NAIST-IS-DD0761002

Doctoral Dissertation

A Study on TCP Adaptation Methods for Heterogeneous Wireless Networks

Kenichi Ishibashi

February 4, 2010

Department of Information Systems
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Kenichi Ishibashi

Thesis Committee:

Professor Hideki Sunahara	(Supervisor)
Professor Suguru Yamaguchi	(Co-supervisor)
Associate Professor Kazutoshi Fujikawa	(Co-supervisor)
Professor Hiroshi Esaki	(Co-supervisor)
Associate Professor Youki Kadobayashi	(Co-supervisor)

A Study on TCP Adaptation Methods for Heterogeneous Wireless Networks*

Kenichi Ishibashi

Abstract

Enhancing the quality of communication in the mobile environment is an essential step towards establishing a sophisticated information society. Efforts made to establish high quality communication in the mobile environment have yielded diverse kinds of wireless communication technologies. The rapid progress and deployment of diverse wireless communication technologies provide for mobile nodes the capacity to form *heterogeneous wireless networks*, in which mobile nodes can employ different kinds of wireless technologies simultaneously to achieve both fast communication and a wide communication range. In the mobile environment, seeking optimized utilization of heterogeneous wireless networks is an effective way to enhance the communication quality. However, the traditional TCP/IP model cannot reap the full benefit of heterogeneous wireless networks since it requires additional solutions to cope with several issues, such as mobility support, fragile data-link characteristics, and different link characteristics of each individual wireless data-link. As an important part of a comprehensive study on the effective utilization of heterogeneous wireless networks, this dissertation studies the issues arising from the different link characteristics of each wireless technology.

When a mobile node conducts a *vertical handoff*, i.e., the mobile node switches from one upstream data-link to another, this usually involves drastic changes in link characteristics. Since the traditional TCP/IP protocols have evolved for stationary networks, the protocols did not envisage such drastic changes and they may not work as expected for heterogeneous wireless networks. In particular, the TCP congestion control mechanism will suffer from such drastic changes and will not work properly. This dissertation presents

*Doctoral Dissertation, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0761002, February 4, 2010.

two different approaches to address the problem. One is to reset the TCP congestion control parameters after a vertical handoff to adjust the congestion control mechanism to the after-handoff data-link. This scheme supposes senders can detect the occurrence of the vertical handoff immediately and called *Teppi*. Another is to introduce a TCP tunneling based multi-homing mechanism to allow a mobile node to use multiple data-links simultaneously and to manage the congestion control parameters separately for each available data-link. This scheme called *Luct*. These proposed schemes are implemented on actual operating systems and evaluated in an emulation environment and actual heterogeneous wireless networks. The experimental results show that these schemes can successfully handle vertical handoffs in their target environments.

Keywords:

Mobile Networking, Wireless Networks, Transport Layer Adaption, Vertical Handoff, Multi-homing

Acknowledgments

I thank very much my family for understanding and supporting my long research life. They have worried about my life since I quit my last job and went to NAIST. I am able to stand here because of their unqualified support.

I thank Professor Hideki Sunahara, for his warm guidance, generosity and extensive knowledge. I could never have met all my research goals without his powerful support. I am grateful to my dissertation committee for their warm support in the improvement of this dissertation. Professor Suguru Yamaguchi provided me with deep and extensive suggestions throughout my research activities. His excellent insights improved this dissertation. I thank Associate Professor Kazutoshi Fujikawa, for his helpful comments to improve my dissertation. I appreciate Professor Hiroshi Esaki of the University of Tokyo, for his valuable and sharp suggestions. I also express my gratitude to Associate Professor Youki Kadobayashi, for giving me a broader point of view on my research activities.

I especially thank my mentor, Dr. Naoto Morishima, for giving me research direction and brilliant suggestions on my research activities. I have learned a lot from him: research thinking, scientific writing, network operation, programming, kernel hacking, and the spirit of engineering. I owe my situation today to his dedicated leadership.

I spent a lot of time with all members of the Laboratory for Internet Architecture and Systems, also known as inet-lab, NAIST. I thank the faculty members of inet-lab, Associate Professor Atsuo Inomata, Dr. Michiko Izumi, Dr. Masatoshi Kakiuchi and Dr. Satoshi Matsuura, and Dr. Naomi Terada, research division digital library, NAIST, for their helpful suggestions and comments. I also thank Ms. Yuuki Ro, who is a secretary of the inet-lab, for her warm support. I totally enjoyed my life with the inet-lab members and I thank all past and present members of inet-lab.

