articles that relate to mSCTP's main open points: proposals for appropriate handover strategies, support from lower layers to the handover scheme, and enhancements introduced during the handover process.

Fig 3.11 illustrates the distribution of handover-related research over the remaining dimensions of the taxonomy. First thing to notice is that there is a small variation regarding applications considered in the analyzed literature. If the application is explicitly specified, usually it is bulk transfer. Signaling or multimedia applications are less frequent and more specific solutions influencing the design of the proposed handover scheme, e.g., Voice over IP (VoIP) in the work by Fitzpatrick et al. [2006]. In contrast, among the network scenarios analyzed there is much more diversity, with almost every wireless category being represented in the scatter plot. The most typical handover scenarios include heterogeneous networks, and wireless general or wireless unspecified category in case of conceptual works. mSCTP-based handover schemes are also evaluated in homogeneous networks, mainly in WLAN, and considerably less often in cellular networks. Future trends in handover-related research may also consider MANET environments that so far has had only a few contributions. Unsurprisingly, the vast majority of the research dedicated to transport layer mobility uses the ns-2 SCTP model to evaluate the proposed ideas. Only a few works go beyond this scheme and provide results from emulated environments, most of them using the Linux kernel implementation [SCTP-LK].

Research on transport-layer mobility based on mSCTP dates back to late 2003. Most of the

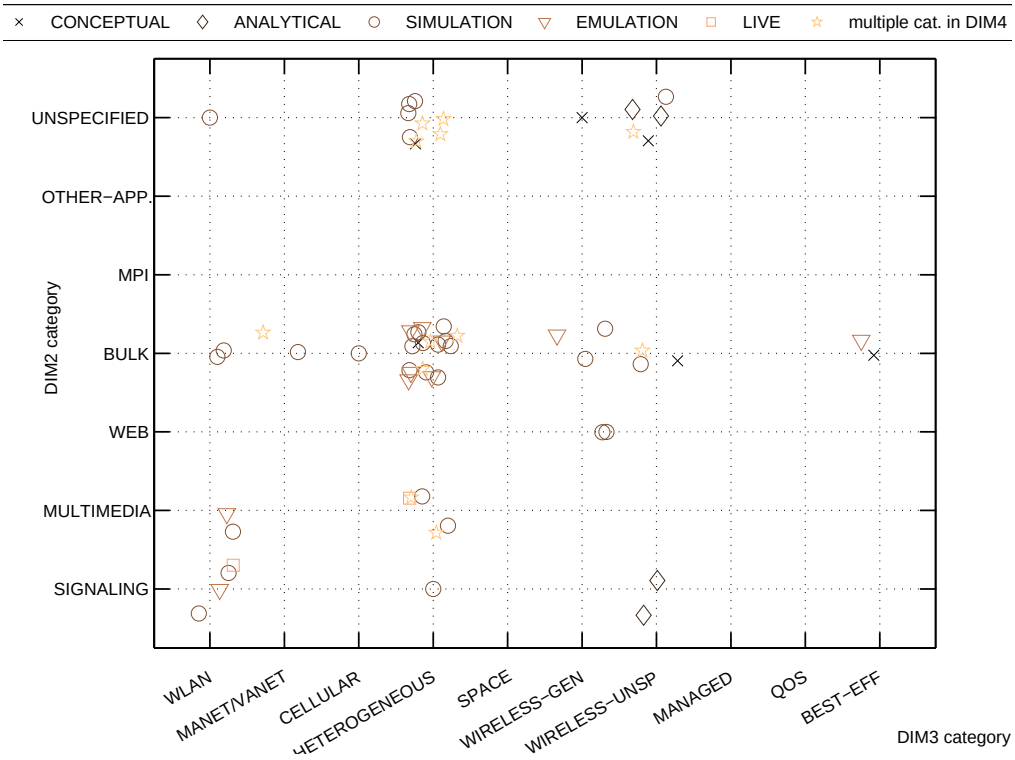


Figure 3.11: Scatter plot of all handover-related articles.

initial works focused on a definition of the proposed solution and identification of its main open points. One of the examples of such conceptual works is the article by Koh et al. [2004] stating that mSCTP can provide seamless handover support, when refined with an appropriate *handover policy*[3]. To mitigate that Koh et al. propose a simple handover decision process (*handover triggering*), based on received signal strength criterion. Another of the initial works on mSCTP, by Ma et al. [2004], provides a comprehensive description of UMTS-to-WLAN and WLAN-to-UMTS mSCTP-based handover schemes, as well as a detailed evaluation of such handover procedures using handover delay as a metric. Further mSCTP evaluation, in terms of handover latency, signaling cost, dropping probability, and overall throughput is given by Argyriou and Madisetti [2007]. Argyriou and Madisetti relate the obtained results to the MIP and HMIP schemes, discussed already in this work in section 2.2.2, finding that the mSCTP scheme is capable of achieving results similar to HMIP while providing a more scalable solution. This work can be complemented by an empirical evaluation of mSCTP in comparison to MIP and SIP schemes performed by Zeadally and Siddiqui [2007]. Zeadally and Siddiqui conclude that although mSCTP can outperform both schemes in terms of handover delay and throughput, an important shortcoming is the inability to operate in networks that use the Network Address Translator (NAT) devices. During a handover to a network that use NAT (e.g., network with dynamic, private IP addressing), NAT assigns a new port number that causes SCTP association to be dropped. There are already some on-going works devoted to provide SCTP support to NATs [Stewart et al., 2008; Xie et al., 2007].

Among the investigated handover strategies proposals, the most simple approach is to rely on the standard SCTP failover mechanism. This approach has been deeply analyzed in Chapter 4 of this work, here only brief examples will be given. Failover in a handover context is considered by Budzisz et al. [2008, 2006b] who suggest decreasing the default `PMR` and `RTO.Min` values of standard SCTP in order to adjust the failover mechanism to the handover needs. Budzisz et al. conclude that the standard SCTP failover mechanism is completely unsuitable for real-time applica-

[3]In this work, term *handover policy* refers to a handover strategy, together with a handover mechanism to accomplish this strategy. Handover strategy defines an objective to achieve when performing a handover, whereas handover mechanism includes a decision/triggering mechanism (i.e., when to start handover signaling) and an execution/signaling mechanism.

tions, however does not completely rule it out for non real-time applications. A similar approach to failover mechanism is evaluated by Noonan et al. [2006] who propose an additional improvement, a so called association routing tables (ART) at the transport layer. ART are created in a way that each destination has assigned a different source address (if possible), and are synchronized at both multihomed endpoints in order to improve the efficiency of the standard SCTP multihoming in presence of network failures. Also, schemes like WiSE [Fracchia et al., 2005, 2007] or AISLE [Casetti et al., 2006], although defined to optimize failover detection within a context more related to robustness, can serve as valid handover solutions, too.

Already at an early stage the researchers looked for a possible lower-layer support in the handover decision-making process. Chang et al. [2004] extends the proposal introduced by Koh et al. [2004] to use link-layer signal strength information to govern the address manipulation process and trigger the handover. A similar approach is also reflected in work by Budzisz et al. [2005], where all handover-related decisions are based on a relative signal-strength criterion with certain hysteresis. An interesting finding is made in [Budzisz et al., 2008] where the authors show that an inappropriately adjusted handover decision mechanism based on a link-layer information, may result in a decrease of the performance, as compared to a handover policy based on the standard SCTP failover. An example of a more complex design, based on collecting the link-layer events (e.g., interface up/down, address manipulation request) and processing them in the handover decision module at the transport layer, is shown by Kim et al. [2006]. An alternative approach to handover triggering is presented by Chang et al. [2007], where the handover decision is based upon information on available wireless bandwidth calculated using link-layer information, and contention probabilities obtained from the periodically sent heartbeat probes.

Another way to improve the mSCTP-based handover performance is to introduce upgrades during an on-going handover process. One of the early studies by Kashihara et al. [2004], proposed severe changes to mSCTP including changes to the congestion control, the path error accounting algorithm and the corresponding `PMR` limits, and finally a modification of the retransmission mechanism to duplicate packets when more than one path is available. The idea of sending duplicate packets among simultaneously available paths has also been discussed in works by Aydin et al. [2003]; Aydin and Shen [2005] that introduce a scheme called cellular SCTP (cSCTP). cSCTP sets the congestion window on both the old and the newly obtained path to half of the value it had on the old path before the handover, and starts sending duplicate packets. This scheme does not provide any kind of estimation of available bandwidth on the newly obtained path, before starting the transmission. Therefore, a proposal firstly introduced by Goff and Phatak [2004], to use load balancing instead of sending duplicated packets seems to be a more reasonable approach. A more detailed analysis of load balancing in handover context is provided in Chapter 5. Modification of the retransmission mechanism during an on-going handover is also proposed by Ma et al. [2007]. Ma et al. propose the so-called smart fast retransmission mechanism reducing the risk of sending retransmitted packets to an already unavailable path. In contrast, Lee et al. [2006] discusses adjusting the flow control in handover scenarios to minimize abrupt throughput changes during the handover process.

Finally, mSCTP-based handover schemes can be also complemented with resource reservation and efficient QoS provisioning as demonstrated in a recent proposal by Ahn et al. [2007].

3.3.2 mSCTP use cases

Before presenting handover scenarios for mSCTP in more detail, an important comment on the naming convention must be made. As already mentioned in section 2.1, the descriptions of the scenarios strictly follow the IETF naming convention defined in [Manner and Kojo, 2004]. According to this naming convention, when addressing the use of mSCTP in 2G/3G cellular systems such as GPRS/UMTS, a base station would be referred to as AP and the role of AR would correspond basically to that of the 2G/3G network gateway (i.e., Gateway GPRS Support Node, GGSN in GPRS/UMTS) in charge of interconnecting the overall cellular network to an external IP packet data network and ultimately providing IP connectivity. In this way, the whole cellular network behaves as a layer-two network (L2-network). On the other hand, the naming convention used on WLANs is already aligned to the IETF naming convention with respect to the AP, but it is interesting to

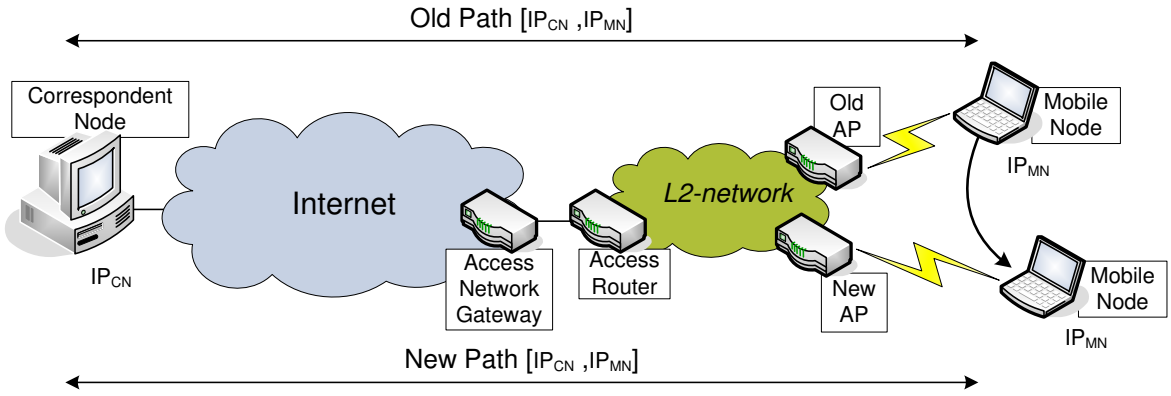


Figure 3.12: Scenario A – The IP address is not changed in the handover process.

remark here that wireless routers (devices with co-located AP and AR functionalities) are the most common to find.

Scenario A. Although it was already pointed out that transport-layer solutions are targeted at scenarios where there is an IP-address change when moving from one PoA to another, we have retained this scenario to emphasize that nowadays the most common situation is that terminal mobility only results in an AP change, while staying in the same IP subnetwork, e.g., intra-system mobility in 2G/3G cellular networks (e.g., GSM or UMTS) or WLAN mobility within the same Extended Service Set (ESS), as presented in Fig. 3.12.

Thus, mSCTP multihoming has no applicability here because the handover does not result in a change of the IP address used in the association. Efficient handover management can be achieved by means of link-layer solutions, so that no specific functionalities are strictly required within the transport layer to cope with the AP change. Nevertheless, despite this possible isolation of the transport layer from the cell-change process under such scenarios, cross-layer design constitutes an appealing research challenge to improve transport-layer performance by means of the information available from lower layers.

Scenario B. Future heterogeneous wireless networks, however, will bring a lot of diversity to the network structure, and terminal mobility among different radio access networks will most likely result in an IP-address change. Under such an assumption, the key feature of this scenario is that terminals can only use a single radio network interface at a time (see Fig. 3.13). This limitation is further referred to, as the single-homed MN.

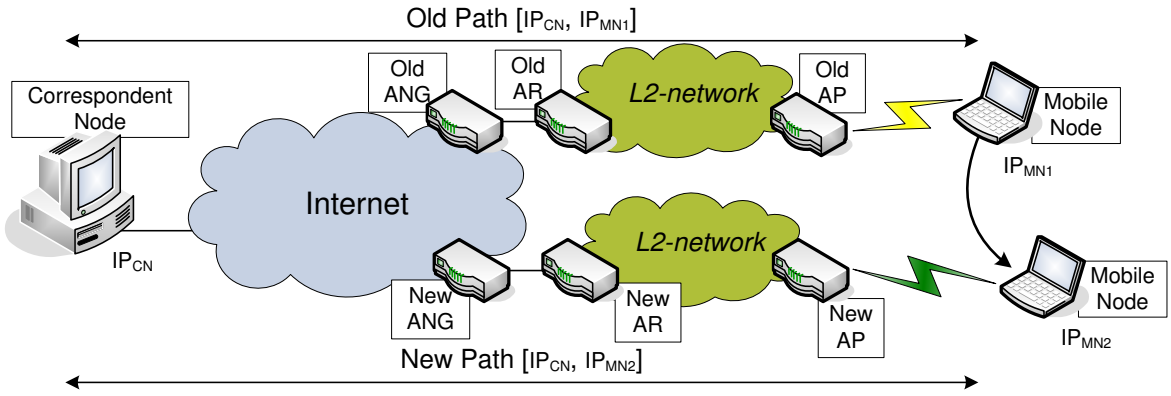


Figure 3.13: Scenario B – Single-homed MN.

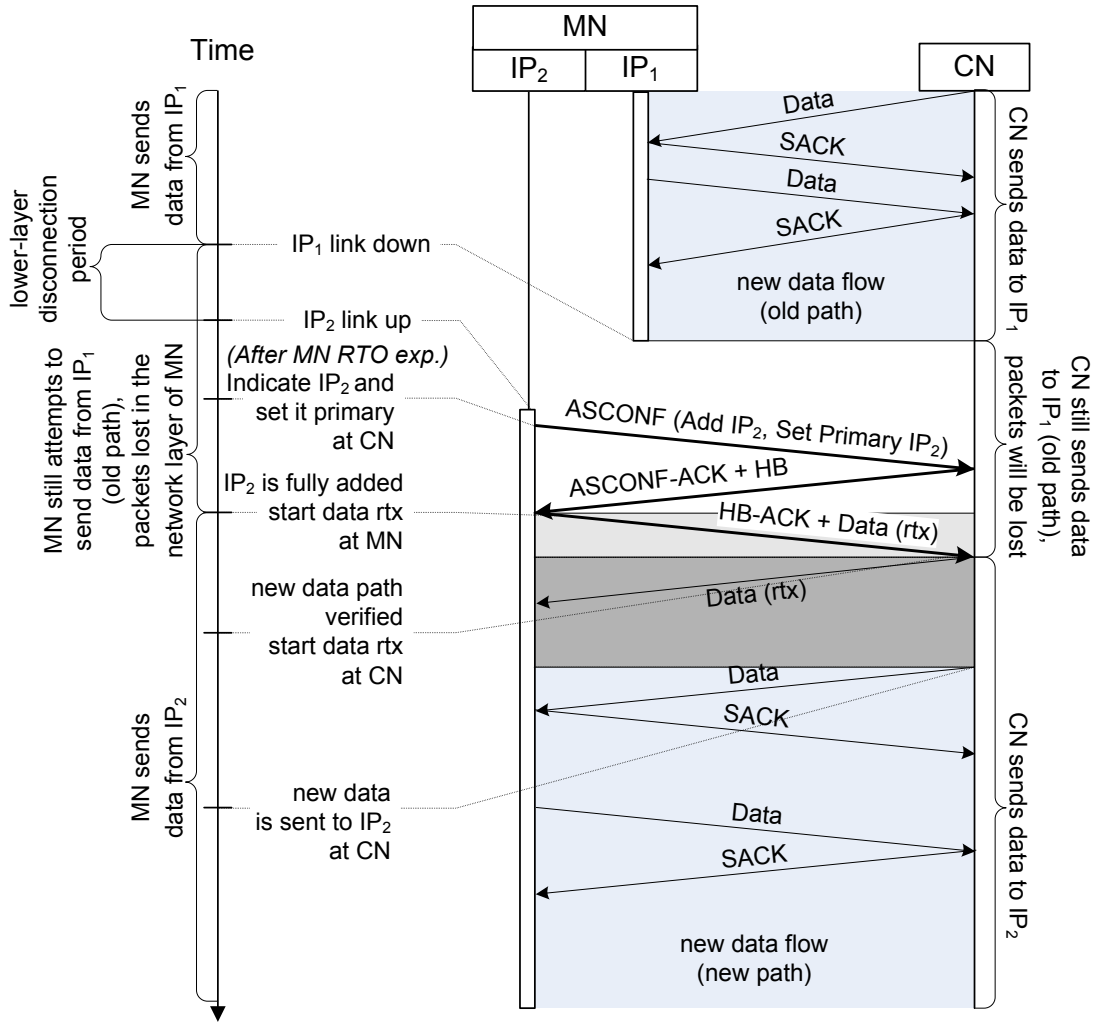


Figure 3.14: mSCTP handover signaling for a single-homed MN (scenario B).

A typical situation in such a scenario is the handover of a WLAN terminal equipped with a single network card between two WLAN APs located in different IP subnetworks. Another situation is a dual-mode terminal (e.g. WLAN/GPRS) that cannot operate both interfaces simultaneously. This single-homed condition implies that the MN is not able to communicate with the new access point when using the old one and vice versa.

mSCTP applied to such a single-homed scenario does not require any infrastructure modifications in expense of the disconnection period required to bring up the new link (notice that both links are not active simultaneously) and configure its IP related settings. Fig. 3.14 illustrates handover signaling scheme based on the current version of the SCTP reference implementation [SCTP-FreeBSD]. The MN disconnects from the old AP, obtains the new IP address from the new AP and sends the ASCONF chunk from that new IP address, even though the new IP address has not yet been included into the association. The packet containing the ASCONF chunk, to avoid being discarded by the receiver (the source address in the IP header is the new IP address), must contain the correct association verification tag in the SCTP-header Verification-Tag field, and the old IP address (the only valid address being part of the association) in the address parameter of the ASCONF chunk. Such a configuration will allow the receiver to recognize the association the chunk belongs to. During the lookup process, the receiver will first check the source address in the IP header. Upon failure of this

attempt (the new address is still not a part of the association), the receiver will eventually get the appropriate IP address from the ASCONF-chunk address-parameter field. The lookup process is followed by the validation of the verification tag, and chunk processing. Bundling of all operations of address manipulation in one chunk is allowed by DAR extension specification [Stewart et al., 2007], and is subject to a handover policy. The only requirements is that the ASCONF chunk parameters should be transmitted in the appropriate order, that is: Add new-IP address, Delete old-IP address, and Set primary address to new-IP address, in order not to be discarded when processing at the receiver. The receiver must then reply to the source IP address of the packet that, after processing all of the ASCONF chunk parameters, is the only IP address included in the association. The described mechanism has strong implications on security, and, as mentioned in Section 3.1.3, an authentication procedure is required before processing the ASCONF chunk to avoid the risk of association hijacking, as described in [Stewart et al., 2007] and [Stewart, Tuexen, and Camarillo, 2007].

The scheme shown in Fig. 3.14 is prone to long handover delays. Apart from the lower-layer disconnection period, also transport-layer contributes to an overall long handover latency. The transport-layer part of the handover latency can be mitigated, if several handover-oriented optimizations are considered. An example of such a handover-optimized scheme is SmSCTP, introduced by Honda et al. [2007], illustrated here in Fig. 3.15. Honda et al. propose to send the ASCONF chunk with the request to add the new IP address, as soon as the new address becomes available, i.e., without waiting for the RTO expiration on the current primary path. This is possible with a slight modification in the approach to the path definition, and thus to the congestion handling. New source address from which the ASCONF chunk will be sent, may be treated as the new path, if only the path is identified by the pair of source-destination address, instead of destination address only, as in case of the reference implementation. Such a modification is still in line with protocol specification, which simply does not require the control of the source address in the path definition. Consequently, retransmitted data can be sent as soon as the new path is available, that is in case of data sent from the MN after the ASCONF-ACK chunk arrives, and for data sent from CN as soon as the new path is verified (when HB-ACK arrives).

Apart from the commented modification, the main challenge in such a handover scenario is to keep system performance during the path-transition phase (i.e., from the moment, the MN has sent a message with the Add IP Address and Set Primary Address option, until the transmission has started on the new address). In particular, to minimize the layer-two disconnection period, and consequently avoid mSCTP timeout retransmissions being sent on the already inactive path.

Alternatively, to overcome limitations of the single-homed scenario, the radio access network can contain an extension facilitating the acquisition of a new IP address through the old AP. An example of such an extension is the Candidate Access Router Discovery (CARD) protocol that provides communication with the new AP through the old AP in order to obtain the new IP address, as shown in [Liebsch et al., 2005].

Scenario C. In this scenario, the MN is multi-homed, that is, more than one network interface can be operated simultaneously. However, regarding CN connectivity, a single-homed CN is assumed so that the CN is only reachable through a unique IP address. Fig. 3.16 illustrates this *asymmetric scenario* involving a dual-homed MN connected to a single-homed CN.

As most Internet servers nowadays are configured with only one IP address, this scenario is likely to become the most common in today's heterogeneous landscape. Under such conditions, mSCTP can be applied to provide seamless handover between two APs connected to different subnets. The main phases in the handover process are shown in Fig. 3.17 and explained below:

1. Once a candidate AP has been selected, an IP address valid for the new location must be obtained. New addresses can be obtained via, e.g., DHCP, DHCPv6 or stateless IPv6 autoconfiguration in the new location.
2. The new IP address is signaled to the mSCTP stack of the MN node. The time period when the two different network addresses are effectively available at the transport layer (i.e., including necessary transport-layer signaling before any DATA chunks can be transmitted) will be further referred as a *dwelling time* in this work.

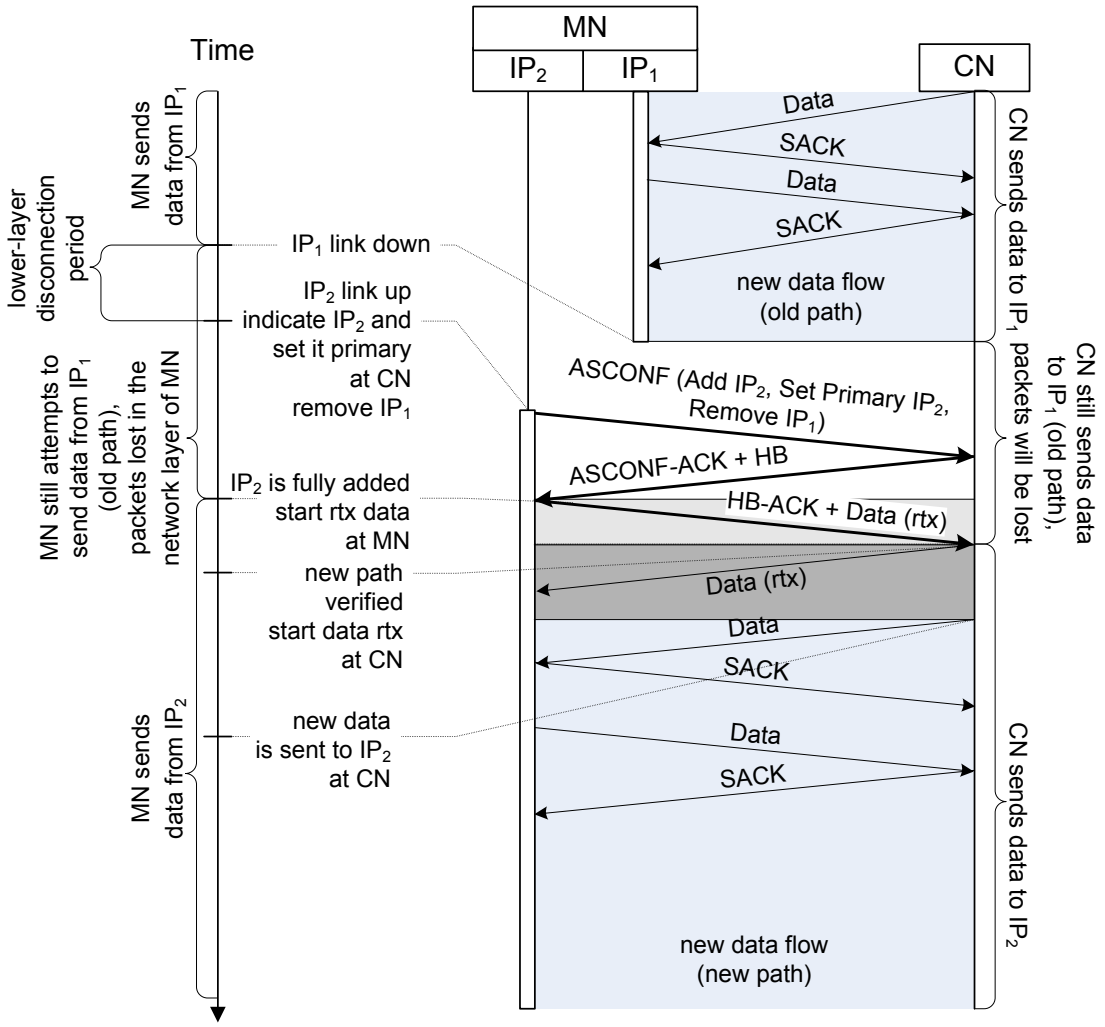


Figure 3.15: Optimized mSCTP handover scheme for a single-homed MN (scenario B).

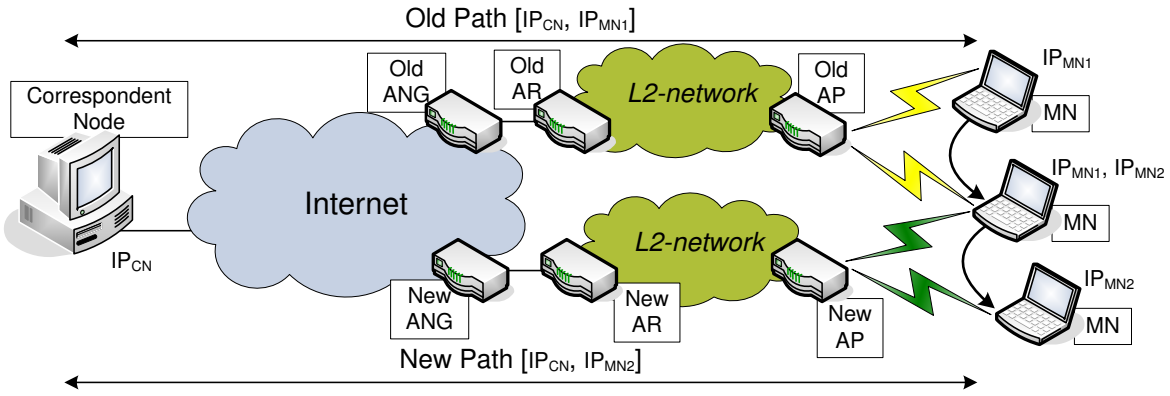


Figure 3.16: Scenario C – Single-homed CN, Dual-homed MN.

3. mSCTP on the MN must notify the CN of the new address — the Add new /IP address request is sent.
4. The CN confirms the incorporation of the new address into the association. As in case of single-homed MN, ASCONF-ACK chunk is usually bundled with the HB chunk used in path verification process. At that time, data from the CN towards the MN is still sent on the old path, as the primary-path change has not been requested yet. On the other hand, once the ASCONF-ACK chunk arrives at the MN, data coming from the MN can be sent to the CN from any of the two available source addresses, depending on MN's routing table configuration.
5. Then, at an adequate moment (possibly the most important open issue is *how to determine when to switch the primary path*) the primary-path change is triggered by the MN, as it is commonly assumed that the handover decision is mainly related to the status of the radio link between the MN and the APs (handover decision can be based on any other criteria, depending on the handover policy that is applied). The primary path change procedure is started by sending an ASCONF chunk with the Set Primary Address parameter pointing to the new path.
6. The CN switches to the new path as soon as the ASCONF chunk with the primary-path-change-request arrives at the CN. The new path had already been verified, so it is used immediately, first to retransmit any chunks that need to be retransmitted (if any), and then to send new data to the MN.
7. Also, as soon as the connection with the old AP is lost, the unnecessary IP address should be removed by means of a Delete IP Address request. Depending on the handover policy, removing of the old IP address can be bundled with the primary-path-change-request. Again it is important to follow the appropriate order of parameters (same as described in a single-homed case) to avoid discarding of the ASCONF chunk.

Consequently, the main challenges in this scenario are related to path management (i.e., criteria to trigger the primary path change) and path transition optimization (e.g., reducing the slow-start phase on the new path, etc.). In that sense the handover policy plays the crucial role in many ways, e.g., (1) handover strategy aims to optimize the use of available resources; (2) handover triggering mechanism may reduce handover latency by indicating the most appropriate time-instance to initiate the handover; and (3) handover execution mechanisms may be simplified by bundling a Set Primary with Add/Delete IP Address parameter within one ASCONF chunk. Inappropriate handover policy may result in degradation of the overall performance. To give the reader more insight on this issue, a range of possible adjustments of handover policies is illustrated with two extreme situations, referred to as *best case* (Fig. 3.18a) and *worst case* (Fig. 3.18b), respectively. The sample handover scenario in the presented examples consists of a MN moving from the slower (IP_1) to faster AN (IP_2). Thus, in such a handover scenario, the best case policy results in the path change, as soon as the new interface is available, whereas the worst case policy waits excessively long, leading to a lower-layer disconnection period, before switching to the new path.

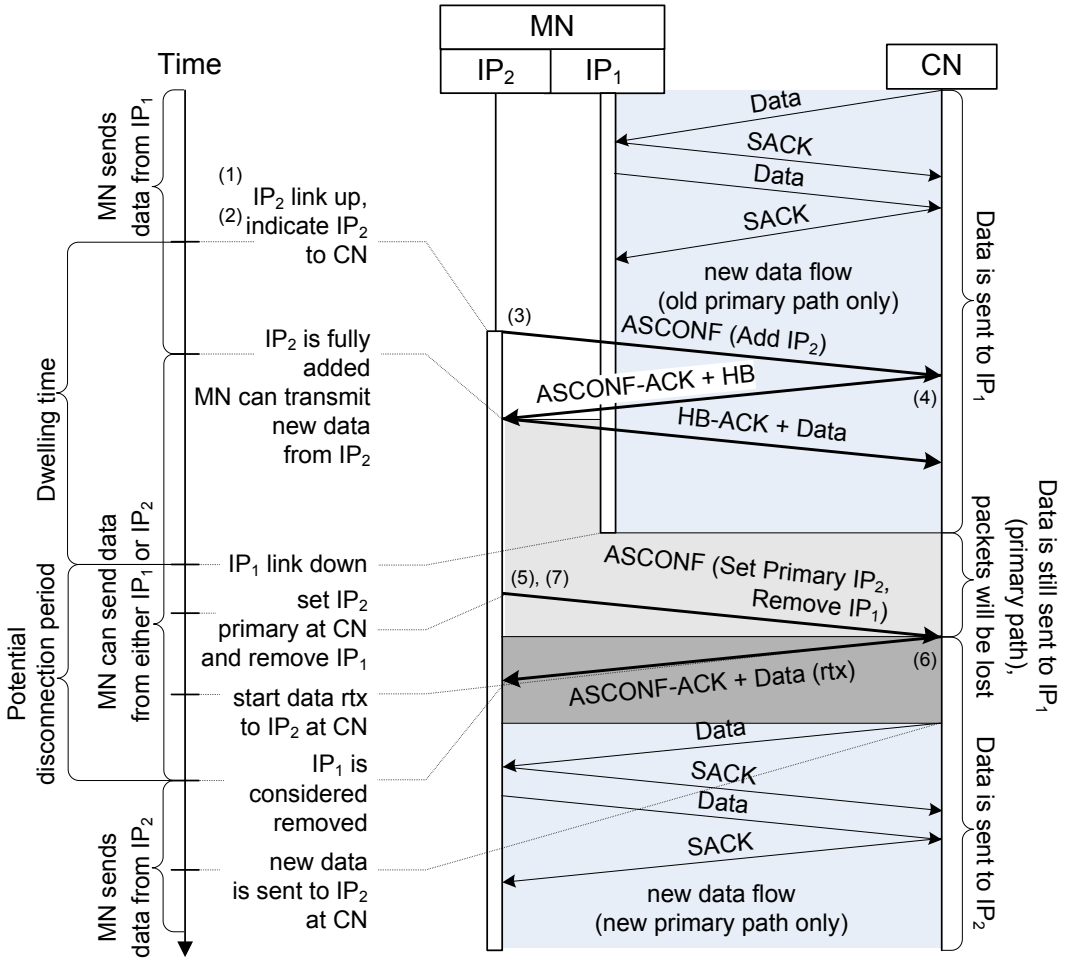
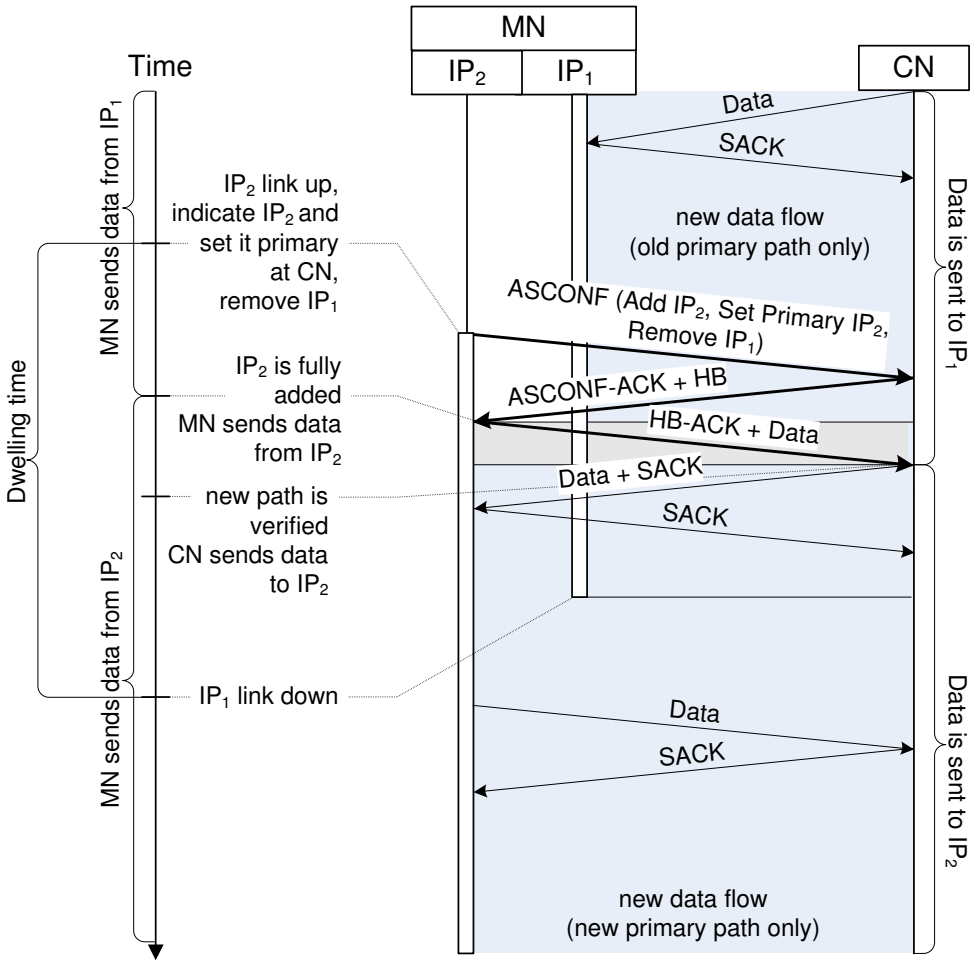
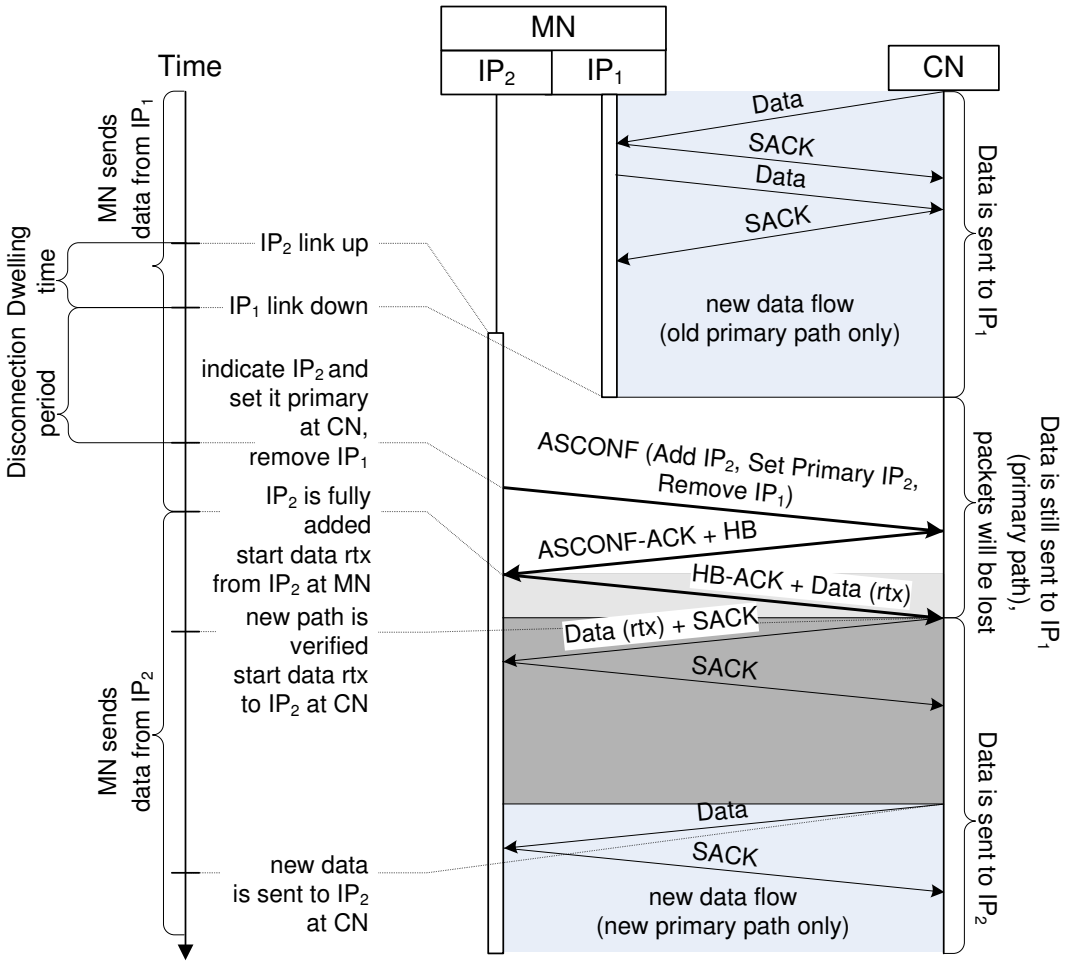


Figure 3.17: mSCTP handover signaling for the asymmetric scenario (scenario C).



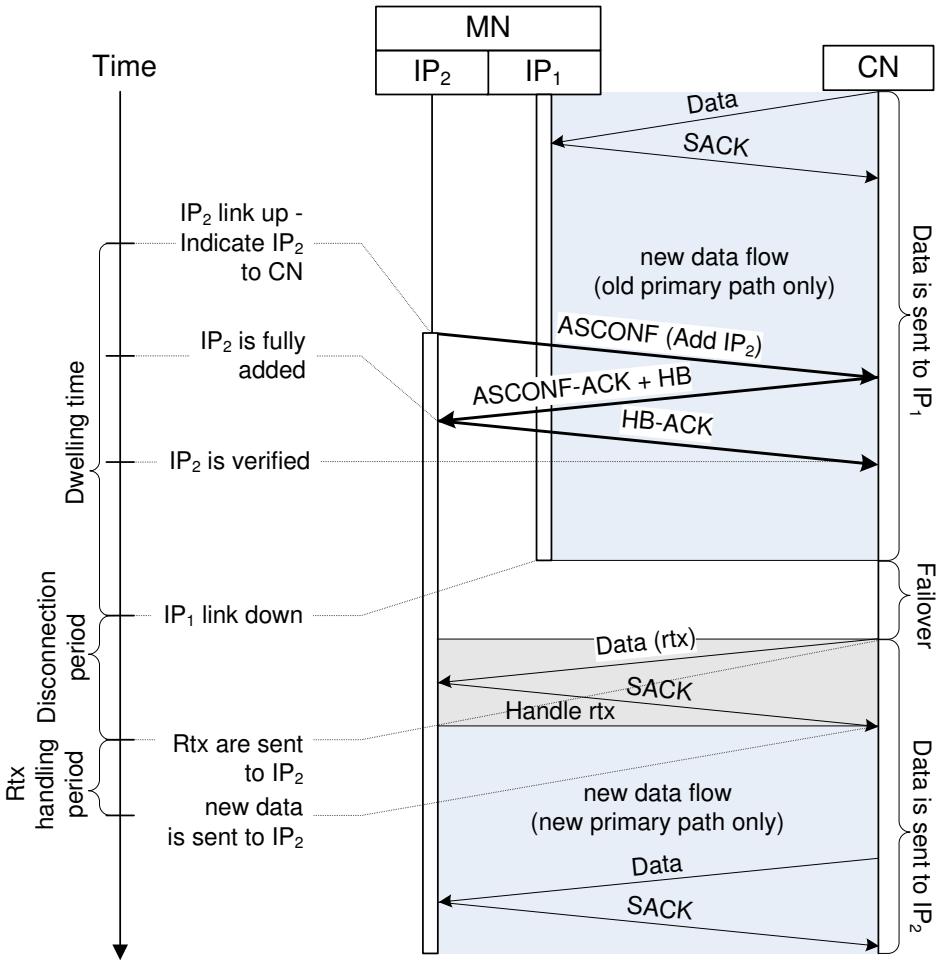
(a)

Figure 3.18: mSCTP handover scheme for the asymmetric scenario with different handover policies (scenario C): (a) best case.



(b)

Figure 3.18: mSCTP handover scheme for the asymmetric scenario with different handover policies (scenario C): (b) worst case.



(c)

Figure 3.18: mSCTP handover scheme for the asymmetric scenario with different handover policies (scenario C): (c) failover-based policy.

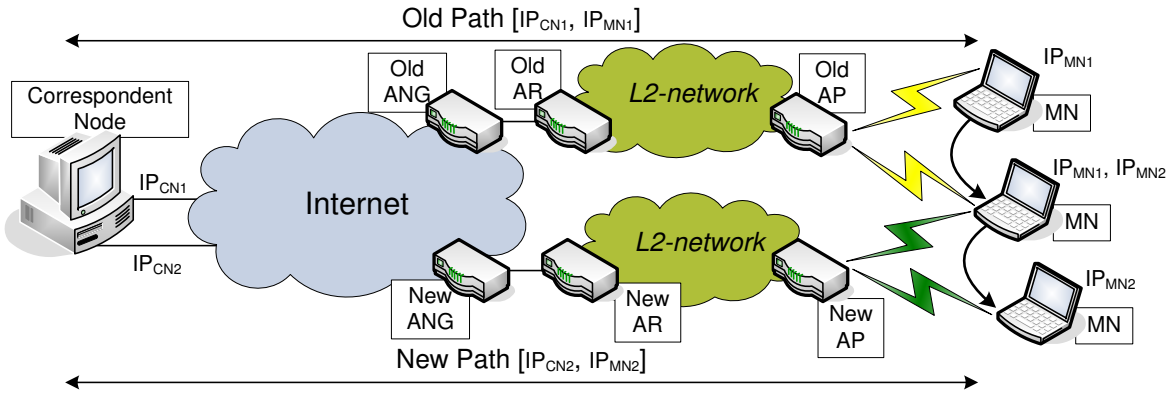


Figure 3.19: Scenario D – Dual-homed CN, Dual-homed MN.

Alternatively, mSCTP handover policy may be based on the standard SCTP failover mechanism, as shown in Fig. 3.18c), what will be explained and evaluated in detail in Chapter 4. Note however that the usage of SCTP's failover mechanism to trigger the primary path change impedes sending data while switching the paths.

Moreover, a loadsharing scheme, e.g., the Concurrent Multipath Transfer (CMT) [Iyengar et al., 2006] can also be exploited in this scenario in the downlink direction. During the dwelling period, the CN can wisely transmit new data towards more than one of the MN's IP addresses included in the association. In the uplink direction, however, the mSCTP association has only one path leading to the unique CN IP address, and as a result loadsharing can not be employed within the mSCTP scheme. Loadsharing issues will be discussed in Chapter 5 in more details.

Scenario D. In order to employ loadsharing in both the uplink and the downlink direction, a multi-homed CN must be considered in a *symmetric scenario* as presented in Fig. 3.19. Handover signaling scheme will be similar to the one presented for scenario C (asymmetric scenario), offering better flexibility as a consequence of having multiple paths between the endpoints. Indeed, the exploitation of multiple interfaces may be useful not only in mobility context, but also in case of degraded performance of the active interface (due, e.g., to congestion).

Symmetric scenario encounters the same open points (triggering criteria) as in the single-homed fixed server case (scenario C), i.e., there will be room for the performance improvements, such as transport-layer performance optimizations (path selection, slow-start-phase reduction) that will be discussed in Chapter 5.

Summary

Table 3.3 summarizes the discussion presented in this section. For the analysis presented further in this thesis, the scenario with single-homed CN and multi-homed MN (scenario C) has been chosen as being the most representative for todays networking.

3.4 Conclusions

SCTP very quickly migrated from the signaling to a general purpose transport protocol. Two new features multihoming and multistreaming, together with a range of well-known features adopted from TCP, made the SCTP a really versatile proposal. In part thanks to multihoming, it is also attractive to the wireless scenarios, and capable of providing handover management at the transport layer, after incorporating the DAR extension. The most important scenarios in the context of handover management were described in Section 3.3.2 to provide the reader with essential information for the analysis presented in the following chapters.

Table 3.3: Summary of mSCTP application scenarios.

Feature	Scenario A	Scenario B	Scenario C	Scenario D
IP address change during handover	No	Yes	Yes	Yes
Number of active interfaces on the MN	Multihoming not available at the transport layer	One (Single-homed	Many (Multi-homed)	Many (Multi-homed)
Number of IP addresses on the CN	Not important	Not important	Single-homed	Multi-homed
Typical scenarios	Intra-system handover in 2G/3G Cellular or WLAN networks	Inter-system handover in heterogeneous networks	Inter-system handover in heterogeneous networks	Inter-system handover in heterogeneous networks
Main open issues	Cross-layer design to enhance transport-layer performance during handover process (e.g., link-layer short interruption, delay spikes, packet bursts losses)	— IP configuration of the new link without affecting the SCTP operation at the current link — Path transition optimization	— Loadsharing on downlink only — Handover triggering criteria — Path transition optimization	— Loadsharing on up-/downlink — Handover triggering criteria — Path transition optimization

Chapter 4

Failover as a basic handover scheme

Standard SCTP provides a failover mechanism to manage transmission over multiple paths when available, increasing therefore protocol robustness in presence of link failures. Nevertheless, limiting the use of multihoming strictly to the failover upon link failure event seems a drawback of the standard SCTP. Already SCTP failover itself can be reused for other purposes, e.g., as a basic mechanism triggering handover in a mSCTP handover scheme, as shown in Fig. 3.18c. This approach means that switching among the multiple paths between the endpoints is only conducted attending to the information available at the transport layer (e.g., no triggering conditions from lower layers are used). This interesting idea will be evaluated in this chapter quantifying the feasibility of the proposed solution for heterogeneous wireless networks. Before that, a detailed description of the failover mechanism will be given, providing the reader with the necessary background for the presented analysis.