February 4, 2010

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Internet has become the most successful communications platform worldwide during the last few decades. At the dawn of the Internet, a few nodes were connected with each other and transmitted only text-based messages. Today, the Internet has become a communications infrastructure that allows people around the world to interact with rich content such as multimedia. In all fields of human activities, such as economics, politics, educations, and daily life, the Internet has provided a multipurpose communication method and it has become an integral part of these activities.

Human activities are performed any time and anywhere. The growing influence of the Internet in human activities has brought the demand of universal connectivity to the Internet. In addition, providing fast and reliable information transmission is one of the fundamental requirements to improve activities in a modern society that is heavily reliant on information. Enhancing communication quality with the Internet in a mobile environment is imperative for raising the current information society to a higher level. This dissertation consequently explores effective ways to provide universal, fast and reliable connectivity to the Internet in a mobile environment.

1.1 The Growth of the Mobile Internet Environment

Why has the Internet developed in such a rapid fashion? The primary factor is its design concepts. The *End-to-End* design principle[1], which has been adopted as the fundamental design principle of Internet architecture, contributes scalability with respect to the increasing number of connected nodes. The growth in the number of nodes in the Internet creates a desirable feedback loop for large-scale deployment; once the Internet is widely used, it becomes possible to use it for ever more purposes, thus spurring its further development. As a consequence, a wide variety of attractive applications and services have been developed on the Internet. These applications and services have different requirements for data transmission and can be classified by their data transmission demands. For bulk data transmission oriented services, such as file transmission and file sharing, the primary demands are high throughput and correctness of data. For interactive communication oriented services, such as remote login service and instant messaging, the primary demand is quickness of response. For real-time oriented services, such as multimedia streaming and VoIP, reducing jitter is a crucial demand rather than the correctness of data.

To provide solutions for such different communication demands with a unified communication infrastructure, the Internet protocol suite, also known as the *TCP/IP*, consists of four layers: the Link layer, the Internet layer, the Transport layer, and the Application layer. Protocols for each layer have their own role and must provide individual functions for data transmission, using features provided by lower layer protocols. Solutions for different kinds of communication demands can be provided by developing protocols and technologies concerned with the demands of each layer. This hierarchical structure is the key design principle of the Internet for achieving high scalability, extensibility, and general versatility.

The hierarchical structure of the TCP/IP model also provides interchangeability of the Link layer communication technology. As a consequence of successful deployment of the Internet, a demand for connection to the Internet anytime and anywhere naturally arises. Such a demand has motivated the development of various kinds of wireless communication technologies, which provide connectivity to the Internet from anywhere for network nodes, especially mobile nodes. Since the TCP/IP model does not depend on a particular data-link technology, it is easy to introduce wireless communication technology into its Link layer. With development and improvement of wireless communication technology, the mobile internet environment, in which mobile nodes can connect to the Internet anytime and anywhere, has become widespread.

Each wireless communication technology in current use has its own particular characteristics. For instance, consider the relative advantages of two wireless technologies; one is the IEEE 802.11 family, which is a set of wireless local area network (WLAN) standards, and the other is the IEEE 802.16 family, which is a set of wireless broadband standards.

An advantage of IEEE 802.11 is its fast data transmission rates. Data transmission for IEEE 802.11n, which is an amendment of the preceding IEEE 802.11 standard, achieves rates of up to 600 Mbps. On the other hand, an advantage of IEEE 802.16 is its wide communication range. Data transmission rates for IEEE 802.16 extend up to only 70 Mbps; however, its maximum communication range can be 50km. The communication range for IEEE 802.16 is significantly greater than for IEEE 802.11, which allows for a maximum communication range of just 180m.

Such complementary characteristics of wireless communication technologies leads to the idea that mobile nodes be equipped with multiple network interfaces which should then be employed depending on the situation to obtain both high speed communication and a wide communication range. In this dissertation, a network environment in which mobile nodes can use different kinds of wireless communication technologies to improve the quality of the communication is referred to as a *Heterogeneous Wireless Network*. In heterogeneous wireless networks, switching over the data-link currently used by one mobile node to another is referred to as *Vertical Handoff*. Mobile nodes will conduct vertical handoffs depending on their network situation to receive complementary benefits of individual wireless data-links. Seeking an appropriate utilization of heterogeneous wireless networks is one of the effective ways to improve the communication quality of the mobile Internet environment. This dissertation consequently seeks to determine how mobile nodes on heterogeneous wireless networks should operate in order to receive an optimal result from wireless communication technology.

1.2 Research Targets and Objectives

What is the primary factor determining *communication quality* in the mobile Internet environment? This depends strongly on the communication demands of those services and applications that are used by the mobile nodes. Since their communication demands are different, it is difficult to specify a universal factor. For instance, some services may require stable and sustainable connectivity to the Internet rather than operating with fast data transmission. Such services will not want to conduct vertical handoffs, even if a faster data-link becomes active since vertical handoffs may involve instantaneous disconnection from the Internet. On the other hand, some other services may require high-speed data transmission. In this case, they will want to conduct vertical handoffs when a fast data-link becomes active to obtain faster data transmission. Since vertical handoffs are a notable feature of heterogeneous wireless networks, the research target of this study is the latter case. This dissertation focuses on mobile nodes that use services with the following demands; (1) the primary requirement is fast data transmission, and (2) a secondary requirement is retaining connectivity with the Internet. One typical situation is where a mobile node uses modern web applications such as mapping services, on-demand video sharing services, and video streaming services. These applications contain images, videos, and audios, and provide rich user interfaces. Since these rich contents occupy a large bandwidth and require shorter transmission delays, obtaining fast data transmission is an important requirement for using these applications without discomfort. On the other hand, in situations where the mobile node cannot obtain enough bandwidth for using rich web applications without discomfort, retaining connectivity is still desirable. We also make the observation that rich web applications sometimes provide a simplified and qualified service for mobile nodes that are connected by thin data-links. In fact, many services want both high-speed data transmission and also stable connectivity, if this is at all possible. Therefore, mobile nodes can use rich web applications in an effective manner when they have both high-speed data transmission and universal connectivity to the Internet, which can be provided by an effective utilization of heterogeneous wireless networks.

Another possible research target is developing a specialized communication platform in which both high-speed data transmission and high availability are primary requirements. The *Mobile ER*, which is a communications platform for emergency medical care, is an example of such a system development. Chapter 2 will describe the Mobile ER platform in detail.

For heterogeneous wireless networks to be effectively utilized, they will need to provide both high-speed data transmission and a wide communication range. Unfortunately, however, the traditional TCP/IP framework lacks the capability of providing the full benefits of heterogeneous wireless networks. The following three characteristics are major issues for heterogeneous wireless networks.

Drastic changes in link characteristics In heterogeneous wireless networks, each wireless technology has individual link characteristics. Therefore, once a vertical handoff is conducted, the link characteristics of the upstream data-link may be drastically changed. Traditional TCP/IP does not cater for such drastic changes in link characteristics. Such drastic changes will affect the features of each layer, especially the application layer and the transport layer.

At the application layer, drastic changes in link characteristics produce the following two issues. *Service selection*: if a mobile node is running multiple applications and conducts a vertical handoff to a slower data-link, the mobile node may need to determine which application should continue to be running since it may be impossible to continue to run all applications. *Application adaptation*: some applications are sensitive to a change in link characteristics. Once the link characteristics change drastically, such applications may not provide appropriate services to mobile nodes. In order to adapt to drastic changes in link characteristics, applications may need to take some action, e.g., video streaming applications should adapt their bit transfer rate to the condition of the network path subsequent to a vertical handoff.

At the transport layer, *TCP adaptation* will be needed when link characteristics are subject to sudden drastic changes. TCP adaptation is the primary research objective of this dissertation and Chapter 3 will discuss the problem in detail.

Mobility support When mobile nodes conduct a vertical handoff, it usually involves a change of IP address. Since the Internet Protocol (IP) identifies end nodes by their IP addresses, correspondent nodes of a mobile node will become unable to find the mobile node.

Fragile link characteristics Wireless data-links have lossy and fragile link characteristics compared with wired data-links. Such link characteristics will break the assumption of TCP congestion control, which is the assumption that an occurrence of packet losses indicates network congestion. Therefore, the TCP congestion control mechanism may not work in an appropriate manner on wireless data-links.

All of these issues have to be addressed to achieve the ultimate goal of a study of effective utilization of heterogeneous wireless networks. As an important part of the study, this dissertation tackles an issue arising from drastic changes in link characteristics. One of the most significant impacts of such changes is harmful effects on the TCP congestion control mechanism. The TCP congestion control mechanism will be injured by such changes and the TCP may not adapt to the after-handoff data-link quickly. Since TCP is the dominating protocol for traffic on the Internet [2] and vertical handoffs are the key mechanism to improve the communication quality of mobile nodes, it is crucial to adapt

TCP to drastic changes in link characteristics. Therefore, this dissertation is mainly focused on developing methods for adapting TCP to drastic changes in link characteristics caused by vertical handoffs.