4.1 Description of the SCTP failover mechanism

Reliability in SCTP is forced with an important protocol feature, multihoming. To detect a path failure each multihomed SCTP endpoint uses both implicit and explicit probes. Transmitted data serves as an implicit probe used to monitor the availability of the primary path. SCTP keeps an error counter that counts the number of consecutive timeouts. If the error counter reaches a certain tunable threshold, `Path.Max.Retrans` (`PMR`), the primary path is considered unavailable. However, if a SACK chunk is received before the error counter reaches `PMR`, the error counter is reset to zero. A persistent failure to reach the primary destination eventually induces a failover, at which the source endpoint selects one of the available alternate paths as a new primary path.

Since no data chunks are normally sent on the alternate paths, SCTP uses explicit probes, called HEARTBEATs to monitor the availability of such idle paths. HB chunks are periodically sent on the alternate paths at a rate governed by a tunable heartbeat timer, in function of $\texttt{Heartbeat}_{\texttt{Interval}}$ ($\texttt{HB}_{\texttt{Int}}$) parameter. If a HB-ACK chunk is not received before the heartbeat timer expires, an error counter is incremented. Again, if the error counter reaches `PMR`, the corresponding alternate path is considered unavailable.

Subject of a discussion in this chapter, the standard SCTP failover mechanism, illustrated in Fig. 4.1, is based on the retransmission timer (T3-rtx timer) derived from the TCP, together with its managing rules, defined in [Paxon and Allman, 2000]. The retransmission timer is used to clock every data chunk sent to the corresponding peer in order to guarantee a reliable delivery. Upon transmission of a new data chunk the T3-rtx timer is set to the value of RTO, based on the current RTT measurements, according to the formula (4.2).

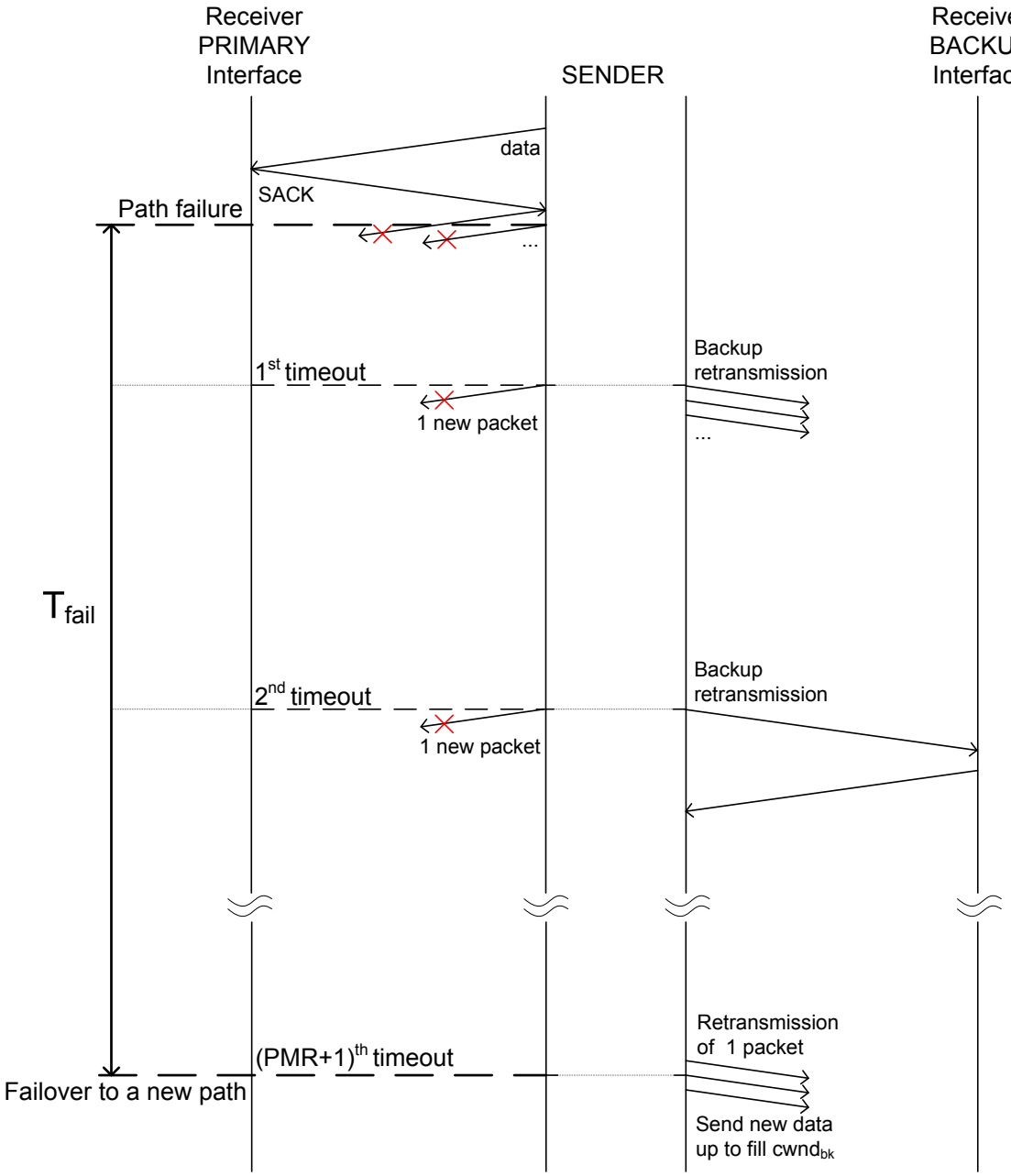


Figure 4.1: SCTP failover mechanism.

Before the first RTT measurement is made: $RTO = \texttt{RTO}_{\texttt{init}}$ (4.1)

If at least one RTT measurement (R′) is made:

$$RTO = SRTT + 4 \cdot RTTVAR \quad (4.2)$$

where:

with first measurement:

$$SRTT = R' \text{ and } RTTVAR = R'/2 \quad (4.3)$$

with any subsequent measurement:

$$SRTT = (1 - \alpha) \cdot SRTT + \alpha \cdot R' \quad (4.4)$$

$$RTTVAR = (1 - \beta) \cdot RTTVAR + \beta \cdot \left| SRTT - R' \right| \quad (4.5)$$

If the T3-rtx timer expires, and the data chunk has not yet been acknowledged by the remote peer, a so-called retransmission timeout (RTO) event, it is assumed that the chunk is lost, and the actual RTO value for the affected path is doubled (*exponential back-off mechanism*). The lost chunk, together with all the chunks that were in transition (in flight) towards the corresponding peer in the moment the T3-rtx timer expired, are marked for retransmission, while the outstanding data counter for that path is reset to zero. Then, SCTP starts retransmitting lost chunks according to its retransmission policy. The SCTP specification [Stewart, 2007] recommends sending timeout retransmissions on the alternate path, provided, of course, that there is an alternate path available. The first retransmission includes all marked chunks that fit into a single data packet. Remaining marked chunks are retransmitted as soon as the congestion window on the alternate path allows it. In order to clock the retransmitted data, the T3-rtx timer on the alternate path is restarted, with the current RTO value of that path. This is the main difference with respect to TCP, which has to send all retransmissions on the same, unique path. Although retransmissions are sent on the alternate path, new data is still sent on the primary path. Thus if the path failure is persistent, as the one shown in Fig. 4.1, additional timeouts will occur. The maximum number of consecutive timeouts on the primary path is limited by PMR. Once this threshold value is exceeded, the path is considered inactive, and a new primary path is selected among the alternate paths that are currently available[1]. The protocol *fails over* to the selected path (path selection is implementation specific, RFC recommends round robin fashion), and from this point on, all data chunks are sent to the new primary path. Following this discussion, and as seen in Fig. 4.1, the *failover time* (T_{fail}) can be defined as the interval between the time when the primary path becomes unavailable, and the time, at which the first new data packet is transmitted on the new primary path.

In case of a persistent failure on the primary path, the SCTP literature usually proposes a rough estimation of the failover time [Stewart and Xie, 2001]. This rough estimation is based on the sum of consecutive timeout periods, analogously to the back-off timer mechanism of SCTP's ancestor, TCP. Indeed, the sum of the PMR consecutive timeouts, called here *total primary path RTO expirations time* (T_{RTO}) is the most important factor in the SCTP failover time estimation. T_{RTO} is given by equation (4.6), combining the RTO value at the time of the path failure (RTO_{fail}), and both upper and lower bound for the RTO value: RTO.Max ($\texttt{RTO}_{\texttt{Max}}$) and RTO.Min ($\texttt{RTO}_{\texttt{Min}}$) accordingly, as recommended by the SCTP specification [Stewart, 2007].

$$T_{fail} \approx T_{RTO} = \sum_{i=0}^{\texttt{PMR}} \min\Big(\max\left(\texttt{RTO}_{\texttt{Min}}; 2^i \cdot RTO_{fail}\right); \texttt{RTO}_{\texttt{Max}}\Big) \quad (4.6)$$

Obviously, the maximum number of allowed retransmissions (PMR) determines the impact of the exponential back-off mechanism in SCTP. Protocol specifications recommend setting PMR to five, which for the lowest RTO_{fail} allowed by the $\texttt{RTO}_{\texttt{Min}}$ default setting (1 s), yields a 63 second-long failover time. This is unacceptably long for the majority of communication applications. Caro [2005] demonstrates that an improvement of this behavior may be achieved by decreasing the PMR value.

[1] The total number of consecutive timeouts for the entire SCTP association is handled by the Association.Max.Retrans parameter. The SCTP specifications recommend that this value should not be larger than the sum of the PMR values of all the paths of the association.

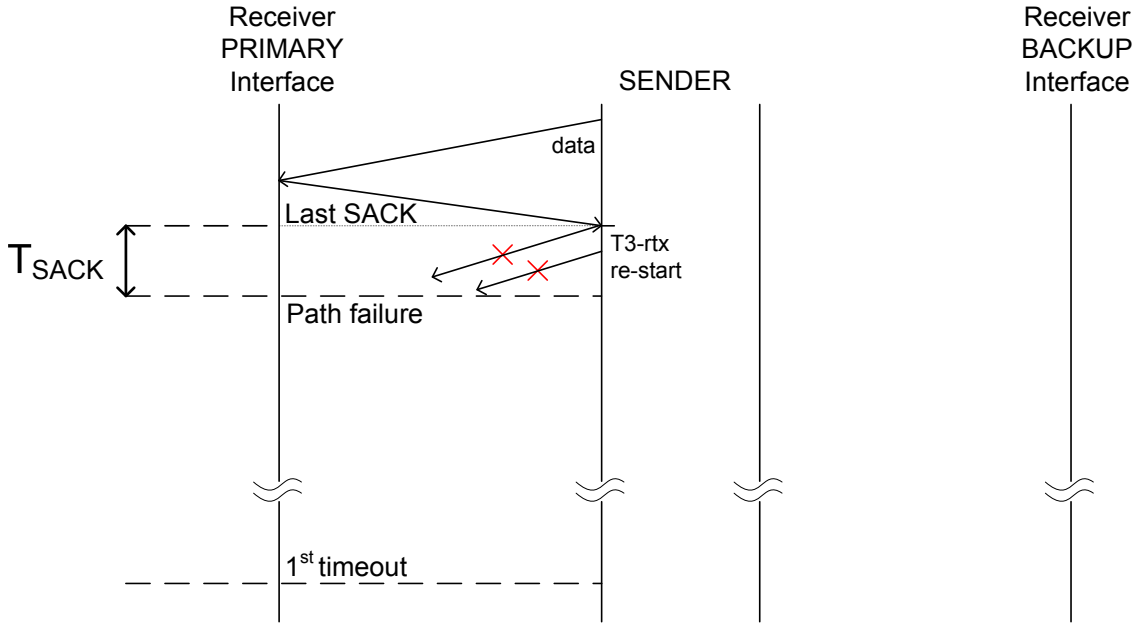


Figure 4.2: The last SACK offset.

As will be shown later in Section 4.3.1, transport layer mobility applications typically should have the PMR value reduced from the default value of five to lower values (i.e., between 0 and 2) in order to achieve shorter failover times. Therefore in such cases, there is a need to take into account additional factors in the estimation of the failover time.

4.2 Reference study: analytical estimation of failover time

The scheme presented in equation (4.6) for the failover time estimation has been applied in most of the SCTP publications so far. In paper [Budzisz et al., 2007] author aims at demonstrating that the accuracy of the estimation based only on T_{RTO} is not always satisfying, and specially in transport layer mobility scenarios. The proposed update seems critical, mainly because of the two following reasons: the lack of an exact indication of the moment when the failure occurred, and the ambiguity of the failover definition in the protocol specification. In this section both aspects will be examined in more detail. This reference study will also serve to bring the reader more details of the failover scheme.

The T3-rtx timer manages the data sending process and is re-started each time, a SACK arrives that acknowledges data with the earliest outstanding TSN (rule R3 in RFC 4960 [Stewart, 2007]). Thus, it is important to relate the moment of the path failure to the instance when the last SACK was successfully received on the primary path, and the RTO expiration accounting was started, as illustrated in Fig. 4.2. Hereafter this time will be called the last SACK offset (T_{SACK}). Depending on the network state at the moment of failure, T_{SACK} may be either negative (the T3-rtx timer was restarted for the last time before the link failure) or positive the last T3-rtx timer reset took place after the path failure, as some SACKs still managed to arrive at the sender).

Another reason for the estimation update arises from the ambiguity of the definition of the failover mechanism in the SCTP specification. The intention of the protocol authors [Stewart, 2007] was that SCTP should be able to send new data packets to the primary path simultaneously with the retransmission of the packets that timed out, handled on the alternate path, as it was presented in Fig. 4.1 in Section 4.1. Since the main scope of the RFCs developed by the IETF is the protocol specification, but not the implementation issues, there is no unique way the protocol should behave, as far as the implementation does not violate the specification rules. Author has examined the most important of the existing SCTP implementations [SCTP-FreeBSD; SCTP-LK; SCTP-SS7], as well as a

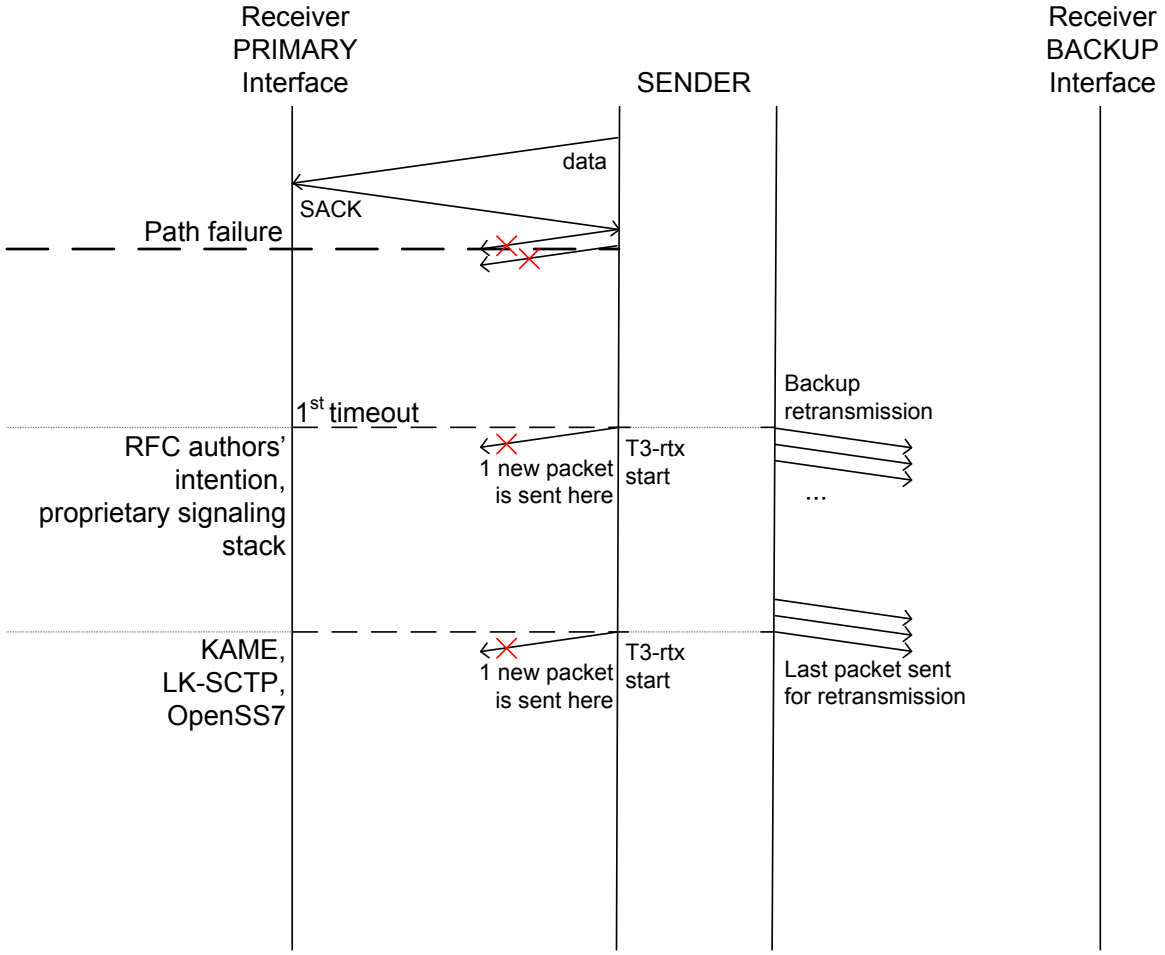


Figure 4.3: Different SCTP implementations behavior.

proprietary signaling stack, and found out that most designers have interpreted the protocol's definition differently than it was intended by the RFC authors. In particular, all major implementations seem to wait until all marked packets are retransmitted on the backup path, before transmitting any new data on the primary path. Both interpretations are compared in Fig. 4.3.

The explanation of such implementations behavior lies in the use of a single output queue. All analyzed implementations have only one output queue, where both types of packets (those marked for retransmission and the new ones) are directed. Further, as protocol specifications prescribe, all retransmissions must be sent prior to any new data, which indeed results in the observed behavior.

Consequently to address this fact, a second new factor should be introduced into the failover time estimation, the total backup path retransmissions time (T_{rtx}), defined as the total time devoted to retransmit pending packets on the alternate path. It should be noted that this factor exists only if retransmissions are allowed (PMR > 0) at the transport layer, and that it is implementation dependent. T_{rtx} may have vital importance in the transport layer mobility scenarios that include long thin networks (LTN), i.e., networks with high RTT values and medium/low bit rates. If retransmissions at the transport layer are allowed, but the PMR value is low (i.e., 1 or 2), the time devoted to retransmissions of the pending packets during the failover period will be relatively long, thus influencing the total failover time.

Summing all up, a new accurate estimation of the failover time was developed in [Budzisz et al., 2007] that can be expressed as the sum of the three aforementioned components, as presented in formula (4.7). It is claimed that this is a general-use model, better reflecting the protocol behavior than the estimation based only on the T_{RTO}. Although applicable to all kinds of network scenarios, equation (4.7) particularly targets long-thin networks. Notably, in scenarios with high PMR values,

equation (4.6) gives a sufficient estimation of T_{fail}, as both newly introduced factors will have less relevance due to the dominant role of the exponential back-off mechanism.

$$T_{fail} = T_{RTO} + T_{SACK} + T_{rtx} \tag{4.7}$$

4.2.1 Best-worst case analysis

Analyzing the proposed estimation formula (4.7), it must be emphasized again that the most important factor is T_{RTO}. However, once the number of available retransmissions (`PMR`) is known, and the decisive T_{RTO} factor determined (4.6), the overall failover time can be considered as a function of the two newly introduced factors. From that perspective, the best case (the shortest failover time) corresponds to the event when the last SACK is received at the earliest possible moment before the failure occurs ($T_{SACK} < 0$), and the new data packets are sent simultaneously with the retransmissions ($T_{rtx} = 0$). In this case, the lower limit for the last SACK offset can be estimated as the time necessary for data packet and its corresponding SACK to traverse the primary path (the round trip time, RTT) plus the value of `SACK delay`, due to the delayed SACK mechanism used by SCTP (the receiver sends a SACK every second packet received, see Section 4.3.1 for more details). So if there is only one packet correctly received, and the primary path failure occurs right before the corresponding SACK is about to be received at the sender, this produces the largest time interval between the last SACK received and the moment of the path failure. Next if the worst case is considered, the last data packet received correctly should arrive at the receiver shortly before the primary path failure occurs. Then the corresponding SACK would still be able to traverse the path, increasing therefore the last SACK offset to half of the RTT ($T_{SACK} > 0$). However, more important here is the influence of the backup path retransmission time. If the SCTP implementation waits until all packets pending retransmission are sent on the backup path, the resulting failover time will be considerably larger. Then, T_{rtx} can be estimated as a function of the number of packets pending retransmission, and the RTT expressed by means of network parameters, such as available bandwidth and latency. An example of such estimation will be presented in Section 4.2.2, whereas here the T_{fail} best-worst case analysis is summarized in Table 4.1.

Table 4.1: Failover time estimation boundaries

Parameter	Best case	Worst case
T_{SACK}	$-(\texttt{SACK delay} + RTT)$	$\frac{RTT}{2}$
T_{rtx}	0	$f(packets\ pending\ rtx; RTT)$ see example in Section 4.2.2

4.2.2 Estimation example

In this section, an example of an accurate estimation of T_{fail} will be developed applying a more detailed analysis to the general-use model introduced before. Such accurate failover time estimation is especially important when using the SCTP multihoming feature as the basis for achieving transport layer mobility in wireless networking scenarios, an application field being subject to intensive research nowadays. The reason is twofold. First, most mobile wireless networks can be categorized as long-thin networks. Second, an accurate knowledge of the transition time between available paths is one of the key prerequisites for handover optimization.

A common scenario, when analyzing SCTP multihoming performance in the context of transport layer mobility, consists of two paths (primary and backup) between two network hosts, e.g. a mobile node and a server, as depicted in Fig. 4.4. Under such scenario it is assumed that the server is sending data to the mobile node that is initially reachable through two IP addresses. Suddenly, due to the problems on the wireless link or as a consequence of user mobility, the address used as the primary becomes unreachable, and the server has to fail over to the alternate path using the standard SCTP failover mechanism.

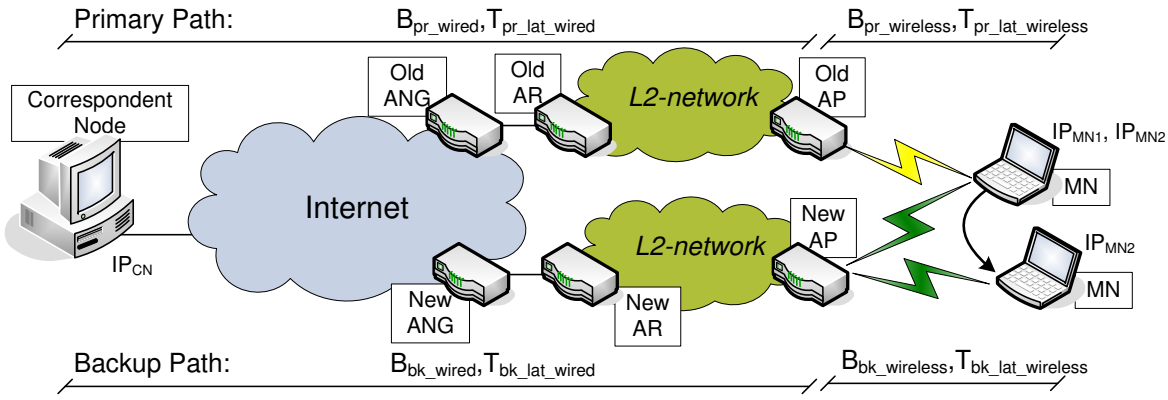


Figure 4.4: Simulation scenario.

In the presented example, the proposed estimation of the failover time is based on the following parameters that model the overall network behavior:

- Link bandwidth for the primary and backup paths denoting separately the wired (B_{pr_wired} and B_{bk_wired}), and wireless parts ($B_{pr\_wireless}$ and $B_{bk\_wireless}$),
- Latency corresponding to the wired and wireless (bottleneck) part on each of the paths (primary path: $T_{lat\_pr\_wired}$, $T_{lat\_pr\_wireless}$, and backup path: $T_{lat\_bk\_wired}$, $T_{lat\_bk\_wireless}$, respectively). Moreover, the total path latency for each path is also indicated (T_{lat_pr} and T_{lat_bk}).

In addition, the following two parameters are assumed to be known at the time of the path failure:

- Congestion window ($cwnd_{fail}$),
- Retransmission time out value (RTO_{fail}).

It is also assumed that the receiver window size (rwnd) is not affecting the failover performance, i.e., it is not limiting the number of packets retransmitted on the alternate path.

Now, when examining the first of the two introduced factors, T_{SACK}, it is important to relate the moment of the path failure to the instance when a SACK was received for the last time, and the RTO expiration accounting was started. In the instance, when the path failure occurs, all SACKs in flight that already managed to pass the bottleneck (it is reasonable to assume that the failure happens in the wireless part) will arrive at the sender, and will cause a re-start of the T3-rtx timer. As follows from the discussion in Section 4.2, the accuracy of T_{SACK} depends on the SACK generation interval. Aiming at the average value of T_{SACK} in the presented estimation, a half of the SACK generation rate is used to obtain formula (4.8). In this equation L_{MTU} and L_{SACK} denote the size of data packets and SACK packets, respectively. As can be seen, presented example fits in the estimation boundaries described in Section 4.2.1.

$$T_{SACK} = \left(T_{lat\_pr\_wired} + \frac{L_{SACK}}{B_{pr\_wired}} \right) - \frac{L_{MTU}}{B_{pr\_wireless}} \tag{4.8}$$

The second of the newly introduced factors, the estimate T_{rtx} is formulated in expression (4.9) by means of two separate components. The first one ($T_{1^{st}\_rtx}$) deals with the estimation of the time spent for the first timeout retransmissions on the backup path, whereas the latter component (T_{other_rtx}) sorts out the time devoted to the remaining consecutive timeout retransmissions until the threshold `PMR` is reached.

$T_{1^{st}\_rtx}$ period is completed, as soon as the last packet marked for retransmission is sent on the alternate path, and new data may be sent on the primary path, restarting therefore the T3-rtx timer. Then, in each of the following consecutive timeout retransmissions only one packet is marked for retransmission. Thus, in these periods new data can be immediately sent on the primary path, and the retransmission timer on the primary path restarted accordingly. Therefore T_{other_rtx} is equal to zero, and the formula estimating T_{rtx} can be reduced to $T_{1^{st}\_rtx}$ only.

$$T_{rtx} = T_{1^{st}\_rtx} + T_{other\_rtx} = T_{1^{st}\_rtx} \tag{4.9}$$

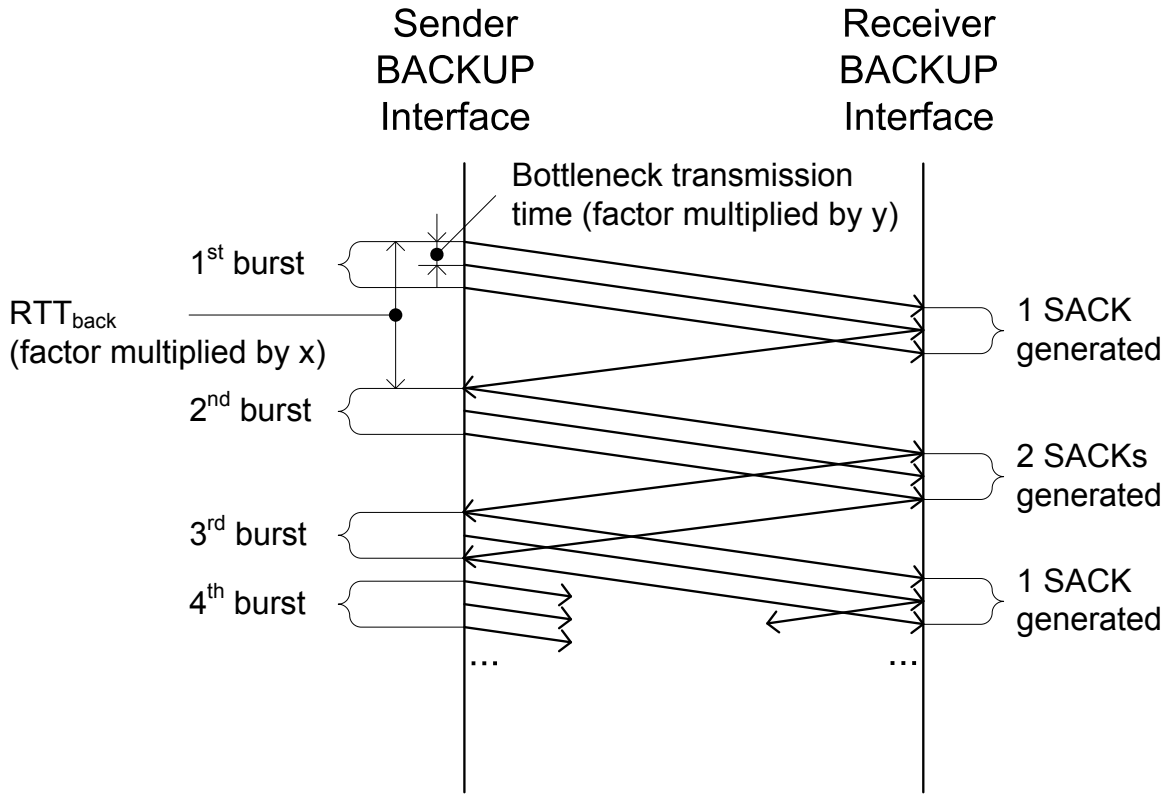


Figure 4.5: First timeout retransmission scheme.

Now focusing on the estimation of first retransmission period. To simplify the calculations, it is further assumed that all the packets pending retransmission will be retransmitted within the slow start phase on the alternate path, so each SACK received at the sender triggers a burst of three retransmitted packets (SCTP by default uses the delayed SACK mechanism). The number of packets retransmitted in the first burst actually depends on the application type that fixes the relation between the packet size (L_{MTU}) and the initial cwnd value on the path that handles the retransmission ($cwnd_{init\_bk}$). Hence, if the bulk file transfer is considered as an example application, the most typical MTU size is 1500 Bytes, which corresponds to three packets sent within the first burst. The described retransmission scheme is depicted in Fig. 4.5.

Consequently, analyzing presented retransmission scheme, $T_{1^{st}\_rtx}$ can be expressed as the function of the number of round trip times (parameter x) and the number of bottleneck transmission times[2] (parameter y), as shown in (4.10).

$$T_{1^{st}\_rtx} = x \cdot RTT_{bk} + y \cdot \frac{L_{MTU}}{B_{bk\_wireless}} \tag{4.10}$$

The parameters x and y are, as pointed up in Fig. 4.5, functions of the number of bursts of three packets, n, that needs to be successfully sent. The factor n, in turn, depends on the number of outstanding packets at the time of the path failure, which is assumed to be equal to the congestion window at that time, $cwnd_{fail}$. The equations (4.11), (4.12), and (4.13) provide the formulas for computing n, x, and y, respectively.

[2]Usually the transmission time through the wireless link.

$$n = \left\lfloor \frac{cwnd_{fail} - 1}{3} \right\rfloor \tag{4.11}$$

$$x = \left\lfloor \frac{1 + \sqrt{1 + 8 \cdot (n-1)}}{2} \right\rfloor \tag{4.12}$$

$$y = 2 \cdot n - \sum_{i=1}^{x} i \tag{4.13}$$

The round-trip time on the backup path (RTT_{bk}) in equation (4.10) can be estimated as a function of network parameters, as presented in (4.14).

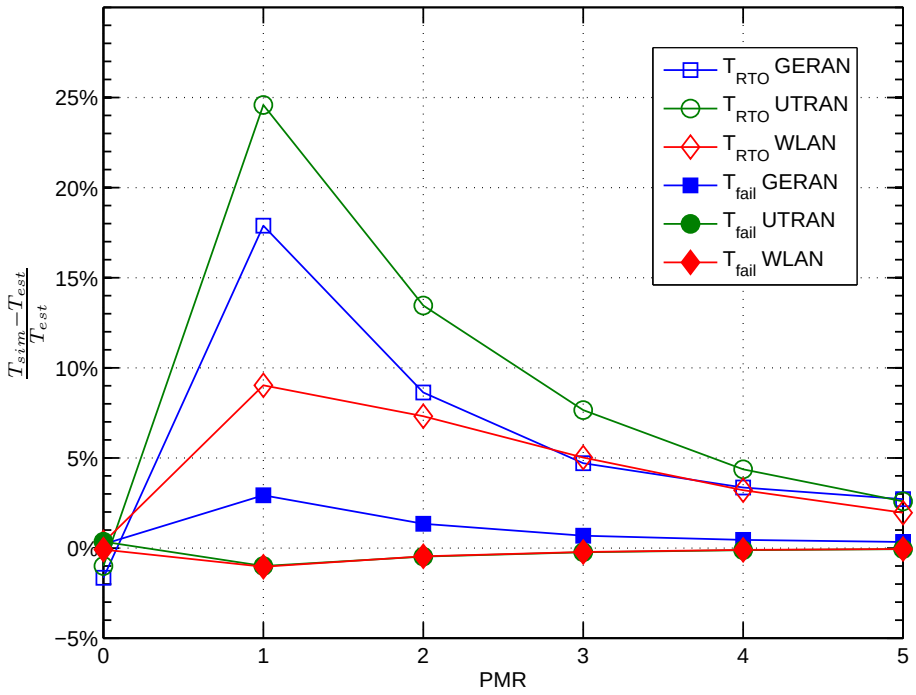
$$RTT_{bk} = 2 \cdot T_{lat\_bk} + \frac{L_{MTU} + L_{SACK}}{B_{bk\_wired}} + \frac{L_{MTU} + L_{SACK}}{B_{bk\_wireless}} \tag{4.14}$$

4.2.3 Estimation verification

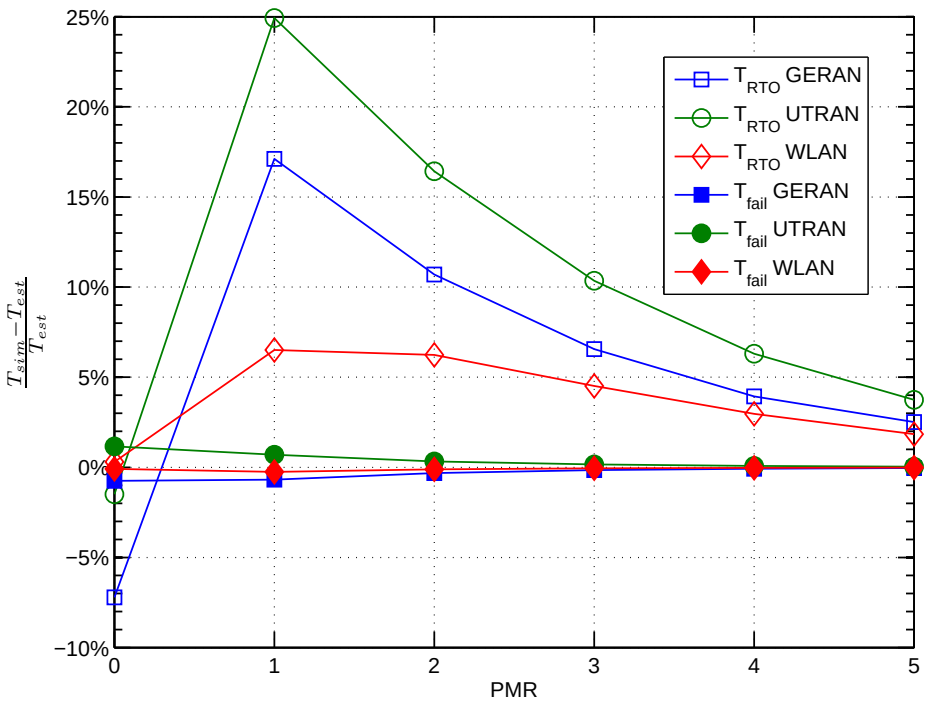
In order to evaluate the proposed formula for estimating the failover time (4.7), and compare it to the one currently used (4.6), the University of Delaware's SCTP model for the ns-2 network simulator [NS-2] (described in more details in Appendix A.2.1) is used, assuming the topology as presented in Fig. 4.4. The wired links have 100 Mbps bandwidth and 5 ms propagation delay. For wireless links three typical LTNs are considered: WLAN (11 Mbps bandwidth and 15 ms propagation delay), UMTS (384 kbps and 80 ms), and GERAN (80 kbps and 80 ms), respectively. Fig. 4.6 presents the outcome of the evaluation, the average estimation error (i.e., relative error between the simulation result T_{sim} and the estimated value of T_{fail}, called here T_{est}) as a function of the PMR parameter. Provided results are for two different queue sizes: one fairly small, 10 packets, and one larger than the bandwidth delay product (BDP) of any of three considered network types, 50 packets.

As shown in Fig. 4.6, the biggest error in the failover time estimation, using the most common formula (4.6), is obtained for PMR $= 1$ or 2 (the typical configuration of SCTP in wireless multihoming scenarios), when the retransmissions on the alternate path have the most significant influence, no matter what the queue size is. The new estimation proposal (4.7) provide considerably better results, having reduced resulting error below 1% for small queue size and below 3% for bigger queue (more packets to retransmit), respectively. Not surprisingly, in area of big PMR values, (i.e., PMR $= 4$ or 5) T_{RTO} completely dominates T_{fail}, hiding the influence of all new factors introduced to the estimation. In such a scenario, the proposal (4.7) gives nearly the exact value of the failover time (the error value below 0,5%), granting slightly better estimation for the smaller queue size again. Yet, there is no need to improve the commonly used formula (4.6), which works well enough (the error value below 3%). Finally, for PMR $= 0$ (i.e., $T_{rtx} = 0$) equation (4.7) also gives an improvement in the failover time estimation, due to the introduction of T_{SACK} component, however, this improvement is rather small, and still better seen for the smaller queue size.

Concluding, a new general-use estimation formula for an accurate estimation of the failover time proposed in Section 4.2 seems crucial in transport layer mobility scenarios based on SCTP multihoming. The estimation commonly used in the literature (4.6), based only on the sum of consecutive timeouts, is not always appropriate, mainly because of the following reasons: (1) the number of allowed retransmissions (PMR) is usually low, reducing the impact of the exponential back-off mechanism, (2) the lack of an exact indication of the moment when the failure occurred, and (3) implementation dependent behavior of new data chunk transmission caused by the ambiguous specification of the failover mechanism. New estimation formula accounts for the influence of network and implementation dependent parameters, and introduces two new factors into the proposed estimation. Both factors have general scope of use, however the focus here was on a typical transport layer mobility scenarios, since this application has great importance for current SCTP research.



(a)



(b)

Figure 4.6: Comparison of the normalized failover time in long-thin networks with the queue size of: (a) 50 packets; and, (b) 10 packets.

4.3 Failover as a basic mechanism to provide mobility

As it was mentioned in Chapter 3, SCTP was primarily designed as a signaling transport protocol over IP. Thus the initial design goal was to meet strict reliability requirements of telephony signaling, and SCTP made it also thanks to the failover mechanism. Nevertheless the application of the failover mechanism and SCTP itself is not limited to transporting signaling messages. Once SCTP was named by IETF a general purpose transport protocol the approach to the failover should be revised accordingly. SCTP failover mechanism can be also treated as a basic mechanism to provide mobility (together with the DAR extension) and as a benchmark for performance evaluation of other, advanced handover solutions based on SCTP that will be presented in the following sections of this work.

4.3.1 Main parameters

To study the feasibility of SCTP failover mechanism for triggering mSCTP-based handover in heterogeneous wireless access networks, the most relevant protocol parameters should be analyzed. Here, with a set of simple experiments the influence of the most important protocol parameters for a failover behavior will be illustrated.

All presented experiments in this section are held in a symmetrical scenario, shown on Fig. 4.7. Both paths (primary and backup) from the sender to the receiver consist of the wired and wireless parts. Wired part on each path has 100 Mbps bandwidth and 5 ms propagation delay, whereas the wireless parts have 11 Mbps bandwidth and 15 ms propagation delay respectively, so the BDP is adjusted to the typical 3G networks values.

Wireless link failure is modeled with an ideal, two-state channel model, shown in Fig. 4.8. Before failure, link is considered fully available, the packet error rate (PER) is 0%, and immediately after the failure all the packets are lost (link becomes permanently unavailable, PER is 100%). In the presented scenario the basic SCTP performance is evaluated, when sending a 16 MB file via FTP, looking for the best possible failover threshold in the overall file transfer time. The initial setting in each experiment comprises of the default set of the protocol parameters, as listed in Table 4.2, and if not stated otherwise, only the parameter under the examination in a given experiment is changed.

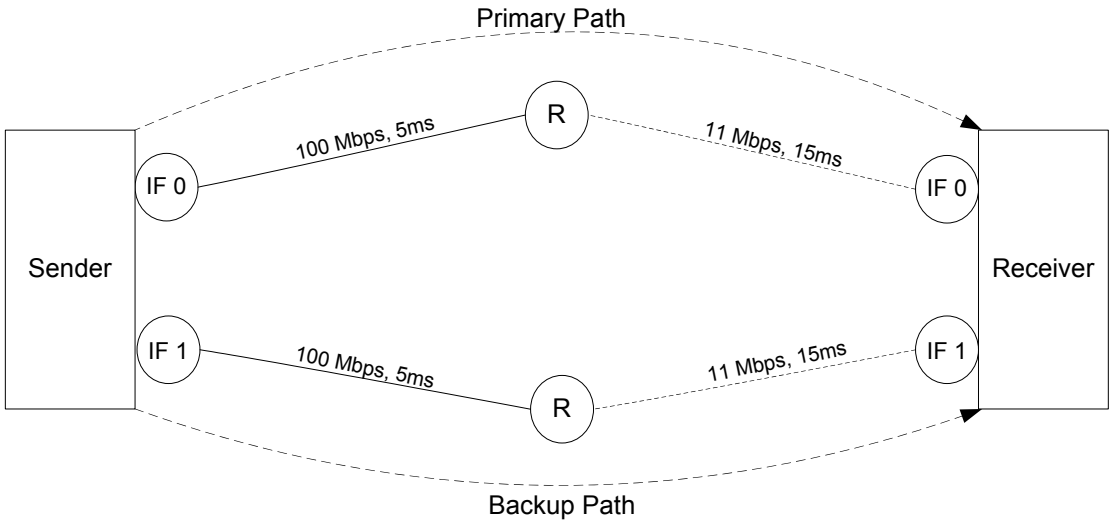


Figure 4.7: Simulation topology.

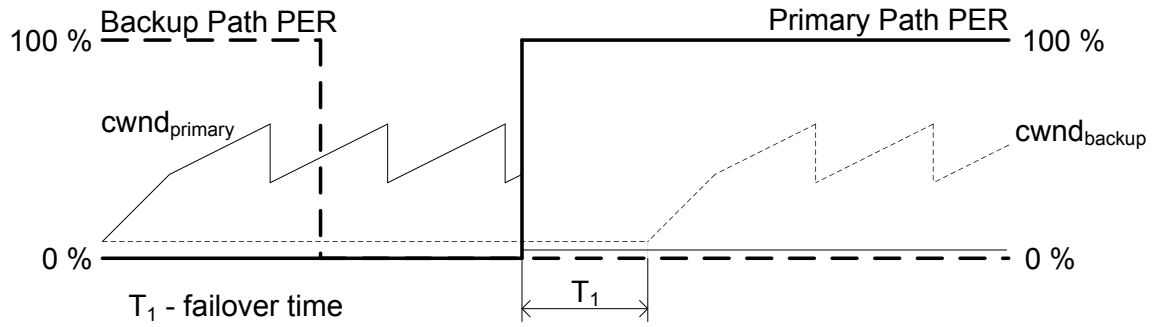


Figure 4.8: Ideal channel model.

Table 4.2: Default set of SCTP parameters

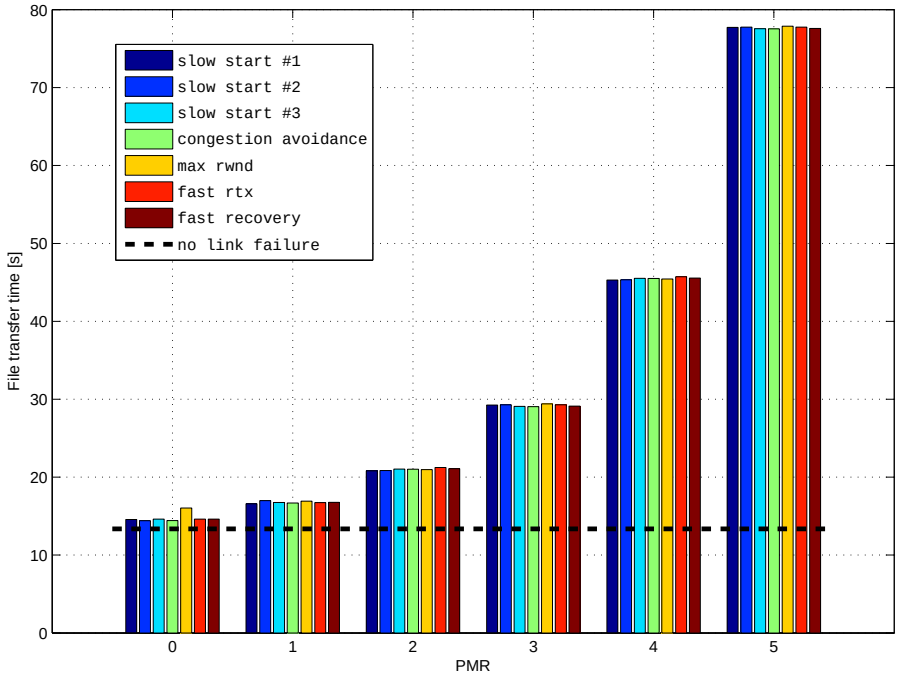
Parameter name	Value
PMR	5
RTOMin	1 s
SACK delay	200 ms
HBInt	30 s
Retransmission policy	FastRtx Same path TimeoutRtx Alternate path

PMR

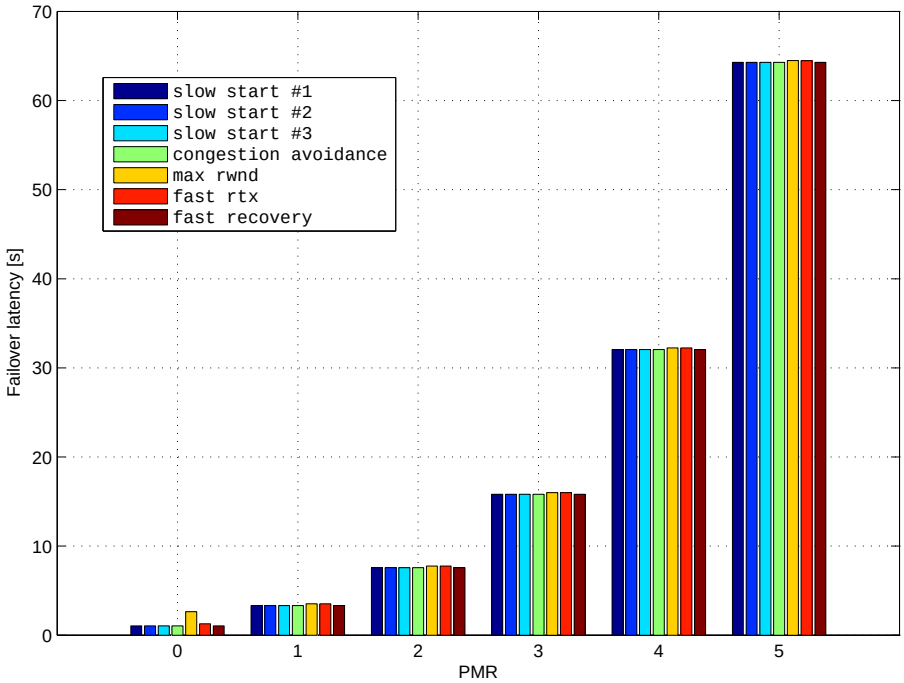
First, and the most important parameter to discuss is the PMR that governs the maximum number of allowed retransmissions and determines the impact of the exponential back-off mechanism in SCTP. Protocol specifications recommend setting PMR to five, which results in unacceptably long failover time (see Section 4.1) for the majority of communication applications. An improvement of this behavior may be achieved by decreasing the PMR value. As it was shown in [Caro et al., 2004a], the faster performance is traded off against a probability of spurious failovers, or even permanent oscillations between the available paths (so called *ping-pong effect*), if the PMR value is decreased too much. More extense study by Caro [2005] devoted to evaluate failover behavior prevailed that using the most aggressive approach (i.e., the PMR set to 0 or 1) albeit may trigger additional spurious failovers not only does not degrade the performance, but counter-intuitively often improve the application goodput.

Results of a set of experiments analyzing the influence of the PMR are presented in Fig. 4.9 in terms of overall file transfer and corresponding failover latency. When examining the PMR parameter, also the influence of the cwnd development in the instance when the failure occurred is evaluated, taking into account three different cases for the slow-start phase (one third, two thirds and full ssthresh value), two cases for the congestion avoidance phase (75% and full rwnd size) and two cases for fast retransmission phase (during fast rtx of the first outstanding packet, and in fast recovery), respectively. All mentioned cases are benchmarked with the transmission without failure.

PMR demonstrates clearly to be the most influencing factor in handover scenarios, if the handover is triggered with a failover mechanism. Transmission time of a 16 MB file in a scenario without loses takes about 13,4 seconds. In the rest of presented scenarios permanent failure on a primary path happens early enough to have a considerable impact on the performance. Still, the default PMR setting results in about six times longer transmission (78 seconds) than without failures, no matter in which cwnd development stage the failure occurs. Obviously, most of that time (about 64 s) is wasted on the failover latency, due to unnecessary retransmission attempts on the path that permanently went down. Only decreasing the PMR value to 0 or 1 can in practice challenge the non-loss case, resulting in 14,7 and 16,7 second-long transmissions, and one and three second-long

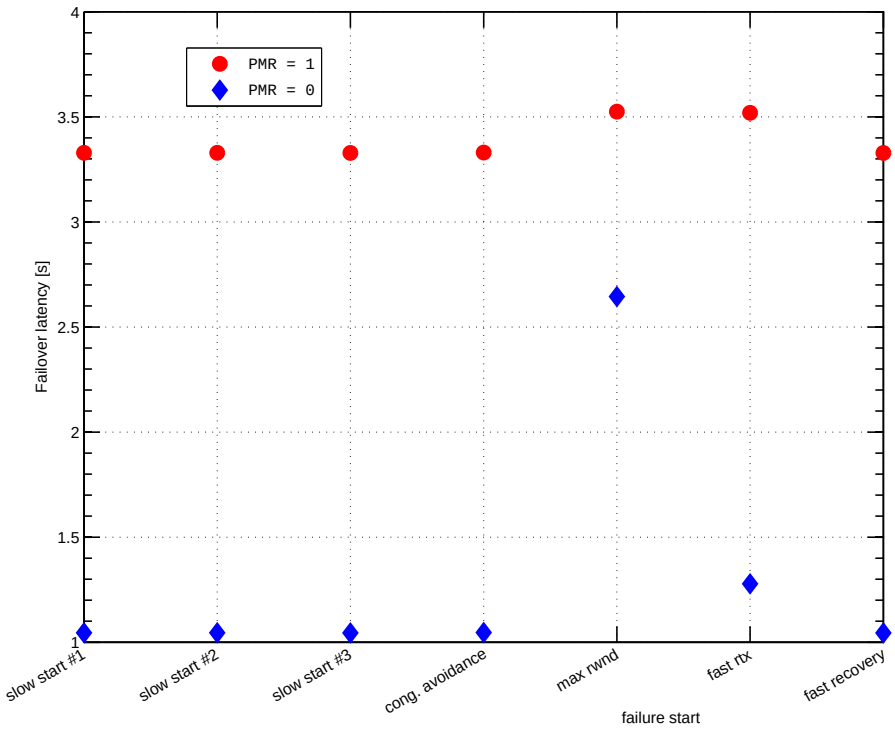


(a)

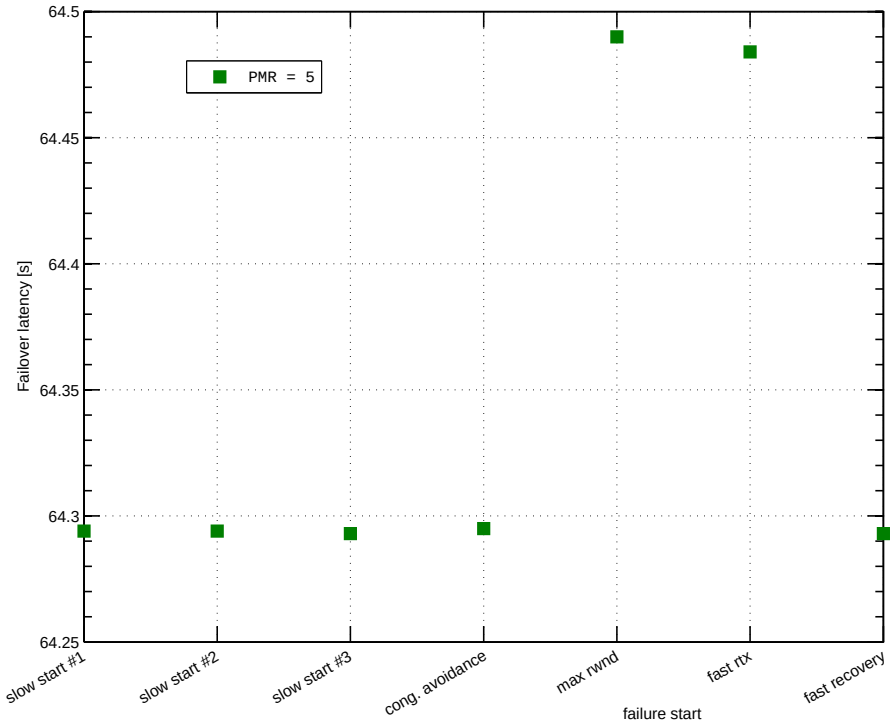


(b)

Figure 4.9: Influence of PMR parameter: (a) file transfer time; and, (b) failover latency.



(a)



(b)

Figure 4.10: Influence of PMR and cwnd evolution in the moment when the failure occurred: (a) for low PMR values; and, (b) for default PMR value.

failover delays, respectively.

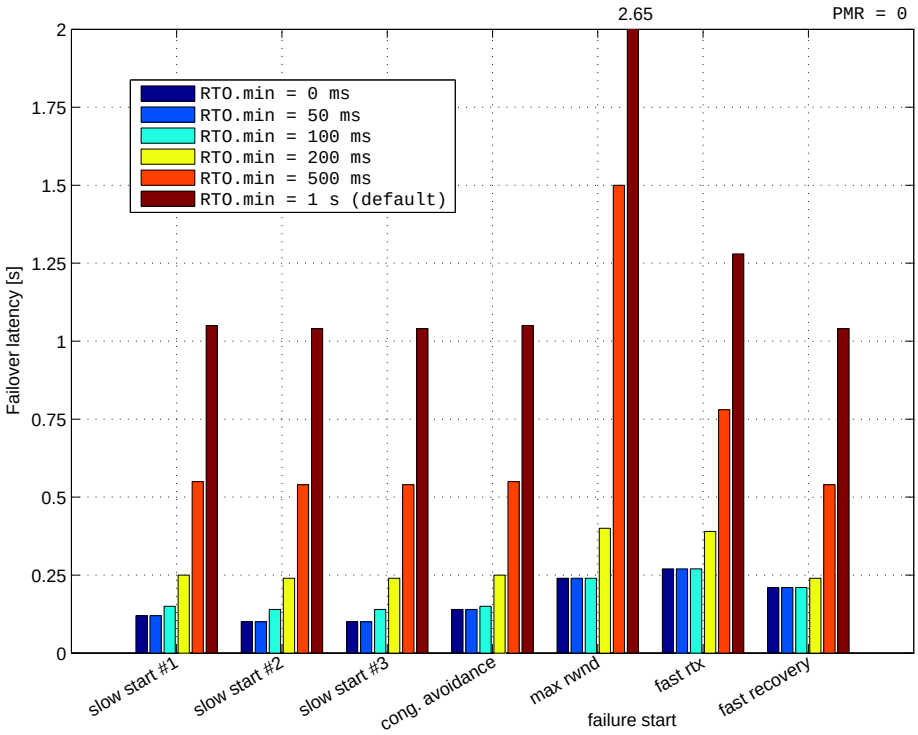
Failover latency, as a more important metric in terms of failover evaluation in a handover application context, is presented in more detail Fig. 4.10. Since the impact of PMR factor is so strong there is a need to examine each value separately, to catch also the influence of the cwnd size in the moment of failure on the overall failover behavior. Here the results for the most interesting cases are presented: PMR set to 0, 1, and the default 5, correspondingly. The SCTP failover mechanism in all analyzed cases failed to provide handover latencies acceptable for real-time applications that typically require latencies of less than 300 ms. This is not surprising, provided that SCTP was configured here with a default value of RTOMin (1 second) that yields the lower bound for a failover latency with low PMR settings (i.e, 0 or 1). This is a very important conclusion in terms of handover application of the failover mechanism. PMR is the key parameter that must be reduced to the low values, but simultaneously also the RTOMin must be adjusted accordingly in order to reflect the RTT measurement and fit the mobility application requirements. Another important consideration is the resiliency against spurious failovers. As it is shown in Section 4.3.2 to prevent sensitive applications the PMR can be reduced to 1 at most, what further limits the use of failover to a soft-real-time, interactive applications that can cope with handover latencies not exceeding three seconds.

Finally, a few comments regarding the cwnd window size in the moment of failure. For the default PMR setting, the instance when the failure happens is insignificant (variance in total transmission time is below 0,5%). This trend is maintained for PMR set to one, but the variance is slightly bigger (about 1%). If no retransmissions are allowed on the path that failed, there is a considerable difference in failover latencies due to a different number of packets that need to be retransmitted before sending new data.

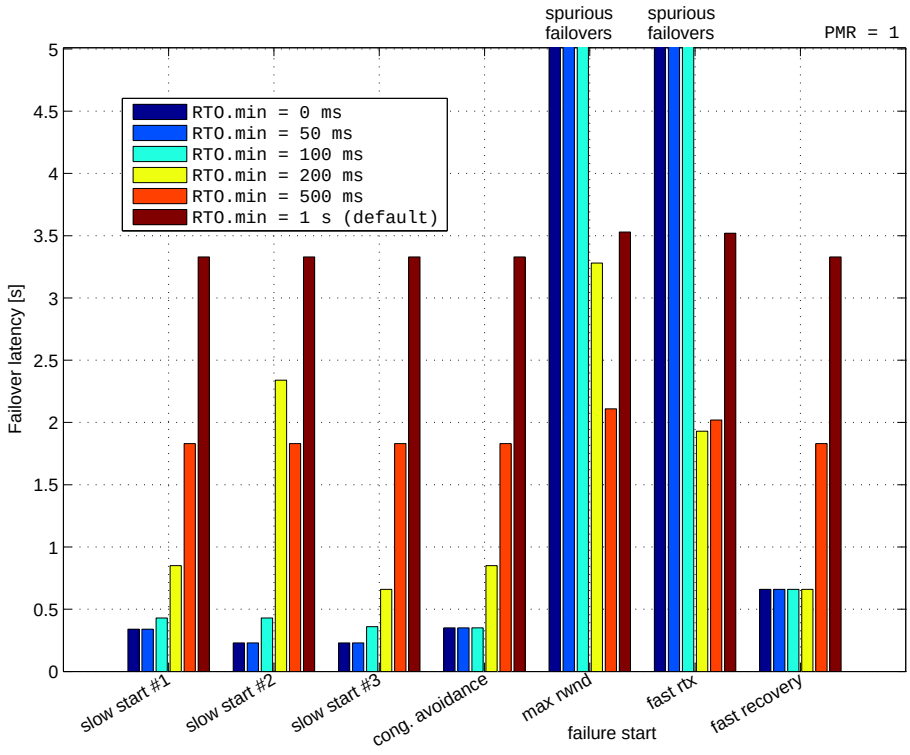
RTOMin

As stated couple of times already, the RTOMin parameter introduces the lower bound for the RTO value, absolutely unacceptable in its default shape for handover scenarios. Even if the PMR value is adjusted for the handover requirements as discussed above, the RTOMin firmly prevents the use of failover for any real time applications. Therefore, if the failover has to be used as a handover triggering mechanism the RTOMin has to be decreased, if not removed at all (resilience to spurious failovers) to let the RTT measurements effectively govern the RTO settings (RTO value is set up upon the RTT measurements). Problem of limitations introduced by the RTOMin has already been studied, and not only in handover context. Apparently one of the first works devoted to SCTP in signaling scenarios [Jungmaier et al., 2002] discussed adjustments to a very liberal, default SCTP parameter settings. Jungmaier et al., apart from reducing PMR, proposed decreasing RTOMin to improve the effectiveness of the failover mechanism. Yet an important conclusion, RTOMin should not be plunged below $2 \cdot RTT$ if spurious failovers has to be avoided. Decreasing RTOMin was further discussed in [Grinnemo and Brunstrom, 2005], in signaling context as well. Grinnemo and Brunstrom conclude also in their study that the default value of RTOMin is inadequate, and should be lowered. Accordingly, changes to RTOinit and RTOMax parameters, should be considered too. However, modifications of these two parameters are not usually mentioned in the handover context, first because removing RTOMin helps avoiding big values of RTO anyway, second, it is reasonable to assume that the handover based on failover happens at least one RTT measurement after association initialization, so the RTOinit is not affecting the handover performance of failover.

Fig. 4.11 illustrates the influence of the RTOMin setting on a failover latency for a given PMR value. Again, as for the PMR analysis, the same set of cwnd evolution cases is also considered. If no retransmissions are allowed by the PMR (Fig. 4.11a), reducing the RTOMin value to the RTT level (slightly above 100 ms) decreases significantly failover latency. Below this level, it is the current RTT value of a given path and not the RTOMin that influences the failover latency. An interesting finding in the examined scenario is the lack of the spurious failovers for the lowest PMR value. However, spurious failovers are present for the second lowest PMR (Fig. 4.11b) for all three RTOMin values that are below the RTT. Actually, the lowest RTOMin that produces stable performance for all analyzed cwnd cases is the RTOMin set to 500 ms, being the lowest of the examined values that meets the condition ($> 2 \cdot RTT$) postulated by Jungmaier et al. [2002]. Therefore in handover context, where low PMR values are envisaged, reducing RTOMin seems a clear choice, even at the expense of

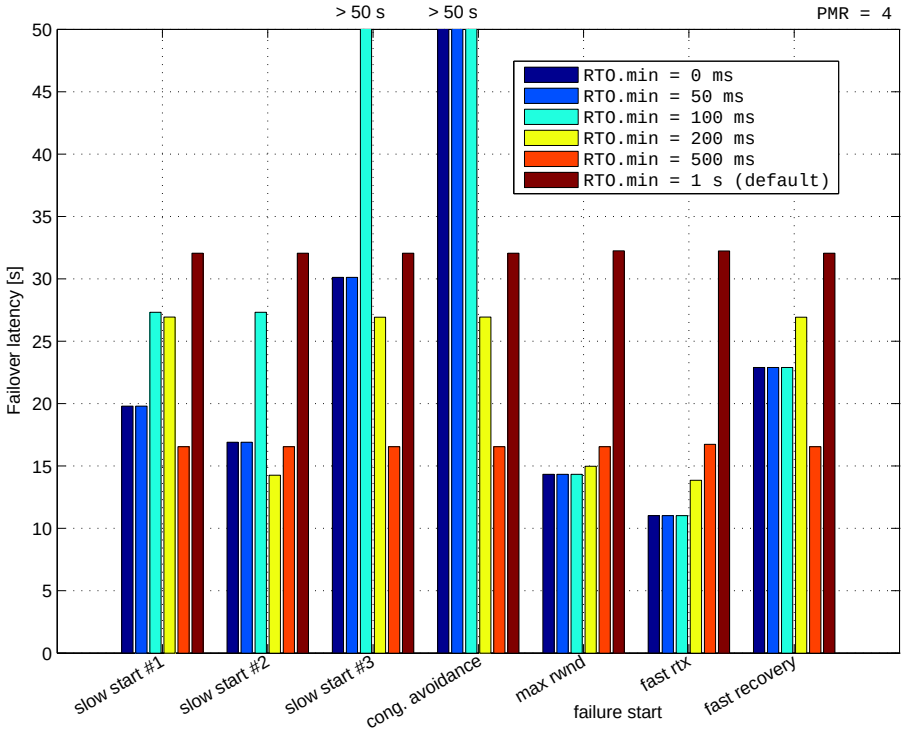


(a)

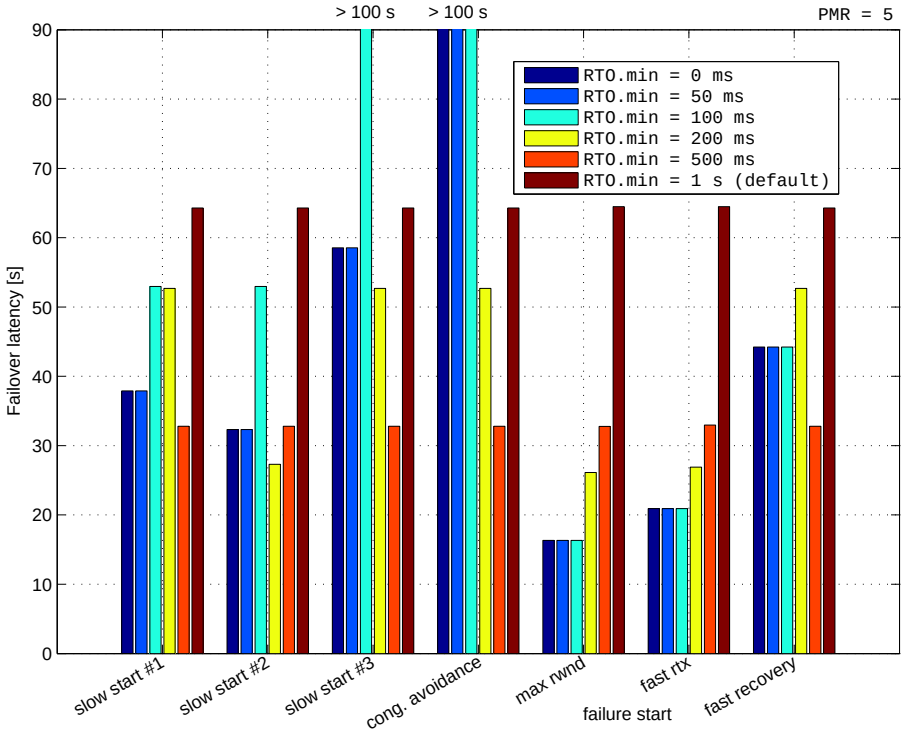


(b)

Figure 4.11: Influence of RTOMin parameter for failover latency: (a) for PMR = 0; and, (b) for PMR = 1.



(c)



(d)

Figure 4.11: Influence of RTOMin parameter for failover latency: (c) for PMR = 4; and, (d) for default PMR value.

increasing possibility of spurious failovers. The gain produced from decreasing the RTOMin setting is rather evident also for the biggest PMR values (Fig. 4.11c and Fig. 4.11d). With no risk of spurious failovers the failover latency can be decreased to only 25% (in the best case) of the value for the default PMR setting. Still, among all analyzed cwnd cases the RTOMin set to 500 ms is the lowest value that guarantees stable improvement. Below that value two out of seven examined cwnd settings resulted in unstable performance.

SACK delay

Following its ancestor TCP, SCTP has also the delayed SACK algorithm, designed in line with the guidelines drawn by the RFC2581 [Paxon et al., 1999]. According to that document, resuming TCP congestion control state of the art as for late 1990s, the SACKs should be generated at least every second packet received and at most 500 ms from the arrival of the first unacknowledged packet (so called SACK delay). SCTP specification suggests 200 ms SACK delay to report unacknowledged data, leaving room for being more conservative, but not beyond the 500 ms TCP limit. On the other hand there is no space within the protocol specification for more aggressive behavior. In that sense study devoted to evaluate the influence of the SACK delay parameter on the failover behavior in signaling transport [Eklund and Brunstrom, 2006] seems quite interesting. Eklund and Brunstrom check the failover performance for SACK delay values below 200 ms and conclude that indeed for managed signaling telephony networks having no SACK delay at all may improve failover performance considering even the cost of introduced traffic overhead. However, in handover context possible gain from decreasing SACK delay is rather uncertain, bearing in mind an extra overhead traffic produced.

Presented set of experiments devoted to analyze the SACK delay parameter (Fig. 4.12) follows the approach proposed by Eklund and Brunstrom, and takes into account all range of values from not having the SACK delay at all, to the biggest value allowed by the protocol specification. Compared to the default setting (200 ms) there is no significant gain for any of the two lowest PMR values, if the SACK delay value is decreased or even removed at all. The same trend is followed for the default PMR setting, the maximum gain reported did not exceed 2%. Therefore, the default value of 200 ms can be clearly preserved, as not harming the protocol performance in the handover context.

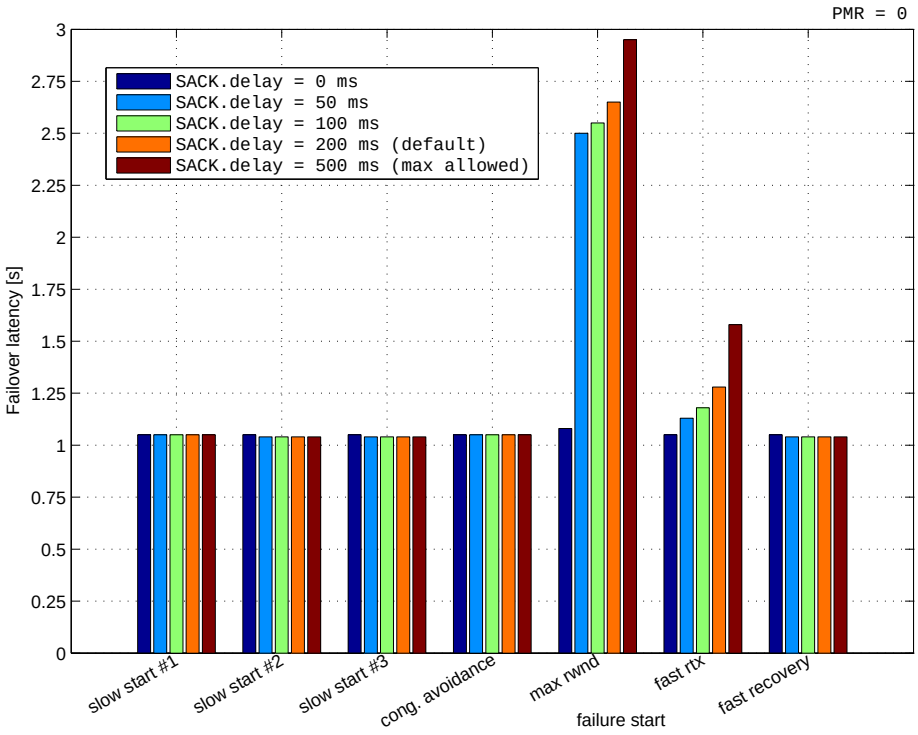
HBInt

Another issue that has to be taken into account while discussing the use of failover for handover scenarios is the heartbeat mechanism. Heartbeat chunks are periodically sent on the idle paths regardless of the link status (whether is marked as ACTIVE or INACTIVE), within the HB_{period} that is calculated according to the formula (4.15). A tuned heartbeat mechanism may serve for faster provisioning of the link state information to the endpoint and as a result induce the failover to a new link. Heartbeat mechanism also prevents from switching to the paths that are already unavailable.

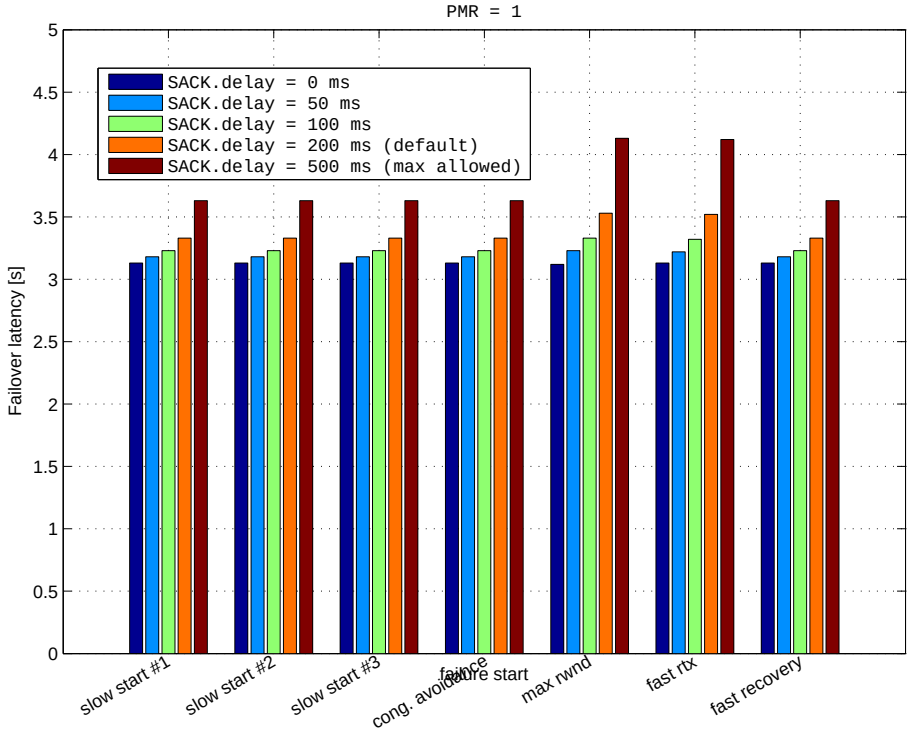
$$HB_{period} = \text{HBInt} + (RTO_{dest} \pm 50\% RTO_{dest} jitter) \tag{4.15}$$

The key parameter governing the heartbeat timer is the HBInt that for the default setting excludes using heartbeat mechanism to probe highly variable wireless scenarios. In order to effectively use the heartbeat mechanism in the handover scenarios the value of the HBInt should be decreased significantly. However, this adjustment must be made carefully enough in order to avoid introducing too much overhead related to the heartbeat traffic. An example study [Kashihara et al., 2004] shows design of an algorithm based on more frequent heartbeat probing of wireless links. In that work Kashihara et al. also propose modifications to PMR settings and the error counting algorithm of a standard SCTP in order to provide support for seamless handover using SCTP and the DAR extension.

In this work the idea of modifying the HBInt parameter will be discussed with more details during the design of a SCTP-based handover scheme with CMT support in Section 5.3.2.

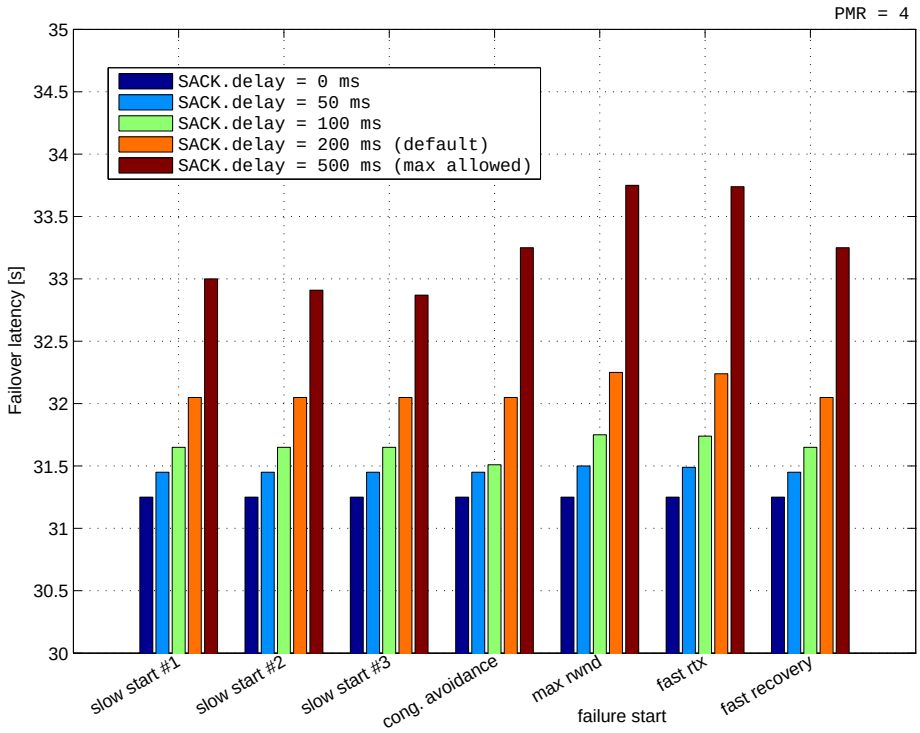


(a)

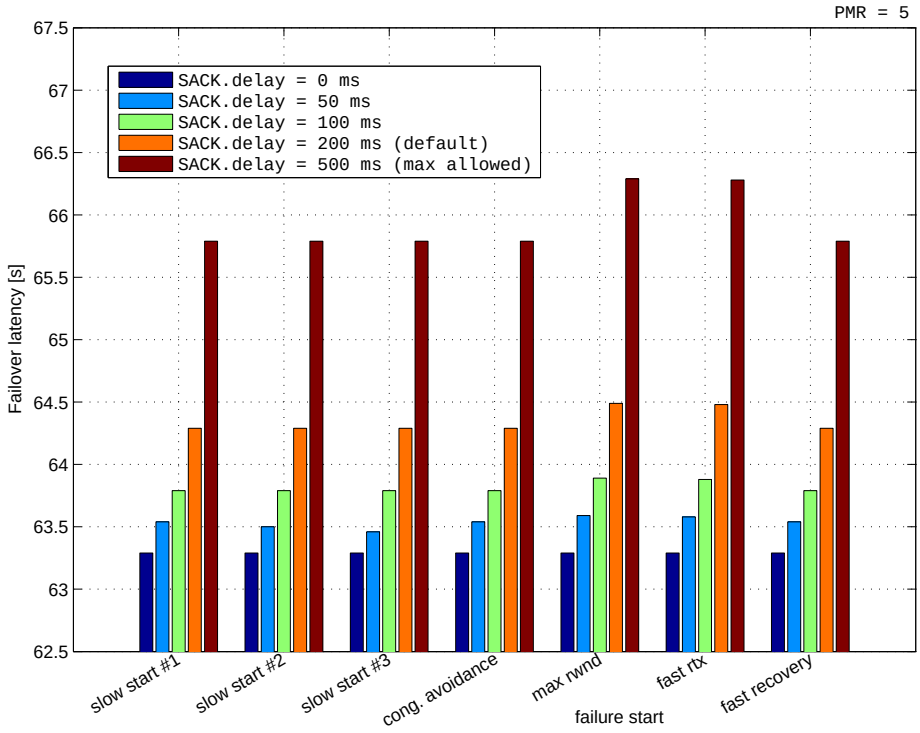


(b)

Figure 4.12: Influence of `SACK delay` parameter for failover latency: (a) for `PMR` $= 0$; and, (b) for `PMR` $= 1$.



(c)



(d)

Figure 4.12: Influence of `SACK delay` parameter for failover latency: (c) for `PMR` = 4; and, (d) for default `PMR` value.

Retransmission policy

Within current protocol specification [Stewart, 2007] there is a clear indication of the recommended retransmission policy (Fast Rtx is handled at the same path and Timeout Rtx is sent to the alternate path if the peer is multihomed - FRSameToRAlt) that determines the protocol behavior when a permanent failure occurs and therefore influences the handover latency if the basic failover mechanism is used. However, to gain some insight on the retransmission policy issue it is necessary to mention related work conducted by Caro [2005] that discusses pros and cons of various retransmission policies and their impact on failover. In his work Caro reconfirms that the existing FRSameToRAlt policy is the most universal one for various types of networks considered. On the same time Caro recommends interesting extensions that are namely: Multiple FastRtx (MFR), Heartbeat after RTO (HAR), and Time Stamp (TS). Especially the first extension used in combination with the current policy results in significant improvements to the overall protocol performance and may have impact on the SCTP performance in handover scenarios. However, as this issue is not directly related to the handover process and still has not been reflected in the protocol specification, it will not be discussed here in more details.

Recommended parameter settings

To conclude the presented discussion a revised version of SCTP parameter set is provided in Table 4.3. Such indication serves as a road map for further failover evaluation.

4.3.2 Performance evaluation

Once the framework for SCTP parameters has been established, a detailed failover evaluation can be given presenting a series of simulation experiments conducted in ns-2 [NS-2]. Presented simulations introduce a dynamically changing wireless channel. In contrast to most performance analysis published so far, such as [Caro, 2005], conditions established at the beginning of the simulation (i.e., bandwidth, latency and losses) are varying during the entire simulation instance. Such approach helps to reflect better the variable nature of wireless channels.

Scenario description

Simulation setup for presented experiments reuses the symmetrical scenario topology presented in Fig. 4.7. Also the same type of application is used, an FTP transfer of a 16 MB file from fixed correspondent node to a mobile host. A novelty in presented evaluation is the dynamic channel model illustrated in Fig. 4.13.

The dynamic channel model, first presented in [Budzisz et al., 2006a], aims at capturing the progressive degradation of a radio network interface that finally leads to the link failure, when a threshold value is exceeded. In the simplest approach, this behavior is achieved by means of varying PER. The radio conditions move gradually from a steady state, where a minimum PER can be satisfied, toward another steady state with a higher PER value beyond the threshold that makes the channel blocked. Parameters describing dynamic channel model on the primary path are, T_1, time from transmission start when link starts the transition, and T_2, transition time between two steady states. The counterparts on the backup path, bearing in mind that the backup path becomes available with time, are T_3 and T_4, respectively. The upper limit of the PER values in the channel is set to PER_{max}; beyond this limit the channel becomes unavailable.

The analysis metrics will take into account the overall effectiveness (average file transfer time), and stability of the transmission (average number of primary path changes). Each experiment is repeated 10 times in order to obtain the average performance (corresponding 95% confidence intervals were calculated to check the estimate of the underlying average, however in order to improve the readability of the graphs these results are not included in the plots presented in this chapter). To make the metrics more trustworthy an upper limit of the average transmission time was set up at 900 seconds (about 68 times the value for the channel without errors). Had the transmission not been completed within this time, the sample is discarded, and the average result is calculated

Table 4.3: Recommended values of SCTP parameters in handover context

Parameter name	Default value	Recommended value	Comments
PMR	5	0-2	In handover scenarios PMR value must be decreased to 0 or 1, as reliability is not a dominant issue. The key aspect is to decrease failover latency to the values acceptable for most of the applications. If inducing spurious failovers is still a concern, some PMR protection is recommended, with $PMR = 2$ as a recommended value.
RTOMin	1 s	0-200 ms	To make the failover feasible for handover scenarios 1 s long lower bound for a failover latency must be removed.RTOMin should be plunged below 200 ms or removed at all if spurious failover resiliency is not an issue. Recommended setting to protect sensitive applications: $RTO_{Min} > 2 \cdot RTT$.
SACK delay	200 ms	200 ms	SACK delay parameter in its current shape is a compromise between frequent feedback from the receiver (implicit link probes) and SACK traffic overhead. Reasonable compromise, based on all experience with TCP. Default value should be preserved.
HBInt	30 s	$\ll$ 30 s	If heartbeat mechanism is used to probe a wireless link, the HBInt value should be reduced drastically. If information about the link state is gathered from lower layers there is no need to change the default value, although more information about the link state is desirable.
Rtx policy	FR: Same, ToR: Alt. path	FR: Same, ToR: Alt. path	Retransmission policy currently used in SCTP is the most versatile approach. However it still can be improved with MFR algorithm [Caro, 2005]. Here the default settings are preserved.

Table 4.4: Simulation parameters for failover evaluation

Parameter name	Value / Range
Transition start time (T_1, T_3)	0,5-10 s
Transition period (T_2, T_4)	0-40 s
PER	0,1-10%
PER threshold (PER_{max})	20%
Queue size	50 packets
PMR	0-5
HBInt	30 s
SACK delay	200 ms
Retransmission policy	FR: Same, TO: Alt. path
MTU size	1500 Bytes
Payload size	1468 Bytes
Downloaded file size	16 MB
Maximum allowed transmission time	900 s

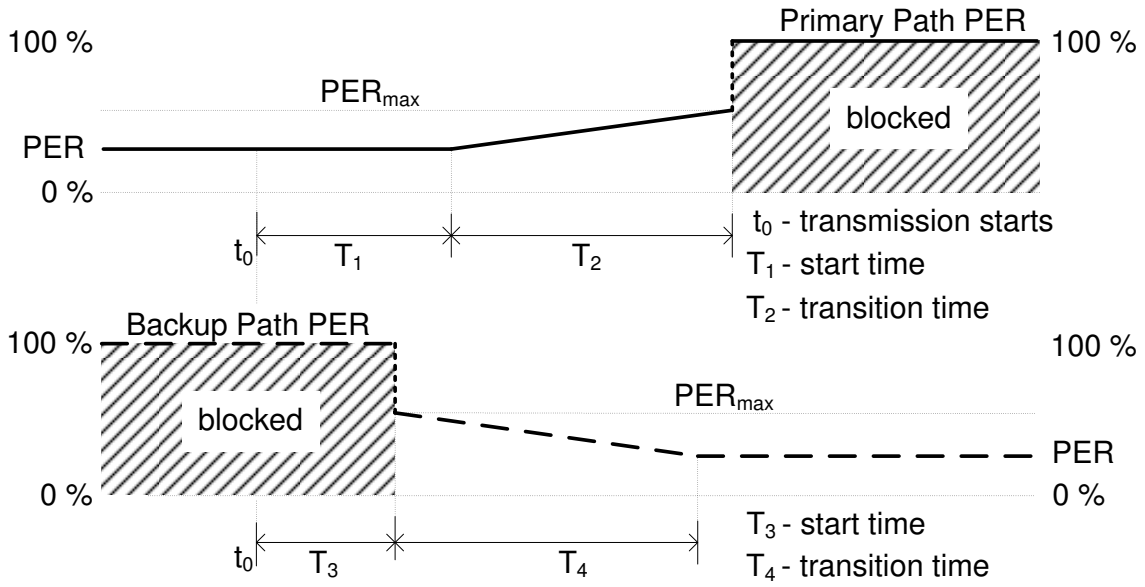


Figure 4.13: Dynamic channel model.

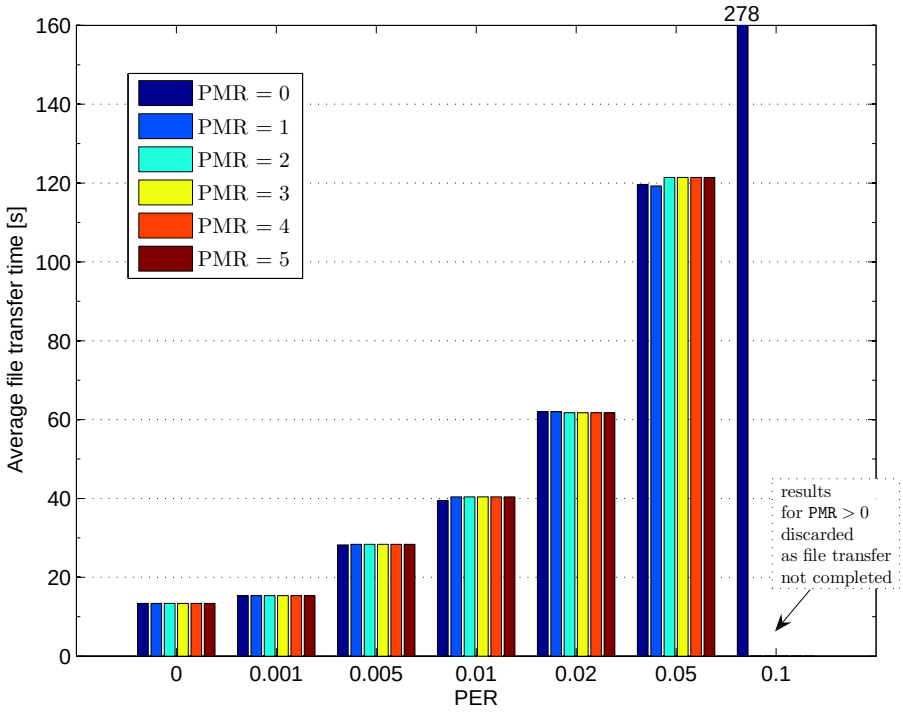
on smaller population of samples. The summary of the most important simulation parameters is presented in Table 4.4.

Reference scenarios

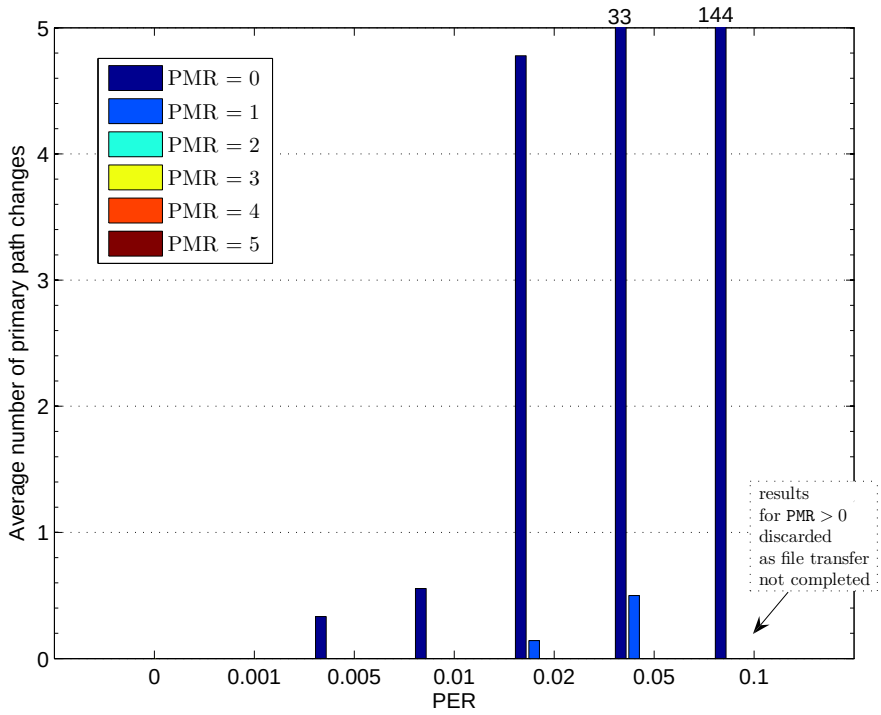
To provide a fair comparison for the presented analysis, a pair of reference scenarios will be considered. First reference scenario comprises of a static channel with a constant PER on each path (i.e., $T_1 = T_3 \rightarrow \infty$), a typical scenario used in performance analysis presented in [Caro, 2005]. Fig. 4.14 presents the results in the reference scenario for different PER values, within the range 0-10%, and also in function of the PMR parameter.

As seen in Fig. 4.14, for *low PER* values (PER $<$ 1%) protocol performance is quite stable, whereas for the biggest PER value tested (10%), the results obtained are at least, 20(!) times bigger (in the best case, when PMR $= 0$) than transmission time in a channel without errors (average transmission time about 13,4 s). This trend is also reflected in number of collected samples that can be taken into account. For low PER values all ten samples are valid, for PER $<$ 2% at least seven out of ten, whereas for the biggest PER considered only for PMR $= 0$ the transmission completes in five out of ten experiments. The rest of PMR settings for PER $=$ 10% result in unending oscillations between both paths, but the file transfer is never finished. That is why the upper limit of the transmission time was set to 900 seconds, since if the transmission can not be completed within this time, it is very likely that the association will shut down without completing the transfer. Also for that reason beyond 900 seconds samples are not taken into account when calculating both metrics, average file transfer time and number of primary path changes. Now, when considering the average number of the primary path changes, for low PER rates file transmission is completed without any failover, if PER $=$ 2% or 5% some failovers may happen if the PMR value is decreased to 0, and for the biggest PER the ping-pong effect can be observed. Interesting observation that setting PMR $= 0$ induces spurious failovers but is the only way to complete the file transfer if both paths have the same huge PER rate (PER $=$ 10%). Summing all up, it is important to state that standard SCTP does not offer any stable solution for high PER values (PER about 10%).

Second reference scenario for a failover benchmark contains a simple modification of previous reference scenario: dynamic channel model is applied to the primary path only, the static channel model on the backup path is preserved. Of course, such approach will result in shorter transmission times and the number of primary path changes tending to one (once the change occurred the transfer should be completed, following the first reference scenario), as shown in Fig. 4.15.

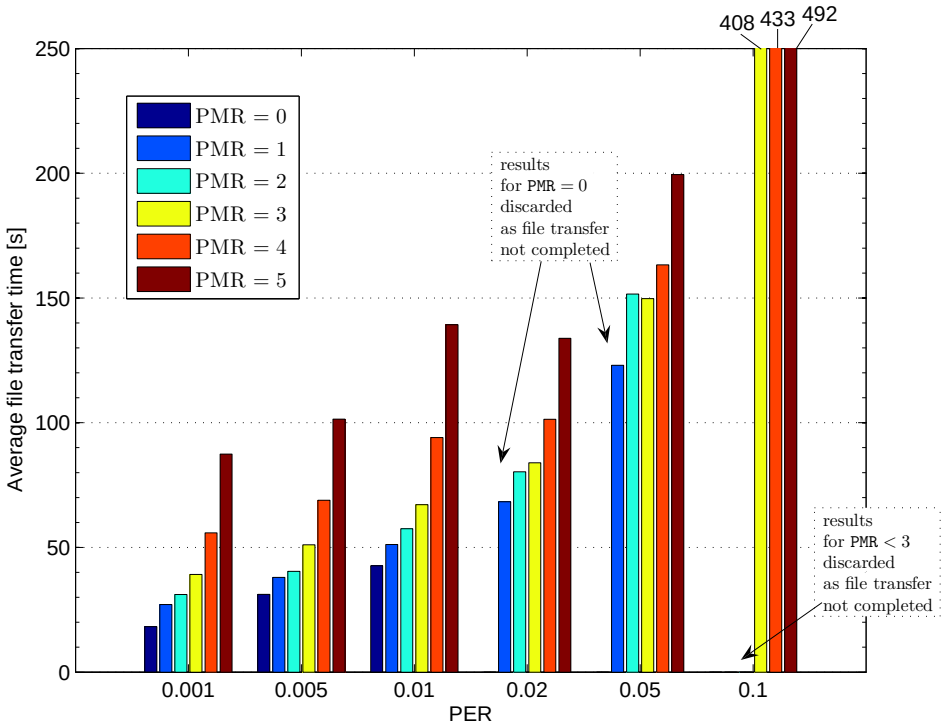


(a)

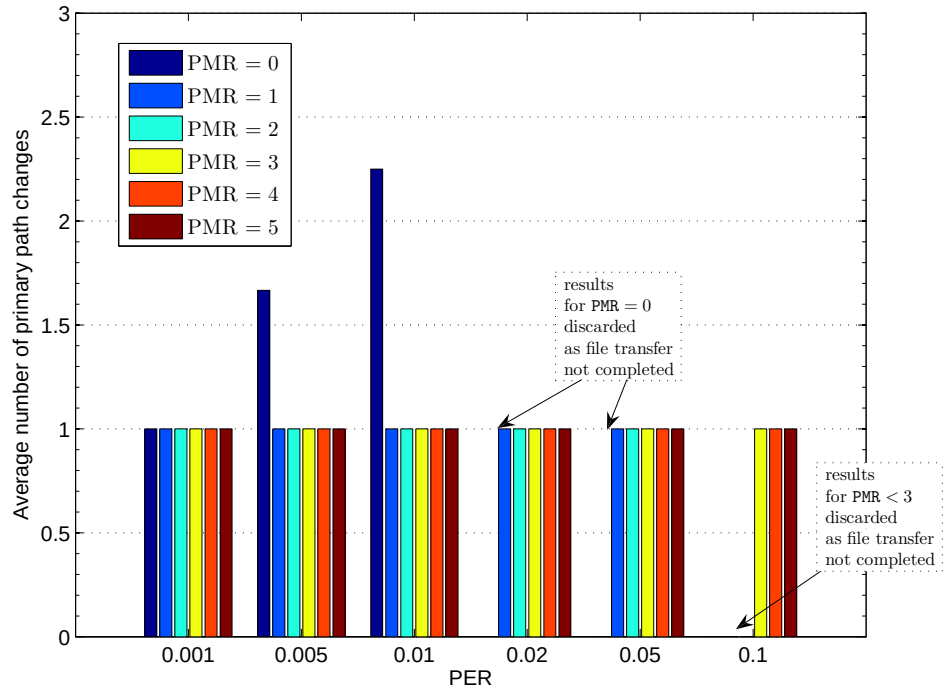


(b)

Figure 4.14: SCTP failover performance in static channel conditions: (a) average transfer time; and, (b) number of primary path changes.



(a)



(b)

Figure 4.15: SCTP failover performance in a scenario with deteriorating primary path and static channel conditions on the backup path: (a) average transfer time; and, (b) number of primary path changes.