Mobility support is also a secondary research objective. There are several standardized mobility support protocols that have been developed, such as Mobile IPv6 and Network Mobility Basic Support. Therefore, most of this dissertation assumes that mobility support is provided by these protocols, except for the study described in Chapter 5. Chapter 5 proposes a multi-homing mechanism which provides not only transport layer adaptation features but also mobility support. In contrast with the above two issues, this dissertation does not tackle fragile link characteristics. Developing specialized technologies and protocols to cope with fragile and lossy link characteristics is a significant part of the study of heterogeneous wireless networks. With specific protocols for heterogeneous wireless networks, users of mobile nodes could receive communications of higher quality. However, since such specific protocols are specialized for a particular situation and technology, it is not always efficient for all possible situations that mobile nodes could face. Once a protocol or technology becomes dependent on another particular technology, it debilitates the scalability, extensibility, and versatility of the TCP/IP model. Therefore, this dissertation does not aim to develop protocols and technologies to address fragile link characteristics.

1.3 Criteria for TCP Adaptation to Heterogeneous Wireless Networks

As mentioned above, the primary purpose of this dissertation is to investigate how to adapt TCP for different link characteristics on heterogeneous wireless networks. Before considering concrete approaches to TCP adaptation, design criteria must be developed to achieve practical, efficient and integrated solutions for the research target. The following principles should be considered as criteria for developing TCP adaptation schemes on heterogeneous wireless networks.

- *Keeping the design concept of the Internet*. The rise of the today's Internet is due to its high scalability and extendability, which are derived from its design principles, in particular, the end-to-end principle. In addition, independence from any particular communications technology, which is another significant feature of the Internet gained from its hierarchical structure, also contributes to versatility and easy deployment. The Internet is still in the transition phases of growth. Therefore, respecting the end-to-end principle and independence from any particular communication technology is important in considering the further development and deployment of the Internet.

- *Ease of integration with other technologies*. There are many research issues in the study of effective utilization of heterogeneous wireless networks, such as mobility support, handoff strategies, fast handoff schemes, application adaptation, service selection and TCP adaptation. The ultimate goal of the study of heterogeneous wireless networks is to have all of these issues resolved in an integrated fashion. For ease of integrating solutions, developing technologies, protocols, and schemes for each issue should generally be isolated from the details of the particular functionality of other solutions. In other words, each solution should be moderately independent from other technologies and should not require other existing technologies to modify their internal mechanisms.

- *Fairness*. In the Internet, one of the important characteristics of end nodes is to share network resources fairly. Even if a scheme for TCP adaptation can provide brilliant performance, such as providing high throughput to an end node, it is harmful to other end nodes when it operates in a self-centered manner. Since TCP has the responsibility for congestion control and flow control, schemes for TCP adaptation should act conservatively. When there are N TCP connections in a network, each connection would occupy exactly $1/N$ of the total network resources in an idealized environment. However, this is difficult to achieve in a real environment since the situation for a real network changes dynamically. Therefore, this dissertation supposes that schemes for TCP adaptation would be considered fair when they treat the semantics and behavior of the TCP congestion control mechanism with respect and do not change its behavior drastically.

- *Applicability to a real environment*. In the engineering field, including studies of the Internet, practical realizations must be kept in mind in all research activities. Ultimately, all research contributions should be applicable to real environments. As an essential design criterion for developing schemes for TCP adaptation, this dissertation places emphasis on both theory and practice.

The study of this dissertation endeavors to follow these design criteria. Chapter 3 discusses methodologies for TCP adaptation and investigates which methodologies are appropriate, given the design criteria above.

1.4 Motivation, Approaches and Contributions

Although the necessity of TCP adaptation to heterogeneous wireless networks has been mentioned, we have not yet discussed the concrete motivation for adapting TCP to drastic changes in link characteristics. Chapter 3 presents the motivation in detail, however, this section briefly describes why this dissertation addresses TCP adaptation to heterogeneous wireless networks.

The motivation for adapting TCP to drastic changes in link characteristics can be summarized as follows.

- *Avoiding network congestion*. When a mobile node conducts a vertical handoff from a relatively fast data-link to a relatively slow data-link, the mobile node may transmit packets too aggressively because the TCP connections on the mobile node have adapted its transmission rate to the fast data-link to achieve a high throughput. This will cause network congestion on the slow data-link.

- *Improving transitional behavior on vertical handoffs*. When a mobile node conducts a vertical handoff, values of the TCP congestion control parameters will not be suitable for the after-handoff data-link because link characteristics will be drastically changed by the vertical handoff. This will cause short term performance degradation after the vertical handoff.