Again, the most important tendency that can be observed in Fig. 4.15 is that for low PER rates the standard SCTP failover mechanism performs stable, also if the default PMR value is decreased to 0 in order to achieve faster file transmission. The channel model used on the primary path forces the failover, so that the transmission times from the first scenario can be challenged only if low PMR settings are used. Not surprisingly, even for the lowest PER rate tested, setting $PMR = 0$ results in 18 second long transmission, $PMR = 2$ yields already 31 s, but the default PMR beats them all with 87 s (most of that spent on a failover), almost seven times longer than the first reference scenario (without failover). Further on, as the PER value increases to 2%, stable transmission is guaranteed by the PMR value not lower than 1 (all ten samples for $PMR = 0$ were discarded), whereas for the biggest tested PER rate (10%) $PMR = 3$ was the lowest value that allowed successful file transmission in less than 900 seconds. Unlike the first scenario were for PER = 10% all cases but $PMR = 0$ resulted in unending oscillations between two highly loss paths, having only one path active (the dynamic channel model blocked the primary path after $T_1 + T_2$ from transmission start) led to successful file transfer if there was enough retransmissions at the transport layer ($PMR \geq 3$). As expected, in such deterministic scenario only $PMR = 0$ could lead to increment of the number of average primary path changes above one, still significantly limited by the PER influence PER $< 2\%$. Concluding, a clear trade-off between transmission time and stability can be drawn here. Comparing protocol performance to the first reference scenario, the lower values of PMRare excluded, as unstable for bigger PER values.

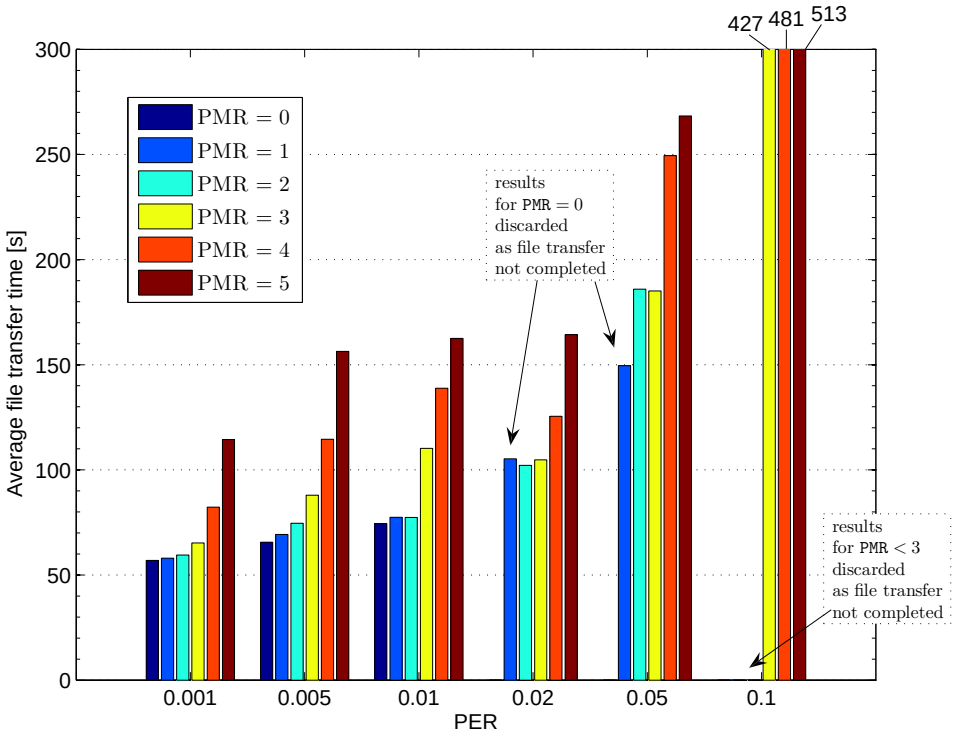
Results

Now, the proper part of the analysis can be started. The results obtained if the dynamic channel model is applied on both paths will be compared with the reference scenarios. For this case, the important point to expose is how the previously discussed results change, depending on when the backup channel becomes available. As seen in Fig. 4.16, the trend for stable performances for low PER rates, no matter how low is the PMR value, is maintained if dynamically changing channel is applied on both paths. Should the PER value increase, a similar trend as for second reference scenario is observed. Stable transmission threshold is set at $PMR = 1$ and at $PMR = 3$ for PER = 2% and PER = 10%, respectively. The trade-off is paid with the transmission time that even for the lowest PER value tested (PER = 0.1%) is at least four times longer(57 s) than in first reference scenario. Interesting from stability point of view that for PER $< 2\%$ the transfer time not varies much within three lowest values of the PMR.

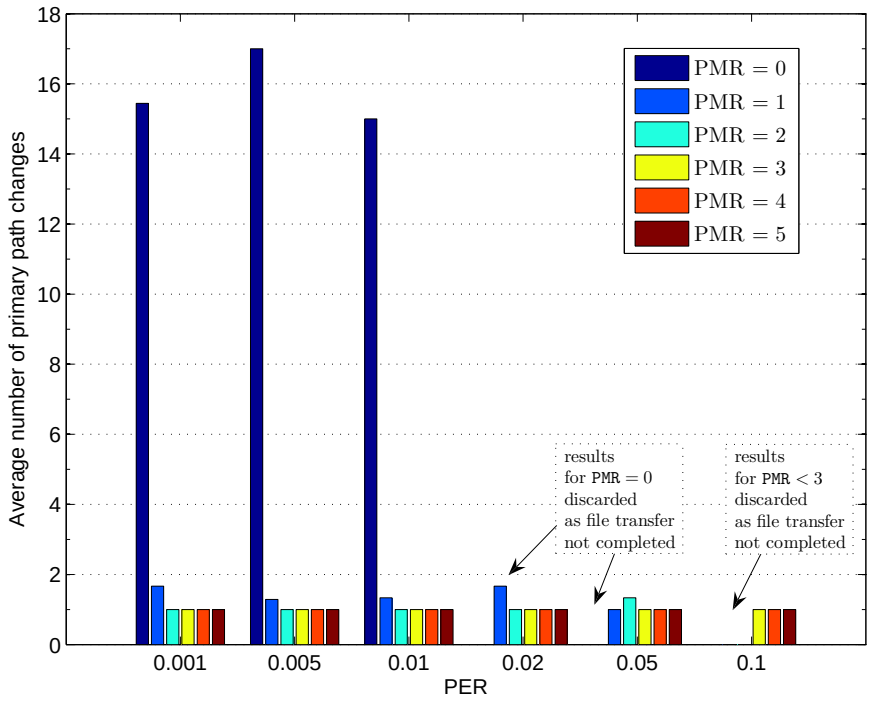
Comparing to the first reference scenario, proposed channel model excludes low PMR settings (i.e., $PMR < 3$), as unstable for the biggest PER values tested (PER $> 5\%$). Yet as a result, the number of primary changes is also reduced to one for all cases but $PMR = 0$, as can be seen at Fig. 4.16, a similar trend as in second reference scenario. In contrast, for $PMR = 0$ presented scenario allows more oscillations (on average 15-17 primary path changes for the three lowest PER rates), as both paths have the dynamic channel model, and active status is maintained longer, but again only within the low PER values. The number of collected samples is also comparable to the second scenario, having at least seven out of ten probes for low PER values, and only four for the biggest PER rate.

Influence of link-layer retransmissions

In noisy wireless environments, since SCTP preserves the TCP's congestion control rules, it is exposed to encounter a lot of non-congestion losses that can provoke unnecessary congestion window reduction at the source. The most common solution to correct errors at the wireless link while preserving the transport protocol end-to-end semantic is a combination of FEC (Forward Error Correction) and ARQ (Automatic Repeat Request) [Chockalingam et al., 1999; Vacirca et al., 2006]. FEC consumes some extra bandwidth to transport the redundant information that helps recovering errors, however does not interfere with transport layer parameters such as round-trip time and retransmission timeout. Since FEC is not the scope of this analysis, it is assumed that PER parameter referred in the channel model already includes the effect of FEC corrections. On the other hand ARQ, used to repeat the packets that FEC is unable to recover, does not consume much bandwidth,

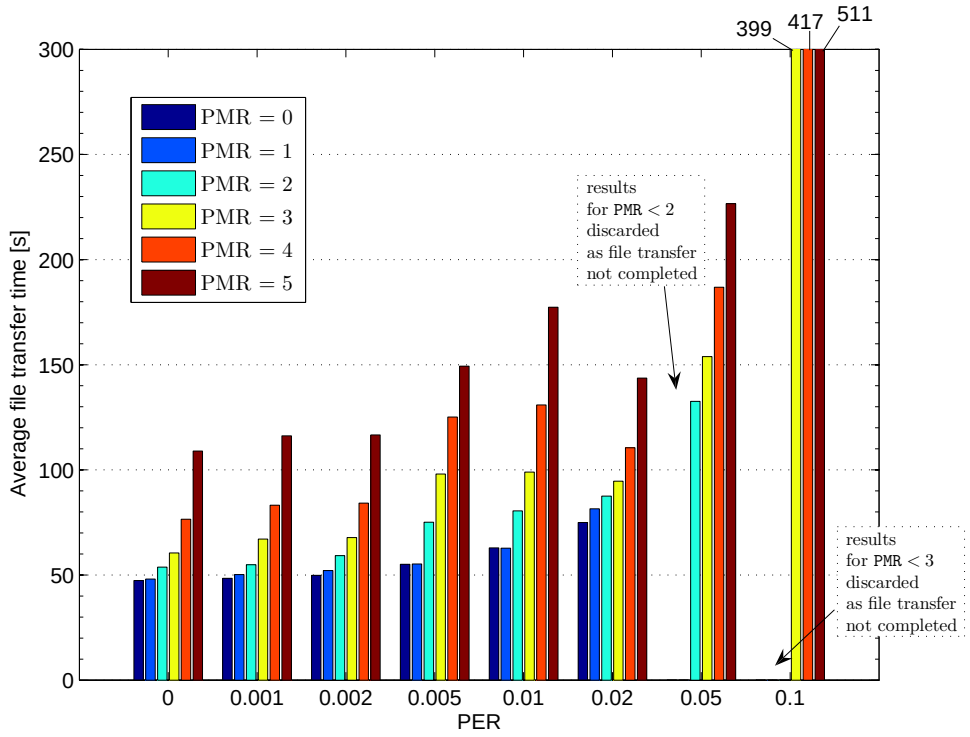


(a)

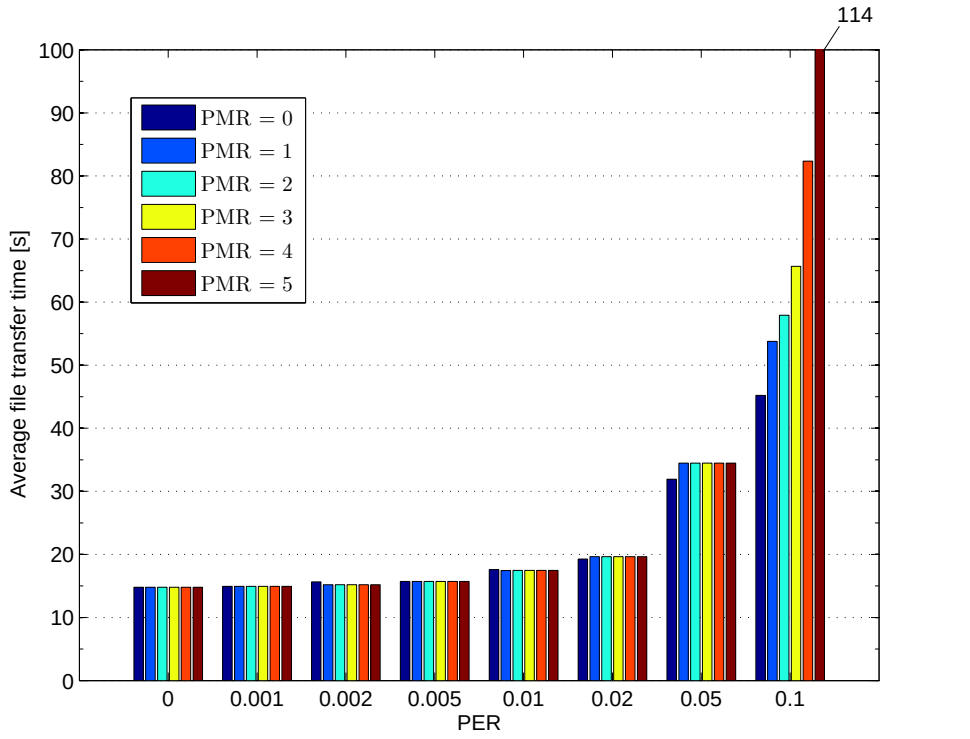


(b)

Figure 4.16: SCTP failover performance in a dynamic changing channel on each path: (a) average transfer time; and, (b) number of primary path changes.

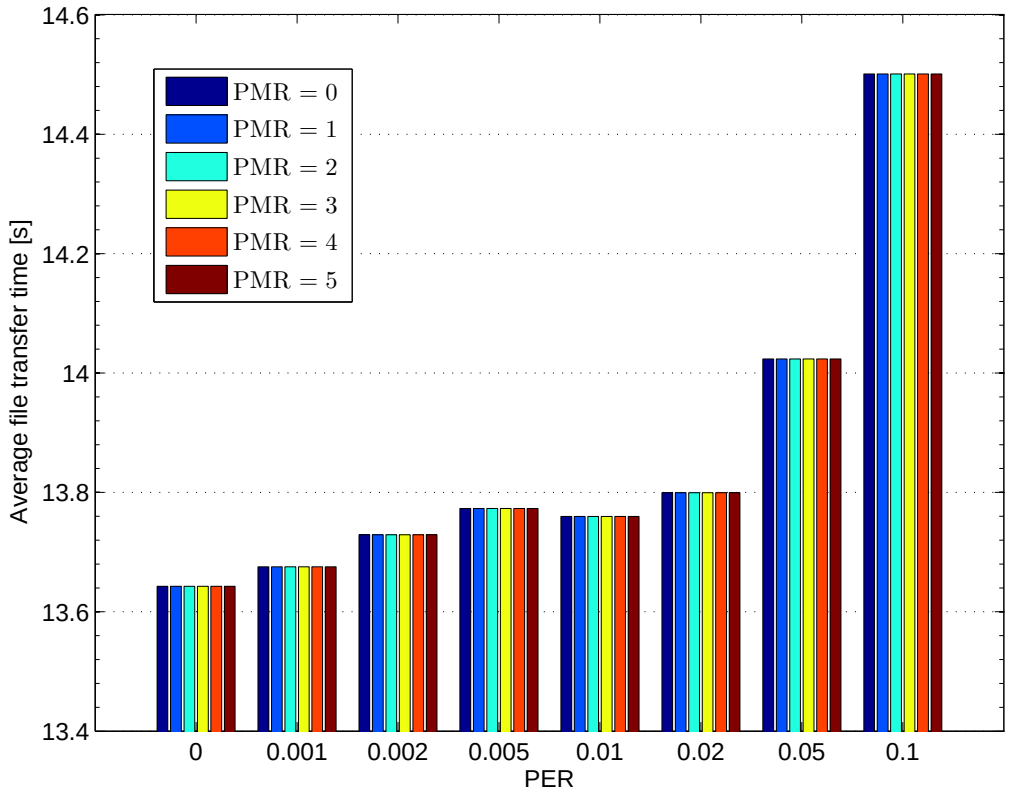


(a)



(b)

Figure 4.17: Influence of ARQ on the SCTP failover performance - average file transfer time: (a) for $\delta = 0$; and, (b) for $\delta = 1$.



(c)

Figure 4.17: Influence of ARQ on the SCTP failover performance - average file transfer time: (c) for $\delta = \infty$.

but indeed may affect SCTP performance in terms of increasing and variable round-trip time (thus decreasing the throughput) and interfering with timeout, while retransmitting corrupted packets.

Therefore, to extend the scope of presented analysis, the influence of the ARQ on the SCTP failover performance will be evaluated, following the study [Budzisz et al., 2006b]. The key parameter here is the number of retransmissions allowed at the link-layer, called the *persistence* of the ARQ (δ). Consequently, three different cases are considered:

- $\delta = 0$ - no link-layer retransmissions allowed. All the impact of the fluctuating channel conditions goes directly to the transport layer, reducing the available congestion window.
- Finite $\delta > 0$ - channel with losses, shielded at the link-layer, with varying delay. Packet transmission delay is affected by the number of retransmissions. Packet losses are still possible if reaching a number of retransmissions equal to delta.
- $\delta = \infty$ - channel without losses with varying delay. Long link-layer retransmissions may provoke spurious retransmissions on the transport layer, or even timeouts.

First, in Fig. 4.17, the results for the three mentioned channel types are presented: channel not shielded at the link layer ($\delta = 0$), channel shielded with losses ($\delta = 1$), and channel without losses ($\delta = \infty$), accordingly.

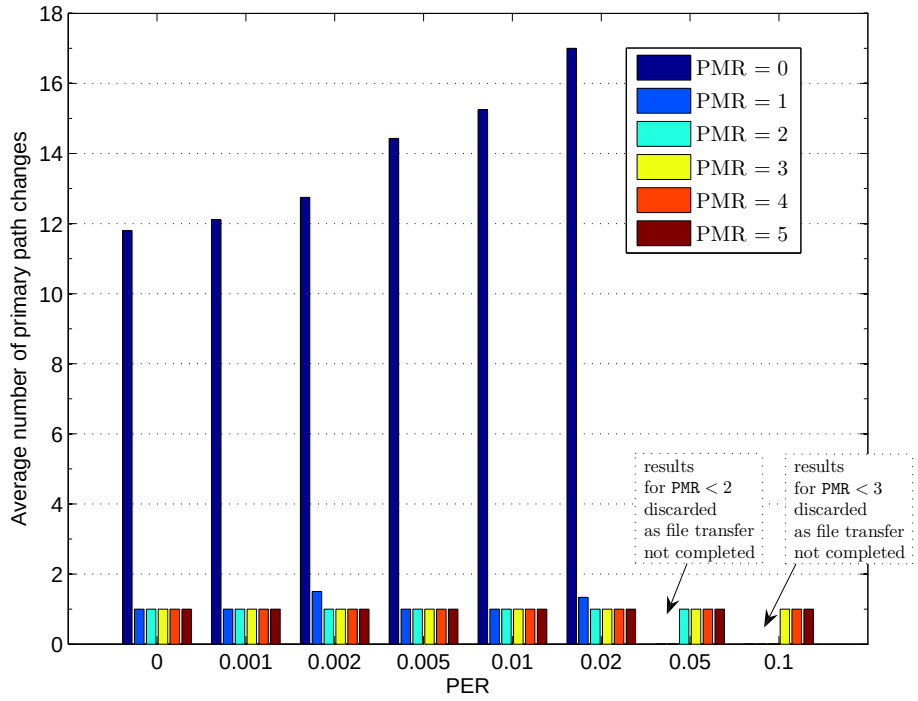
Following the TCP's experience with noisy radio channels, channel not protected at the link layer leads to weak SCTP performance as well, forcing transmission times 400-500 second-long (depending on `PMR` setting) for the highest 10% PER rate (well more than 30 times longer than in channel without errors - 13,4 s), and about 55-120 s for the lowest PER rate investigated 0,1% (4-9 times bigger than error-free transmission, respectively). Introducing any shielding at the link layer improves

significantly the overall SCTP performance, leading finally to a very stable and fast performance, when many retransmissions are allowed. Here are the examples. With as little as one link-layer retransmission the transmission time is reduced below 20 seconds for all PER $< 5\%$, and the number of the collected samples increases to at least nine out of ten for all PER values tested. More, ideally shielded channel has the transmission time for the biggest PER rate tested only 8% longer than the transmission without losses (also no failover is forced), and of course, all ten samples collected.

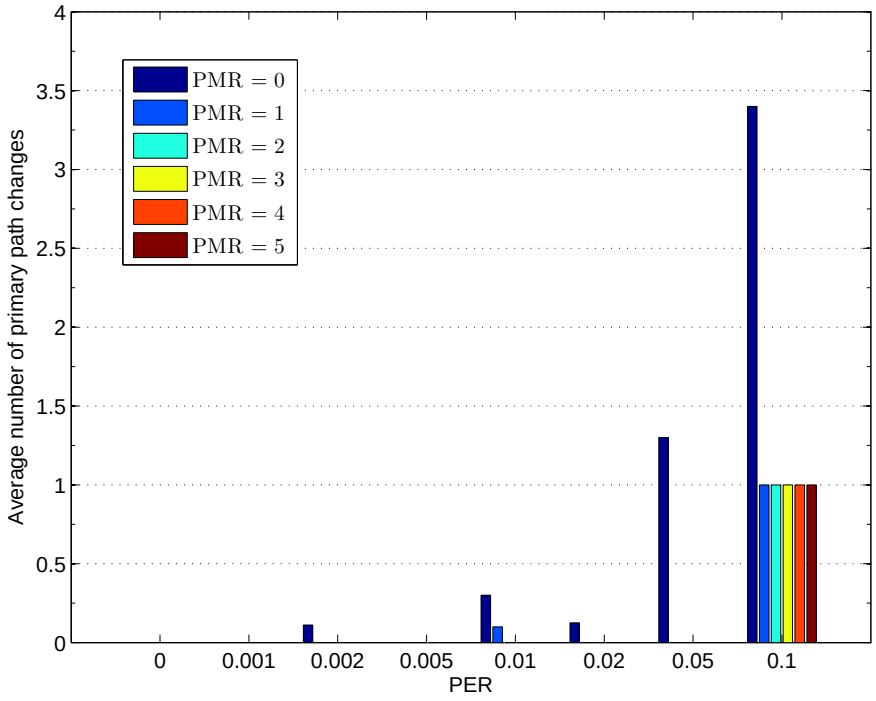
Shielding on the link layer is also visible in terms of number of the primary path changes presented in the Fig. 4.18. In a non-shielded wireless channel, when the entire impact of non-congestion losses goes directly to the transport layer, PMR parameter plays the role of the stabilizing factor. For PER $< 2\%$ it is possible to obtain fairly high throughput rates decreasing the PMR value from the default 5 down to 1. PMR set to 0 even for low PER rates provokes ping-pong effect and the gain in throughput is not that significant, as if compared to the PMR $= 1$. Further on, as the PER value increases beyond 2%, only higher PMR values guarantee stable file transmission, however as it was mentioned above the trade-off results in fairly long transmission time, because of exponential back-off mechanism that triggers handover. Meanwhile, with low PMR values (PMR < 2) file transfer cannot be completed within 900 seconds time in a non-shielded channel. For partially shielded channel ($\delta = 1$), also the lower PMR values can result in successful file transmission, even if the PER achieves rates as high as 10%. That practically guarantees reliable and fast file transmission. As for channel without losses ($\delta = \infty$), all the impact of varying radio channel is handled on the link layer, and therefore preventing from forcing any failover at the transport layer. Such policy however, could result in spurious retransmissions or even timeouts for very noisy channels. Nevertheless, in the analyzed case, the highest PER rate taken into account 10% (that corresponds to PER rates varying between 10 and 20% in the proposed channel model) was not big enough to provoke that.

4.4 Conclusions

SCTP failover mechanism can be reused to grant handover support. The main goal of the analysis presented in this chapter was to provide the quantitative response to that issue, as several doubts were raised in beforehand. At least a pair of them is of merit. First, as it was rather evident the most of the congestion control rules that were inherited from TCP had negative effect on SCTP performance in wireless scenarios, in presence of non-congestion losses. Second concern was the SCTP design goal that forced signaling tune-up of protocol parameters. However, once the appropriate adjustments were made, either by introducing FEC-ARQ shielding at the link-layer of the wireless link or directly to the protocol parameters (see Tab. 4.3), the performance analysis could give the response to the main question, whether SCTP failover mechanism is suitable for handover provisioning or not. Experiments present clearly that even after tune-up of the most important protocol parameters, namely PMR and RTO_{Min}, SCTP failover mechanism is rather unsuitable for real-time applications. On the other hand, non real-time applications have not ruled out completely the SCTP failover mechanism. In fact, provided that few-second handover latencies are acceptable, experiments suggest that the SCTP failover mechanism could indeed be used for these applications, after necessary PMR and RTO_{Min} parameter adjustments. Yet, as will become evident in the following sections, there are still better ways of using SCTP for handover than using the failover mechanism.



(a)



(b)

Figure 4.18: Influence of ARQ on the SCTP failover performance - average number of primary path changes: (a) for $\delta = 0$; and, (b) for $\delta = 1$.

Chapter 5

Improving handover with transport-layer loadsharing

Transport-layer loadsharing is another application that extends the use of SCTP multihoming, relative to what is defined within the standard protocol specification. If compared to loadsharing schemes in other layers [Goff and Phatak, 2004], transport-layer loadsharing has a considerable potential to improve protocol performance (in terms of overall throughput) and provide an useful solution from the application point of view. The major challenge arising from simultaneous data transfer over multiple paths is packet reordering at the receiver. This issue may deteriorate SCTP performance, since congestion control algorithms in standard SCTP are derived from TCP, and hence do not work well when reordering is common. Congestion control in standard SCTP is applied to the entire association (i.e., as an inheritance from TCP, a unique TSN numeration is used for all destinations), however separate sets of congestion control variables (cwnd, sstresh and partial_bytes_acked) are kept for each of the destination addresses of a multihomed peer. Thus, to accomplish loadsharing the SCTP send-buffer management and congestion control must be updated to take into account the problems of sending data over multiple paths using a single sequence-number space, and the consequences of sender-introduced reordering. So far, there is no commonly defined extension that facilitates loadsharing for SCTP. Therefore, the most important proposals will be examined here in detail. Then, the applicability of the most common solution (Concurrent Multipath Transfer) to distribute data among two end-to-end paths of a mSCTP association during the handover transition process will be evaluated. To that end, the design principles of a protocol extension that joins Concurrent Multipath Transfer and mSCTP are given. The proposed new handover scheme is benchmarked with pure mSCTP handover scheme, demonstrating its potential benefits: smoothing the handover transition process, and improving the application's overall throughput.

5.1 Related work on transport-layer loadsharing with SCTP

One of the first proposals for loadsharing with SCTP, called LS-SCTP has been brought up by Abd El Al et al. [2004a,b]. LS-SCTP separates flow control, handled per association, from congestion control that for loadsharing need to be handled per path. Therefore, Abd El Al et al. propose introduction of two additional chunk types to carry data and related acknowledgment in LS-SCTP. Both chunks are backward-compatible with corresponding standard SCTP chunks, the only difference being the additional sequence numbers added to facilitate the loadsharing congestion control. The proposed solution offers also a modified path monitoring mechanism, with more frequent heartbeat probing, to avoid stalling the application on an inactive path. The additional per path numbering introduced by LS-SCTP results in an unnecessary overhead as similar information can be inferred from the sender state variables and SACK chunks in their standard shape. This is the approach of another loadsharing proposal, called independent per path congestion control SCTP (IPCC-SCTP) introduced by Ye et al. [2004]. IPCC-SCTP, instead of using explicit per path numbering (as in case of LS-SCTP), provides local, per path mapping for each SCTP packet. This information is necessary

only at the sender-side to control the congestion, thus sending of redundant information is avoided. Thanks to this local mapping, IPCC-SCTP can govern congestion control, SACK processing, and retransmission handling on each path separately, instead of doing it for the entire association (when using just TSN information) as in standard SCTP.

5.1.1 Concurrent multipath transfer

IPCC-SCTP's implicit per path sequence numbering approach has been followed in the design of the most common loadsharing scheme so far, *Concurrent Multipath Transfer* (*CMT*), fully described in [Iyengar et al., 2006]. The idea of CMT was first introduced in [Iyengar et al., 2004a], however in contrast to IPCC-SCTP, CMT has been further developed in the following years. To accommodate CMT, Iyengar et al. propose a new sender architecture, where each path has a virtually separate buffer to guarantee path independence. This modification preserves TCP-friendliness under the assumption that the bottleneck is not shared by the paths.

Of course, such a *multibuffer* sender structure has its implications on congestion control, and therefore several changes to standard SCTP must be considered. All algorithms cited here were proposed and tested in fixed networks by Iyengar et al. [2006], and in their current shape will be incorporated to wireless scenarios in the analysis presented in Section 5.2:

1. To handle congestion control per-path, not per association, a sender cwnd growth algorithm (cwnd update for CMT - CUC) has been proposed. Thus, SACKs updating the CumTSN received in-order per path and out-of-order per association increase the cwnd on that path.
2. Fast retransmission needs slight modification as reordering introduced on the sender side can provoke unnecessary spurious fast retransmissions with cwnd implications. Elimination of spurious fast retransmissions is handled by the Split Fast Retransmit (SFR) algorithm that takes into account not only SACK information, but also transmission destination for each TSN when triggering the retransmission to a given path. The missing report counter is increased only at the destination where considered TSN was sent.
3. The CMT receiver should not send immediate SACKs irrespectively of whether arriving packet has been received in order or not, as networks may be vulnerable for the increased ACK traffic. As the SCTP receiver does not distinguish loss from reordering introduced by a CMT sender, therefore to correctly infer losses at the sender, an algorithm called Delayed ACK for CMT (DAC) was applied. On the receiver side, DAC algorithm adds to SACK information about the number of data PDUs received since the last SACK was sent (a flag in chunk header indicates either one or two PDUs received). This information is processed on the sender side, to indicate the SFR algorithm by how many the missing report counter should be increased.
4. CMT requires an appropriate policy to handle retransmissions. This topic has been investigated in more detail in [Iyengar et al., 2004b], with five retransmission policies being proposed:
   (a) RtxSame - send retransmission to the same path,
   (b) RtxAsAp - send retransmission to any destination that has cwnd space available,
   (c) RtxCwnd - path with the highest cwnd is chosen,
   (d) RtxSsthresh - path with the highest ssthresh is chosen,
   (e) RtxLossRate - path with the lowest loss rate is chosen; loss rate of each path is known in advance.

   According to Iyengar et al. [2004b], the best results for bulk applications were achieved by the loss rate-based policies, i.e. the policies that either try to estimate the loss rate of each path, namely RtxCwnd and RtxSsthresh, or know the loss rate in advance (RtxLossRate). Following this conclusion, and taking into account that cwnd reacts faster than ssthresh to change of the link conditions, the RtxCwnd policy (path with the highest cwnd handles the retransmission), has been chosen to be used in the experiments presented in this work.

Despite of an extense work spent on its development, CMT can provoke the following problems, which relevance should be assessed when considering handover scenarios:

1. Receiver buffer (rbuf) blocking (receiver buffer is filled with out-of-order data) caused by complete or short-term failures. This problem has already been tackled by Iyengar et al.

[2005, 2007] who focused in their work on wired-only scenarios. A solution that partially mitigates receiver buffer blocking, called CMT Potentially Failed (CMT-PF) has been proposed in [Natarajan et al., 2009, 2006]. CMT-PF marks the path that has experienced a failure (single timeout) as potentially failed, and stops transmitting data on such a path, until a positive heartbeat probe is returned. In case the PMR threshold was exceeded (with PMR + 1 consecutive failures) the path is marked as inactive (same as in standard SCTP). The PF state prevents the PMR parameter settings from degrading the throughput performance during failure scenarios, as the exponential backoff mechanism clocks only the HB packets. CMT-PF proposal is dedicated to lossy scenarios, e.g. wireless networks, thus making feasible the idea of applying CMT to improve transport-layer handover.

2. An ambiguity at the sender for the SACKs with the same CumTSN that acknowledge various Gap ACK blocks: first more Gap ACK blocks are acked on the faster path followed then by a packet with fewer Gap ACK blocks received on the slower path. This can lead to an unnecessary retransmission in case the difference between paths' bandwidth is high.
3. Incorrect RTT estimate on a slower path that comes from the ambiguity of the SACK received on the faster path that also acknowledges the packet marked for a RTT estimation on the slower path.

5.1.2 Scheduling algorithms

Additionally, loadsharing support can be complemented with source scheduling algorithm. A source scheduling algorithm picks the optimal path for transporting a DATA chunk, estimating which of the available paths is most likely to deliver (deliver first) the DATA to the receiver. An estimate takes into account current path conditions, e.g., number of outstanding chunks, available bandwidth, etc. The work by Casetti et al. [2004] provides an initial idea for load balancing based on a bandwidth-aware source scheduling extension to SCTP. Casetti et al. suggest sending back-to-back a pair of heartbeat packets to estimate the available bandwidth, and picking the fastest path to transmit data, as the simplest approach named Sender-Based Packet-Pair (SBPP) SCTP. This idea is further developed with a design of Westwood-like SCTP (W-SCTP) proposed by Fiore and Casetti [2005]; Fiore et al. [2007]. Apart from introducing a multibuffer structure and per-path congestion control, modifications similar to these described for CMT, Fiore and Casetti employ a packet scheduler that maximizes the chance that packets sent on paths with different bandwidths will arrive in order at the receiver, thus minimizing the receiver buffer blocking problem. The bandwidth estimation is made in a Westwood-like manner, giving the name for the proposed scheme. Moreover, an explicit advance acknowledgment algorithm and a minimum bandwidth estimate threshold are provided to increase the robustness of the presented approach. An exhaustive comparison of both approaches (SBPP and W-SCTP) based on emulation results is provided by Perotto et al. [2007]. W-SCTP is also used together with the PR-SCTP extension (the entire scheme is named W-SCTP-PR) to provide support for real-time applications, as presented in [Fiore and Casetti, 2005]. Further studies on multimedia traffic have been conducted by Rossi et al. [2006]. The use of load balancing to support the needs of video applications has also been evaluated by Abd El Al et al. [2007].

5.1.3 Taxonomy

To complement the review of the related work on transport layer loadsharing with SCTP, Fig 5.1 presents distribution of the loadsharing-related research over the dimensions of the taxonomy introduced by Budzisz et al. [2008], described already in more details in this work in Section 3.2.1. What possibly differs loadsharing from the other two analyzed uses of multihoming (i.e., originally considered robustness and transport layer mobility) is the variety of possible applications (contribution is witnessed in each category but Other applications, with the most important application being bulk transfer) and considerable versatility of analyzed network scenarios (more or less equally distributed between categories of wired and wireless domain). Again, simulation proves to be the most common study approach, whereas articles based on the emulation results are the second main group, with a considerably lower share.

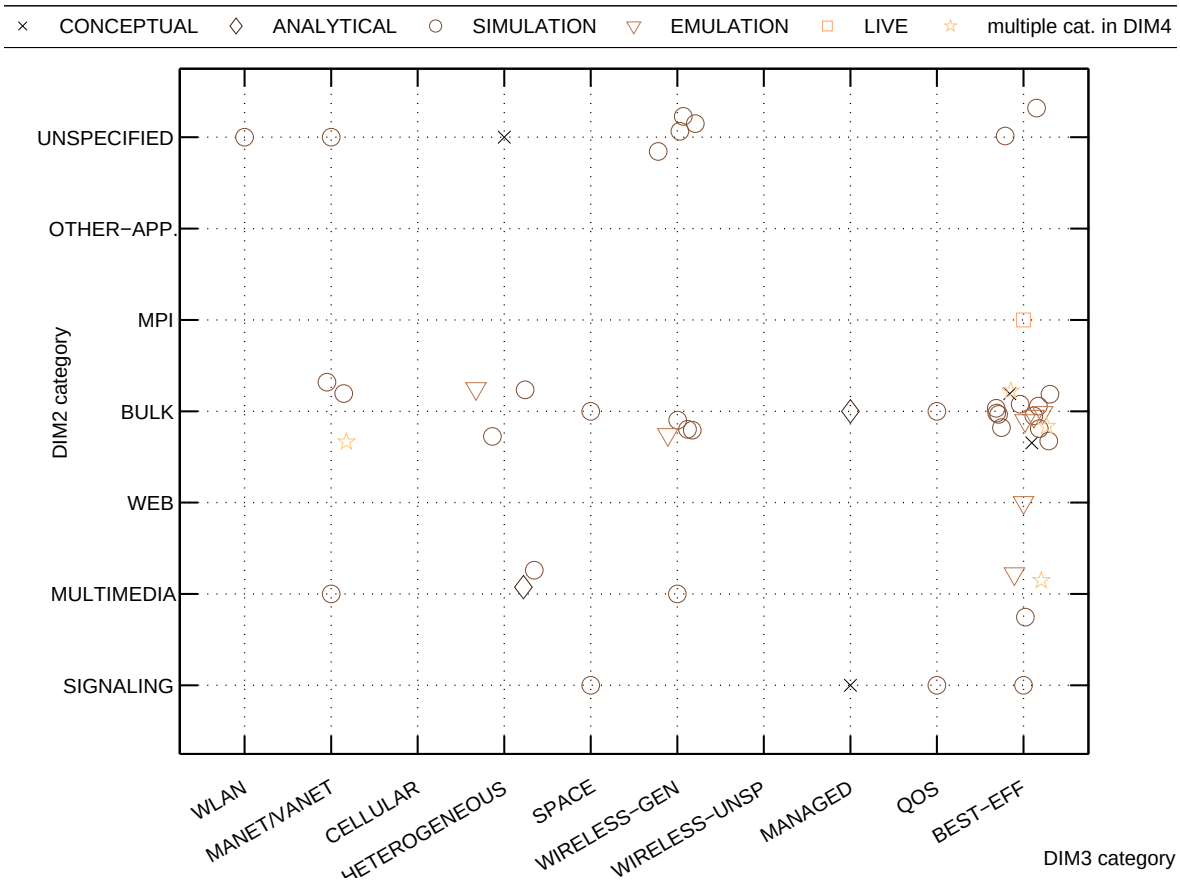


Figure 5.1: Scatter plot of all loadsharing-related articles.

5.2 CMT to improve transport layer mobility: the mSCTP-CMT scheme

The CMT-PF proposal is dedicated to lossy scenarios, although not particularly designed with wireless networks in mind, thus making feasible the idea of applying CMT to improve transport-layer handover. Using loadsharing in such context was originally proposed by Goff and Phatak [2004]. Goff and Phatak in the initial experiments use the multistreaming feature to facilitate loadsharing (for ease of the practical implementation), so that each stream is handled on a separate path. As a continuation of these works, Huang and Tsai [2007] present the design of a complete transport-layer mobility scheme that takes into account loadsharing as a possible enhancement. In this section a framework for using mSCTP-CMT scheme, which includes CMT-PF, but herein will be referred to as mSCTP-CMT, is initially explained, followed by the description of the evaluating approach, theoretical analysis and performance evaluation by means of simulations.

5.2.1 Scenario description

Fig. 5.2 shows a general handover scenario in heterogeneous wireless networks, where a MN is traversing one particular radio access network (RAN #1) coverage area towards the coverage area of a neighboring RAN #2. The RANs have an overlap area, i.e., an area where both RANs provide coverage. It is assumed that the MN is capable of handling transmissions on multiple links simultaneously. This assumption is fairly reasonable, as in the near future, nearly all mobile multimedia devices will be equipped with multiple network interfaces, despite the current power consumption constraints. Consequently, once an MN enters the overlap area, multiple links are physically

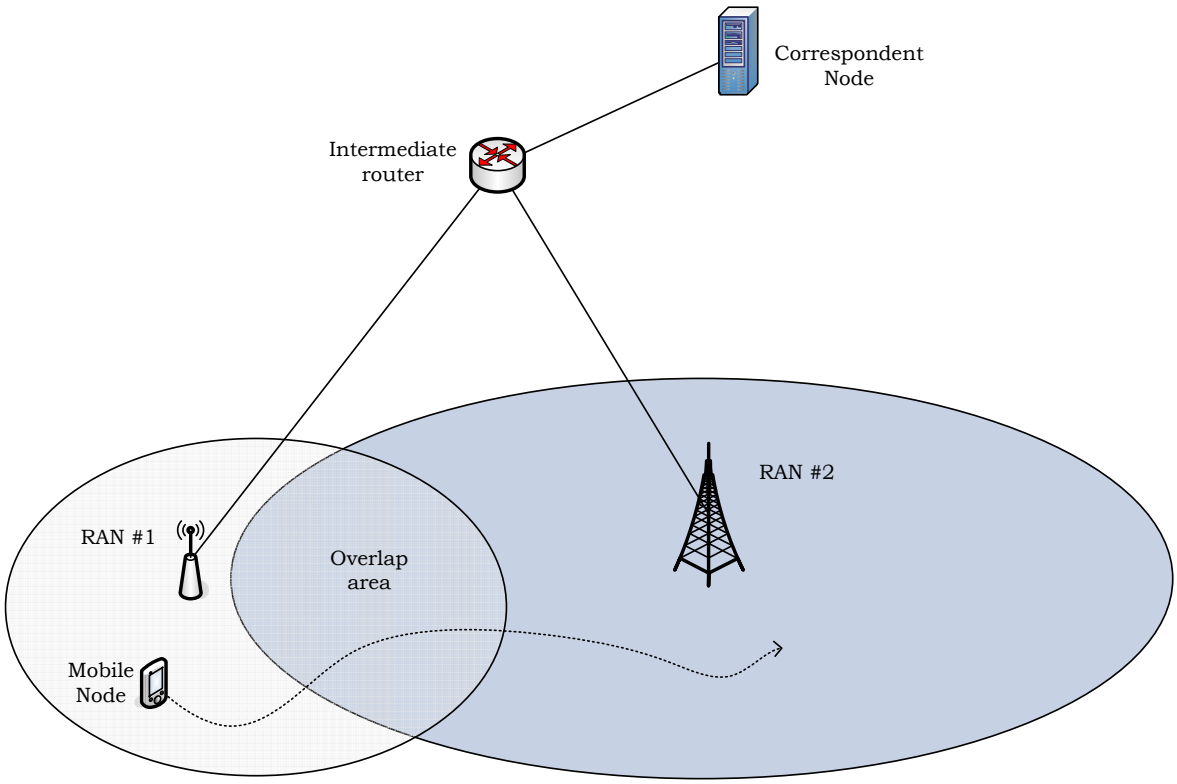


Figure 5.2: Proposed CMT scenario.

available for simultaneous data transmission.

It is further assumed that the different paths do not share bottlenecks, i.e., the radio link of each path is the bottleneck. This is a coherent supposition taking into account the mixed wired-wireless topology of the envisaged scenarios, and as it was explained in Section 5.1.1, it is vitally important for the use of CMT. As a consequence, the bottlenecks on each path are independent and a sender can consider a per-path congestion approach, while still preserving overall *TCP-friendliness*.

Now focusing on the proposed mSCTP-CMT handover scheme, this work considers one directional bulk data flow from a CN to the MN. As presented in Fig. 5.3, a MN configured initially with IP_1 address before entering an overlap area is using the mSCTP protocol to transfer data on a single link. When the MN enters the overlap area, the coverage of RAN #2 is discovered. To get the new link operational, the MN undergoes the correspondent network registration procedure. Both the network discovery process and registration procedure details [Honda et al., 2007] are outside the scope of this work. As soon as the network address IP_2 in RAN #2 is operational, the CN must be informed about the new destination (by means of ASCONF chunk), and has to verify its availability (sending HB chunk). Once the new destination is confirmed, the IP_2 address is considered available for normal data transfer. At this point, CMT can be exploited while having two paths available. Finally, when the MN leaves the overlap area, it is necessary to: (1) quit CMT mode, (2) handle any retransmissions of packets that were in flight on the link that just went down, and (3) perform all necessary congestion adjustments on the current path for the once again single-homed MN.

The main goal of this work is to evaluate whether is possible to apply CMT in the presented handover scenario, what gains can be achieved, and in which situations, if any, might the use of CMT degrade service. To this end, the described mSCTP-CMT handover scheme will be compared to two benchmark schemes:

- a handover based on the optimized failover mechanism of standard SCTP, as illustrated in Fig. 3.18c. To optimize for handover scenarios, standard SCTP's failover mechanism was tailored with low `PMR` settings and the `RTOMin` limitation was removed, as described in Sec-

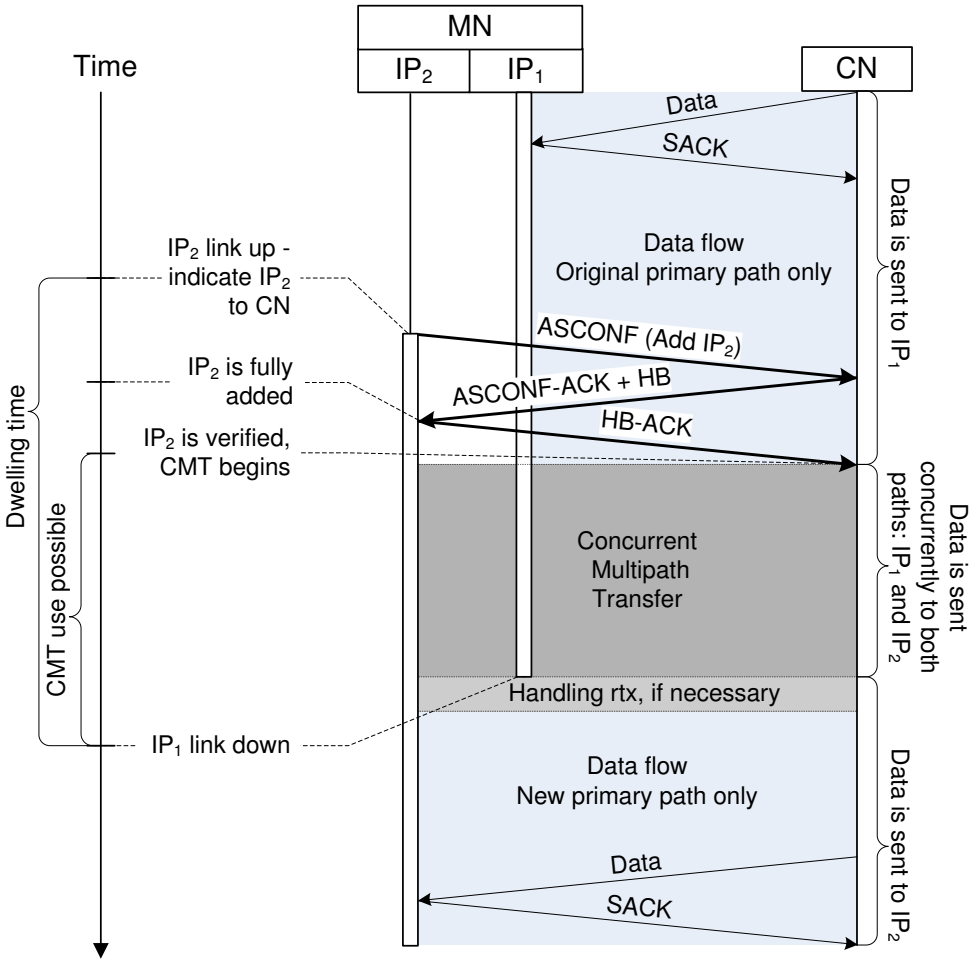


Figure 5.3: mSCTP-CMT handover scheme.

tion 4.3.1 and in [Budzisz et al., 2008]. Again it should be reminded that the usage of SCTP's failover mechanism to trigger primary path change impedes sending data while switching the paths.

- a handover based on mSCTP, as illustrated in Fig. 3.18. As briefly mentioned in Section 3.3.2, an important challenge for mSCTP is to provide an appropriate handover policy, and in particular, to optimally select the instant when the ASCONF chunk with Set Primary parameter should be sent. Here two ideal schemes reflecting a *range* of possible adjustments to the envisaged scenario are provided: (1) the *best case* (Fig. 3.18a), a policy resulting in MN staying in the better quality RAN (in terms of bandwidth, propagation delay or both) as long as possible, and (2) a *worst case* (Fig. 3.18b) keeping the MN in the poorer quality RAN for the maximum duration (e.g., as a result of a policy based on the cost of the link utilization).

Lastly, two important parameters to evaluate mSCTP-CMT performance are named:

1. *dwelling time* (t_{dwell}), defined already in this work in Section 3.3.2, as the effective time (both destinations are available at the transport layer) a MN remains in the overlap area, i.e., including transport-layer signaling, before transfer of DATA chunks can be started. Dwelling

Table 5.1: Basic simulation parameters

Parameter name	Value / Range
Wired part (each path)	bandwidth: 100 Mbps one-way propagation delay: 5 ms *(modified in advanced analysis, Section 5.2.4)*
fast RAN	bandwidth: ($bw_{ratio}\times$ 384) kbps one-way propagation delay: 15 ms
slow RAN	bandwidth: 384 kbps one-way propagation delay: 80 ms
scenario pattern	RAN #1: fast RAN RAN #2: slow RAN *(modified in advanced analysis, Section 5.2.4)*
bw_{ratio}	1-14
dwelling time (in function of bw_ratio)	bw_{ratio} = 1-2: 2-80 s bw_{ratio} = 3-4: 2-40 s bw_{ratio} = 6-8: 2-20 s $bw_{ratio} \geq$ 10: 2-10 s
rbuf size	16-256 kB (ideal buffer up to 2 MB)
`RTOMin`	50 ms
`PMR`	Optimized failover, mSCTP: 1 CMT: 5
`SACK delay`	200 ms
Retransmission policies	mSCTP FastRtx: Same path mSCTP TimeoutRtx: Alternate path CMTRtx: path with largest cwnd
MTU size / Data payload	1500 / 1468 Bytes
File size	8 MB (5778 DATA chunks)

time is affected by the speed of the MN, as well as its movement pattern, and therefore may be crucial for using CMT. Depending on the scenario considered, t_{dwell} can vary in practice from a few seconds for fast MNs going across the overlap area to tenths of seconds for slow MNs traversing the overlap area.

2. *bandwidth ratio* (bw_{ratio}), defined as a ratio of the bandwidths available in the neighboring RANs (in this work always the faster bandwidth is related to the slower one, so the $bw_{ratio} \geq 1$). bw_{ratio} reflects the asymmetry of a handover scenario.

To study the feasibility of mSCTP-CMT for transport-layer handover, a series of simulation experiments in ns-2 (ver. 2.32) [NS-2] was conducted, adjusting an existing CMT-PF implementation to work with heterogeneous wireless environments (refer Appendix A.2.1). The most important simulation parameters in their basic configuration for various experiments performed in the scenario under test (Fig. 5.2) are presented in Table 5.1. More detailed specification of the parameters specific to a particular test is provided with each set of the experiments. However, before looking for the simulation results of the mSCTP-CMT in such a defined handover scenario, the maximum possible gain that can be achieved will be estimated, and related to mentioned mSCTP-based handover schemes.

5.2.2 Analytical model

The proposed scenario under test (Fig. 5.2) considers the following basic mobility pattern: (1) first the MN moves within the coverage area of RAN #1 (faster of the two), (2) after t_1 from the transmission start, the MN enters the overlap area where there is a possibility of applying CMT scheme

during t_{dwell}, and finally (3) MN leaves the overlap area and remains in RAN #2 (slower RAN), where again only one path is available for data transmission. Then, the minimum time necessary to transmit a file of size L (given that L is large enough, so that the transmission is not completed before leaving the overlap area) depends on the available bandwidth that the MN can achieve in each of the discussed regions. Hence:

$$T = \sum_{i=1}^{3} t_i = \sum_{i=1}^{3} \frac{L_i}{bw_i} = \frac{L_1}{bw_1} + \frac{L_{overlap}}{bw_{overlap}} + \frac{L_2}{bw_2} \tag{5.1}$$

As can be easily seen from formula (5.1) the main factor differentiating the performance of all discussed handover schemes is the bandwidth available in the overlap area ($bw_{overlap}$). If CMT is applied in the analyzed scenario with two paths available for using CMT during the t_{dwell}, the $bw_{overlap}$ can be estimated as:

$$bw_{overlap} \leq bw_1 + bw_2 \tag{5.2}$$

The maximum gain in terms of file transfer time reduction will be produced when the $bw_{overlap}$ would be equal to the sum of the values of each link bandwidth, bw_1 and bw_2, respectively. Therefore, theoretically the minimum time necessary to transmit the entire file of size L when CMT is applied in the overlap area is:

$$T_{cmt\_th} = t_1 + t_{dwell} + \frac{L - t_1 \cdot bw_1 - t_{dwell} \cdot (bw_1 + bw_2)}{bw_2}$$

leading to:

$$T_{cmt\_th} = t_1 + \frac{L}{bw_2} - (t_1 + t_{dwell}) \cdot bw_{ratio} \tag{5.3}$$

For mSCTP-based handover schemes the handover policy will influence the value of $bw_{overlap}$, changing its value from $\min(bw_1, bw_2)$ (bandwidth of the slower of the two RANs) in the worst case to $\max(bw_1, bw_2)$ in the best case, during the time the MN stays in the overlap area (t_{dwell}). Therefore, corresponding file transfer times are,

for the mSCTP worst case:

$$T_{msctp\_worst} = t_1 + \frac{L}{bw_2} - t_1 \cdot bw_{ratio} \tag{5.4}$$

and for the mSCTP best case:

$$T_{msctp\_best} = t_1 + t_{dwell} + \frac{L}{bw_2} - (t_1 + t_{dwell}) \cdot bw_{ratio} \tag{5.5}$$

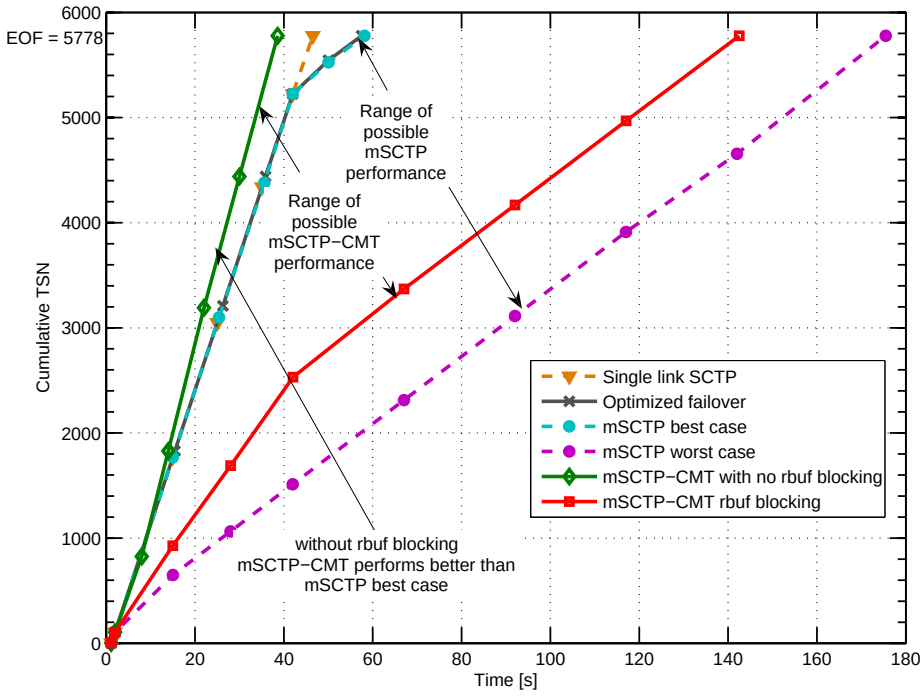
Finally, the maximum theoretical gain of mSCTP-CMT over mSCTP schemes (ΔT) in terms of file transfer time reduction can be expressed as:

$$\begin{aligned} \Delta T \ &\in \ [\Delta_{min}, \Delta_{max}] = \\ &\quad \left[T_{msctp\_best} - T_{cmt\_th}, T_{msctp\_worst} - T_{cmt\_th}\right] \\ \Delta T \ &\in \ [t_{dwell}, t_{dwell} \cdot bw_{ratio}] \end{aligned} \tag{5.6}$$

Moreover, equation (5.3) expressing the best theoretical time for the mSCTP-CMT handover scheme will be used to benchmark the results obtained in the simulations.

5.2.3 Basic performance evaluation

A basic analysis will examine not only (1) the possible gain that can be achieved using mSCTP-CMT handover scheme, as presented in [Budzisz, Ferrús, Casadevall, and Amer, 2009], but also (2) the potential of smoothing the transition process, as well as (3) the influence of dedicated CMT retransmission policies.



(a)

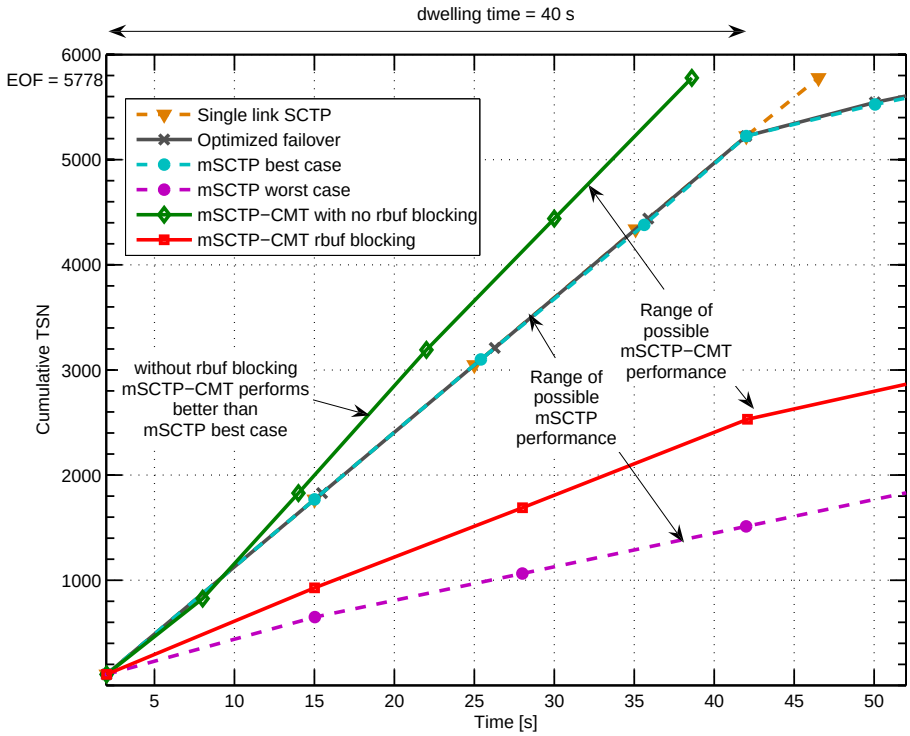
Figure 5.4: Performance comparison of all SCTP versions for $bw_{ratio} = 4$, and $t_{dwell} = 40\,s$: (a) entire file transfer.

TSN-time evolution

Fig. 5.4 presents a performance comparison in terms of TSN-time evolution diagram for all SCTP protocol versions described in Section 5.2.1, namely: (1) optimized failover, (2) mSCTP-based solution (both best and worst case), and (3) the mSCTP-CMT scheme (with two different values of rbuf size, 32 kB and 512 kB, to illustrate receiver buffer blocking problem). Additionally, obtained results are related to a single link SCTP performance (i.e., considering a hypothetical case where faster of the two RANs is available for the complete file transfer). As can be seen in Fig. 5.4b, the overlap area (2-42 s in the time scale) is the zone of special interest, witnessing different slope values for the presented SCTP flavors. Nearly all possible mSCTP-CMT gain over pure mSCTP-based handover schemes will be produced here, if strict constraints on rbuf size are met. The range for possible mSCTP-CMT performance gain is significant, from being much worse than the best mSCTP policy (almost as bad as the worst one) in presence of rbuf blocking, to gaining over the best mSCTP policy (as well as failover-based scheme and faster of the two links) for an appropriate rbuf size adjustment. Still in any of the considered cases mSCTP-CMT did not perform any worse than the worst mSCTP policy. In an effort to identify the application area for mSCTP-CMT, all important factors such as, t_{dwell}, bw_{ratio} and rbuf size will be analyzed now in more detail.

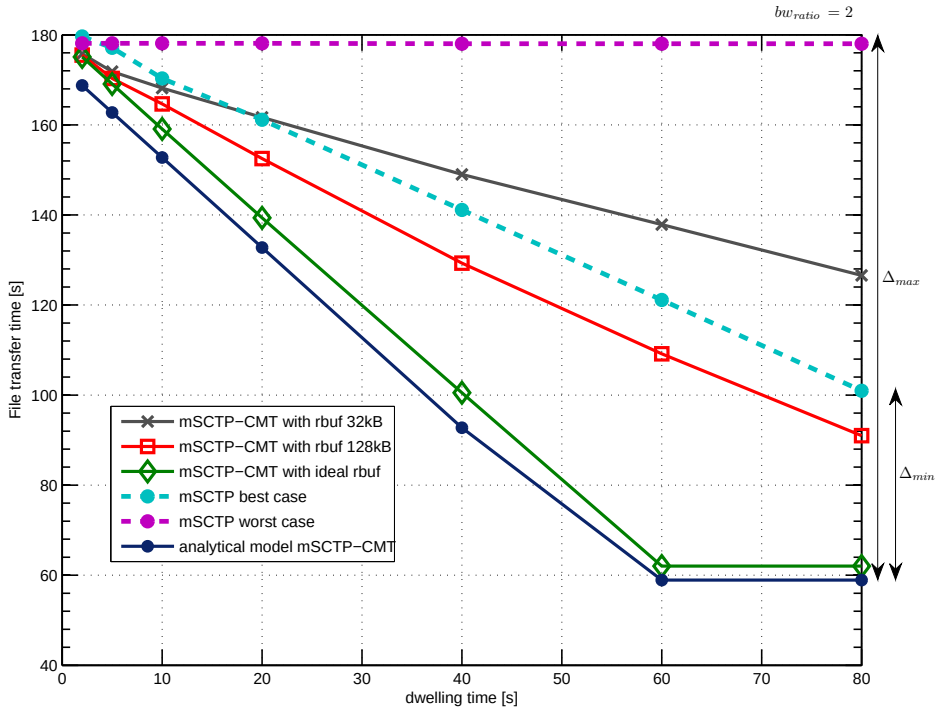
Dwelling time

As seen in dwelling time graph (Fig. 5.5), users with short t_{dwell} would have no significant gain from using mSCTP-CMT when compared to mSCTP-based schemes in any of the presented cases. In contrast, having long t_{dwell} can effectively benefit from mSCTP-CMT. This conclusion follows the results of theoretical analysis from Section 5.2.2, and so do the respective gains of mSCTP-CMT over both extreme cases of mSCTP (correspondent gains are marked at each graphics). For $bw_{ratio} = 2$ the supposition made in the theoretical analysis that the file is large enough, so that the

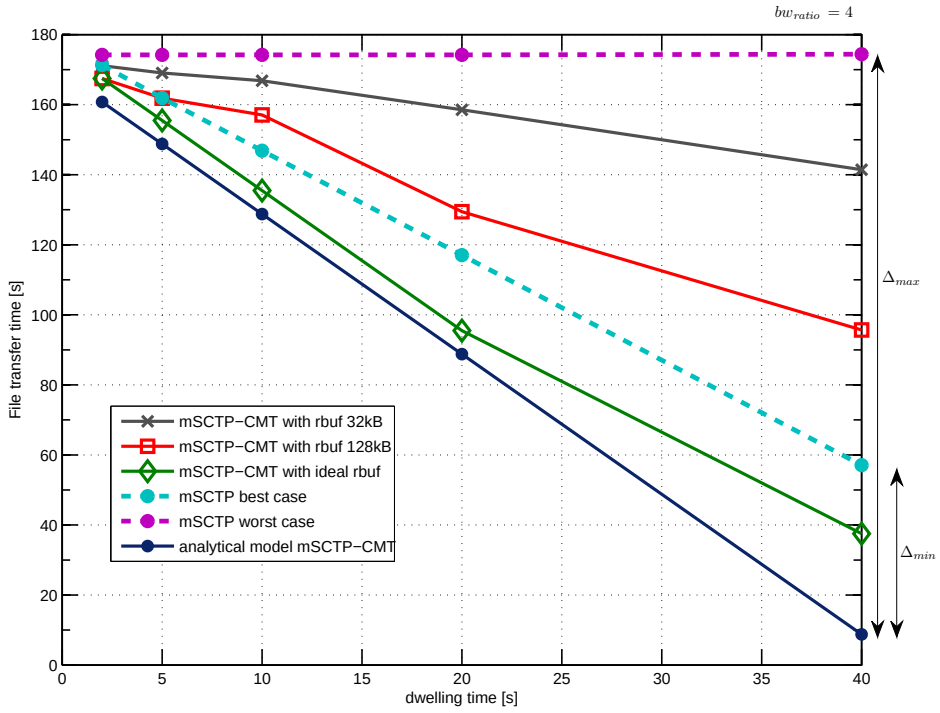


(b)

Figure 5.4: Performance comparison of all SCTP versions for $bw_{ratio} = 4$, and $t_{dwell} = 40\,s$: (b) overlap area only.

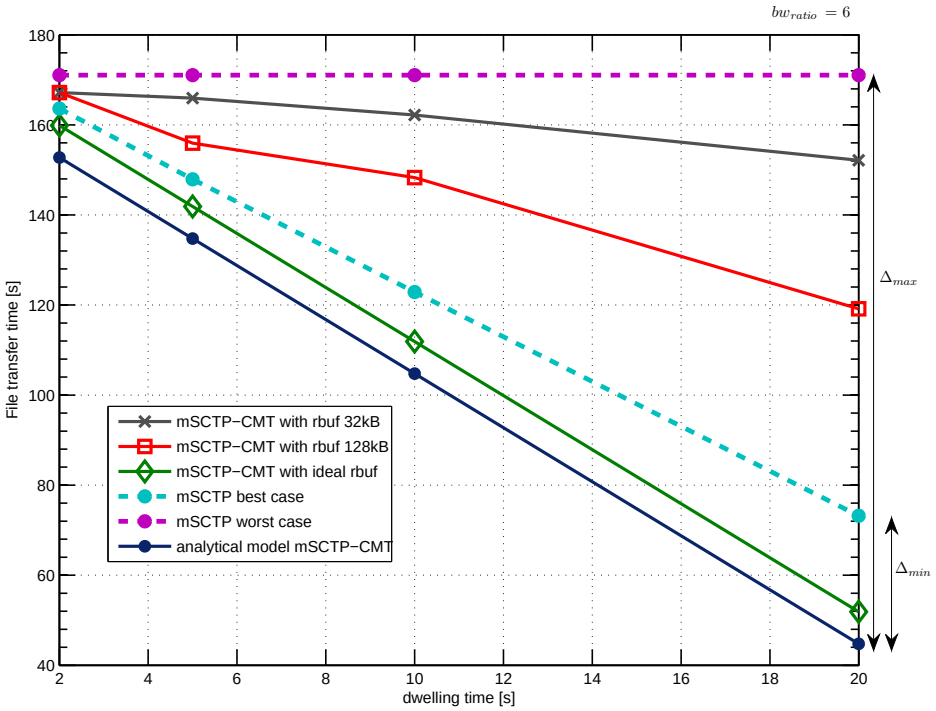


(a)

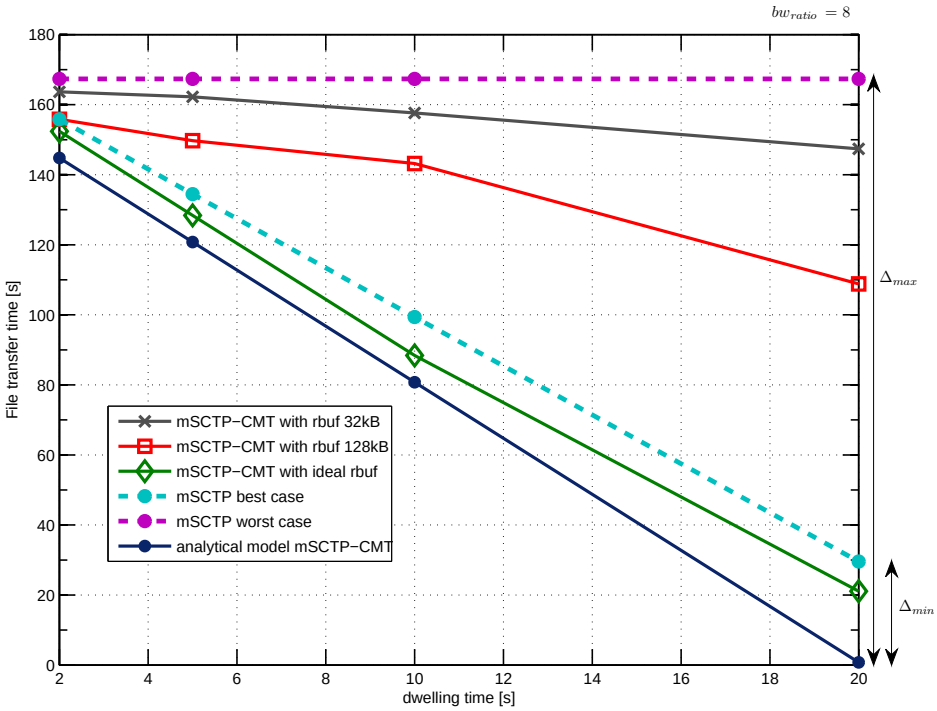


(b)

Figure 5.5: Comparison in function of dwelling time for: (a) $bw_{ratio} = 2$; and, (b) $bw_{ratio} = 4$.

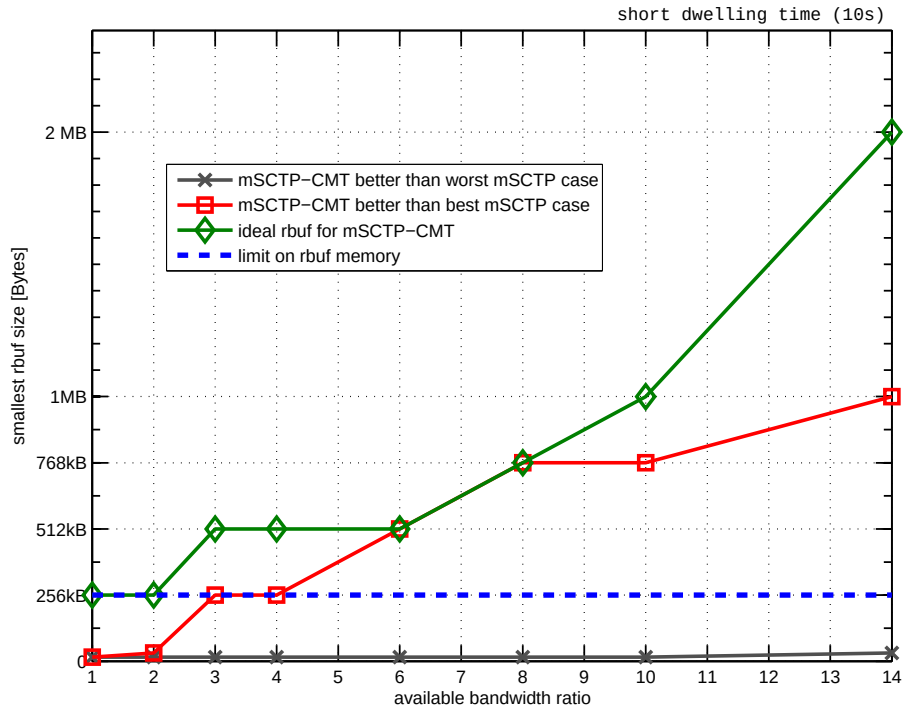


(c)



(d)

Figure 5.5: Comparison in function of dwelling time for: (c) $bw_{ratio} = 6$; and, (d) $bw_{ratio} = 8$.



(a)

Figure 5.6: Rbuf size constraints for: (a) short (10s) dwelling time.

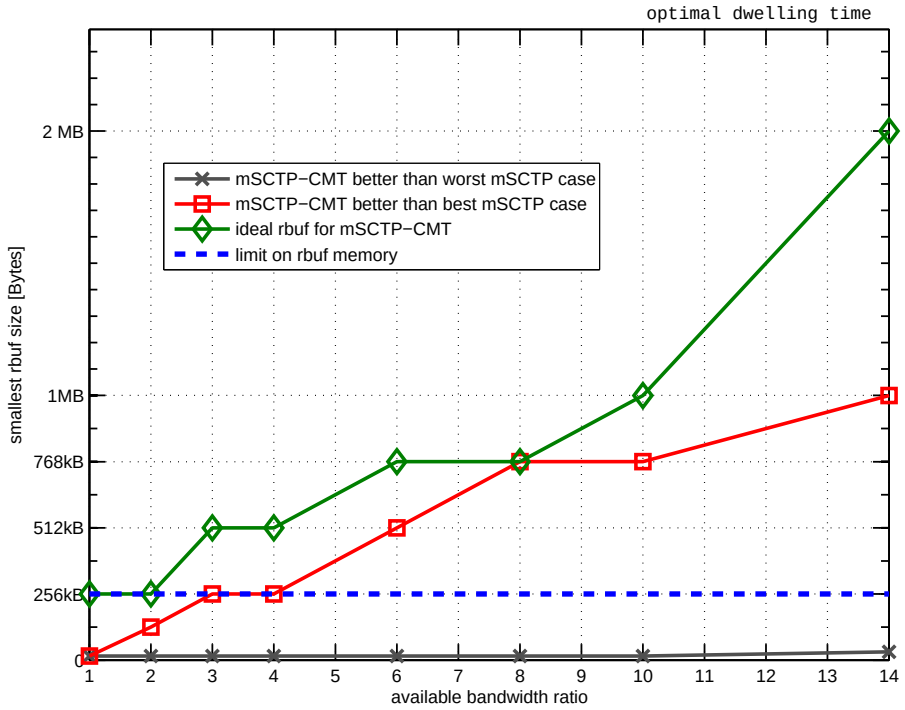
transmission is not completed before leaving the overlap area is meet in all cases but for $t_{dwell} = 80\,s$, where formula (5.3) has to be modified accordingly. It is important to stress that again the impact of rbuf blocking can be witnessed. Within the tested scenario mSCTP-CMT was not capable of outperforming the best mSCTP case for a rbuf not exceeding 256 kB. Having an *ideal rbuf* (i.e., large enough to avoid rbuf blocking) led to nearly optimal performance marked by the theoretical trend based upon equation (5.3).

Receiver buffer blocking

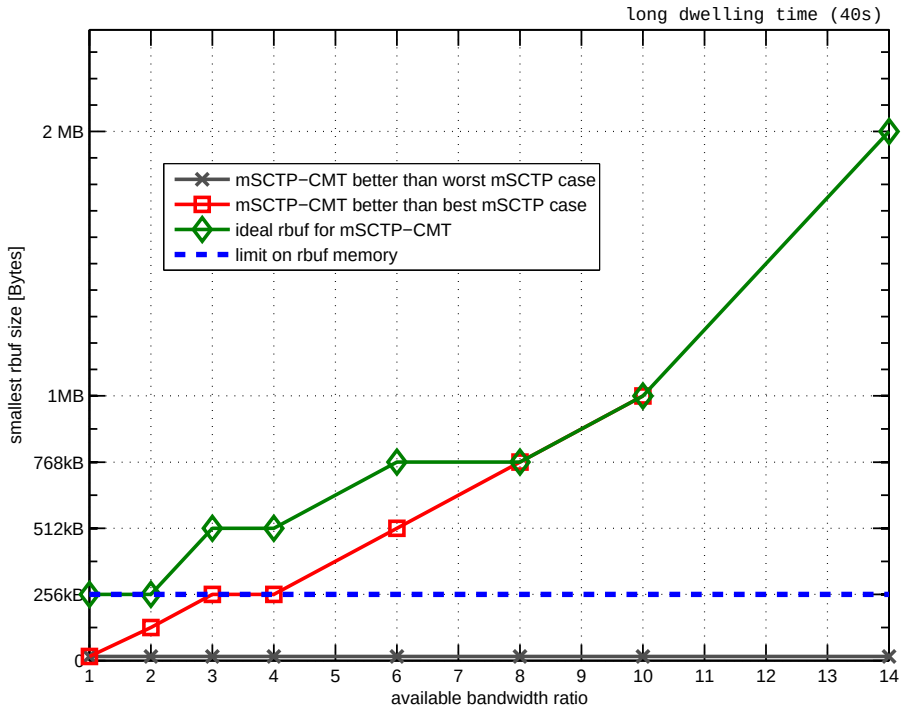
To better understand the *receiver buffer blocking problem*, in Fig. 5.6, the following metrics are provided: the smallest rbuf size that guarantees mSCTP-CMT outperforms mSCTP in terms of overall file transfer, for worst and best policy accordingly, as well as the smallest rbuf size without the rbuf blocking problem at all. The result is clear, not much asymmetry between two paths is allowed. Assuming a 256 kB limitation on rbuf memory at MN is reasonable nowadays, only a $bw_{ratio} \leq 2$ makes the application of mSCTP-CMT feasible, if the design concern is not having rbuf blocking at all. Less conservatively, if improvement over the mSCTP best case is the sole design goal, mSCTP-CMT scope of use extends to $bw_{ratio} \leq 4$, a value that would correspond, for instance, to a handover from WLAN to UMTS. Within the outlined limits, mSCTP-CMT can shorten considerably the file transfer, allowing to exploit the availability of multiple links during the entire dwelling time. Beyond these limits, the difference between both links makes the application of mSCTP-CMT pointless.

Smoothing the transition process

A potential gain from using CMT is not only related to the fact of simultaneous data transmission while the MN stays in the overlap area. The improvement offered by the mSCTP-CMT scheme can



(b)



(c)

Figure 5.6: Rbuf size constraints for: (b) optimal (longest possible, see Table 5.1); and, (c) long (40s) dwelling time.

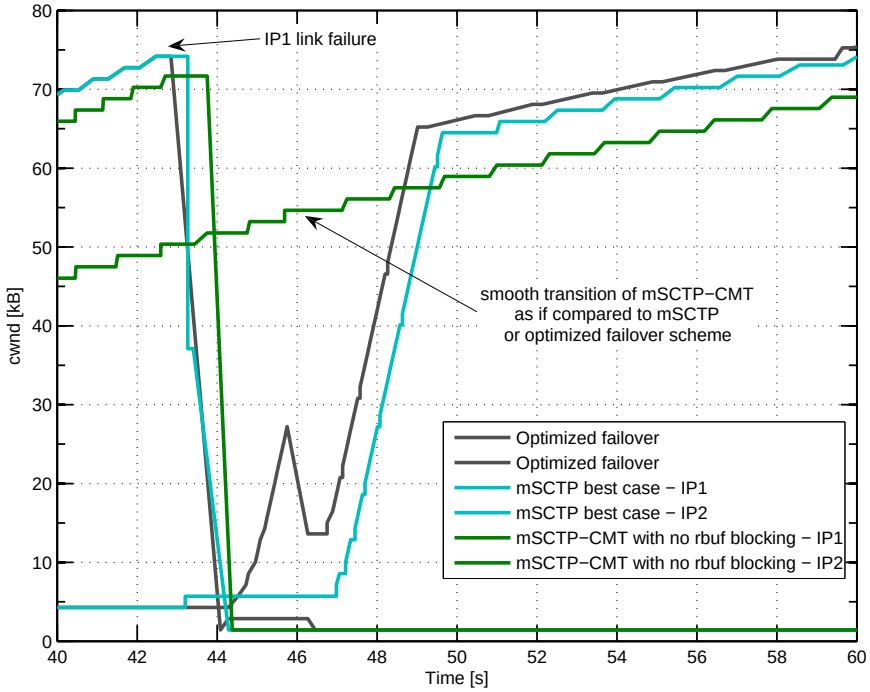
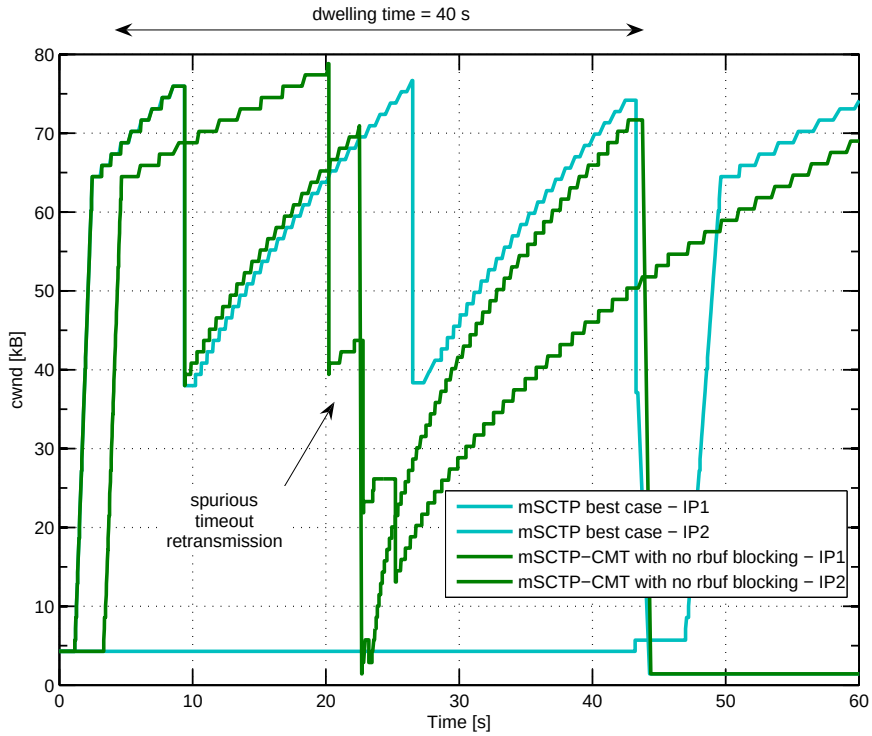


Figure 5.7: Smoothing effect.

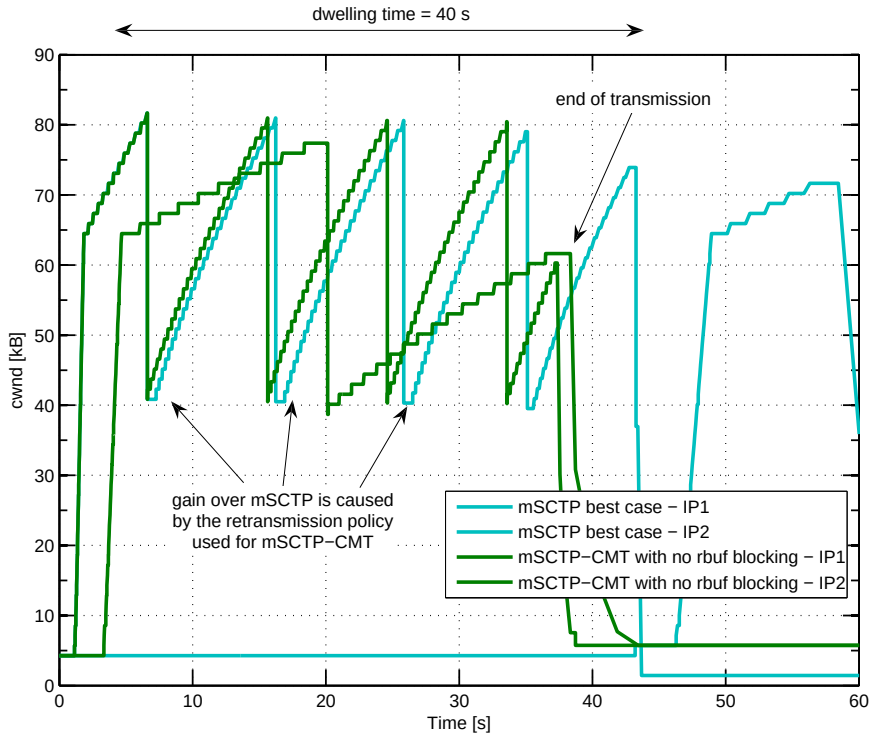
be also witnessed during the transition process, as shown in Fig. 5.7. In the presented example, the overlap area ends around 42 s, when the IP_1 link fails. Both failover- and mSCTP-based schemes, as soon as the failure is encountered, switch the paths and start to transmit data on the new address from the `cwnd.init` value in the slow start phase. Meanwhile, mSCTP-CMT quits the CMT, and switches back to a transmission on a single link using the current value of the cwnd on the IP_2 link. This uninterrupted transmission of mSCTP-CMT, called here a *smooth transition*, results in the overall file transfer gain over both failover- and mSCTP-based scheme during the time spent by the latter on: first (1) retransmission of the packets lost because of the IP_1 failure, and then (2) increasing of a cwnd to the value of that of mSCTP-CMT scheme in the moment of failure. In the example shown in Fig. 5.7 it takes about 6-7 s for failover- and mSCTP-based scheme to recover the value of cwnd to the level of that of mSCTP-CMT at the moment the IP_1 link went down. During this time mSCTP-CMT scheme has the cwnd value already adjusted to the current network conditions and therefore can sent more packets that the other two discussed schemes.

Influence of the retransmission policy

Out of five different retransmission policies designed to be used with CMT by Iyengar et al. [2006] that were described in this work in Section 5.1.1, the policy RtxCwnd, recommended in [Iyengar et al., 2004b] as the most appropriate for the bulk transfer has been chosen to perform experiments evaluating the behavior of mSCTP-CMT scheme. Fig. 5.8 illustrates how the CMT retransmission policy can affect the gain obtained from the simultaneous use of all available links. For a scenario with little asymmetry ($bw_{ratio} = 2$), such as the one presented in Fig. 5.8a, handling retransmissions on the path that has the biggest value of cwnd leads to a spurious timeout caused by the loss of packet being currently fast retransmitted. Consequently, one of the links is brought down to one MTU and has to start again in the slow start phase. A sample tracing of such a behavior is given below. In contrast, in the scenario with higher bandwidth asymmetry ($bw_{ratio} = 4$), shown



(a)



(b)

Figure 5.8: Influence of the retransmission policy RtxCwnd on the mSCTP-CMT behavior: (a) spurious timeout retransmission; and, (b) additional gain.

in Fig. 5.8b, the proposed retransmission policy results in better performance when mSCTP-CMT is compared to the failover- or mSCTP-based scheme (both handle fast retransmission on the same path where it occurred). With mSCTP-CMT using RtxCwnd policy, the faster path can start new data transmission without needing to wait until the retransmission ends on the slower path.

Occurrence of spurious timeout retransmissions in function of retransmission policy applied to CMT has already been investigated in more detail by Iyengar et al. [2006]. As shown in the example provided here, the choice of the loss-based policy aiming to pick the path with the lowest estimated loss rate, and thus minimize the probability of a spurious failover, does not guarantee avoiding completely this type of events. An open question, left for the future research, is further evaluation whether relying on the cwnd value to select the path for the retransmission is the most appropriate approach in the handover scenarios.

Tracing for mSCTP-CMT with RtxCwnd policy leading to a spurious timeout

Sample of simplified ns-2 tracing provided here has the following format in the subsequent fields: 1 - event: pkt enqueued (+), dequeued (-), received (r) or dropped (d); 2 - time; 3 - source node id (CN: 1,2; intermediate router: 6,7; MN: 4,5); 4 - destination node id (node 4 has address IP_1, and node 5 has address IP_2); 5 - protocol name (sctp); 6 - packets size; 7 - chunk type: DATA (———-D), SACK (———-S), HB (———-H) and HB-ACK (———-B); 8 - TSN; 9 - packet id; 10 - stream id (SID); 11 - stream sequence number (SSN).

Explanation of the symbols additionally used in the provided tracing example: saddr - source address; daddr - destination address; pba - partial_bytes_acked; out - number of outstanding bytes on a given destination; peerRwnd - peer rwnd size; rto - RTO value.

Comments (in italics) are marked in-line.

```
packet is initially lost at daddr: 5 (node ids for this path: 2-7-5)
+ 18.475849 7 5 sctp 1500 -------D 1690 2497 0 1689
d 18.475849 7 5 sctp 1500 -------D 1690 2497 0 1689
saddr: 2  daddr: 5  cwnd: 79248 pba: 216 out: 77804 ssthresh: 65536
peerRwnd: 289408 rto: 3.134
...

drop detection:
r 18.502478 7 2 sctp 168 -------S 1527 2488 65535 65535
r 18.564978 7 2 sctp 168 -------S 1534 2498 65535 65535
r 18.627478 7 2 sctp 168 -------S 1540 2507 65535 65535
r 18.689978 7 2 sctp 168 -------S 1546 2516 65535 65535
...
r 20.002395 7 2 sctp 164 -------S 1674 2707 65535 65535
r 20.064895 7 2 sctp 164 -------S 1680 2716 65535 65535
r 20.127395 7 2 sctp 164 -------S 1686 2725 65535 65535
...
r 20.158188 6 1 sctp 168 -------S 1689 2736 65535 65535
r 20.189978 7 2 sctp 168 -------S 1689 2734 65535 65535
r 20.205104 6 1 sctp 172 -------S 1689 2745 65535 65535
r 20.220688 6 1 sctp 168 -------S 1689 2746 65535 65535

3 Dup Gap ACK reports on the same path, reduce cwnd to half at daddr: 5
(start fast rtx, fast recovery exit point: 1855)
time: 20.22069  saddr: 1  daddr: 4  cwnd: 66782
time: 20.22069  saddr: 2  daddr: 5  cwnd: 40358

FAST RTX to daddr: 4 is triggered (node ids for this path: 1-6-4):
time: 20.22069  saddr: 1  daddr: 4  RTX TSN: 1690
time: 20.22069  saddr: 1  daddr: 4  cwnd: 66782 pba: 60404 out: 66060
```

```
ssthresh: 38890 peerRwnd: 276196 rto: 0.922
+ 20.220688 1 6 sctp 1500 -------D 1690 2751 0 1689
- 20.220688 1 6 sctp 1500 -------D 1690 2751 0 1689
r 20.225808 1 6 sctp 1500 -------D 1690 2751 0 1689
+ 20.225808 6 4 sctp 1500 -------D 1690 2751 0 1689
- 20.887049 6 4 sctp 1500 -------D 1690 2751 0 1689
r 20.917674 6 4 sctp 1500 -------D 1690 2751 0 1689
+ 20.933299 4 6 sctp 164 -------S 1772 2831 65535 65535
r 20.955021 6 1 sctp 164 -------S 1772 2831 65535 65535
FAST RTX completed successfully before timeout

meanwhile another packet on the slower link is dropped:
+ 20.158188 2 7 sctp 1500 -------D 1855 2743 0 1854
- 20.158188 2 7 sctp 1500 -------D 1855 2743 0 1854
r 20.163308 2 7 sctp 1500 -------D 1855 2743 0 1854
+ 20.163308 7 5 sctp 1500 -------D 1855 2743 0 1854
d 20.163308 7 5 sctp 1500 -------D 1855 2743 0 1854
saddr: 2  daddr: 5  cwnd: 80716 pba: 240 out: 79272 ssthresh: 65536
peerRwnd: 287940 rto: 2.936

drop detection:
r 21.844638 6 1 sctp 68 -------S 0 1854 2959 65535 65535
r 21.875387 7 2 sctp 68 -------S 0 1854 2957 65535 65535
r 21.891513 6 1 sctp 68 -------S 0 1854 2967 65535 65535
r 21.907138 6 1 sctp 68 -------S 0 1854 2968 65535 65535
3 Dup Gap ACK reports on the same path, do not reduce cwnd at daddr: 5
(still in fast recovery)
time: 21.90714  saddr: 1  daddr: 4  cwnd: 71186
time: 21.90714  saddr: 2  daddr: 5  cwnd: 43294

FAST RTX triggered:
time: 21.90714  saddr: 1  daddr: 4  RTX TSN: 1855
time: 21.90714  saddr: 1  daddr: 4  cwnd: 71186 pba: 14198 out: 71932
ssthresh: 38890 peerRwnd: 312896 rto: 0.777
+ 21.907138 1 6 sctp 1500 -------D 1855 2973 0 1854
- 21.907138 1 6 sctp 1500 -------D 1855 2973 0 1854
r 21.912258 1 6 sctp 1500 -------D 1855 2973 0 1854
+ 21.912258 6 4 sctp 1500 -------D 1855 2973 0 1854
- 22.621424 6 4 sctp 1500 -------D 1855 2973 0 1854
r 22.652049 6 4 sctp 1500 -------D 1855 2973 0 1854
+ 22.683299 4 6 sctp 64 -------S 1989 3087 65535 65535
- 22.683299 4 6 sctp 64 -------S 1989 3087 65535 65535
r 22.698966 4 6 sctp 64 -------S 1989 3087 65535 65535
+ 22.698966 6 1 sctp 64 -------S 1989 3087 65535 65535
- 22.698966 6 1 sctp 64 -------S 1989 3087 65535 65535
r 22.703971 6 1 sctp 64 -------S 1989 3087 65535 65535
FAST RTX completed successfully but after timeout expiration!!!

Timeout expires on daddr: 4
time: 22.68463  saddr: 1  sport: 0  daddr: 4  dport: 0  DataTimeout,
peerRwnd: 199860 rto: 0.913
+ 22.684631 1 6 sctp 56 -------H -1 3089 65535 65535
- 22.684631 1 6 sctp 56 -------H -1 3089 65535 65535
r 22.689635 1 6 sctp 56 -------H -1 3089 65535 65535
+ 22.689635 6 4 sctp 56 -------H -1 3089 65535 65535
```

```
- 23.402674 6 4 sctp 56 -------H -1 3089 65535 65535
r 23.418257 6 4 sctp 56 -------H -1 3089 65535 65535
+ 23.418257 4 6 sctp 56 -------B -1 3170 65535 65535
- 23.418257 4 6 sctp 56 -------B -1 3170 65535 65535
r 23.433841 4 6 sctp 56 -------B -1 3170 65535 65535
+ 23.433841 6 1 sctp 56 -------B -1 3170 65535 65535
- 23.433841 6 1 sctp 56 -------B -1 3170 65535 65535
r 23.438845 6 1 sctp 56 -------B -1 3170 65535 65535

Timeout RTX triggered:
time: 22.68463  saddr: 2  daddr: 5  RTX TSN: 1855
time: 22.68463  saddr: 2  daddr: 5  cwnd: 44762 pba: 24980 out: 45508
ssthresh: 40358 peerRwnd: 199860 rto: 2.965
+ 22.684631 2 7 sctp 1500 -------D 1855 3088 0 1854
- 22.684631 2 7 sctp 1500 -------D 1855 3088 0 1854
r 22.689751 2 7 sctp 1500 -------D 1855 3088 0 1854
+ 22.689751 7 5 sctp 1500 -------D 1855 3088 0 1854
- 23.521465 7 5 sctp 1500 -------D 1855 3088 0 1854
r 23.632715 7 5 sctp 1500 -------D 1855 3088 0 1854
+ 23.632715 5 7 sctp 72 -------S 2076 3196 65535 65535
- 23.632715 5 7 sctp 72 -------S 2076 3196 65535 65535
r 23.714215 5 7 sctp 72 -------S 2076 3196 65535 65535
+ 23.714215 7 2 sctp 72 -------S 2076 3196 65535 65535
- 23.714215 7 2 sctp 72 -------S 2076 3196 65535 65535
r 23.719221 7 2 sctp 72 -------S 2076 3196 65535 65535
```

5.2.4 Extended performance evaluation

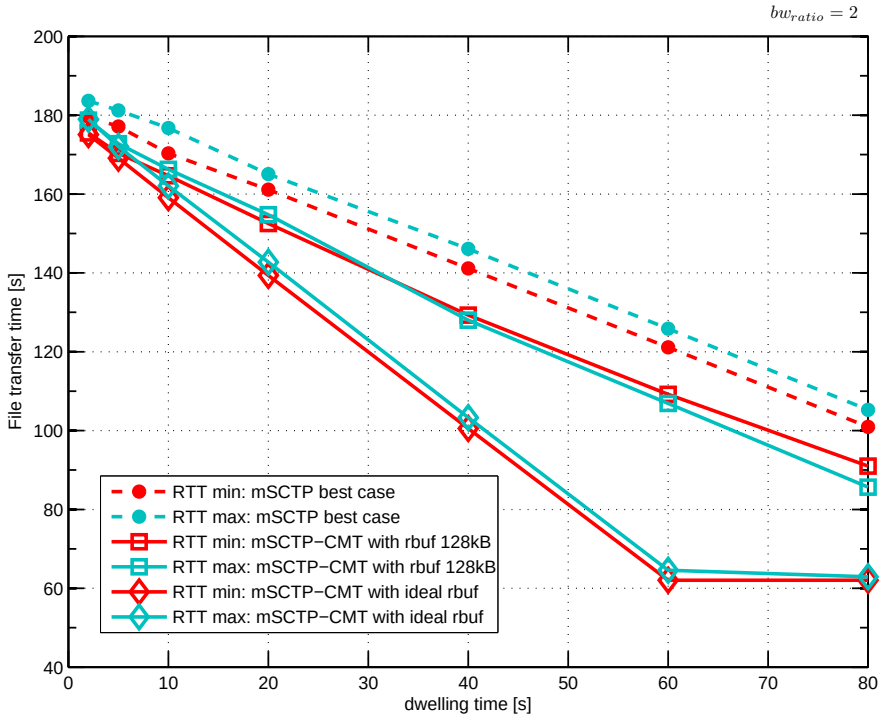
In further evaluation of the mSCTP-CMT scheme, small modifications will be introduced to the proposed scenario under test (Fig. 5.2). First, four different values of propagation delay are provided to evaluate the mSCTP-CMT performance in various RTT conditions. Then, two different patterns of the mobility scenario are considered: (1) transition from fast to slow RAN, and (2) from slow to fast RAN, accordingly. Table 5.2 summarizes all parameters that were modified with respect to the basic scenario configuration, described in Table 5.1. Some of the result provided by this analysis were published in [Budzisz, Ferrús, and Casadevall, 2009].