This dissertation takes two different approaches to developing TCP adaptation schemes. One is adapting TCP to drastic changes in link characteristics caused by vertical handoffs to address the above issues. Another is introducing a multi-homing mechanism to eliminate vertical handoffs. A brief summary of these approaches is as follows:

Approach 1: Adapting TCP to drastic changes in link characteristics When the link characteristics have changed drastically as the result of conducting a vertical handoff, the TCP congestion control mechanism should be adapted to the after-handoff data-link as soon as possible to prevent the occurrence of network congestion and lower utilization of the data-link capacity. For adapting the TCP congestion control mechanism to drastic changes in link characteristics, this dissertation presents a method that resets congestion control parameters after the occurrence of the vertical handoff. Although the strategy of the proposed method is quite simple, it is not trivial to determine when the congestion control parameters should be reset. Harmful side effects involved by the vertical handoff will interfere with resetting the parameters if the parameters are reset immediately after the occurrence of the vertical handoff. The proposed method seeks the point where such harmful side effects are no longer occurring, and then the method resets the congestion control parameters at this point.

Approach 2: Introducing a TCP-based multi-homing mechanism for seamless vertical handoff Multi-homing mechanisms provide a capability to use multiple data-links simultaneously for network nodes. In principle, if mobile nodes can use a multi-homing mechanism, vertical handoffs are no longer necessary to optimize the performance of heterogeneous wireless networks. Since multi-homing mechanisms, which are developed in the Internet layer aggregate multiple data-links into one Internet layer entity, they cannot manage Transport layer information separately for each data-link. This could be a serious drawback for heterogeneous wireless networks because data-links have significantly different characteristics and the Transport layer information, especially the TCP congestion control parameters, must be managed separately for each data-link. This dissertation presents a TCP tunneling based multi-homing mechanism for heterogeneous wireless networks. Although the mechanism also provides only one Internet layer entity at the upper layer, it can manage the TCP congestion control parameters separately for each data-link since the mechanism establishes TCP tunnels for each data-link. The proposed mechanism automatically selects the best data-link to transmit packets and transmits packets mostly through the selected data-link.

Figure 1.1 summarizes the positions of the research targets of this comprehensive study of heterogeneous wireless networks. There are many research targets to enhance the communication quality on heterogeneous wireless networks. As part of a comprehensive study of heterogeneous wireless networks, this dissertation mainly studies TCP related research objectives, such as adapting the TCP congestion control mechanism for vertical handoffs and introducing multi-homing mechanisms to eliminate the need of vertical handoffs.

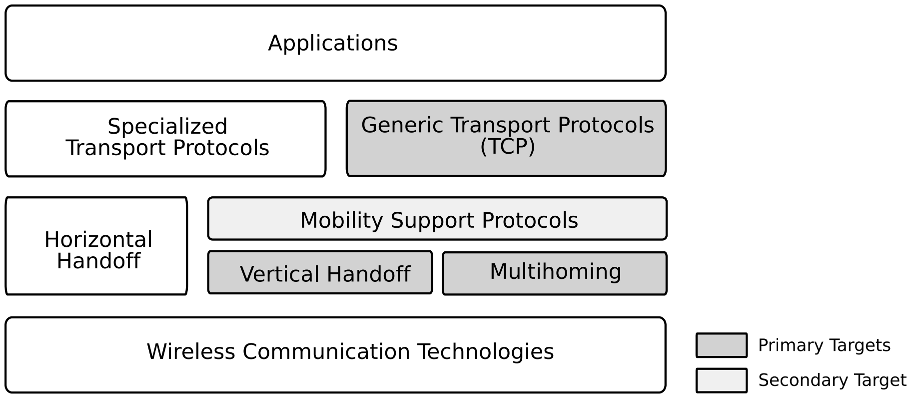


Figure 1.1: Research targets of this dissertation

1.5 Organization of this Dissertation

The rest of this dissertation is organized as follows. Chapter 2 describes the motivation for constructing heterogeneous wireless networks and presents a communication platform for emergency medical care, called Mobile ER, as an empirical application of the effective utilization of heterogeneous wireless networks. Adapting the TCP congestion control mechanism to drastic changes in link characteristics is the primary issue of this dissertation. Chapter 3 investigates the reasons that the TCP congestion control mechanism does not work as expected in heterogeneous wireless networks. The following Chapters 4 and 5 present different approaches which provide schemes for TCP adaptation to heterogeneous wireless networks. Chapter 6 discusses open issues and future work towards the comprehensive research goal of enhancing the mobile Internet environment. Finally, Chapter 7 concludes this dissertation.

Chapter 2

Heterogeneous Wireless Networks and Their Application

This Chapter reviews today's mobile internet environment and introduces the concept of *Heterogeneous Wireless Networks*. Several kinds of wireless networks can be used for connecting to the Internet in the current mobile environment. Their different link characteristics offer both high-speed data transmission and a wide communication range for mobile nodes. Mobile nodes can be equipped with multiple networks interfaces and can use them simultaneously to enhance communication quality. In this dissertation, a mobile environment in which mobile nodes have multiple interfaces to improve communication quality is referred to as a*Heterogeneous Wireless Network*. For heterogeneous wireless networks, mobility support is one of the primary requirements to identify mobile nodes. This Chapter briefly describes two standardized mobility support protocols.