RTT value

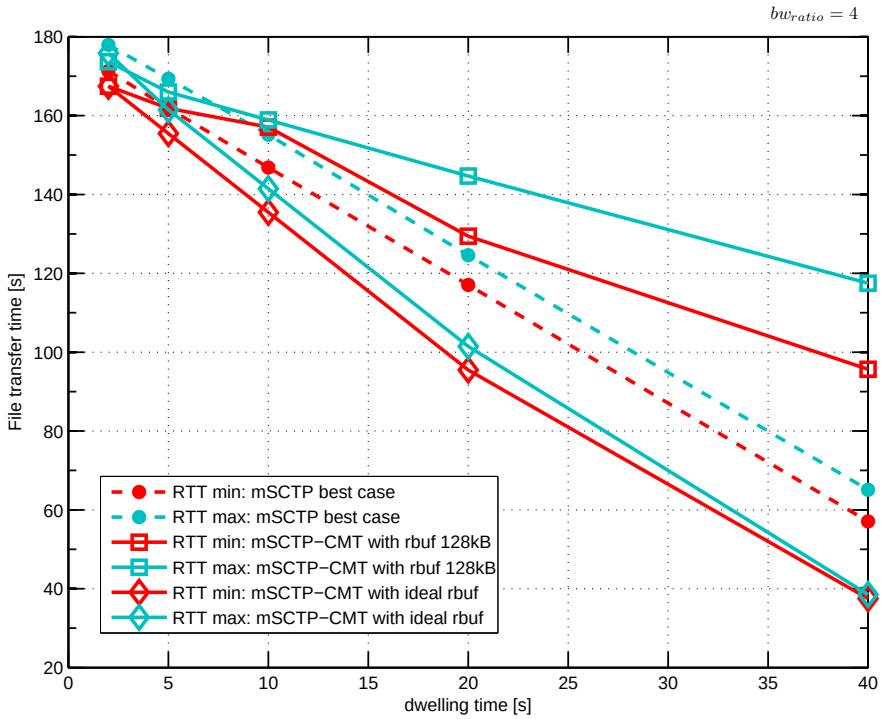
As there was not much difference in the performance of mSCTP-CMT for four considered RTT cases (according to Table 5.2) only the results for both ends of the considered RTT range were marked in Fig. 5.9. Again, as in case of the basic performance analysis (Section 5.2.3), the dwelling time graph reveals that mSCTP-CMT does not perform any better (but any worse either) than the remaining handover schemes in case of short t_{dwell} values (i.e., $t_{dwell} \leq 5\,s$). For $t_{dwell} \geq 10\,s$ a considerable

Table 5.2: Parameters modified for further evaluation of mSCTP-CMT

Parameter name	Value / Range
Wired part (each path)	bandwidth: 100 Mbps one-way propagation delay: 5-20-45-90 ms
scenario pattern #1: fast-to-slow change	RAN #1: fast RAN RAN #2: slow RAN
scenario pattern #2: slow-to-fast change	RAN #1: slow RAN RAN #2: fast RAN

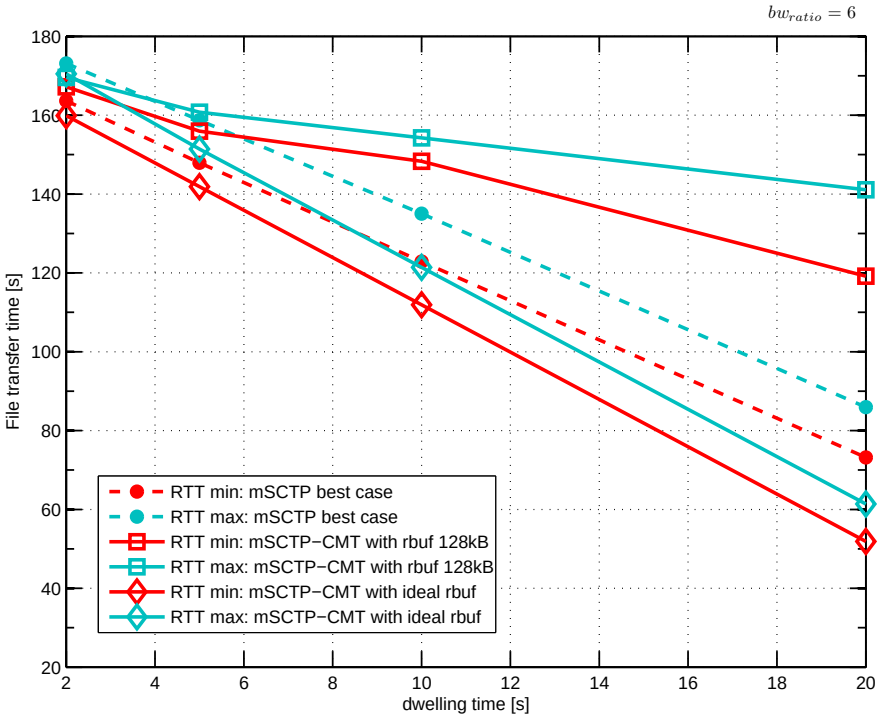


(a)

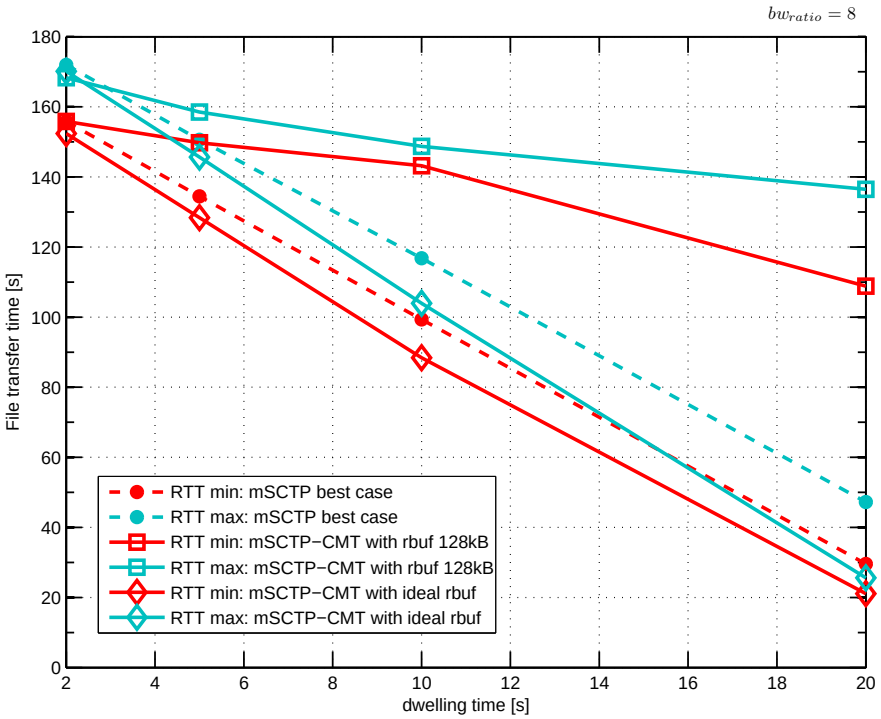


(b)

Figure 5.9: Comparison for different RTT values in function of dwelling time for: (a) $bw_{ratio} = 2$; and, (b) $bw_{ratio} = 4$.

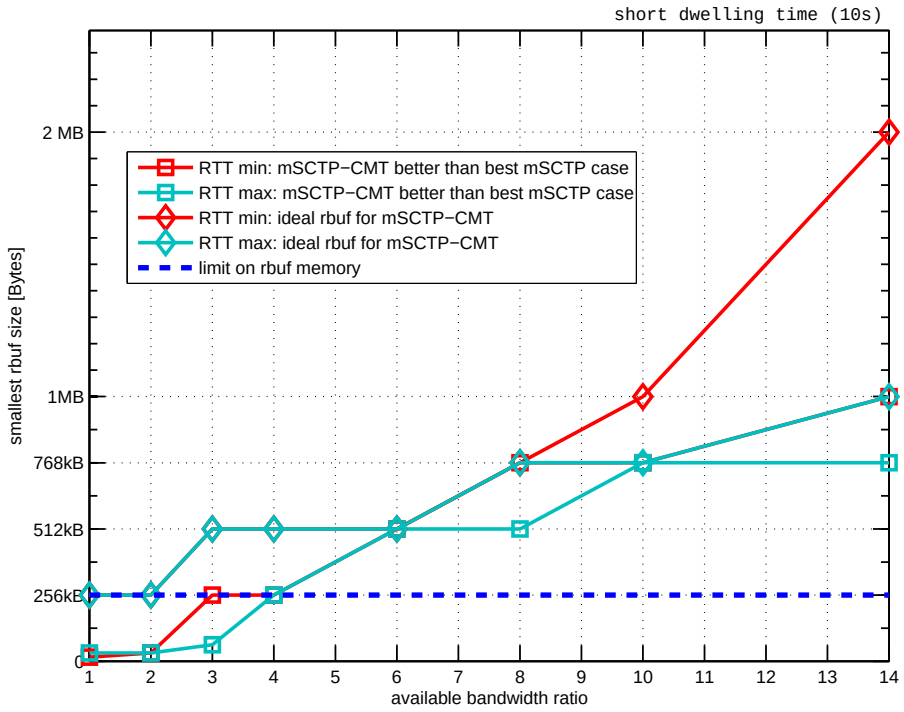


(c)



(d)

Figure 5.9: Comparison for different RTT values in function of dwelling time for: (c) $bw_{ratio} = 6$; and, (d) $bw_{ratio} = 8$.



(a)

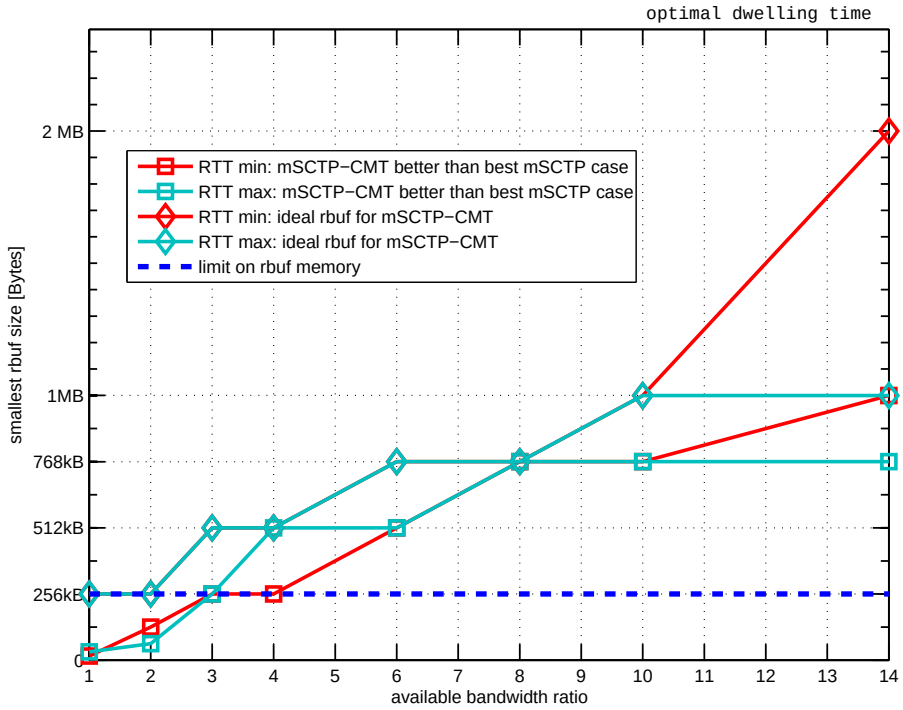
Figure 5.10: Rbuf size constraints for different RTT values: (a) short (10s) dwelling time.

gain can be noticed, however the performance is strongly constrained by the rbuf blocking problem. Varying RTT did not change this tendency: mSCTP-CMT was not capable of outperforming best mSCTP case for a rbuf not exceeding 256 kB for bw_{ratio} equal to 4. For $bw_{ratio} = 2$ already 128 kB buffer guaranteed much better performance than mSCTP for all investigated values of RTT. For the two biggest analyzed values of bw_{ratio} (i.e, 6 and 8), rbuf blocking problem forced having nearly ideal rbuf size, significantly exceeding 256 kB in order to provide improvement over the best mSCTP scheme.

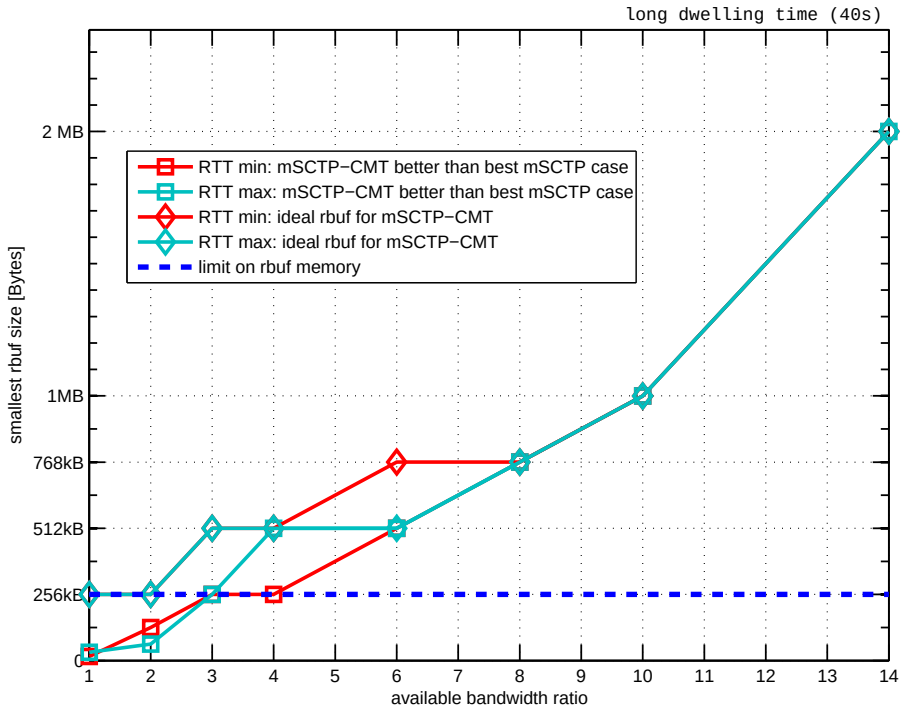
To quantify the constraints introduced by rbuf blocking problem for a given limitation on the rbuf memory size set to 256 kB, the same metrics as in case of basic analysis are used: the smallest rbuf size that guarantees mSCTP-CMT outperforms mSCTP in terms of overall file transfer, for worst and best policy accordingly, as well as the smallest rbuf size without the rbuf blocking problem at all. Fig. 5.10 shows clearly that, independently of the RTT value, always that the $bw_{ratio} \leq 2$, the mSCTP-CMT algorithm can bring a given gain over the best mSCTP strategy, if the design concern is not having rbuf blocking at all. If the design approach allows some rbuf blocking aiming only at improving the result of mSCTP, the scope of use of mSCTP-CMT extends to bw_{ratio} of three or even four in function of considered RTT. Beyond this limit, the difference between both links makes the application of mSCTP-CMT pointless. This is hold true for all considered t_{dwell} values.

Mobility pattern

Last aspect that will be evaluated in the presented analysis of mSCTP-CMT is the influence of the transition type. Similarly to Fig. 5.4, a performance comparison in terms of TSN-time evolution diagram for slow-to-fast RAN transition is presented in Fig. 5.11. As can be seen the transition type is a factor not influencing much the presented scenario. In both cases (fast-to-slow and slow-to-fast transition) the gain will be produced in the overlap area (2-42 s in the time scale) if the rbuf size is appropriately adjusted. Also bw_{ratio} and t_{dwell} settings are considerably more important than the

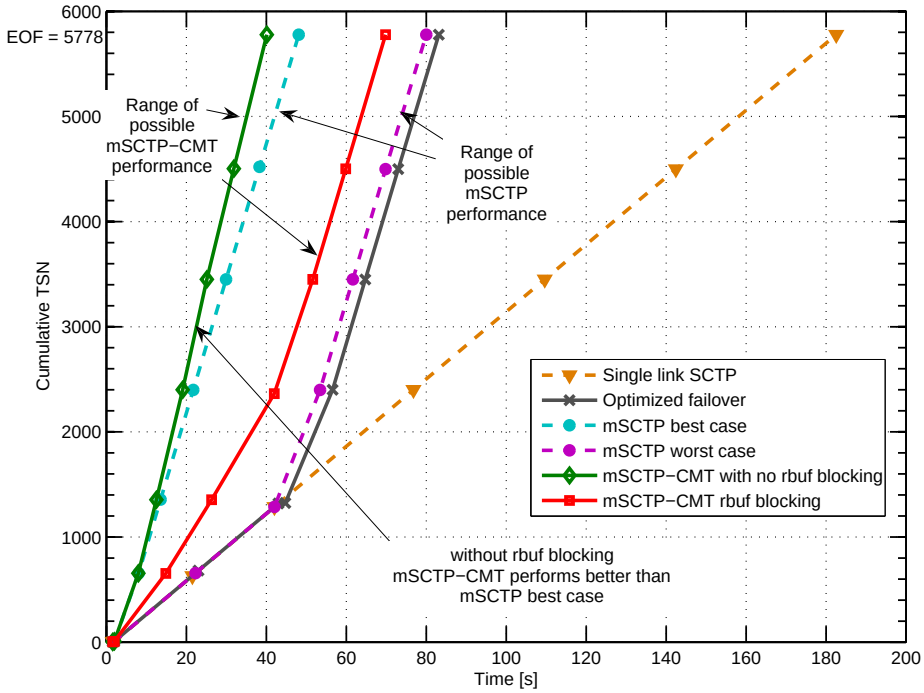


(b)

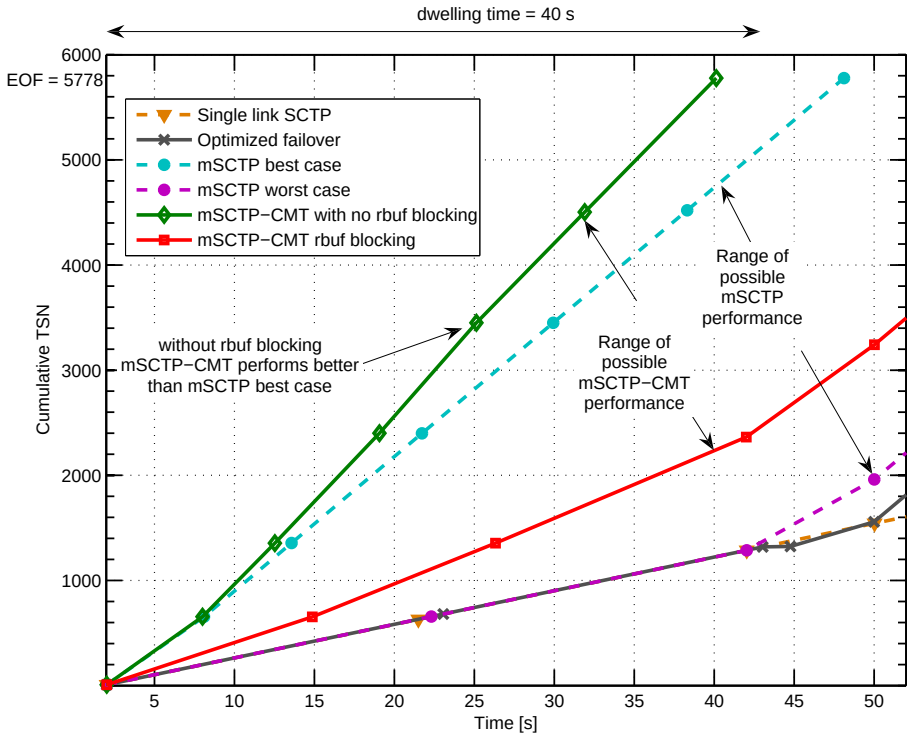


(c)

Figure 5.10: Rbuf size constraints for different RTT values: (b) optimal (longest possible, see Table 5.1); and, (c) long (40s) dwelling time.



(a)



(b)

Figure 5.11: Performance comparison of all SCTP versions for $bw_{ratio} = 4$, and $t_{dwell} = 40\,s$: (a) entire file transfer; and, (b) overlap area only.

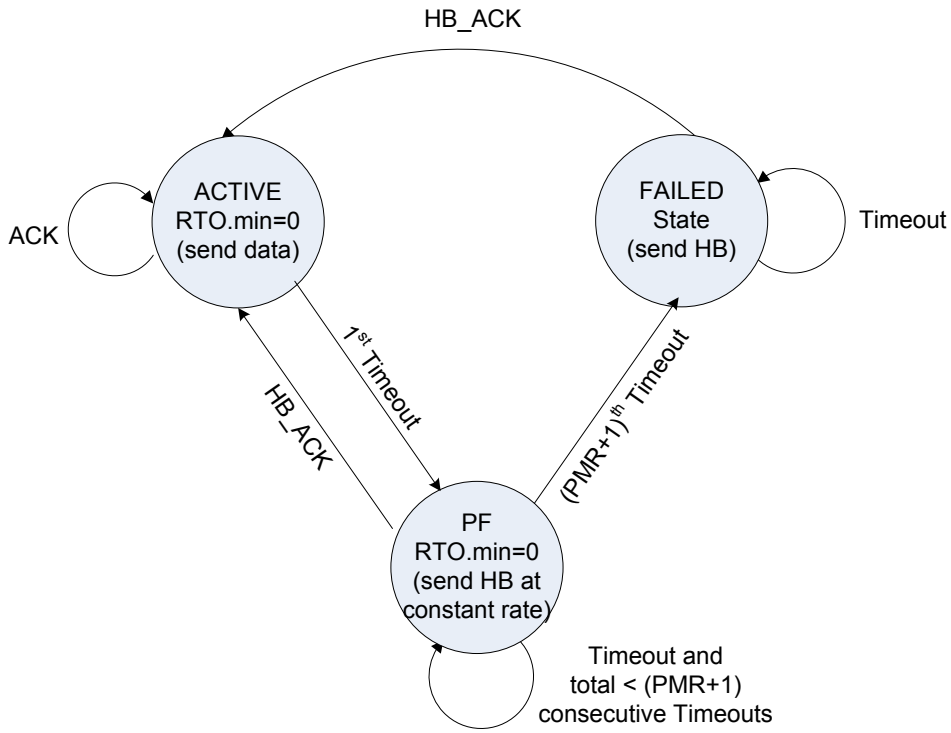


Figure 5.12: Failure detection in CMT-PF with frequent line probing.

analyzed transition type.

5.3 Future work

Readers interested in further development of the CMT application in the handover scenarios will find in this section some ideas that may be helpful for their future work.

5.3.1 ABC in slow start

In order to increase the gain introduced when using the CMT, it may be useful to improve the slow start phase on the new path (to get it fully running as fast as possible).

The first idea that comes into mind is to remove the delayed SACK mechanism (within the standard SCTP, a SACK is sent every second packet), already described here when discussing similar issue in failover context in Section 4.3.1, in order to increase the cwnd growth pace. The cost of such decision is a more intense SACK traffic on the link. In wireless scenarios the wasted mobile terminal energy spent on unnecessary SACK transmissions may have serious consequences, such as reduction of battery lifetime. Another approach to this problem, and possibly a better solution in the context of wireless scenarios is to modify slightly protocol behavior and preserve the delayed SACK mechanism. SCTP's Free-BSD implementation [SCTP-FreeBSD] includes a parameter called the $sctp_{\text{abc}}$ (ABC stands for Appropriate Byte Counting) that manages the growth of the congestion window in the slow start phase. Standard SCTP, on the reception of the SACK that advances CumTSN, increases the window by the smaller of two: $sctp_{\text{abc}} \times Path_{\text{MTU}}$ and the number of Bytes acknowledged, with $sctp_{\text{abc}}$ set by default to 1. Setting the $sctp_{\text{abc}}$ to 2 will lead to the same effect, as no delayed SACKs without the increase of SACK traffic.

5.3.2 More frequent link probing schemes

Nevertheless the CMT-PF reduces rbuf blocking problem, in case of mobility scenarios the most crucial issue is to get the information about the state of the path as frequent as possible due to the changes experienced at the radio link. Therefore, the idea, the author propose to discuss (Fig. 5.12) is to sent the HB on a constant rate of the RTO from the moment of failure, instead of on an exponentially-increasing-RTO basis, once the path is marked as the PF. This will guarantee frequent, `PMR`-independent path probing, and once the `PMR` limit is hit the path is considered inactive, and the standard SCTP's HB sending pace will be applied.

5.4 Conclusions

Provided that mSCTP itself lacks handover policies, CMT can be seen as an added value to such a transport-layer handover scheme. CMT (or strictly speaking CMT-PF) introduced by the author to the mSCTP-based handover scenarios in a scheme called mSCTP-CMT, has proved to have the potential of smoothing the handover process between two paths, as well as providing an additional gain due to the simultaneous data transmission over multiple paths available in the overlap area. An initial evaluation reports that a significant gain over the best possible mSCTP strategy can be achieved (in the range of up to 35% of total file transmission time). The main drawback is owed to the strong influence of the receiver buffer blocking on any scenario where the mSCTP-CMT is used, resulting in a firm limitation of possible application area in terms of receiver buffer size. The next important factor, scenario asymmetry, measured here as the available bandwidth ratio, puts also strong constraints on design of such system. Still, as initial results demonstrate, all mentioned limitations allow the mSCTP-CMT to fit into the future heterogeneous scenarios.

Chapter 6

Extended mobility analysis and performance comparison to network layer schemes

So far, the main focus of this work has been put on demonstrating how to design and improve a transport layer handover scheme based on SCTP. Consequently, the corresponding evaluation was performed using relatively simple scenario models in order to clearly assess the main benefits and limitations of the proposed schemes. Over such a basis, this chapter provides an extended analysis addressing the performance of all described SCTP variants in additional network scenarios. Furthermore, the proposed transport-layer schemes are now compared to other existing solutions, and in particular to the network-layer-based schemes that, as already stated in Section 2.2.2, are nowadays the most common handover schemes for heterogeneous wireless networks. In this context, two main-stream network layer solutions, namely: MIPv4-RO [Perkins, 2002; Perkins and Johnson, 1998] and MIPv6 [Johnson et al., 2004], are introduced in the analysis to have a fair comparison between transport and network layer approaches.

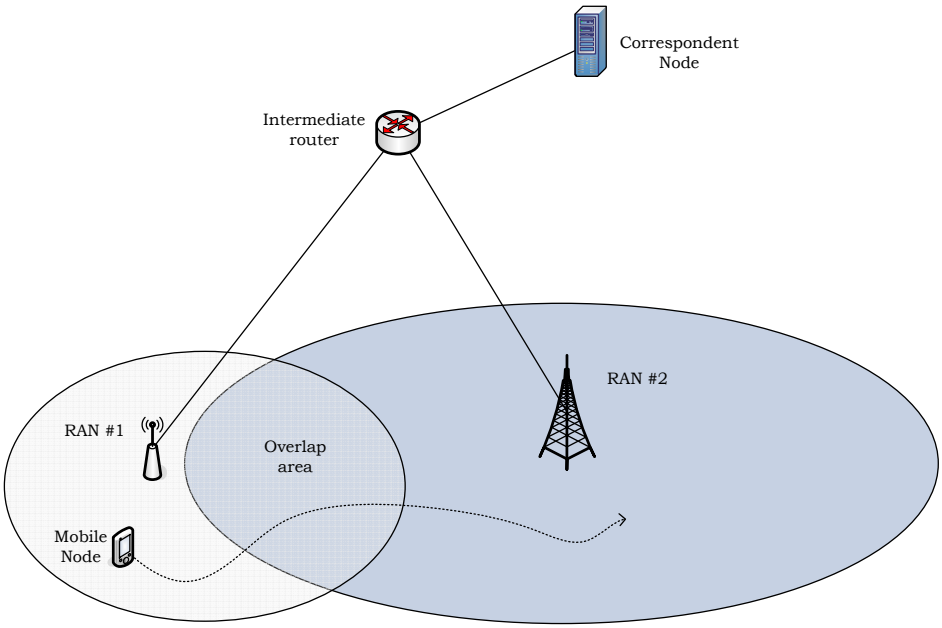


Figure 6.1: Scenario under test.

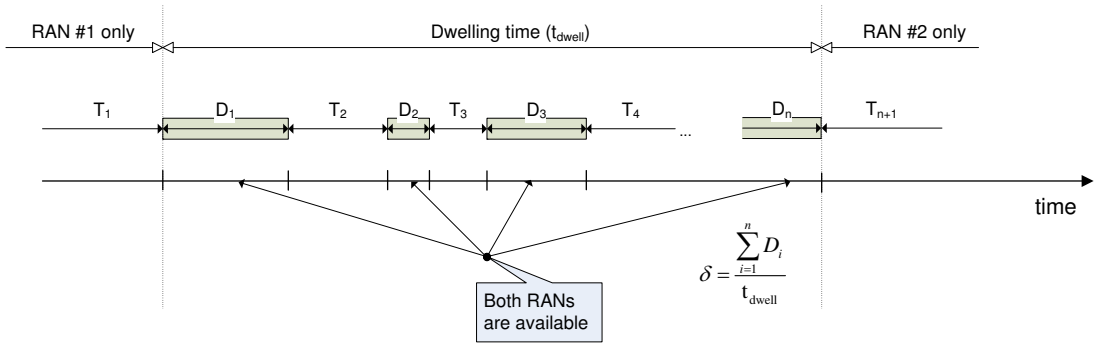


Figure 6.2: Mobility model for the presented analysis.

Table 6.1: Mobility patterns

Value of n	Pattern	Description
1	N/A	simple scenario: both RANs are available during entire dwelling time *(Section 6.2.1)*
3	1-1 (2-2)	advanced scenario #1: stable availability of one of the RANs in the overlap area *(Section 6.2.2)*
3	1-2	advanced scenario #2: changing availability of RANs inside the overlap area *(Section 6.2.3)*

6.1 Preeliminaries

6.1.1 Scenario description

The scenario under test (Fig. 6.1) proposed for the series of analysis presented in this chapter reuses the topology presented already in Chapter 5. Analogously, a MN is moving from RAN #1 coverage area towards the coverage area of a neighboring RAN #2 traversing across the overlap area. However this time, the model of the overlap area has been improved to better reflect the characteristics of a handover scenario in future wireless heterogeneous networks [Zahran et al., 2008]. Fig. 6.2 illustrates the overlap area model that will be applied. The main parameter is n, number of periods when both RANs are simultaneously available. Two different values of n (1 and 3) are proposed to capture different mobility patterns, and Table 6.1 further describes the details of each pattern analyzed. t_{dwell} counts for the entire period when MN can be connected to either one or both RANs. Periods of availability of each RAN inside the overlap area (pattern's naming convention reflects the availability of each RAN, e.g., 1-1 means that RAN #1 is the only RAN available in both periods between three overlap ares) are adjusted in function of the MN's speed, and are exponentially distributed random variables. Parameter δ reflects the ratio between the time when both RANs are simultaneously available and the dwelling time. Summary of all scenario parameters will be given in Table 6.8, but first all handover schemes are defined and explained.

6.1.2 Analyzed handover schemes

The following five mobility schemes will be taken into account in the presented analysis: (1) failover-based, (2) mSCTP, (3) mSCTP-CMT-PF (additionally mSCTP-CMT without the PF extension [Natarajan et al., 2009, 2006] is tested in the simplest scenario in Section 6.2.1), (4) MobileIPv4-RO, and (5) MobileIPv6. Next, each of the presented schemes is described in more details, with parameters

specific for each scheme given in a separate table, correspondingly.

Scheme #1: optimized failover-based mSCTP handover scheme. A handover scheme with the standard SCTP failover mechanism adapted for the handover scenarios (necessary parameter tune-up) and used to trigger the handover. This scheme has been introduced and described in Chapter 4, here the most important details are given in Table 6.2.

Table 6.2: Failover-based mSCTP handover scheme details

Parameter name	Value / Range
PMR	1
RTO.min	0 ms
Handover strategy	no handover policy defined primary path change is triggered with failover,
Transport layer signaling	Add IP policy: sent if the new interface has been up for τ_{AddIP}, and its address does not belong to the association Set Primary Address policy: not sent Delete IP Address policy: sent when the interface has been down for $\tau_{DeleteIP}$ (most typically sent bundled with data chunk)
Network layer signaling	N/A

Scheme #2: mSCTP handover scheme, with strategy to choose always the faster RAN. As mentioned in Section 3.3.2, handover schemes based on mSCTP lack an appropriate handover policy. Provided that possible handover policies can be based on the information passed to the transport layer from lower layers (e.g., signal strength information), the policy proposed for this scheme has the strategy that chooses always the fastest RAN (in terms of available bandwidth) out of all the access networks that are currently available. Detailed description of this scheme is given in Table 6.3.

Table 6.3: mSCTP handover scheme details

Parameter name	Value / Range
PMR	5
RTO.min	200 ms
Handover strategy	use always faster RAN
Transport layer signaling	Add IP policy: sent if the new interface has been up for τ_{AddIP}, and its address does not belong to the association Set Primary Address policy: sent after *hysteresis time* $\tau_{ChangeIP}$, from the moment a faster interface is available Delete IP Address policy: sent when the interface has been down for $\tau_{DeleteIP}$ (most typically sent bundled with data chunk)
Network layer signaling	N/A

Scheme #3: mSCTP-CMT-PF handover scheme. A novel design, introduced and described in Chapter 5, where the CMT-PF loadsharing scheme is used as an enhancement of the handover management based on mSCTP. Table 6.4 presents more details of this scheme.

Table 6.4: mSCTP-CMT-PF handover scheme details

Parameter name	Value / Range
PMR	5
RTO.min	200 ms
Handover strategy	as long as two links are available use the CMT-PF
Transport layer signaling	Add IP policy: sent if the new interface has been up for τ_{AddIP}, and its address does not belong to the association Set Primary Address policy: not sent Delete IP Address policy: sent when the interface has been down for $\tau_{DeleteIP}$ (most typically sent bundled with data chunk)
Network layer signaling	N/A

Scheme #3a: mSCTP-CMT handover scheme. A subversion of scheme #3 provided here as a benchmark to demonstrate that the PF extension of the CMT proposed by Natarajan et al. [2006] fits better handover scenarios than the "pure" CMT scheme. Therefore mSCTP-CMT scheme, described in Table 6.5 is only used in the simplest possible scenario (Section 6.2.1).

Table 6.5: mSCTP-CMT handover scheme details

Parameter name	Value / Range
PMR	5
RTO.min	200 ms
Handover strategy	As long as two links are available use the CMT
Transport layer signaling	Add IP policy: sent if the new interface has been up for τ_{AddIP}, and its address does not belong to the association Set Primary Address policy: not sent Delete IP Address policy: sent when the interface has been down for $\tau_{DeleteIP}$ (most typically sent bundled with data chunk)
Network layer signaling	N/A

Scheme #4: Network-layer scheme: MIPv4-RO. First of the two analyzed network-layer-based schemes MIPv4-RO is characterized in Table 6.6. As already described in section 2.2.2, handling mobility at the network layer transparently for the adjacent SCTP and with minimum routing overhead, requires sending binding update (BU) messages to both HA and CN, informing about the current IP address of the MN. Consequently, each change of the IP address results in a handover delay $\tau_{mipv4-ro}$, as stated in [Saha et al., 2004]. Value for $\tau_{mipv4-ro}$ shown in Table 6.8 is calculated using this formula and considered scenario parameters.

The handover strategy proposed for this scheme selects always the fastest RAN in terms of available bandwidth) out of all the access networks that are currently available.

Table 6.6: MIPv4RO network-layer handover scheme details

Parameter name	Value / Range
PMR	5
RTO.min	200 ms
Handover strategy	use always faster RAN
Transport layer signaling	Add IP/Set Primary/Delete IP Address policy: N/A
Network layer signaling	The MN sends a registration request to the FA that forwards the BU message to the HA. Then the HA informs the CN about the change. Such a procedure is performed: (1) once a faster interface is up when the MN has a slow interface and (2) when a faster interface is down and transfer can continue over existing slow interface Total signaling delay: $\tau_{mipv4-ro} = t_{MN-HA} + t_{HA-CN} + t_{CN-MN}$

Scheme #5: Network-layer scheme: MIPv6. Second of the analyzed network-layer-based schemes MIPv6, reduces significantly the signaling necessary to indicate to the CN the change of the IP address (τ_{mipv6} is much lower than the corresponding $\tau_{mipv4-ro}$) [Saha et al., 2004]. The handover strategy is similar to the previously described schemes and selects always the fastest RAN in terms of available bandwidth out of all the access networks that are currently available. Parameters describing this scheme are shown in Table 6.7, whereas the corresponding value of τ_{mipv6} is calculated in Table 6.8.

Table 6.7: MIPv6 network-layer handover scheme details

Parameter name	Value / Range
PMR	5
RTO.min	200 ms
Handover strategy	use always faster RAN
Transport layer signaling	Add IP/Set Primary/Delete IP Address policy: N/A
Network layer signaling	BU message is sent from the MN to the CN: (1) once a faster interface is up when the MN has a slow interface and (2) when a faster interface is down and transfer can continue over existing slow interface Total signaling delay: $\tau_{mipv6} = 2 \cdot t_{CN-MN}$

6.1.3 Simulation parameters

Table 6.8 summarizes the list of the most important simulation parameters, including parameters specific to each of the five handover schemes, calculated for the given scenario topology. The details of the radio channel model are described in Appendix A.

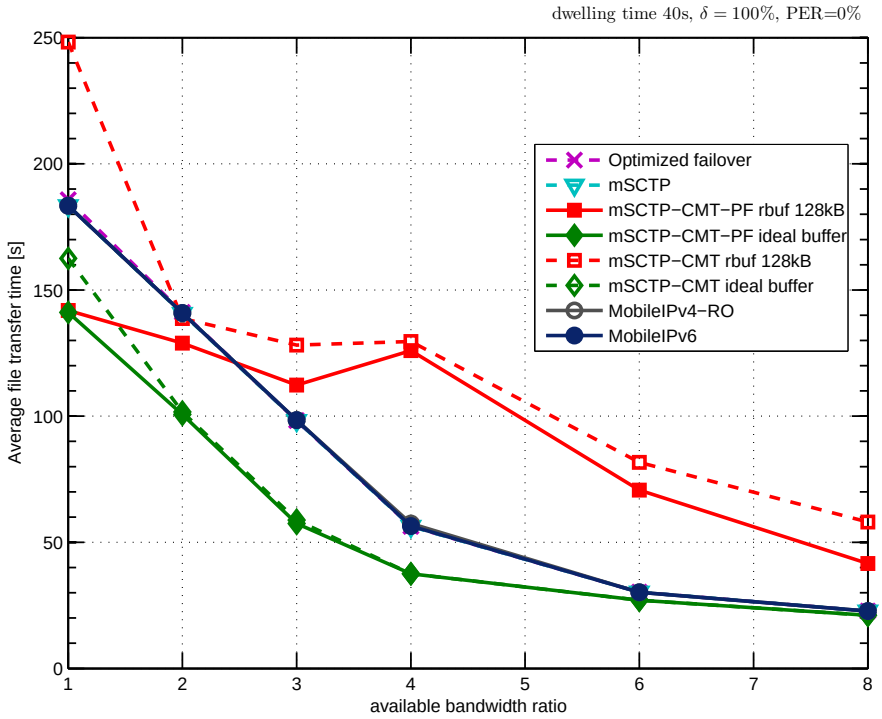
Table 6.8: Simulation parameters for comparison with the network layer schemes

Parameter name	Value / Range
Wired part (each path)	bandwidth: 100 Mbps one-way propagation delay: 5 ms
fast RAN	bandwidth: (bw_ratio × 384) kbps one-way propagation delay: 15 ms
slow RAN	bandwidth: 384 kbps one-way propagation delay: 80 ms
scenario pattern	RAN #1: fast RAN RAN #2: slow RAN
Average PER	0-1-2-5%
mobility patterns	simple scenario: N/A advanced scenario #1: 2-2 advanced scenario #2: 1-2 *(as shown in Table 6.1)*
δ	simple scenario: 100% advanced scenarios: 25-50-75%
t_{dwell}	20-40-60-80 s
bw_{ratio}	1-8
$\tau_{mipv4-ro}$	500 ms
τ_{mipv6}	200 ms
τ_{AddIP}	100 ms
$\tau_{ChangeIP}$	100 ms
$\tau_{DeleteIP}$	60 s
rbuf size	128-256-512 kB (ideal buffer: up to 2 MB)
`RTOMin`	50 ms
`PMR`	Optimized failover, mSCTP: 1 CMT: 5
`SACK delay`	200 ms
Retransmission policies	mSCTP FastRtx: Same path mSCTP TimeoutRtx: Alternate path CMTRtx: path with largest cwnd
MTU size / Data payload	1500 / 1468 Bytes
File size	8 MB
Random seed	15 sets of exponentially distributed random variables per simulation

6.2 Analysis results

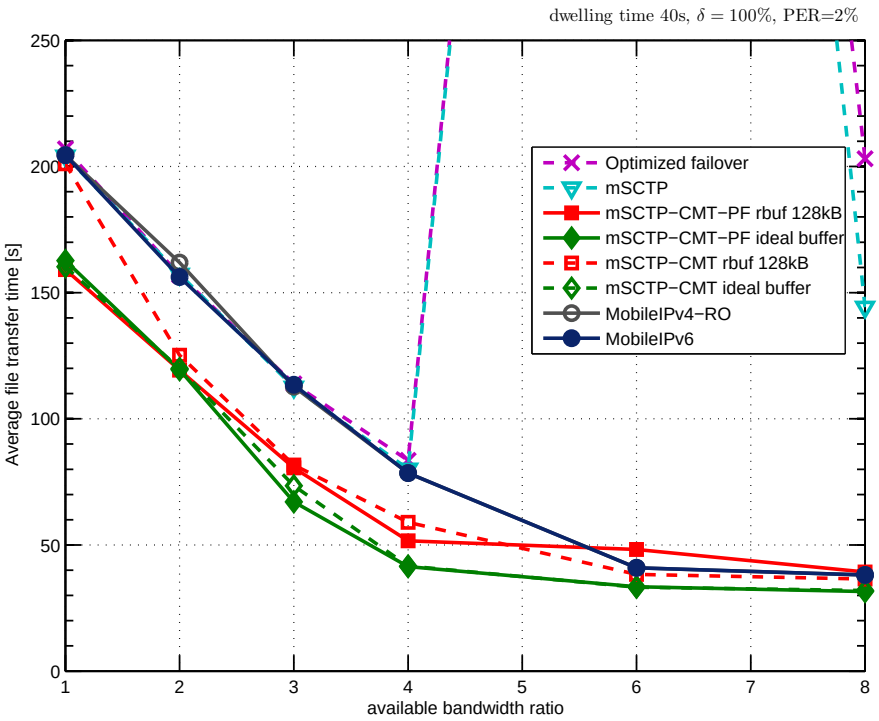
6.2.1 Simple scenario (n = 1)

First, a comparison of all described handover schemes is made in a simple scenario, where both RANs are simultaneously available for the entire dwelling time. Fig. 6.3 presents the results for the average file transfer time as a function of available bw_{ratio} and packet loss rate for a given t_{dwell}. $t_{dwell} = 40\,s$ has been selected as the most interesting case, given that for $t_{dwell} = 20\,s$ the time both links were simultaneously available was too short to effectively use the loadsharing and thus have a fair comparison to other schemes, whereas for both $t_{dwell} = 60\,s$ and $80\,s$ the advantage from the use of loadsharing-based scheme was too evident. Consequently, the application of the mSCTP-CMT-PF scheme is highly recommended for the scenarios, where the MN is likely to stay for a relatively long time in the overlap area. Now, the influence of the remaining factors will be analyzed for $t_{dwell} = 40\,s$. In a scenario without losses, shown in Fig. 6.3a, both CMT-based schemes with the ideal buffer size (the size of the buffer big enough to avoid rbuf blocking) provide the best performance over the entire scope of analyzed bw_{ratio}. Both mSCTP-CMT-PF and mSCTP-CMT perform equally good in all asymmetrical scenarios (in terms of bandwidth), whereas the only difference appears when paths are symmetrical. The PF extension stops sending data packets to a path where the loss occurred, sending instead a HB packet to probe the path availability. On the positive response the path gets back to the active state, and the data transmission is resumed. Meanwhile the lost packet (in this case, due to the overflowing of the buffer queue size) is retransmitted on the only path currently available. This algorithm turns out to be more efficient than keeping the failed path active as in case of CMT without PF extension. Much more difference between both CMT schemes can be seen for smaller buffer sizes. Results presented here for the buffer size of 128 kB, show clearly that in presence of rbuf blocking, the PF extension increases significantly protocol's performance.

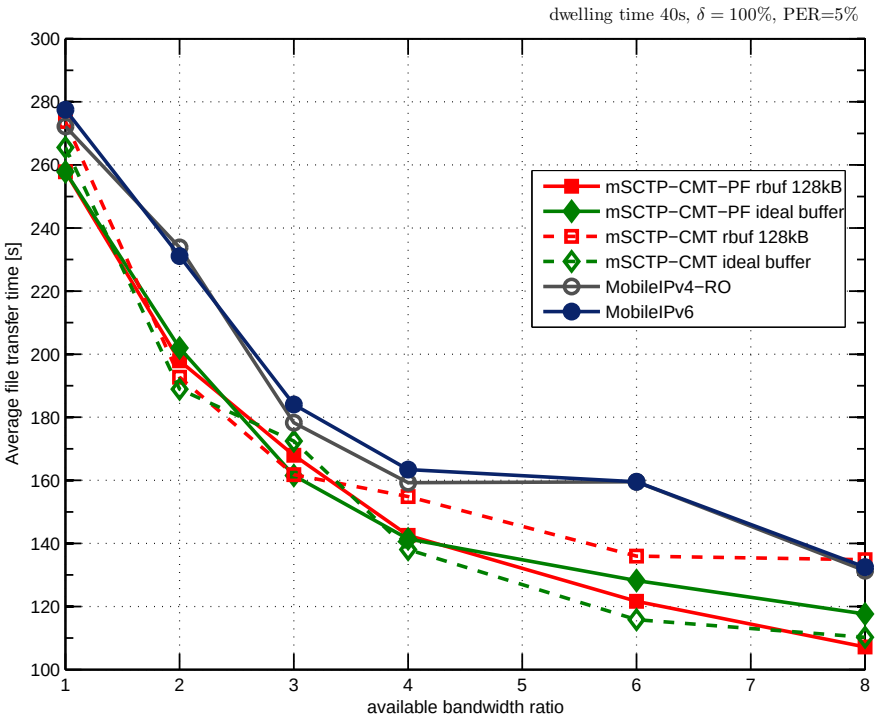


(a)

Figure 6.3: Comparison of various handover schemes in a simple scenario, for the following PER values: (a) no losses.



(b)



(c)

Figure 6.3: Comparison of various handover schemes in a simple scenario for the following PER values: (b) 2% losses; and, (c) 5% losses.

Still, even the mSCTP-CMT-PF with only 128 kB of buffer size was not able to outperform the remaining non-loadsharing schemes for $bw_{ratio} > 2$. Both network-layer schemes, as well as mSCTP and failover-based scheme, performed almost in the same way in the given scenario, meaning that handling the mobility at the network layer, transparently to the transport layer, did not do any harm to the SCTP cwnd evolution. Thus, both network-layer handover delays $\tau_{mipv4-ro}$ and τ_{mipv6} were small enough to provoke just a single packet loss, recoverable by the fast retransmission, and not leading to any timeout.

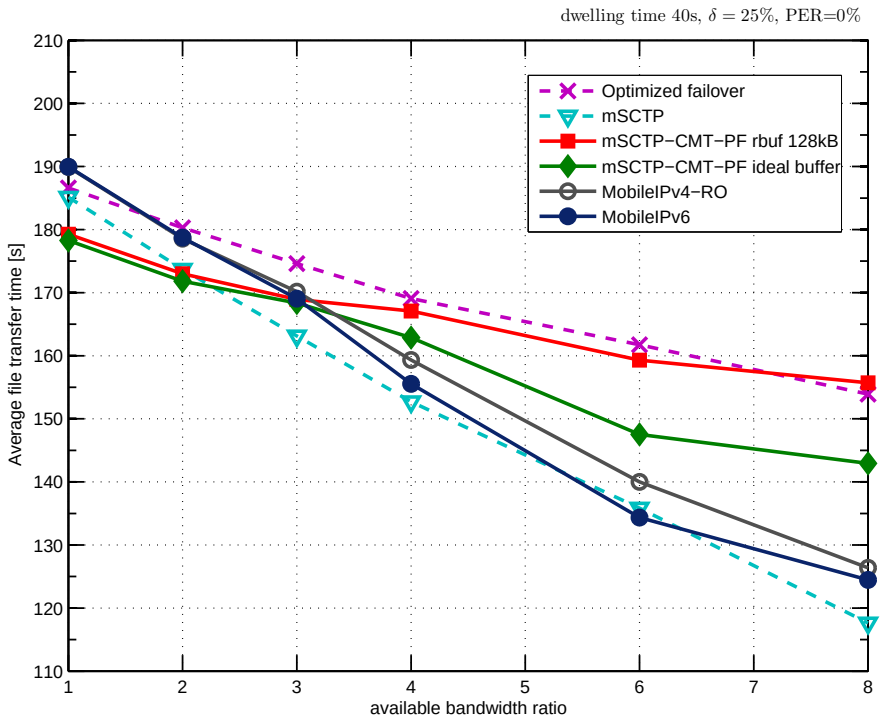
For a scenario where packet losses are present, both failover-based and mSCTP scheme perform unstable, specially when the scenario asymmetry increases, as shown in Fig. 6.3b. Also there are much more differences between the other schemes. The range of the bw_{ratio} where the CMT-based schemes with a 128 kB receiver buffer perform better than the network-layer schemes extends to $bw_{ratio} < 6$. Naturally, the mSCTP-CMT-PF scheme with an ideal buffer size works best of all presented schemes in the entire analyzed scope.

In case the packet loss rate increases to 5% (Fig. 6.3c), for both failover-based and mSCTP-based scheme there was no single situation where the transmission could be completed in less than 900 s time. This fact, plus the high instability for already 2% of the loss rate, leads to the conclusion that any of these schemes should not be recommended as the possible solution, in scenarios where losses are present. As for the remaining schemes, both network-layer schemes perform noticeably worse than the mSCTP-CMT-PF, even with the rbuf size of 128 kB in the entire scope of experiment. The mSCTP-CMT-PF scheme is not only the best one in terms of minimum transfer delay, but also presents the most stable performance among all presented solutions. Not having the PF extension available yields slightly worse performance, but still permits prevail the network-layer counterparts for almost the entire range of the analyzed bandwidth asymmetry ($bw_{ratio} < 8$, corresponding to a scenario with the MN moving from the WLAN to the UMTS network), also with the small size of the receiver buffer.

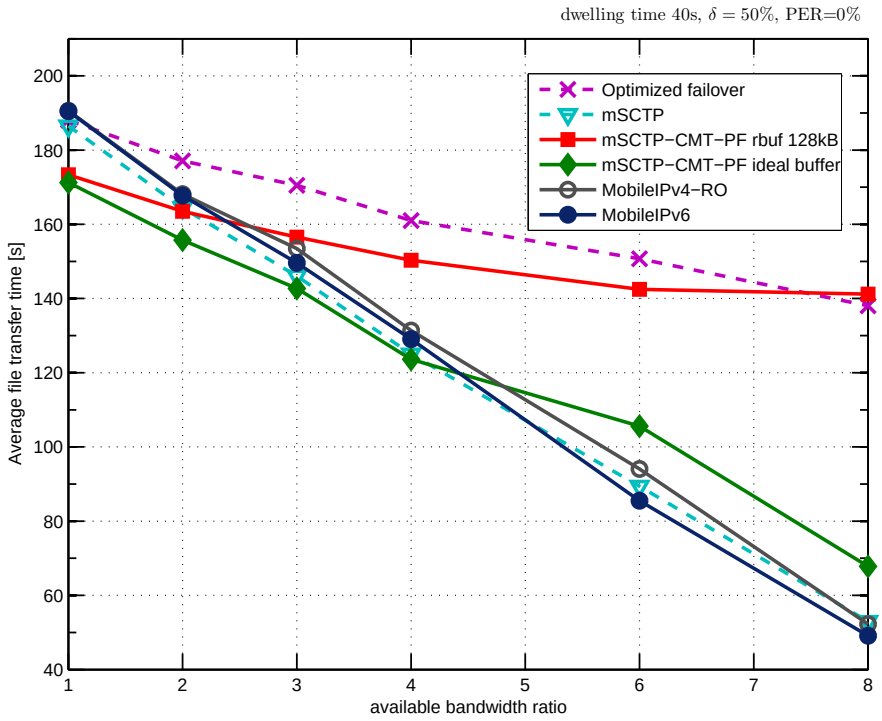
6.2.2 More complex mobility pattern (n = 3) - part 1

Having analyzed the simplest scenario, with both links simultaneously available all the time the MN stays in the overlap area, it is now time to check the influence of the mobility patterns on the performance of proposed handover schemes. First, a scenario, with stable availability of the RAN #2 in the overlap area and periodical availability of RAN #1 (faster of the two), denoted here as 2-2, is analyzed. Fig. 6.4 presents the results for average file transfer time as a function of bw_{ratio} in a scenario without losses for three different values of the parameter δ. Parameter δ reflects the fraction of the t_{dwell} that both RANs are simultaneously available in the overlap area, incurring worse performance of the mSCTP-CMT-PF scheme for the lower values of δ. Consequently, the following trend can be noticed: for $\delta = 25\%$ (Fig. 6.4a) the mSCTP-CMT-PF scheme gets worse than any scheme but failover-based, for the $bw_{ratio} > 3$ no matter what the receiver buffer size is. For $\delta = 50\%$ (Fig. 6.4b) it is already $bw_{ratio} > 4$, whereas for $\delta = 75\%$ (Fig. 6.4c) the mSCTP-CMT-PF is the best scheme in the entire analyzed scope of bw_{ratio}. For the purpose of comparison it is also worth to recall here the results from Section 6.2.1 ($\delta = 100\%$) that gave the mSCTP-CMT-PF scheme a clear advantage for all analyzed bw_{ratio} values. Thus, the main conclusion is that an instable coverage of both RANs in the overlap area, clearly reduces the scope of use of the mSCTP-CMT-PF, when compared to other mobility schemes. MobileIPv6 seems to deal best with such situations as the solution having the lowest latency out of all the remaining schemes. In contrast, MobileIPv4-RO due to the big value of the $\tau_{mipv4-ro}$, performs even worst than the mSCTP scheme, but still guarantees relatively stable performance, no matter what δ value was.

For 2% packet loss rate (Fig. 6.5) the mSCTP-CMT-PF performs slightly worse than in the same scenario without losses, being better only if both paths have equal bandwidth in case of the lowest δ analyzed (Fig. 6.5a), $bw_{ratio} < 4$ for $\delta = 50\%$ (Fig. 6.5b), and preserving the best score among all schemes in the entire scenario for the biggest value of δ (Fig. 6.5c). Therefore, an overall conclusion for the mSCTP-CMT-PF application in the analyzed handover scenario is that for a comparable bandwidth ratio of both paths, i.e., bw_{ratio} not exceeding 2, the use of this scheme improves the performance of the mSCTP even for a very unstable availability of both paths in the overlap region. Outside of the stated application area, it can be clearly seen that the introduction of the loadshar-

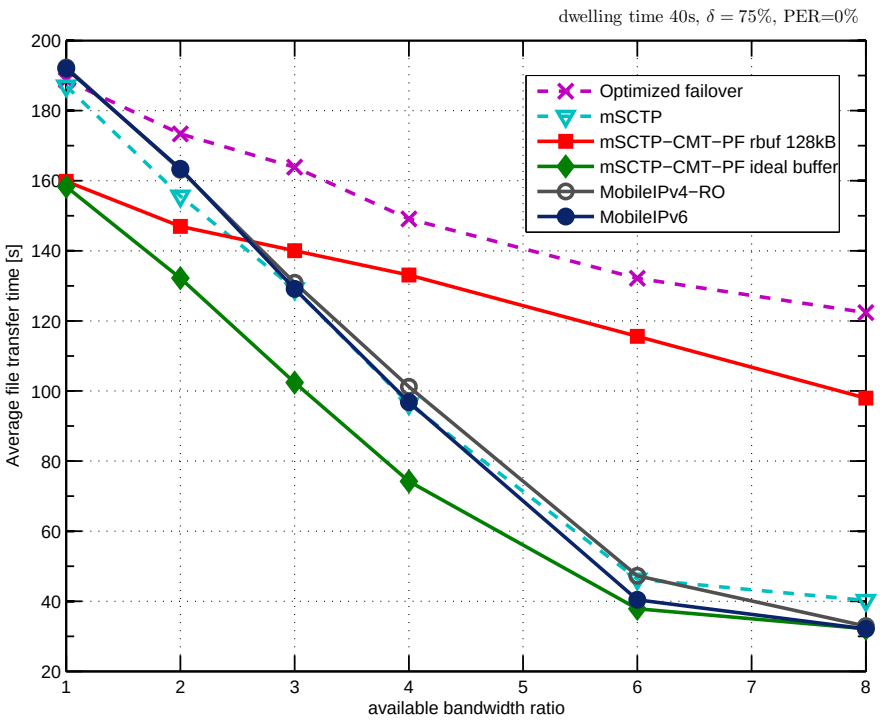


(a)



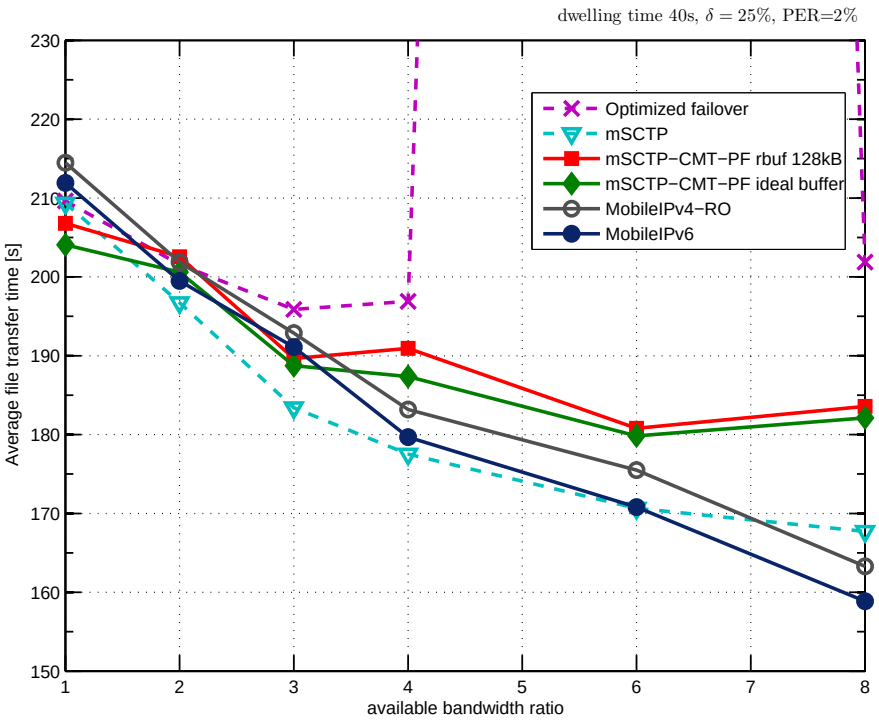
(b)

Figure 6.4: Comparison of various handover schemes for different values of δ in a scenario with pattern 2-2 and without losses, for: (a) $\delta = 25\%$; and, (b) $\delta = 50\%$.

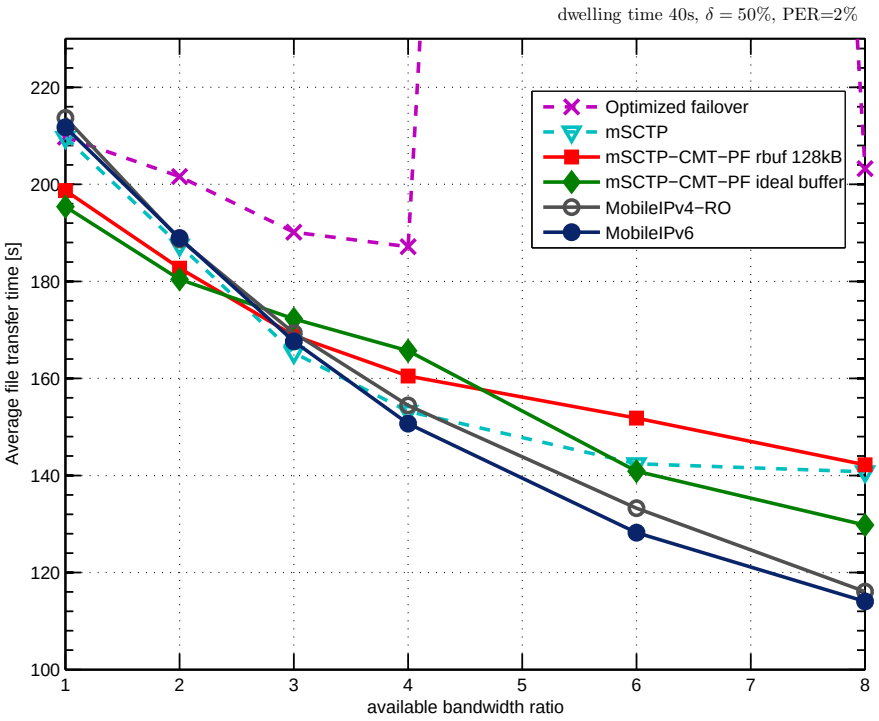


(c)

Figure 6.4: Comparison of various handover schemes for different values of δ in a scenario with pattern 2-2 and without losses, for: (c) $\delta = 75\%$.

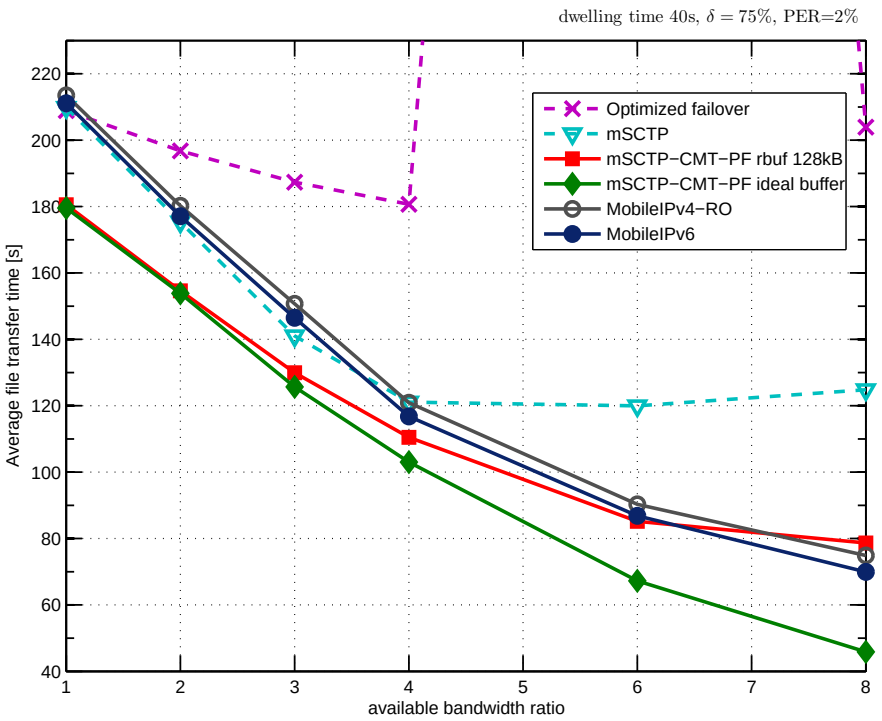


(a)



(b)

Figure 6.5: Comparison of various handover schemes for different values of δ in a scenario with pattern 2-2, and 2% of losses for: (a) $\delta = 25\%$; and, (b) $\delta = 50\%$.



(c)

Figure 6.5: Comparison of various handover schemes for different values of δ in a scenario with pattern 2-2 and 2% of losses, for: (c) $\delta = 75\%$.

ing scheme to the handover transition process deteriorates the performance of the mSCTP. Having already numerous retransmissions that must be handled, not only because of the temporal unavailability of one of the paths, but also because of the losses present in available radio channels, any additional retransmission due to the loadsharing decrease the throughput significantly. Nevertheless, the mSCTP scheme performs very well only for low value of δ, improving much less than the other schemes when the δ increases, and becoming the worst of all stable schemes for the biggest value of δ. Failover-based scheme was again not able to offer any stable performance in presence of losses, making impossible the trade-off between stable and fast performance due to reduction of PMR value. In that sense the only valid alternatives become the network layer schemes, as being able to deal with moderate losses in all three analyzed scenarios.

6.2.3 More complex mobility pattern (n = 3) - part 2

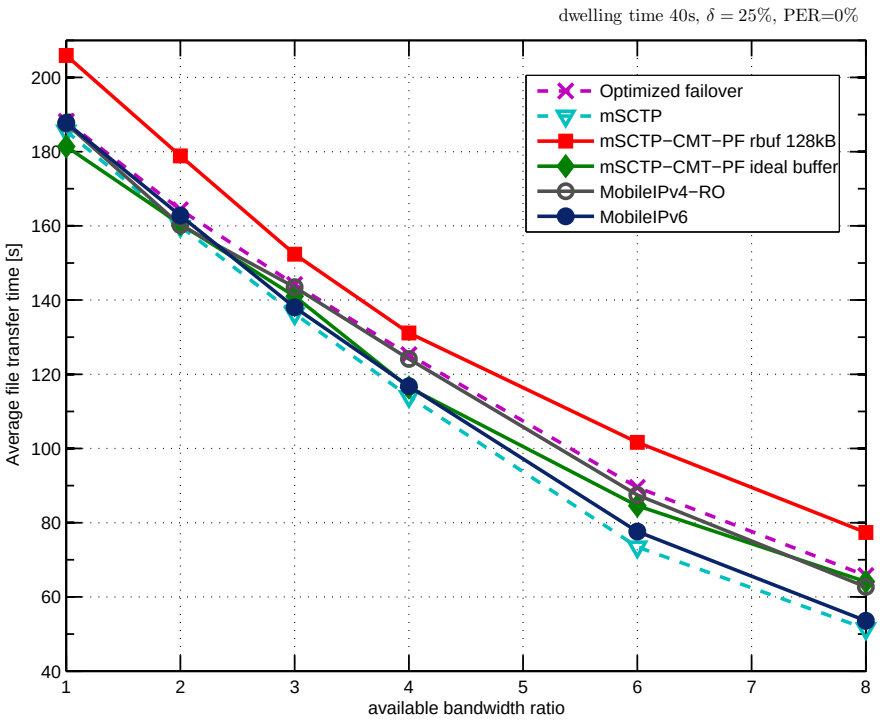
In order to fully understand the influence of handover policies on the overall performance of handover scheme it is necessary to test different mobility patterns. This is the most important difference between scenarios presented in this Section, were users using always-faster-RAN strategy will be forced to perform at most 2 handovers instead of 3 forced handovers as it was in case of scenarios analyzed in Section 6.2.2.

Fig. 6.6 represents the results for the average transfer time in a scenario without losses. The mSCTP-CMT-PF with the ideal buffer size is able to achieve the best performance whenever the $bw_{ratio} < 3$ for the lowest value of δ (Fig. 6.6a). As δ increases, this thresholds extends up to $bw_{ratio} < 6$ for $\delta = 50\%$ (Fig. 6.6b), and further to $bw_{ratio} < 8$ for the highest value of delta (Fig. 6.6c). The constraints introduced by a limited receiver buffer space would decrease the observed benefits, e.g., having a 128,/kB limitation on the receiver buffer memory leads to the worst performance from all presented schemes (worse even than failover based scheme) for the lowest value of δ and all values of bw_{ratio}. In such a case there are too many on-going retransmissions that such a small buffer can handle. Having $\delta = 50\%$ was not enough to improve this situation if $bw_{ratio} > 3$, and even for the highest value of δ, the failover over-performed the mSCTP-CMT-PF scheme for $bw_{ratio} > 4$. Observed behavior is significantly different than the results for the 2-2 scenario presented in Section 6.2.2, due to a decreased amount of mandatory handovers. Nevertheless, the mSCTP-CMT-PF can still be recommended to be applied in the scenarios where $bw_{ratio} < 3$, if only an appropriate receiver buffer space is available. Otherwise, the scope of use of the mSCTP-CMT-PF scheme is quite limited and highly dependent on the given scenario. Another important consideration is the noticeable improvement of both network layer schemes, caused by the reduced latency with lower number of handovers performed in the overlap area.

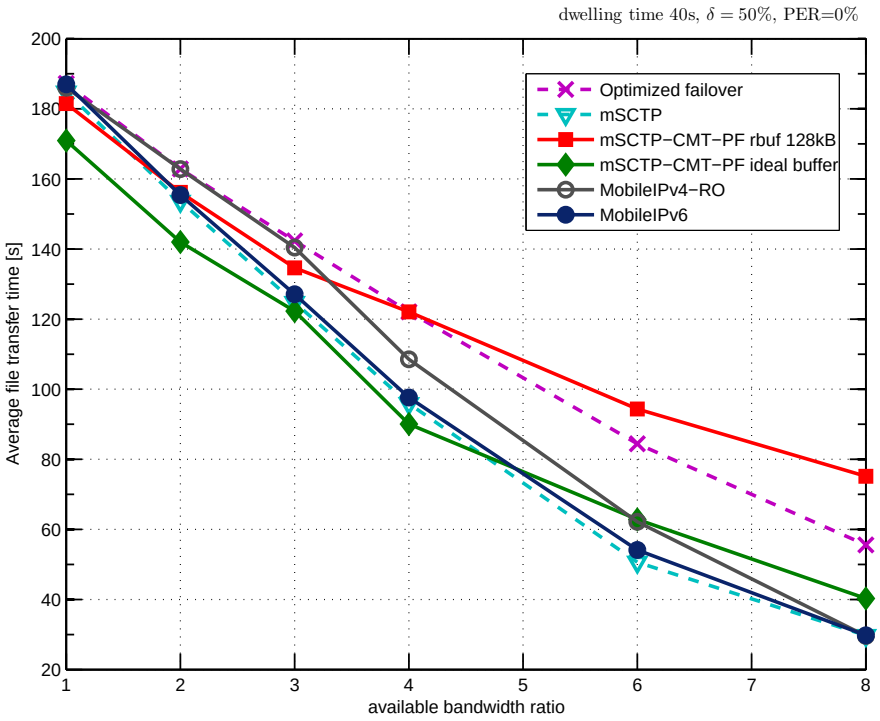
With a 2% packet loss existing in the analyzed scenario (Fig. 6.7) further limitation of the use of the mSCTP-CMT-PF scheme can be observed. For the lowest value of δ (Fig. 6.7a), the mSCTP-CMT-PF is the worst scheme out of the stable ones, no matter how big the size of receiver buffer is. For $bw_{ratio} > 6$, the mSCTP-CMT-PF is outperformed even by the failover based-scheme, again being a consequence of a limited number of handovers. Only the biggest value of δ (Fig. 6.7c) provides the mSCTP-CMT-PF with an acceptable performance, and again the best of all available schemes. Therefore both MIPv6 or MIPv4-RO can be seen as valid alternatives, especially in the scenarios similar to the presented one.

6.3 Conclusions

Comparison of all SCTP-based schemes introduced and described in this work was aiming to provide an answer for the question, whether there is a real need for such solutions, given that the network-based schemes are the most common mobility schemes nowadays. Looking at the results of the presented analysis, some initial conclusions can be drawn, but any more straightforward answer to the aforementioned question, require further evaluation. Provided that there are small asymmetries between both RANs (i.e., $bw_{ratio} < 3$) and reasonably stable availability of both in the overlap area (i.e., $\delta > 50\%$) the mSCTP-CMT-PF scheme was able to outperform all other analyzed schemes in all presented experiments, also taking into account the difficulties introduced by

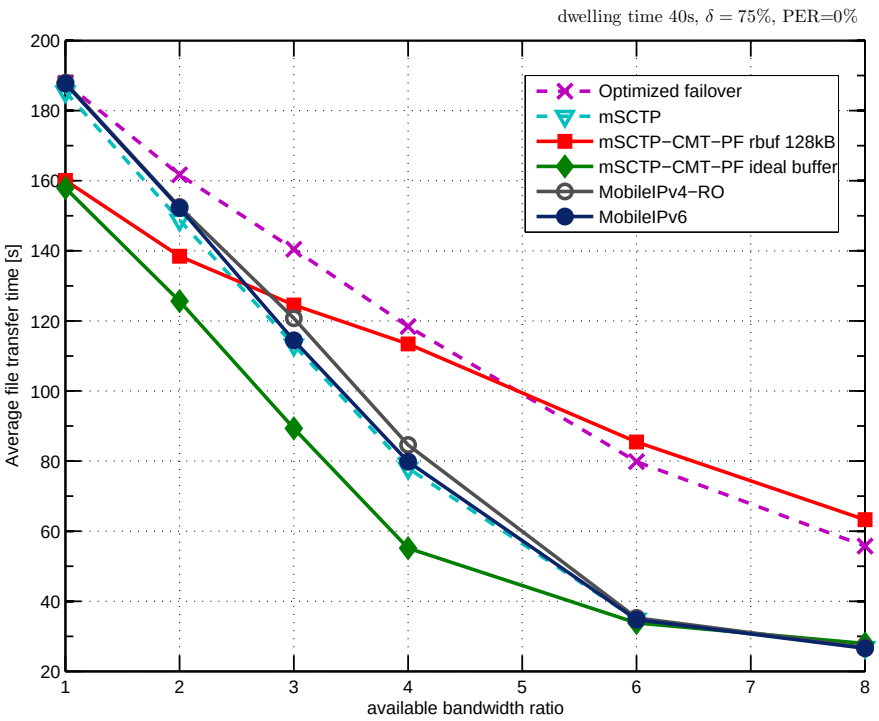


(a)



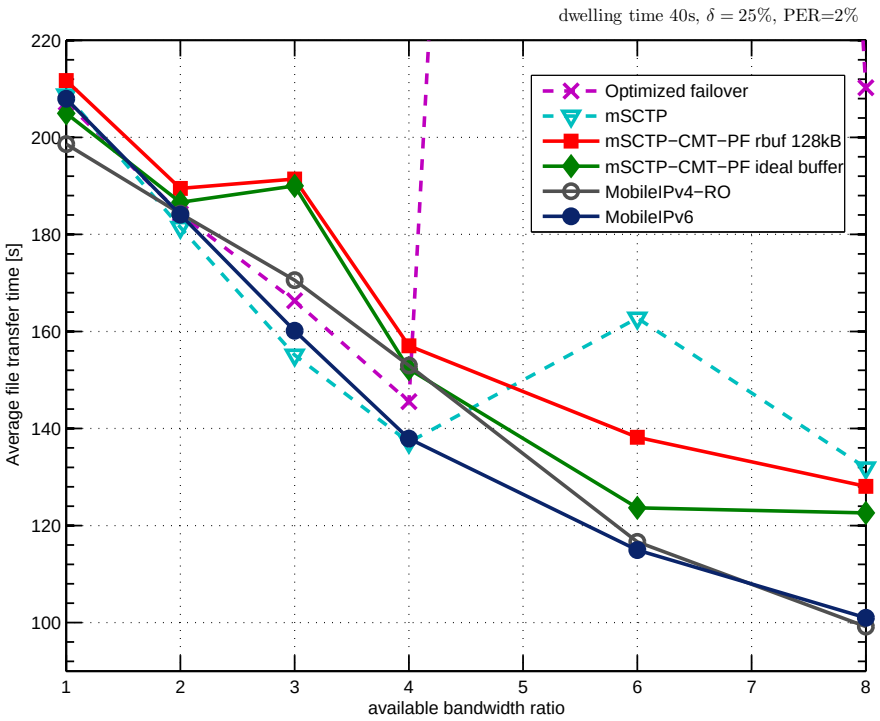
(b)

Figure 6.6: Comparison of various handover schemes for different values of δ in a scenario with pattern 1-2 and without losses, for: (a) $\delta = 25\%$; and, (b) $\delta = 50\%$.

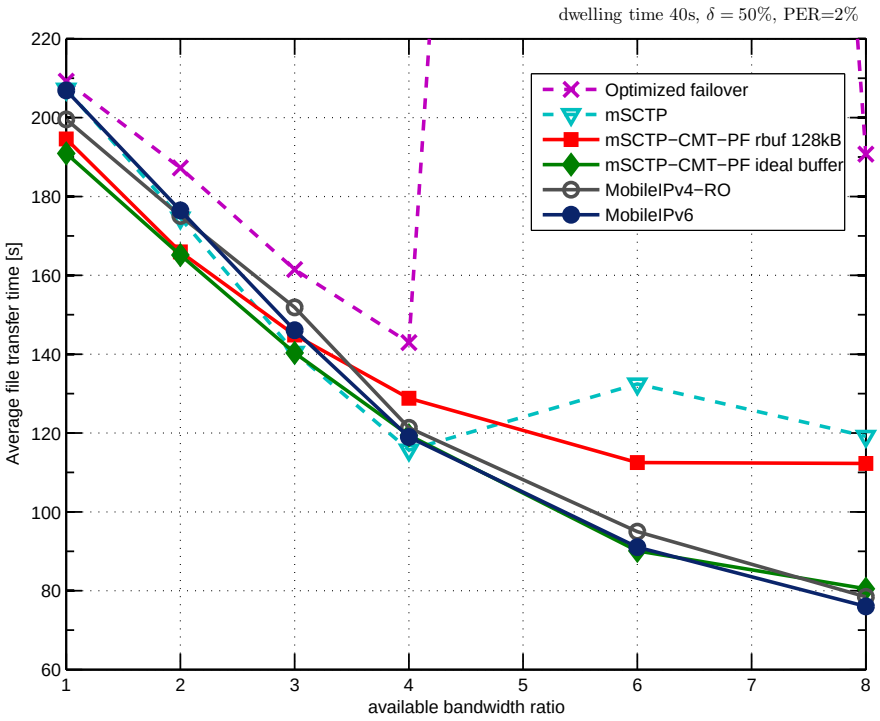


(c)

Figure 6.6: Comparison of various handover schemes for different values of δ in a scenario with pattern 1-2 and without losses, for: (c) $\delta = 75\%$.

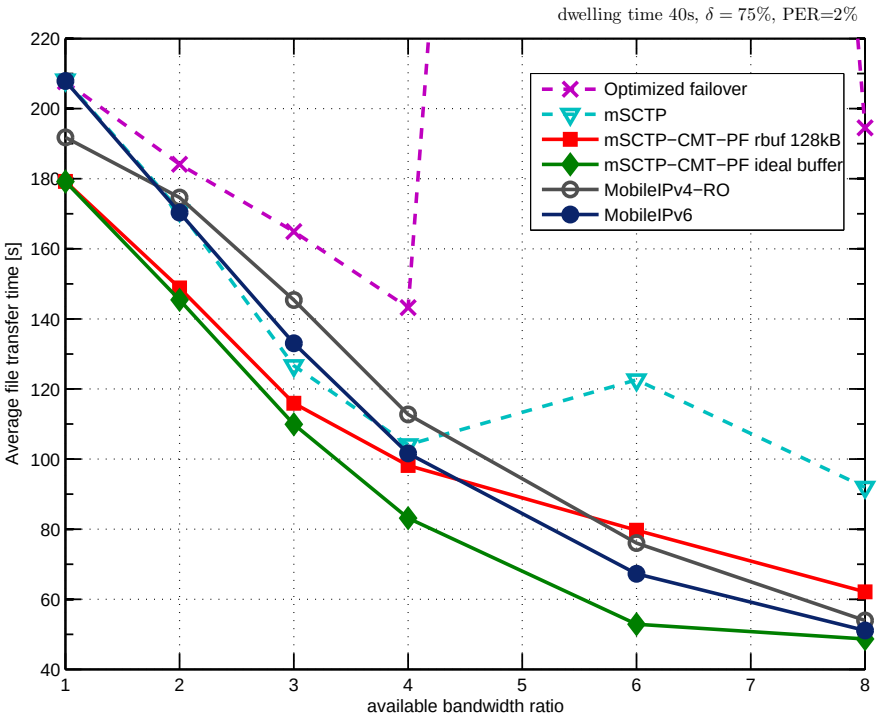


(a)



(b)

Figure 6.7: Comparison of various handover schemes for different values of δ in a scenario with pattern 1-2 and 2% of losses, for: (a) $\delta = 25\%$; and, (b) $\delta = 50\%$.



(c)

Figure 6.7: Comparison of various handover schemes for different values of δ in a scenario with pattern 1-2 and 2% of losses, for: (c) $\delta = 75\%$.

a limited receiver buffer space. However, there is some dependency on possible handover scenario conditions that can lead to suboptimal performance of this scheme, especially with low number of mandatory handovers, and big asymmetry between neighboring RANs. In contrast, it is quite clear that failover-based scheme can not be recommended as a possible solution, given its unstable performance in a presence of packet losses, very typical for wireless scenarios. The usefulness of the mSCTP-based scheme also stays in question. In a no-, or little-loss-scenario, mSCTP is able to compete with both MobileIP-based schemes, being unable to deal with greater number of losses (i.e., 2% or higher). In the presented comparison both network-layer-based schemes present acceptable results, being very competitive to the mSCTP-based schemes, despite the inferred handover latency. In this sense, both well-established mobility schemes can be seen as viable solutions, good enough not to be substituted with any scheme, which scope of use is limited.

Chapter 7

Conclusions

7.1 Summary

The SCTP protocol, which was initially proposed for transporting signaling messages over IP networks, has had a considerable impact on the research community during the first eight years since its first specification was published. SCTP is now an established general transport protocol, and the presented dissertation shows that SCTP has a great potential for application in diverse fields. The main SCTP research interest circles around two of SCTP's new features: multihoming and multistreaming, which are involved in about 70% of the identified, SCTP-related research. Multihoming that was originally designed to increase robustness, can also be used for transport layer handovers, as well as for loadsharing, as highlighted in this work.

The number of Internet drafts, protocol implementations for major OSes, and simulator models illustrate the dynamics of the SCTP research community, and the number of interesting open points promise to keep up the research interest in SCTP. Yet, an open issue is the answer to the question whether the current SCTP research can provide enough impact, so the protocol can break into the well established TCP/IP protocol stack, and thus overcome the main disadvantage of the SCTP, becoming a widely-used transport protocol.

In an effort to promote the SCTP in the networking community this dissertation investigated and evaluated the design considerations in handling mobility at the transport layer, using mSCTP as an example of a handover transport layer protocol. To this end, all important handover scenarios have been identified, along with the detailed discussion about the main considerations on usage of mSCTP as a transport handover solution in each scenario. The most representative scenario in context of the future heterogeneous wireless networks has been selected to conduct presented analysis. Firstly, the reuse of the standard SCTP failover algorithm has been discussed to provide a benchmark for the evaluation of more advanced schemes. During the evaluation of the standard SCTP failover mechanism a new method of the failover time estimation has been devised and proposed as a more accurate solution than sum of the consecutive timeout periods. The provided improvement is seen especially important in transport layer mobility scenarios based on SCTP multihoming. It has been proved that the failover-based solution, despite the necessary parameter tune-up, is not valid for any real-time applications, and practically is strictly limited to long-latency-insensitive applications. Next, an essential improvement in the context of mSCTP-based handover designs has been suggested in the mSCTP-CMT-PF proposal that joins two most important applications not envisaged within the scope of the original SCTP specification. Introduction of the loadsharing into a handover scheme seems very promising idea that may offer not only an additional throughput gain, but also smoother the handover process. Loadsharing scheme in handover scenarios incurs however strict applicability limitations, due to reported receiver buffer constraints. As a continuation, a comparison of all identified mSCTP-based handover schemes, namely: (1) optimized failover, (2) mSCTP-based solution (both best and worst case), and (3) the mSCTP-CMT-PF scheme, with two most representative network-layer solutions (4) MIPv4 and (5) MIPv6 has been given in series of performance analysis that involved simpler and more advanced mobility models. An overall conclusion that has been drawn does not seem favorable enough to advocate the substitution

of the existing, and well-established network layer solutions which perform nearly as well as the proposed transport-layer mobility schemes. Apart from the analysis of the mobility management aspects, this dissertation reports also on the state-of-the-art in SCTP modeling, very important for further protocol development.

Possibly the most important issue, although not directly related with the problematic of mobility management addressed in this dissertation, is the availability and recognition of the SCTP, as a widely-used transport layer protocol. Given the dominant role of TCP and UDP it is not an easy task to find a so-called *killer application* that will help to promote SCTP. Author believes that transport layer mobility can be one of them.

7.2 Most important remarks

Most interesting findings of this dissertation can be shortly summarized with the following conclusions:

1. To provide the seamless transport-layer mobility solution, the mSCTP schemes lack an appropriate handover policy, what was already pointed out at the early stage of the DAR extension [Stewart et al., 2007] specification. Looking for an adequate solution targeted for heterogeneous networks, several design proposals have been demonstrated here. Range of solutions considering support from the link layer in a handover triggering process incurs an interesting finding that an inadequate handover policy (characterized by inappropriate path switching triggers) based on the link layer information, called in this work *the worst case*, yields worse effect than use of the standard failover mechanism to trigger the primary path change.
2. Standard SCTP failover mechanism can be reused to grant the handover support for non real-time applications only. In order to achieve this, the most important protocol parameters `PMR` and `RTOMin` must be adjusted to better fit handover scenarios requirements, as well as presence of non-congestion losses, given that the default protocol settings derive from the wired TCP environments, where such losses are not present. As it was shown in this work also standard failover time estimation formula, based on the sum of consecutive timeouts, must be updated to take into consideration proposed protocol modifications.
3. The mSCTP-CMT-PF scheme introduced by this work, links two alternative applications of multihoming in order to provide a solution that not only guarantees additional throughput gain over previously considered schemes but also brings closer the transport-layer seamless mobility by smoothing the transition process. However, the incorporation of a loadsharing scheme into handover scenarios increases the difficulties, too. A common loadsharing problem, the rbuf blocking, has been presented, and argued that although it can not be eliminated, but when diminished it still let over perform the remaining mSCTP-based schemes, as well as some of the most common network-layer schemes.
4. Finally, it is necessary to remind the reader that an important deficiency of any discussed mSCTP solution is the lack of the location management, as argued already in the introductory discussion of various mobility management approaches. In this dissertation this aspect has not been addressed in more detail rather than just theoretical consideration of possible feedback from either MIP or DDNS.

7.3 Future work

As a continuation, a list of the most important aspects and open points identified that shall be considered as possible future research directions:

- One of the most important concerns for the SCTP-based handover schemes is the development of more accurate handover policies. Basic approaches consider incorporating indications from the link-layer. However, so far only several studies have examined such a design [Budzisz et al., 2005; Chang et al., 2004]. In that sense evaluation of *cross-layer* designs would provide more depth to the analysis.

- Improvement of the initial proposal for the mSCTP-CMT-PF. As already pointed out in Section 5.3 there are several ways of making the proposed solution better fit heterogeneous wireless scenarios, namely to introduce more frequent link probing that provides more reliable information about current link state, or ABC that aims at more aggressive behavior on a newly obtained link in order to use the CMT scheme more efficiently.
- Additional enhancement of a loadsharing scheme that is aimed to form a part of a handover solution can also include a *packet scheduler*. Several loadsharing schemes, described in Section 5.1.2 claimed that using an appropriate bandwidth estimation technique increases the robustness of a loadsharing solution.
- Improvement of the cooperation between mSCTP and MIP can provide a complete mobility scheme that includes location management. As pointed out by Noonan et al. [2002] such a cooperation leads to significant overload of the HA. Still, a joint mSCTP-MIP scheme needs further evaluation.
- In this dissertation, the scope of the analyzed applications has been limited to bulk transfer applications. It may be also beneficial to extend presented considerations with other types of applications that fit handover scenarios, such as multimedia and web traffic.
- The presented evaluation of the mSCTP-based schemes has been based on the ns-2 SCTP module [SCTP-ns2], as described in Section A.2.4. The provided model of a wireless channel that made possible evaluation of the wired-routing-based SCTP module in heterogeneous handover scenarios can be extended twofold: (1) by including more radio-specific characteristics, e.g., variable physical bit-rate, etc.; (2) SCTP module can be adapted to fit into one of the presented multihoming-enabled platforms (see Section A.2.3).
- Furthermore, development of the mSCTP-based mobility schemes shall also include security considerations, and prevention from possible attacks. Existing security specification [Stewart, Tuexen, and Camarillo, 2007; Tuexen et al., 2007] is still not complete, e.g., the use of IPsec to prevent hijacking attacks lacks specifying a detailed procedure.

Appendix A

SCTP support in ns-2 simulator

This appendix provides the reader with all necessary details of the ns-2 simulator, and in particular of the SCTP module. Also, all modifications provided by the author of this dissertation to (1) develop a more accurate wireless channel model being able to provide support for the multihoming feature of SCTP, as well as (2) improve the existing model of SCTP in ns-2, are given here.

A.1 Introduction

A freeware ns-2 [NS-2] is one of the most popular *discrete event* simulators currently available, offering a numerous models of various networking protocols. In comparison to other existing tools, such as OPNET or Qualnet, the strength of the ns-2 lies in its shared libraries and models, developed by various research institutions that provide a substantial support for modeling also the most recent protocols. A good example is the SCTP module for ns-2, built by the Protocol Engineering Labs research group at the University of Delaware [SCTP-ns2], described here in more details in Section A.2.1.

Ns-2 is an object-oriented Tcl (OTcl) script interpreter including network component and network setup module libraries, as well as a scheduler of simulation events. In order to increase its efficiency and reduce processing time, the event scheduler and the most essential network component objects are implemented in C++, as a modular structure that is easy to expand. The C++ objects are made available to OTcl through an OTcl linkage that guarantees control of the C++ objects used during a simulation. To setup and run a simulation in ns-2, a user executes an OTcl script that initiates the event scheduler, configures the network topology using network components and network setup functions in the library, and schedules packet transmission events. As a result of a simulation one or more text-format output files are produced. Simulation output file(s) contain the simulation data according to the instructions specified in the OTcl simulation script and the tracing format used. Output data can be further processed, either to provide a graphical illustration of the simulation (e.g., using a tool called Network Animator (NAM), distributed together with ns-2) or for a detailed analysis of the simulation results (e.g., using script-languages such as AWK or Perl). A very helpful tool for graphical analysis of the simulation results is Tracegraph [Małek]. For more details about ns-2, please refer the ns-2 home page [NS-2] (*ns-manual* can be found there).

Among the ns-2 network components the most important ones are nodes and links. A *node* is a compound object, comprising for an unicast node (a node that performs the unicast routing) of a node entry object, port and address classifiers, as shown in Fig. A.1a, and additionally of a multicast classifier to perform multicast routing in case of a multicast node. Links that connect the nodes are another example of a compound object. The basic *link* component is unidirectional (simplex link), and contains the following objects: an output queue of a node (Queue object), object simulating propagation delay (Link delay), object adjusting time-to-live parameter for each packet received (TTL), and object that frees packets that are dropped at the queue (Null Agent). Additionally, the simplex link structure can be complemented with the trace objects that report each packet received to a specified trace file, according to the trace format selected in the OTcl script. The basic trace

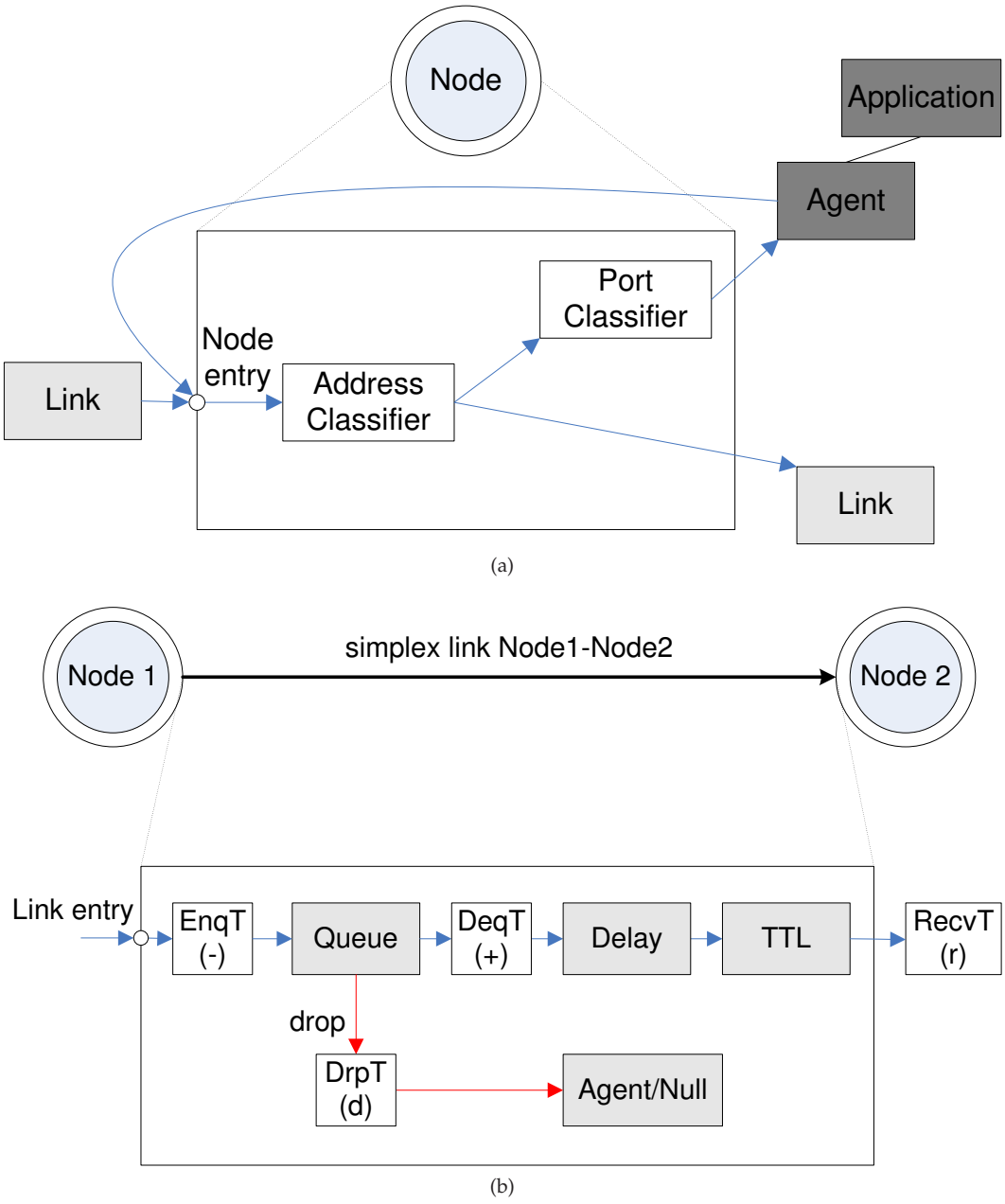


Figure A.1: Basic network components in ns-2: (a) node; and, (b) link.

objects include: EnqT object reporting arrival of the packet to the queue (marked in the trace format as '+' event), DeqT object tracing the packets leaving the queue ('-' event), DrpT object following packets that were dropped at the queue ('d' event) and RecvT object that marks the packets correctly received at the end of the link ('r' event). The entire structure of a simplex link with tracing objects is shown in Fig. A.1b. In practice, to speed up the topology setup process, the duplex-link function that creates two simplex links in both directions between given nodes is used.

The fundamental building blocks to create a protocol stack in ns-2 are *agents*. Agents represent endpoints where network-layer packets are constructed or consumed, and can be extended, using C++ derived classes, to build additional functionality. On the top of transport agents, e.g., TCP, UDP or SCTP, reside *applications*. There are two types of applications in ns-2: traffic generators, generating traffic according to a given distribution (e.g., exponential on/off, pareto on/off, constant bit rate (CBR), etc.) and simulated applications, simulating the behavior of a given application.

Currently there are only two simulated applications available in the main ns-2 distribution: FTP and Telnet.

A.2 Implementation details

This section provides a detailed description of the most important ns-2 simulator modules and objects that were modified in order to facilitate the simulation support for the analysis presented in this work.

A.2.1 SCTP module

SCTP module for ns-2 has been developed by the Protocol Engineering Labs at the University of Delaware [SCTP-ns2]. The first version of a source code has been released as a patch for ns-2 in July 2001. Since then, SCTP module has become an integral part of the main ns-2 distribution (from January 2004), and is subject to periodical updates. The latest version of the source code dates back to July 2007.

The basic functionality of SCTP is provided with the SCTP Agent class (Agent/SCTP). The list of supported features for the current SCTP implementation in ns-2 (release 3.7, version 1.12 of the sctp.cc source code) includes the following sections of the protocol specification (RFC 4960) [Stewart, 2007]:

- 5.1 - Normal establishment of an association (rudimentary handshake)
- 5.4 - Path verification (only primary path)
- 6.1 - Transmission of DATA chunks
- 6.2 - Acknowledgement of reception of DATA chunks
- 6.3 - Management of retransmission timer
- 6.4 - Multihomed SCTP endpoints
- 6.5 - Stream Identifier and Stream Sequence Number
- 6.6 - Ordered and Unordered Delivery
- 6.7 - Report Gaps in received DATA TSNs
- 7.2 - SCTP slow-start and congestion avoidance
- 8.1 - Endpoint failure detection
- 8.2 - Path failure detection
- 8.3 - Path HEARTBEAT (without upper layer control)
- additionally, there is some support of PR-SCTP, however it is not fully-compliant with RFC 3753 [Manner and Kojo, 2004], as the model was developed already at the draft-stage of the PR-SCTP specification

In context of the ns-2 architecture, multihoming support is the most significant novelty introduced by the SCTP module. The ns-2 does not allow creation of nodes with multiple interfaces due to the limitations posed by the routing algorithm used for wired nodes. Therefore, to provide support for multihomed endpoints with SCTP, the following idea has been applied. Each multihomed node is made of one *core node* connected via unidirectional links to multiple *interface nodes* that together form the logical structure of the multihomed node, illustrated in Fig. A.2. The SCTP agent resides on all the nodes that form multihomed node structure, however the data traffic goes only from and to the interface nodes. The link between the core node and interface nodes is never used to transmit data, rather for the route lookup only, to dynamically obtain the information which interface needs to be used to reach the desired destination. Presented solution for multihomed nodes works well as far as wired-only scenarios are concerned, however due to the routing issues it can not be easily extended for the wireless domain, as shown in Section A.2.3.