Although developing systems for heterogeneous wireless networks is not the primary objective of this dissertation, it is important to demonstrate the effectiveness of heterogeneous wireless networks. A ubiquitous communication platform for emergency medical care, called *Mobile ER*, is described in this Chapter as an empirical example of systems developed on heterogeneous wireless networks.

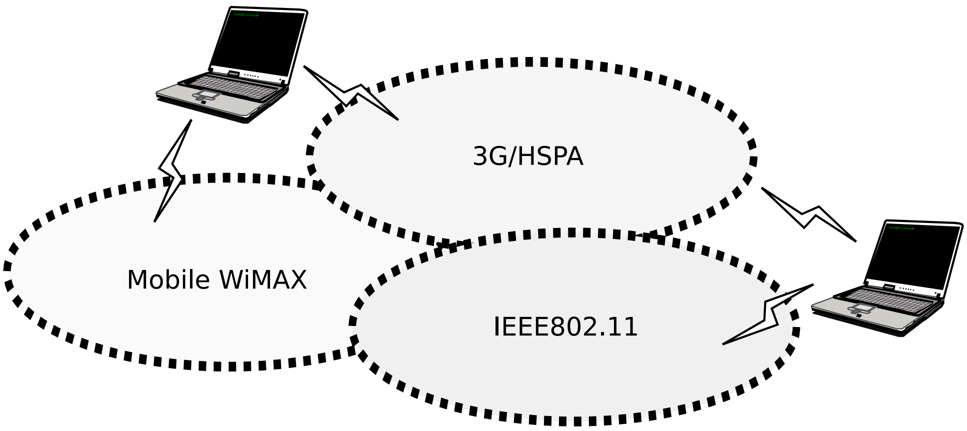


Figure 2.1: A conceptual example of a heterogeneous wireless network

2.1 The Motivation for Constructing Heterogeneous Wireless Networks

As a result of recent progress and deployment of wireless communication technologies, many different kinds of wireless data-links can be used to connect to the Internet in the current mobile environment. The complementary advantages and disadvantages of such wireless data-links allow a mobile node to switch its upstream data-link depending on the situation of the mobile node, or to employ these wireless data-links simultaneously. Throughout this dissertation, the term *heterogeneous wireless network* is used to refer to an environment which has multiple wireless interfaces available for a mobile node to connect the Internet through each wireless data-link and such that the mobile node can employ these wireless data-links separately or simultaneously to enhance the quality and availability of communication. A conceptual example of a heterogeneous wireless network is shown in Figure 2.1.

The aim of this section is to compare the relative advantages and disadvantages of wireless data-links which can be used at present to connect to the Internet and to describe the benefits of heterogeneous wireless networks and issues arising from their deployment. A typical example of the use of a heterogeneous wireless network is also given in this section.

2.1.1 Complementarity Relation of Wireless Technologies

In the last few decades, wireless communication technology has evolved at a rapid pace with regard to its communication speed, coverage, and cost efficiency. Since wireless technology has released mobile nodes from the limitation of a fixed network, it has been used extensively as a communications platform over various ranges of communication. Depending on the communication range, wireless networks, which utilize a particular wireless technology, can be roughly classified as follows.

A wireless network used for communication close to a person is referred to as a wireless personal area network (WPAN). WPAN is typically used for communication among computers and computer devices with network technologies such as IEEE802.15.1 (Bluetooth) and IEEE802.15.4 (ZigBee). Although these technologies could be used to connect to TCP/IP based networks including the Internet, the communication speed and range of these technologies are relatively restricted since their primary focus is on providing a means for linking devices wirelessly in a power saving manner.

A wireless local area network (WLAN) is a small-scale computer network in which each computer is connected by a particular wireless technology. A WLAN has relatively higher communication speeds than other wireless networks. The IEEE802.11 family, which is the de facto standard for WLAN, is operating at 600 Mbps currently. However, the availability of WLAN is relatively restricted. It is mainly used for constructing indoor local networks and there are only a few places where a WLAN is available outdoors.