Some of the functionalities provided by the basic version of the SCTP code, such as the establishment of the association, or path verification procedure are implemented in the limited version that needs to be extended if more advanced setup is to be applied. The author of this dissertation proposes the extension of the path verification scheme, to fully fulfil the SCTP RFC 4960 specification [Stewart, 2007]. Path verification scheme currently implemented in ns-2 provides only the verification of the primary path that is made during the initialization phase of the association. Once

the association is set up all paths declared at the association setup are considered active. This is an almost-correct approach, if for most of the simulation time data is sent on the primary path only. However, for the scenarios involving frequent changes of the paths, e.g., handover scenarios that are subject of the analysis in this dissertation, such situation is not acceptable. Therefore, author proposes introducing a new variable *ePathVerification* that controls the path verification setup, and which can be bound from within OTcl, with three different values (modes):

```
/* Variable that controls path verification process
*/
  enum PathVerification_E
  {
    VERIFY_OFF,          // no path verification procedure used
    VERIFY_ON,           // current path verification procedure used
    VERIFY_ON_CHANGE,    // experimental, path verification is forced
each time when change is performed (handover scenarios, CMT)
  };

  PathVerification_E ePathVerification;// path verification procedure
```

Simulations presented in this dissertation were performed with *ePathVerification* set to value: *VERIFY_ON_CHANGE*.

Apart from the basic Agent/SCTP class, PEL group provides also several experimental extensions to the SCTP, available as separate classes, derived from the basic SCTP's class. Among provided modifications, two main categories can be distinguished: extensions dedicated to improve the standard SCTP retransmission policy, and CMT-related extensions. Experimental retransmission policies, described in more details and evaluated in [Caro et al., 2004b], include the following classes:

- Agent/SCTP/HbAfterRto - after each RTO, a HEARTBEAT is sent on the destination that timed out.
- Agent/SCTP/MultipleFastRtx - allows several fsat retransmissions of the same TSN, if necessary (Multiple Fast Retransmit algorithm).

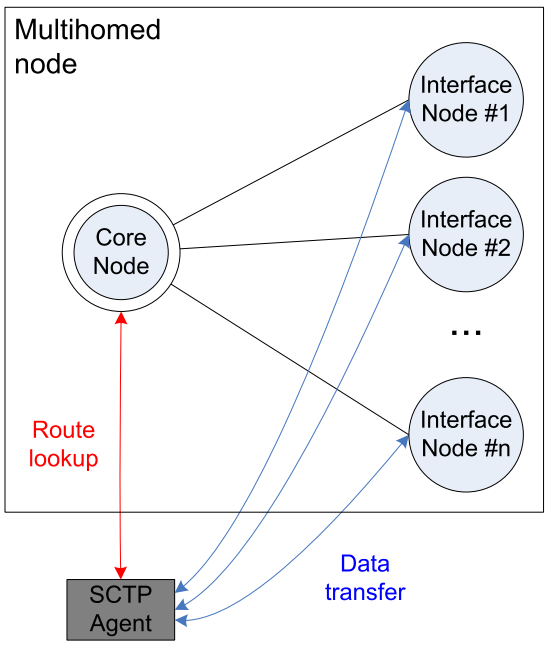


Figure A.2: Multihomed node.

- Agent/SCTP/Timestamp - includes the time stamp into each packet sent, in order to distinguish original transmissions from retransmissions.
- Agent/SCTP/MfrHbAfterRto - combines MultipleFastRtx and HbAfterRto.
- Agent/SCTP/MfrTimestamp - combines MultipleFastRtx and Timestamp.

In contrast, all CMT-related functionality is provided within one class Agent/SCTP/CMT. The latest CMT implementation (release 3.7) includes the following features, explained already in this work in Section 5.1.1 on page 84, and described in more detail in [Iyengar et al., 2006]:

- Cwnd Update for CMT algorithm (version 2).
- Split Fast Retransmit algorithm.
- Delayed Ack for CMT algorithm.
- Five CMT-dedicated retransmission policies: RtxCwnd (default and recommended), RtxSsthresh (recommended), RtxSame (experimental), RtxAsAp (experimental), and RtxLossRate (experimental).
- CMT-PF extension [Natarajan et al., 2006].

A.2.2 State of the art for wireless environments and mobility support in ns-2

Wireless support for ns-2 has been first proposed in 1998 in the mobility extension provided by the CMU's Monarch group [CMU Monarch]. Since then, the wireless functionality has been integrated into the main ns-2 package. The Monarch's wireless model introduced the MN object and few additional features to facilitate the simulation of WLANs, multi-hop ad-hoc networks, etc. A MN object is an extended version of the basic node object. The most important new features are: the ability to move within given topology while sending/receiving packets, and periodic position updates. The main difference is that the MN, unlike a basic node, does not need to use the link object to connect to the other nodes. Each MN can have one or more wireless interface, and each interface is attached to a wireless channel. Packet transmitted by the MN on a given channel is delivered to each interface on that channel, and then each interface checks using its radio propagation model whether it was able to receive given packet. According to the *ns-manual* [NS-2], the list of the features of the wireless model currently available in the main ns-2 distribution can be summarized according to the ISO/OSI protocol stack model:

- Physical Layer:
  - Radio Propagation model - three models available: free space model, two-ray ground reflection model and shadowing model.
  - Antenna model - unity gain omni-directional antenna.
  - Network interface model - a shared media interface, subject to collisions and the propagation model. Each node can overhear packets transmitted by the other nodes.
- Link Layer:
  - The IEEE 802.11 distributed coordination function (DCF) MAC protocol (for unicast packets follows the RTS/CTS/DATA/ACK pattern, uses both physical and virtual carrier sense).
  - A single hop, preamble-based TDMA protocol.
- Network Layer:
  - Four different routing protocols implemented: Dynamic Source Routing (DSR), Destination Sequence Distance Vector (DSDV), Temporally Ordered Routing Protocol (TORA) and Ad-hoc on demand Distance Vector (AODV).

Scope of the application of the original Monarch's model was limited to the simulation of WLANs and multihop wireless networks, due to the unsolved routing conflict: wired nodes use topology-based routing, whereas MNs that do not use links have dedicated routing protocols. In order to make possible coexistence of wireless and wired nodes in the same scenario two important modifications were introduced. First of them is the *base station node* (BS), a gateway node that interconnects the wired and the wireless part of the topology. Second important modification that made possible routing between wireless and wired domain is *hierarchical addressing*. Although, originally devised to reduce the size of the routing tables for large topologies (i.e. topologies with several thousands of nodes), hierarchical addressing is also useful in a combined wired-wireless scenarios.

The default hierarchical addressing format in ns-2 uses the three-level hierarchy: *domains*, *clusters and nodes*, each level being 8-bit long. Hierarchical addressing must be applied to all the nodes in the scenario. BS together with all MNs that are assigned to it form an unique domain. Packets sent to the MN in a given domain would first arrive to the BS that eventually will forward them to the correct destination. Same in the opposite direction, MNs forward packets destined outside their wireless domain to the BS, which further routes the packets into the wired part of the topology.

Introduction of the Monarch's wireless model has sparked the development of numerous models of various wireless networks that better reflect each specific scenario. Table A.1 provides a summary of the most important wireless modules currently available. More information on wireless extensions proposed for ns-2 can be found at the ns-2 homepage in the contributed code section [NS-2].

Table A.1: Most important wireless networks models for ns-2.

Wireless Tech.	Author's info	Description
IEEE 802.11	DaimlerChrysler REDNA, University of Karlsruhe	Provides revised version of WirelessPHY (SNR computation, preamble and PLCP header processing and capture) and MAC802_11 (CSMA/CA mechanism) modules, and new propagation model (Nakagami). Source: `http://dsn.tm.uni-karlsruhe.de/english/Overhaul_NS-2.php`
IEEE 802.11	INRIA	Revised WirelessPHY module (ET/SNRT/BER-based). Added support for 802.11a multirate, 802.11e HCCA and EDCA. Source: `http://spoutink.inria.fr/ns-2-80211/`
IEEE 802.11	NIST	Numerous improvements for the MAC802_11 module (improved backoff and defer timers, added management frames, and scanning for new APs). Added support for MIH (IEEE 802.21). Source: `http://www.nist.gov/`
IEEE 802.16 (WiMax)	Seamless and Secure Mobility Project NIST	Most complete WiMax model for ns-2. Provides implementation of WiMax's WirelessPHY (WirelessMAN-OFDM) and MAC (TDD, additional support for network entry without authentication) modules. Supports also the mobility extension (IEEE 802.16e) and MIH (IEEE 802.21). Source: `http://www.nist.gov/`
UMTS	SEACORN project, Ericsson	Most complete UMTS model for ns-2. Introduces RNC, BS and UE nodes, and support for simulating RACH/FACH and DCH, as well as HS-DSCH (i.e., support for HSDPA) UMTS transport channels. Source: `http://www.ti-wmc.nl/eurane`
Bluetooth	University of Cincinnati	Provides implementation of Bluetooth's WirelessPHY and MAC. Most complete Bluetooth model for ns-2. Source: `http://www.ececs.uc.edu/~cdmc/ucbt`

Having all wireless extensions developed independently has leaded to a serious problem in the coexistence of various network layer interfaces within the same simulation scenario (e.g., heterogeneous handover scenarios), as each wireless implementation provides its own routing solution. A significant effort to unify the most important wireless network interfaces (IEEE 802.11, IEEE 802.16, UMTS and Bluetooth) has been done while developing the IEEE 802.21 Media Independent Handover (MIH) platform [MIH-ns2]. More details on multihomed wireless nodes are given in Section A.2.3.

However, first mobility model in ns-2 has been created even before the Monarch's wireless model was introduced. Sun's model of Mobile IP (MIP), called Mobins2, was based on the wired nodes and links structure that modeled behavior of a wireless channel [Perkins]. Soon after Monarch's

wireless extension had been announced, Sun's MIP model was integrated into it, allowing simulations of the MIP that use wireless nodes. Currently, the wireless MIP model forms part of the core ns-2 distribution. Wireless MIP model introduces MobileNode/MIPMH and MobileNode/MIPBS objects that are modified versions of MN and BS objects, respectively. MobileNode/MIPBS objects serve as HA or FA in the wireless MIP scenario, and include the registering agent entity, responsible for sending out beacon messages to the MNs, responding to the solicitation messages from MNs, and encapsulating/decapsulating packets when necessary. MobileNode/MIPMH objects used to simulate MN have also the registering agent, which in this case receives and responds to beacon messages, as well as sends out solicitations to FA and HA.

The wireless MIP model has already several extensions proposed. The most important in context of simulations of mobility schemes different than MIP is the No Ad-hoc Routing Agent (NOAH) extension [Widmer]. NOAH is the wireless routing agent dedicated to scenarios where wireless multihop routing (e.g., DSDV or DSR) is not desired. NOAH routing does not send any routing related packets, and therefore supports only direct communication between the wireless nodes, or between BS nodes and MN nodes. Apart from providing the routing support for non-MIP mobility solutions, the NOAH extension improves also the wireless MIP model. The modifications include support for overlapping service areas of BS and improved handover mechanism through an intelligent selection of FAs.

FHMIP extension [Hsieh], is an example of a mobility extension based on both wireless MIP model and NOAH extension that provides models for various mobility management schemes, namely MIP, HMIP, FMIP, and FHMIP. FHMIP extension introduces new type of node, a MAP node, which is a wired node with the MAP agent. The MAP node acts as an intermediate node between the MN's HA and current FA. The HA encapsulates packets destined to the MN and sends them to the MAP node. The MAP node decapsulates packets incoming from the HA and encapsulates them again using the FA address. Finally, the FA can decapsulate packets and send them directly to the MN. Additionally, FHMIP extension provides support for fast handovers.

Yet, it is important to stress that within the current state-of-the-art of ns-2, the support for non-network-layer-based mobility schemes is not provided.

A.2.3 Multihoming in wireless scenarios

Simulation of multihomed wireless nodes has been one of the most challenging issues for the ns-2 community in the recent years. Currently, there are two principal extensions available, offering support for simulations of nodes with multiple wireless interfaces. One of them is the IEEE 802.21 Media Independent Handover (MIH) platform [MIH-ns2], aiming at providing the solution that unifies routing algorithm for different wireless interfaces models that were developed independently.

Second important solution is the MIRACLE ns-2 library, developed by the SIGNET Lab at the University of Padova [MIRACLE]. MIRACLE library provides means for the interlayer communication and flexible multi-layer design, allowing not only simulations of wireless nodes with multiple radio interfaces, but also making possible exchange of any type of message/structure/command among modules/protocols.

As stated already in Section A.2.1, SCTP multihoming solution was intended to work exclusively in the wired networks, due to the wired routing limitations. Indeed, if SCTP multihomed node is used in a mixed wired-wireless scenario with the hierarchical routing, the following routing conflict occurs, as reported in [Song, 2005]:

```
num_nodes is set 4
INITIALIZE THE LIST xListHead
Starting Simulation...
can't read "Node_(5)": no such element in array
while executing
"return $Node_($id)"
(procedure "_o3" line 3)
(Simulator get-node-by-id line 3)
invoked from within
```

```
"$self get-node-by-id [lindex $L 0]"
(procedure "_o3" line 14)
(Simulator compute-hier-routes line 14)
invoked from within
"$self compute-hier-routes "
invoked from within
"if [Simulator hier-addr?] {
$self compute-hier-routes
} else {
$self compute-flat-routes
}"
(procedure "_o3" line 2)
(Simulator compute-routes line 2)
invoked from within
"[Simulator instance] compute-routes"
(procedure "Agent/rtProto/Static" line 2)
(Agent/rtProto/Static init-all line 2)
invoked from within
"Agent/rtProto/Static init-all"
invoked from within
"if [info exists rtprotos_] {
foreach proto [array names rtprotos_] {
eval Agent/rtProto/$proto init-all $rtprotos_($proto)
}
} else {
Agent/rtProto/St..."
(procedure "_o191" line 3)
(RouteLogic configure line 3)
invoked from within
"[$self get-routelogic] configure"
(procedure "_o3" line 5)
(Simulator run line 5)
invoked from within
"$ns_ run"
(file "mixed-wired-wireless-sctp.tcl" line 215)
```

Therefore, there are currently two possible solutions to provide simulations of the SCTP in handover scenarios: (1) either to adapt the SCTP code, so it can co-operate with one of the multihomed platforms presented in this section, or (2) to re-use Charlie Perkins' approach from the initial MIP implementation [Perkins], and exploit wired nodes and links that fully cooperate with SCTP, to model wireless channels. Author of this dissertation opted to employ the latter option, mainly because of the fact that at the time he started working on his dissertation any of the two multihome-capable platforms presented here were not yet available.

A.2.4 Proposed solution

In order to supply simulations of SCTP in handover scenarios using wired nodes and links objects, there are several changes that must have been applied to the ns-2 (Section A.3 specifies full list of the modified files). A new model of the wireless channel has been provided, based on a modified link model, named *ARQLinearErr link*, illustrated in Fig. A.3. The most important change lies in adaptation of the link characteristic to that of a wireless link. Therefore, packets arriving at the link are marked as erroneous in the link delay module, but not dropped. Eventually, packets marked as corrupted are dropped at the drop target of the loss module that is attached after the link delay module, so the link delay is applied to all packets, also these corrupted ones. To facilitate packet drop after the link delay module a special inst-procedure endlink-lossmodel was created.

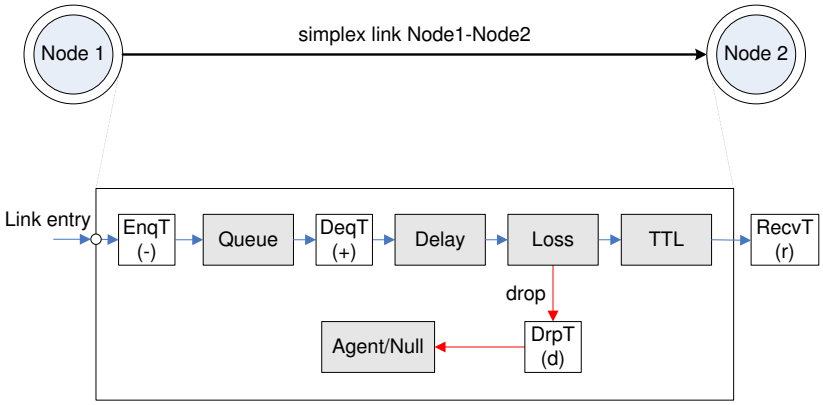


Figure A.3: Wireless channel model - ARQLinearErr link.

The wireless channel simulated using ARQLinearErr link model is characterized by a constant physical bit-rate, mean packet error rate (PER) that is either fixed or linearly degrading in function of time, and number of retransmissions for radio blocks. This model does not deal with radio specific settings for transmission control, signal-to-noise (SNR), etc.

Another important aspects that need to be commented is the construction of the handover schemes. SCTP code provides primitive *set-primary-destination* for forcing the primary path change that can be reused for handover purposes. Using that primitive, simulations for all SCTP-based handover schemes described in this dissertation have been provided.

A.3 List of the modified files

The modifications were applied to the following files (all given paths are relative to $NS_PATH):

/Makefile
/link/delay.cc
/link/delay.h
/queue/errmodel.cc
/queue/errmodel.h
/sctp/sctp.cc
/sctp/sctp.h
/sctp/sctp-cmt.cc
/sctp/sctp-cmt.h
/tcl/lib/ns-default.tcl
/tcl/lib/ns-lib.tcl
/tcl/lib/ns-link.tcl

Bibliography

Abd El Al, A., Saadawi, T., and Lee, M. 2004a. Improving throughput and reliability in mobile wireless networks via transport layer bandwidth aggregation. *Computer Networks 46*, 5 (December), 635–649. <83>

Abd El Al, A., Saadawi, T., and Lee, M. 2004b. LS-SCTP: a bandwidth aggregation technique for stream control transmission protocol. *Computer Communications 27*, 10 (June), 1012–1024. <83>

Abd El Al, A., Saadawi, T., and Lee, M. 2007. Unequal error protection for real-time video in mobile ad hoc networks via multi-path transport. *Computer Communications 30*, 17 (November), 3293–3306. <85>

ACM. The ACM Digital Library. http://portal.acm.org/. <30>

Ahn, B.-H., Kim, D.-Y., Cho, K.-H., Cha, S.-H., and Jo, M. 2007. An efficient resource reservation and qos provisioning mechanism based on mSCTP for next generation network. In *The 5th ACIS International Conference on Software Engineering Research, Management and Applications (SERA 2007)*. 245–252. <39>

Akyildiz, I., Xie, J., and Mohanty, S. 2004. A survey of mobility management in next generation All-IP based wireless systems. *IEEE Wireless Communications 11*, 4 (April), 16–28. <1, 8>

Argyriou, A. and Madisetti, V. 2007. The design and evaluation of an end-to-end handoff management protocol. *Wireless Networks 13*, 1 (February), 61–75. <38>

Atiquzzaman, M. and Reaz, A. 2005. Survey and classification of transport layer mobility management schemes. In *The 16th IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 2005)*. Vol. 4. 2109–2115. <11>

Aydin, I., Seok, W., and Shen, C.-C. 2003. Cellular SCTP: a transport-layer approach to internet mobility. In *The 12th International Conference on Computer Communications and Networks (ICCCN 2003)*. 285–290. <39>

Aydin, I. and Shen, C.-C. 2005. Evaluating cellular SCTP over one-hope wireless networks. In *The 62nd IEEE Vehicular Technology Conference (VTC 2005-Fall)*. Vol. 2. 826–830. <39>

Bhagwat, P., Perkins, C., and Tripathi, S. 1996. Network layer mobility: an architecture and survey. *IEEE Personal Communications 3*, 3 (June), 54–64. <8>

BibFinder. The BibFinder database. http://kilimanjaro.eas.asu.edu/. <30>

Budzisz, Ł., Ferrús, R., Brunstrom, A., Grinnemo, K.-J. Fracchia, R., Galante, G., and Casadevall, F. 2008. Towards transport-layer mobility: Evolution of SCTP multihoming. *Computer Communications 31*, 5 (March), 980–998. <37, 38, 39, 88>

Budzisz, Ł., Ferrús, R., and Casadevall, F. 2005. Study on transport layer handover using SCTP. In *The 8th International Symposium on Wireless Personal Multimedia Communications (WPMC 2005)*. 1551–1556. <39, 130>

Budzisz, Ł., Ferrús, R., and Casadevall, F. 2006a. On the performance of multihoming SCTP in dynamically changing radio channels. In *The 15th IST Mobile and Wireless Communications Summit*. <71>

Budzisz, Ł., Ferrús, R., and Casadevall, F. 2006b. SCTP multihoming performance in dynamically changing channels with the influence of link-layer retransmissions. In *The 64nd IEEE Vehicular Technology Conference (VTC 2006-Fall)*. 2624–2628. <38, 79>

Budzisz, Ł., Ferrús, R., and Casadevall, F. 2009. Design principles and performance evaluation of mSCTP-CMT for transport-layer based handover. In *accepted to the 69th IEEE Vehicular Technology Conference (VTC 2009-Spring)*. <101>

Budzisz, Ł., Ferrús, R., Casadevall, F., and Amer, P. 2009. On Concurrent Multipath Transfer in SCTP-based handover scenarios. In *accepted to the IEEE International Conference on Communications (ICC 2009)*. <90>

Budzisz, Ł., Ferrús, R., Grinnemo, K.-J., Brunstrom, A., and Casadevall, F. 2007. An analytical estimation of the failover time in SCTP multihoming scenario. In *IEEE Wireless Communications and Networking Conference (WCNC 2007)*. 3932–3937. <54, 55>

Budzisz, Ł., Garcia, J., Brunstrom, A., and Ferrús, R. 2008. A taxonomy and survey of SCTP research. *work in progress, to be submitted to ACM Computing Surveys*. <25, 85>

Bush, R. and Meyer, D. 2002. RFC 3439, some Internet architectural guidelines and philosophy. http://www.ietf.org/rfc/rfc3439.txt. <11>

Campbell, A., Gomez, J., Kim, S., Wan, C.-Y., Turányi, Z., and Valkó, A. 2002. Comparison of IP micro-mobility protocols. *IEEE Wireless Communications 9*, 1 (February), 72–82. <9, 11>

Campbell, A., Gomez, J., Wan, C.-Y., Kim, S., Turányi, Z., and Valkó, A. 1999. Cellular IP, IETF draft. http://tools.ietf.org/id/draft-valko-cellularip-01.txt, draft expired. <9>

Caro, Jr., A. 2005. End-to-end fault tolerance using transport layer multihoming. Ph.D. thesis, University of Delaware. <53, 62, 71, 72, 73>

Caro, Jr., A., Amer, P., and Stewart, R. 2004a. End-to-end failover thresholds for transport layer multihoming. In *IEEE Military Communications Conference (MILCOM 2004)*. 99–105. <62>

Caro, Jr., A., Amer, P., and Stewart, R. 2004b. Retransmission schemes for end-to-end failover with transport layer multihoming. In *IEEE Global Telecommunications Conference (GLOBECOM 2004)*. Vol. 3. 1341–1347. <136>

Carpenter, B. 1996. RFC 1958, Architectural principles of the Internet. http://www.ietf.org/rfc/rfc1958.txt. <11>

Casetti, C., Chiasserini, C. F., Fracchia, R., and Meo, M. 2006. AISLE: Autonomic Interface SeLEction for Wireless Users. In *International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2006)*. 42–48. <39>

Casetti, C., Galante, G., and Greco, R. 2004. Load balancing over multipaths using bandwidth-aware source scheduling. In *The 7th International Symposium on Wireless Personal Multimedia Communications (WPMC 2004)*. <85>

Chang, L.-H., Lin, H.-J., and Chang, I. 2007. Dynamic handover mechanism using mobile SCTP in contention based wireless network. In *The 5th International Symposium on Parallel and Distributed Processing and Applications (ISPA 2007)*. Vol. LNCS 4742. 821–831. <39>

Chang, M., Lee, M., and Koh, S. 2004. Transport layer mobility support utilizing link signal strength information. *IEICE Transactions on Communications E87-B*, 9 (September), 2548–2556. <39, 130>

Chockalingam, A., Zorzi, M., and Tralli, V. 1999. Wireless TCP performance with link layer FEC/ARQ. In *IEEE International Conference on Communications (ICC 1999)*. Vol. 2. 1212–1216. <76>

CiteSeer. CiteSeer.IST, Scientific Literature Digital Library. http://citeseer.ist.psu.edu/. <30>

CMU Monarch. The cmu monarch project's wireless and mobility extensions to ns-2. http://www.monarch.cs.cmu.edu/cmu-ns.html. <137>

Coene, L. 2002. RFC 3257, Stream Control Transmission Protocol applicability statement. http://www.ietf.org/rfc/rfc3257.txt. <17>

Deering, S. and Hinden, R. 1998. RFC 2460, Internet Protocol, version 6 (IPv6) specification. http://www.ietf.org/rfc/rfc2460.txt. <9>

Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and Carney, M. 2003. RFC 3315, Dynamic Host Configuration Protocol for IPv6 (DHCPv6). http://www.ietf.org/rfc/rfc3315.txt. <10>

Dutta, A., Famolari, D., Das, S., Ohba, Y., Fajardo, V., Taniuchi, K., Lopez, R., and Schulzrinne, H. 2008. Media-independent pre-authentication supporting secure interdomain handover optimization. *IEEE Wireless Communications 15*, 2 (April), 55–64. <7>

Eddy, M. 2004. At what layer does mobility belong? *IEEE Communications Magazine 42*, 10 (October), 155–159. <1, 11, 15>

Eklund, J. and Brunstrom, A. 2006. Impact of SACK delay and link delay on failover performance in SCTP. In *The 3rd IASTED International Conference on Communications and Computer Networks*. 69–74. <68>

El Malki, K. 2007. RFC 4881, Low-latency handoffs in Mobile IPv4. http://www.ietf.org/rfc/rfc4881.txt. <9>

EngVillage. The Engineering Village database. http://www.engineeringvillage2.com/. <30>

Fiore, M. and Casetti, C. 2005. An adaptive transport protocol for balanced multihoming of real-time traffic. In *IEEE Global Telecommunications Conference (GLOBECOM 2005)*. Vol. 2. 1091–1096. <85>

Fiore, M., Casetti, C., and Galante, G. 2007. Concurrent multipath communication for real-time traffic. *Computer Communications 30*, 17 (November), 3307–3320. <85>

Fitzpatrick, J., Murphy, S., and Murphy, J. 2006. SCTP based handover mechanism for VoIP over IEEE 802.11b wireless LAN with heterogeneous transmission rates. In *IEEE International Conference on Communications (ICC 2006)*. Vol. 5. 2054–2059. <37>

Fogelstroem, E., Jonsson, A., and Perkins, C. 2007. RFC 4857, Mobile IPv4 regional registration. http://www.ietf.org/rfc/rfc4857.txt. <9>

Fracchia, R., Casetti, C., Chiasserini, C.-F., and Meo, M. 2005. A WiSE extension of SCTP for wireless networks. In *IEEE International Conference on Communications, (ICC 2005)*. 1448–1453. <39>

Fracchia, R., Casetti, C., Chiasserini, C.-F., and Meo, M. 2007. WiSE: Best-path selection in wireless multihoming environments. *IEEE Transactions on Mobile Computing 6*, 10 (October), 1130–1141. <39>

Fu, S., Ma, L., Atiquzzaman, M., and Lee, Y. 2005. Architecture and performance of SIGMA: A seamless handover scheme for data networks. In *IEEE International Conference on Communications (ICC 2005)*. Vol. 5. 3249–3255. <13>

Giaretta, G. 2009. Interactions between PMIPv6 and MIPv6: scenarios and related issues, IETF draft. http://tools.ietf.org/id/draft-ietf-netlmm-mip-interactions-02.txt, work in progress. <7, 10>

Goff, T., Moronski, J., Phatak, D., and Gupta, V. 2000. Freeze-TCP: a true end-to-end TCP enhancement mechanism for mobile environments. In *The 19th Annual Joint Conference of the IEEE Computer and Communications Society (INFOCOM 2000)*. 1537–1545. <11>

Goff, T. and Phatak, D. 2004. Unified transport layer support for data striping and host mobility. *IEEE Journal on Selected Areas in Communications 22*, 4 (May), 737–746. <39, 83, 86>

Google Scholar. Google scholar. http://scholar.google.com/. <30>

Grinnemo, K.-J. and Brunstrom, A. 2005. Impact of traffic load on SCTP failovers in SIGTRAN. In *International Conference on Networking 2005 (ICN 2005)*. 774–783. <65>

Honda, M., Nishida, Y., Nakazawa, J., and Tokuda, H. 2007. Performance enhancement of transport layer handover on single-homed mobile nodes. *IEICE Transactions on Communications E90-B*, 10 (October), 2683–2692. <37, 42, 87>

Hsieh, R. FHMIP ns-2 extension. http://mobqos.ee.unsw.edu.au/robert/opcomm/. <139>

Huang, C.-M., Chiang, M.-S., and Lin, J.-W. 2008. A proactive mobile-initiated fast handoff scheme using the multihomed approach. *Wireless Communications and Mobile Computing*. <15>

Huang, C.-M. and Tsai, C. H. 2007. WiMP-SCTP: Multi-path transmission using stream control transmission protocol (SCTP) in wireless networks. In *The 21st International Conference on Advanced Information Networking and Applications Workshops/Symposia (AINAW 2007)*. Vol. 2. 209–214. <86>

Hui, S. and Yeung, K. 2003. Challenges in the migration to 4G mobile systems. *IEEE Communications Magazine 41*, 12 (December), 54–59. <1>

IEEEXplore. The IEEE Xplore database. http://ieeexplore.ieee.org/Xplore/dynhome.jsp. <30>

IETF SIGTRAN. IETF signaling transport (sigtran) working group webpage. <17, 24>

IETF TSVWG. IETF transport area (tsvwg) working group webpage. <4, 17>

ISI. The ISI Web of Knowledge. http://www.isiwebofknowledge.com/. <30>

Iyengar, J., Amer, P., and Stewart, R. 2004a. Concurrent multipath transfer using transport layer multihoming - performance under varying bandwidth proportions. In *IEEE Military Communications Conference (MILCOM 2004)*. 238–244. <84>

Iyengar, J., Amer, P., and Stewart, R. 2004b. Retransmission policies for concurrent multipath transfer using SCTP multihoming. In *The 12th IEEE International Conference on Networks (ICON 2004)*. 713–719. <84, 97>

Iyengar, J., Amer, P., and Stewart, R. 2005. Receive buffer blocking in concurrent multipath transfer. In *IEEE Global Telecommunications Conference (GLOBECOM 2005)*. Vol. 1. 121–126. <84>

Iyengar, J., Amer, P., and Stewart, R. 2006. Concurrent multipath transfer using sctp multihoming over independent end-to-end paths. *IEEE/ACM Transactions on Networking 14*, 5 (October), 951–964. <24, 49, 84, 97, 99, 137>

Iyengar, J., Amer, P., and Stewart, R. 2007. Performance implications of a bounded receive buffer in concurrent multipath transfer. *Computer Communications 30*, 4 (February), 818–829. <85>

Johnson, D. and Perkins, C. 2001. Route optimization in Mobile IP, IETF draft. http://tools.ietf.org/id/draft-ietf-mobileip-optim-11.txt, draft expired. <9>

Johnson, D., Perkins, C., and Arkko, J. 2004. RFC 3775, Mobility support in IPv6. http://www.ietf.org/rfc/rfc3775.txt. <9, 109>

Jung, H., Koh, S., Soliman, H., and et al. 2004. Fast handover for Hierarchical MIPv6 (F-HMIPv6), IETF draft. http://tools.ietf.org/id/draft-jung-mobileip-fastho-hmipv6-04.txt, draft expired. <10>

Jungmaier, A., Rathgeb, E., and Tuexen, M. 2002. On the use of SCTP in failover scenarios. In *The 6th World Multiconference on Systemics, Cybernetics, and Informatics (SCI 2002)*. 363–368. <65>

Kashihara, S., Iida, K., Koga, H., Kadobayashi, Y., and Yamaguchi, S. 2004. Multi-path transmission algorithm for end-to-end seamless handover across heterogeneous wireless access networks. *IEICE Transactions on Communications E87-B*, 3 (March), 490–496. <39, 68>

Kim, D., Koh, S., and Kim, S. 2006. mSCTP-DAC: dynamic address configuration for msctp handover. In *International Conference on Embedded and Ubiquitous Computing (EUC 2006)*. Vol. LNCS 4096. 244–253. <39>

Kim, W., Kim, M., Lee, K., Yu, C., and Lee, B. 2004. Link layer-assisted mobility support using SIP for real-time multimedia communications. In *The 2nd ACM International Workshop on Mobility Management and Wireless Access Protocols (MobiWac 2004)*. 127–129. <14>

Koh, S., Chang, M. J., and Lee, M. 2004. mSCTP for soft handover in transport layer. *IEEE Communications Letters 8*, 3 (March), 189–191. <38, 39>

Koh, S. and Xie, Q. 2004. Mobile SCTP with Mobile IP for transport layer mobility, IETF draft. http://tools.ietf.org/id/draft-sjkoh-mobile-sctp-mobileip-04.txt, draft expired. <14>

Koh, S. and Xie, Q. 2005. Mobile SCTP (mSCTP) for IP handover support, IETF draft. http://tools.ietf.org/id/draft-sjkoh-msctp-01.txt, draft expired. <24, 25>

Kohler, E., Handley, M., and Floyd, S. 2006. RFC 4340, Datagram Congestion Control Protocol (DCCP). http://www.ietf.org/rfc/rfc4340.txt. <11>

Kong, K.-S., Lee, W., Han, Y.-H., Shin, M.-K., and You, H. 2008. Mobility management for all-IP mobile networks: mobile IPv6 vs. proxy mobile IPv6. *IEEE Wireless Communications 15*, 2 (April), 36–45. <10>

Koodli, R. 2008. RFC 5268, Mobile IPv6 fast handovers. http://www.ietf.org/rfc/rfc5268.txt. <10>

Koodli, R. and Perkins, C. 2007. RFC 4988, Mobile IPv4 fast handovers. http://www.ietf.org/rfc/rfc4988.txt. <9>

Lee, K. J., Nam, S. S., and Mun, B. I. 2006. Sctp efficient flow control during handover. In *IEEE Wireless Communications and Networking Conference (WCNC 2006)*. 69–73. <39>

Liebsch, M., Singh, A., Chaskar, H., Funato, D., and Tuexen, M. 2005. RFC 4066, Candidate Access Router Discovery (CARD). http://www.ietf.org/rfc/rfc4066.txt. <42>

Ma, L., Yu, F., and Leung, V. 2007. Performance improvements of mobile SCTP in integrated heterogeneous wireless networks. *IEEE Transactions on Wireless Communications 6*, 10 (October), 3567–3577. <39>

Ma, L., Yu, F., Leung, V., and Randhawa, T. 2004. A new method to support UMTS/WLAN vertical handover using sctp. *IEEE Wireless Communications 11*, 4 (August), 44–51. <38>

Małek, J. Tracegraph tool (ns trace file analyzer). http://www.tracegraph.com/. <133>

Maltz, D. and Bhagwat, P. 1998. MSOCKS: an architecture for transport layer mobility. In *The 17th Annual Joint Conference of the IEEE Computer and Communications Society (INFOCOM 1998)*. Vol. 3. 1037–1045. <11>

Manner, J. and Kojo, M. 2004. RFC 3753, Mobility related terminology. http://www.ietf.org/rfc/rfc3753.txt. <5, 39, 135>

Matsuoka, H., Yoshimura, T., and Ohya, T. 2003. End-to-end robust IP soft handover. In *IEEE International Conference on Communications (ICC 2003)*. Vol. 1. 532–536. <13>

MIH-ns2. The network simulator ns-2 NIST add-on: IEEE 802.21 model. http://www.nist.gov/. <138, 139>

MIRACLE. NS-MIRACLE library: the Multi-InteRfAce Cross-Layer Extension for ns-2 simulator. http://www.dei.unipd.it/wdyn/?IDsezione=3965. <139>

Mockapetris, P. 1987. RFC 1034, Domain names - concepts and facilities. http://www.ietf.org/rfc/rfc1034.txt. <11>

Montenegro, G. 2001. RFC 3024, Reverse tunneling for Mobile IP, revised. http://www.ietf.org/rfc/rfc3024.txt. <9>

Natarajan, P., Ekiz, N., Amer, P., and Stewart, R. 2009. Concurrent multipath transfer during path failure. *accepted for publication in Computer Communications 32*. <85, 110>

Natarajan, P., Iyengar, J., Amer, P., and Stewart, R. 2006. Concurrent multipath transfer using transport layer multihoming: Performance under network failures. In *IEEE Military Communications Conference (MILCOM 2006)*. <85, 110, 112, 137>

Nikander, P., Arkko, J., Aura, T., Montenegro, G., and Nordmark, E. 2005. RFC 4225, Mobile IPv6 route optimization security design background. http://www.ietf.org/rfc/rfc4225.txt. <9>

Noonan, J., Perry, P., and Murphy, J. 2002. A study of SCTP services in mobile IP network. In *The 2nd Annual Information Technology and Telecommunications Conference (ICT 2002)*. 153–161. <14, 131>

Noonan, J., Perry, P., and Murphy, J. 2006. End-point synchronisation and handover for multihomed services. *IEE Proceedings: Communications 153*, 5 (October), 691–696. <39>

NS-2. The network simulator ns-2. http://www.isi.edu/nsnam/ns/. <3, 4, 59, 71, 89, 133, 137, 138>

Ong, L., Rytina, I., Garcia, M., Schwarzbauer, H., Coene, L., Lin, H., Juhasz, I., and Sharp, C. 1999. RFC 2719, Framework architecture for signaling transport. http://www.ietf.org/rfc/rfc2719.txt. <17>

Paxon, V. and Allman, M. 2000. RFC 2988, Computing TCPs retransmission timer. http://www.ietf.org/rfc/rfc2988.txt. <51>

Paxon, V., Allman, M., and Stevens, W. 1999. RFC 2581, TCPs congestion control. http://www.ietf.org/rfc/rfc2581.txt. <22, 68>

Perkins, C. Mobins2 extension to ns-2. http://people.nokia.net/ charliep/mobins2/. <138, 140>

Perkins, C. 1997. Mobile IP. *IEEE Communications Magazine 35*, 5 (May), 84–99. <8>

Perkins, C. 2002. RFC 3344, IP mobility support for IPv4. http://www.ietf.org/rfc/rfc3344.txt. <2, 8, 109>

Perkins, C., Calhoun, P., and Bharatia, J. 2007. RFC 4721, Mobile IPv4 challenge/response extensions (revised). http://www.ietf.org/rfc/rfc4721.txt. <9>

Perkins, C. and Johnson, D. 1998. Route optimization for Mobile IP. *Cluster Computing 1*, 2 (June), 161–176. <9, 109>

Perotto, F., Casetti, C., and Galante, G. 2007. SCTP-based transport protocols for concurrent multipath transfer. In *IEEE Wireless Communications and Networking Conference (WCNC 2007)*. 2971–2976. <85>

Plummer, D. 1982. RFC 826, An Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware, (standard 37). http://www.ietf.org/rfc/rfc826.txt. <8>

Postel, J. 1980. RFC 768, User Datagram Protocol, (standard 6). http://www.ietf.org/rfc/rfc768.txt. <2>

Postel, J. 1981. RFC 793, Transmission Control Protocol, (standard 7). http://www.ietf.org/rfc/rfc793.txt. <2>

Postel, J. 1984. RFC 925, Multi-LAN address resolution. http://www.ietf.org/rfc/rfc925.txt. <8>

Ramjee, R., La Porta, T., Thuel, S., Varadhan, K., and Wang, S. 1999. HAWAII: a domain-based approach for supporting mobility in wide-area wireless networks. In *Proc. of the 7th International Conference on Network Protocols (ICNP 1999)*. 283–292. <9>

Riegel, M. and Tuexen, M. 2007. Mobile SCTP, IETF draft. http://tools.ietf.org/id/draft-riegel-tuexen-mobile-sctp-09.txt, work in progress. <7, 11, 25>

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E. 2002. RFC 3261, SIP: Session Initiation Protocol. http://www.ietf.org/rfc/rfc3261.txt. <14>

Rossi, C., Casetti, C., Fiore, M., and Schonfeld, D. 2006. A partially reliable transport protocol for multiple-description real-time multimedia traffic. In *The 2006 IEEE International Conference on Image Processing*. 1301–1304. <85>

Saha, D., Mukherjee, A., Misra, I., Chakraborty, M., and Subhash, N. 2004. Mobility support in IP: a survey of related protocols. *IEEE Network 18,* 6 (November-December), 34–40. <11, 112, 113>

Schulzrinne, H. and Wedlund, E. 2000. Application-layer mobility using SIP. *ACM Mobile Computer and Communications Review 4,* 3 (July), 47–57. <14>

SCTP-FreeBSD. SCTP reference implementation. http://www.sctp.org/. <41, 54, 107>

SCTP-LK. Linux kernel SCTP (LK-SCTP) implementation. http://lksctp.sourceforge.net/. <37, 54>

SCTP-ns2. Protocol engineering labs (pel), SCTP module for network simulator ns-2. http://pel.cis.udel.edu/. <4, 31, 34, 131, 133, 135>

SCTP-Qualnet. SCTP module for Qualnet simulator. http://degas.cis.udel.edu/SCTP/. <4, 31>

SCTP-SS7. Open SS7 SCTP implementation. http://www.openss7.org/. <54>

SCTP Survey. SCTP Database webpage. http://www.cs.kau.se/cs/prtp/sctpwiki/pmwiki.php?n=Resources.SCTPSurvey. <30>

Shacham, R., Schulzrinne, H., Thakolsri, S., and Kellerer, W. 2007. Session Initiation Protocol (SIP) session mobility, IETF draft. http://tools.ietf.org/id/draft-shacham-sipping-session-mobility-05.txt, RFC Ed Queue. <14>

Snoeren, A. and Balakrishnan, H. 2000. An end-to-end approach to host mobility. In *The 6th International Conference on Mobile Computing and Networking (MobiCom 2000)*. 155–166. <13>

Soliman, H., Castelluccia, C., El Malki, K., and Bellier, L. 2005. RFC 4140, Hierarchical Mobile IPv6 mobility management (HMIPv6). http://www.ietf.org/rfc/rfc4140.txt. <10>

Song, J. 2005. Performance evaluation of handoff between UMTS/802.11b based on Mobile IP and SCTP. M.S. thesis, North Carolina State University. <139>

Stewart, R. 2007. RFC 4960, Stream Control Transmission Protocol (SCTP). http://www.ietf.org/rfc/rfc4960.txt. <17, 18, 19, 22, 23, 53, 54, 71, 135>

Stewart, R. and Amer, P. 2007. Why is SCTP needed given TCP and UDP are widely available? http://www.isoc.org/briefings/017/. <25>

Stewart, R., Arias-Rodriguez, I., Poon, K., Caro, Jr., A., and Tuexen, M. 2006. RFC 4460, Stream Control Transmission Protocol (SCTP) specification errata and issues. http://www.ietf.org/rfc/rfc4460.txt. <17>

Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and Conrad, P. 2004. RFC 3758, Stream Control Transmission Protocol (SCTP) partial reliability extension. http://www.ietf.org/rfc/rfc3758.txt. <17, 19, 24, 25>

Stewart, R., Tuexen, M., and Camarillo, G. 2007. RFC 5062, Security attacks found against the Stream Control Transmission Protocol (SCTP) and current countermeasures. http://www.ietf.org/rfc/rfc5062.txt. <13, 24, 42, 131>

Stewart, R., Tuexen, M., and Ruengeler, I. 2008. SCTP Network Address Translation, IETF draft. http://tools.ietf.org/id/draft-ietf-behave-sctpnat-00.txt, work in progress. <38>

Stewart, R. and Xie, Q. 1998. Multi-network Datagram Transmission Protocol (MDTP), IETF draft. http://tools.ietf.org/id/draft-stewart-xie-mdtp-00.txt, draft expired. <17>

Stewart, R. and Xie, Q. 2001. *Stream Control Transmission Protocol (SCTP) - A reference guide*. Addison Wesley. <53>

Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., and Paxson, V. 2000. RFC 2960, Stream Control Transmission Protocol (SCTP). http://www.ietf.org/rfc/rfc2960.txt, obsoleted by RFC 4960. <17, 23>

Stewart, R., Xie, Q., Tuexen, M., Maruyama, S., and Kozuka, M. 2007. RFC 5061, Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration. http://www.ietf.org/rfc/rfc5061.txt. <19, 24, 42, 130>

Stone, J., Stewart, R., and Otis, D. 2002. RFC 3309, Stream Control Transmission Protocol (SCTP) checksum change. http://www.ietf.org/rfc/rfc3309.txt, obsoleted by RFC 4960. <17>

Thomson, S. and Narten, T. 1998. RFC 2462, IPv6 stateless address autoconfiguration. http://www.ietf.org/rfc/rfc2462.txt. <10>

Tuexen, M., Stewart, R., Lei, P., and Rescorla, E. 2007. RFC 4895, Authenticated chunks for the Stream Control Transmission Protocol (SCTP). http://www.ietf.org/rfc/rfc4895.txt. <19, 25, 131>

Vacirca, F., De Vebductis, A., and Baiocchi, A. 2006. Optimal design of hybrid FEC/ARQ schemes for TCP over wireless links with Rayleigh fading. *IEEE Transactions on Mobile Computing 5*, 4 (April), 289–302. <76>

Vakil, F., Dutta, A., Chen, J.-C., Tauil, M., Baba, S., Nakajima, N., and Schulzrinne, H. 2001. Supporting mobility for TCP with SIP, IETF draft. http://tools.ietf.org/id/draft-itsumo-sipping-mobility-tcp-00.txt, draft expired. <14>

Vixie, P., Thomson, S., Rekhter, Y., and Bound, J. 1997. RFC 2136, Dynamic updates in the Domain Name System (DNS UPDATE). http://www.ietf.org/rfc/rfc2136.txt. <11>

Wedlund, E. and Schulzrinne, H. 1999. Mobility support using SIP. In *The 2nd ACM International Workshop on Wireless Mobile Multimedia (WoWMoM 1999)*. 76–82. <14>

Widmer, J. No Ad-hoc Routing Agent (NOAH) extension for ns-2. http://icapeople.epfl.ch/widmer/uwb/ns-2/noah. <139>

Xie, Q., Stewart, R., Holdrege, M., and Tuexen, M. 2007. SCTP NAT Traversal Considerations, IETF draft. http://tools.ietf.org/id/draft-xie-behave-sctp-nat-cons-03.txt, draft expired. <38>

Yabusaki, M., Okagawa, T., and Imai, K. 2005. Mobility management in All-IP mobile network: end-to-end intelligence or network intelligence? *IEEE Communications Magazine 43*, 12 (December), supl.16–supl.24. <1, 10>

Ye, G., Saadawi, T., and Lee, M. 2004. IPCC-SCTP: an enhancement to the standard SCTP to support multi-homing efficiently. In *The 23rd IEEE International Conference on Performance, Computing, and Communications (IPCCC 2004)*. 523–530. <83>

Zahran, A., Liang, B., and Saleh, A. 2008. Mobility modeling and performance evaluation of heterogeneous wireless networks. *IEEE Transactions on Mobile Computing 7*, 8 (August), 1041–1056. <110>

Zeadally, S. and Siddiqui, F. 2007. An empirical analysis of handoff performance for SIP, mobile IP, and SCTP protocols. *Wireless Personal Communications 43*, 2 (October), 589–603. <38>

Zhang, J., Chan, H., and Leung, V. 2007. A SIP-based seamless-handoff (S-SIP) scheme for heterogeneous mobile networks. In *IEEE Wireless Communications and Networking Conference (WCNC 2007)*. 3946–3950. <15>

Zimmermann, H. 1980. OSI reference model–the ISO model of architecture for open systems interconnection. *IEEE Transactions on Communications [legacy, pre - 1988] 28*, 4 (April), 425–432. <1>

References in this dissertation follow the ACM Transactions bibliography style: in-text citations have the general form "[author name(s) year]", whereas the reference list is sorted alphabetically by the author name(s). Page(s) where the item is referenced are given in angle brackets <p.>.

Index