A wireless metropolitan area network (WMAN) and a cellular network have similar characteristics. A WMAN is a large-scale wireless network that usually covers a city or large facilities. In general, the communication speed of a WMAN is relatively slow compared with a WLAN but the coverage of a WMAN is broader than that of a WLAN. The IEEE 802.16 (also known as WiMAX, an acronym for Worldwide Interoperability for Microwave Access) could offer over 90 Mbps with approximately 50km communication range to a stationary node. The IEEE 802.16e (also known as Mobile WiMAX) provides over 20 Mbps communication speed to a moving node. A cellular network generally has a little narrower bandwidth but a broader coverage. High Speed Packet Access (HSPA), which is an extended technology of third generation (3G) cellular networks, supports a downlink speed of up to 14 Mbps and an uplink speed of up to 5.76 Mbps with a nationwide communication range.

Table 2.1 summarizes the comparison of wireless technologies. A significant aspect

Table 2.1: Comparison of wireless technologies

Classification	Speed	Coverage	Technology
WPAN	slow	very restricted	`IEEE 802.15 family` (Bluetooth, Zigbee)
WLAN	fast	restricted	`IEEE 802.11 family` (WLAN)
WMAN	medium	medium-scale	`IEEE 802.16 family` (WiMAX, Mobile WiMAX)
Cellular network	slow-medium	national-scale	3G, HSPA

of these technologies is their complementary relationship with regard to communication speed and coverage. This relationship leads to the concept of a mobile node being equipped with multiple network interfaces and using them to obtain optimized communication, depending on the situation of the mobile node. As mentioned above, such a communications environment is referred to as a heterogeneous wireless network. In heterogeneous wireless networks, a mobile node can select different strategies depending upon what is required, for example, high-speed communication or sustained connectivity to the Internet, and it can switch between technologies, depending on its situation. The mobile node can also use these technologies simultaneously in heterogeneous wireless networks. This dissertation focuses on the benefits and issues arising from the deployment of such heterogeneous wireless networks.

A typical situation where a mobile node can get a benefit of heterogeneous wireless networks, in other words, the mobile node can achieve both fast communication speed and wide communication range, is as follows. Suppose the mobile node has two interfaces such as WLAN and HSPA. Since the communication range of HSPA is nationwide, it can be expected that HSPA is available everywhere. When both WLAN and HSPA are available, the mobile node will choose WLAN as its upstream data-link because it has relatively high bandwidth. Once the mobile node moves away from the coverage area of WLAN, the mobile node will switch its upstream data-link from WLAN to HSPA to try to keep connectivity to the Internet following the assumption that HSPA could be available even if WLAN is no longer available. In this manner, the mobile node could obtain both high-speed communication where this is possible and also a wide communication range.

2.1.2 Vertical Handoff

In heterogeneous wireless networks, switching over the data-link from one network to another is referred to as a *vertical handoff*. The *before-handoff data-link* is the data-link which is used as the upstream data-link prior to the vertical handoff. Similarly, the *after-handoff data-link* is the data-link which is used as the upstream data-link after a vertical handoff.

Conducting a vertical handoff is a key function of heterogeneous wireless networks. However, the generic Internet Protocol (IP) cannot support vertical handoffs in a natural way. As the following section 2.2 will describe, once a vertical handoff is conducted, the correspondent nodes cannot locate the mobile node so they cannot communicate with it. Therefore, mobility support protocols, which ensure that correspondent nodes can identify and communicate with a mobile node, is crucial for heterogeneous wireless networks.

Although mobility support is essential to obtain the full benefit of heterogeneous wireless networks, it is a secondary issue in this dissertation since there do exist several mobility support protocols, which are also described in section 2.2, that have been standardized by IETF. Most of this dissertation (except for Chapter 5) assumes that mobile nodes are mobility-supported by such mobility protocols unless otherwise noted. In the next section however, we do discuss some aspects of mobility support.

2.2 Mobility Support Protocols

As we have noted in the previous section, heterogeneous wireless networks offer both high-speed communication and a wide communication range. However, generic IP does not support the operation of heterogeneous wireless networks in a natural way. One of the essential problems is lack of mobility support. IP uses an IP address to identify the destination node and IP packets are forwarded by routers, based on the destination IP address. In heterogeneous wireless networks, however, the IP address of a mobile node is not fixed since a mobile node is likely to conduct vertical handoffs, which result in a change of IP address. Two major problems arise when the IP address of the mobile node changes. Firstly, the transport layer or higher layer connections will be disconnected. If the mobile node changes its IP address while it is transmitting some information, this information will be lost. Secondly, the correspondent nodes cannot locate the mobile node at that point, because correspondent nodes can only recognize the mobile node by its IP address. A correspondent node cannot connect to the mobile node while the mobile node is being notified of its new IP address.

The Internet Engineering Task Force (IETF) developed and standardized *mobility support protocols*, to prevent the above problems: Mobile IPv6 [3] for a single mobile node and Network Mobility (NEMO) Basic Support [4] for a mobile network.

Mobile IPv6 Mobile IPv6 (MIPv6) provides mobility transparency for a single mobile node. Mobility transparency is the property that guarantees reachability and connectivity for a mobile node. The basic concept of MIPv6 is to utilize a permanent IP address, which is called a Home Address (HoA) provided by a Home Agent (HA). An HA is placed in the home network of the mobile node and acts as an intermediate node between the mobile node and correspondent nodes.

Figure 2.2 shows an overview of how MIPv6 works. The mobile node sends a binding update (BU) message to HA to specify the whereabouts of the mobile node. The BU message includes a Care of Address (CoA), which is provided by the access router on the visited network, and an HoA for the mobile node. When the HA receives the BU message, the HA creates or updates a binding cache entry that associates the HoA with the CoA and then creates a bidirectional tunnel between the mobile node and the HA. All packets are sent from the mobile node to the correspondent node through this tunnel. Because the HoA is used for the source addresses of these packets, the mobile node and the correspondent node can communicate with each other even though the CoA has changed.

MIPv6 supports route optimization in which the mobile node communicates with a correspondent node directory. However, for simplicity, the route optimization mechanism is not described in this dissertation because route optimization is independent of network mobility basic support, which is one of our key protocols.

NEMO Basic Support A mobile node could contain a network within itself, called a mobile network, when the mobile node contains some communication components, such as a vehicle [5]. Network Mobility (NEMO) Basic Supports are designed as extensions to MIPv6 to provide mobility transparency for mobile networks. With NEMO, a mobile node is called a mobile router (MR) since the mobile node acts as the gateway of the mobile network to provide connectivity to the Internet for the mobile network nodes (MNNs). The BU message in NEMO provides additional information, including a mobile network prefix (MNP), which is a network prefix of mobile networks advertised from the

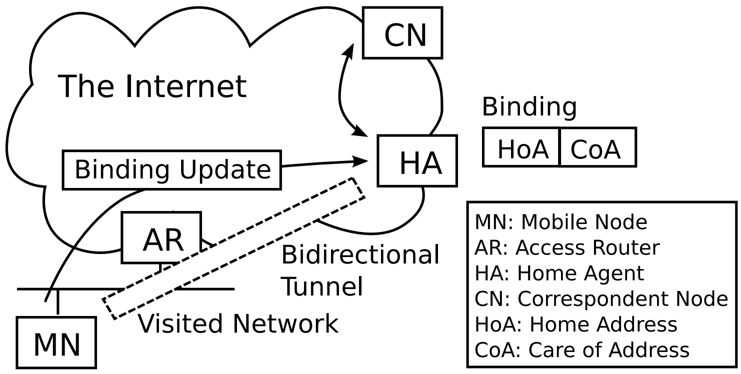


Figure 2.2: Conceptual architecture of Mobile IPv6

MR to the MNNs. The HA maintains a relationship between MR's HoA and the MNP, so each packet sent from the correspondent node to an MNN is appropriately forwarded by the HA and arrives at the MNN through the bidirectional tunnel.

2.3 An Empirical Example: The Mobile ER Platform

This section presents an empirical study of the utilization of a heterogeneous wireless network that is a communication platform for emergency medical care. Although the primary objective of this dissertation is not to develop systems or applications on heterogeneous wireless networks, it is an important part of this study to show the reasons for studying heterogeneous wireless networks and what kind of tasks they can be used for effectively.

The Ikoma119 Project [6] has been developing a communication platform for emergency medical care, called *Mobile ER*, since 2005. The platform uses the Internet as its communication infrastructure and an ambulance for communicating with other nodes, such as a doctor's PC and mobile phone, through heterogeneous wireless networks. Since the NEMO basic support protocol is used for providing mobility support to an ambulance, doctors and emergency medical technicians (EMTs) can always communicate with each other, irrespective of the location of the ambulance.

2.3.1 Motivation and Background

In emergency medical care, information about patients and their conditions greatly influences the quality of medical treatment. In particular, to enable the best treatment, the information has to be timely, accurate and complete. In current information transmission systems for emergency medical care, an exclusive line is commonly used to transmit medical information to doctors from ambulances. For example, information acquired from medical equipment is transmitted by cellular phones. However, information transmission systems that are currently used in emergency medical care have some drawbacks. Firstly, supporting diverse information sources is difficult. Appropriate medical treatment requires exchanging such diverse information as a patient's clinical history, vital signs, and images of the affected areas. However, most services that are currently used to transmit medical information require individual exclusive lines for each piece of information. To support new medical information, we need more exclusive lines, but this increases the system cost. Secondly, information and its transmission lines are strongly bound. Since most current medical information transmitting services do not provide a function that switches transmission lines, the service will not be available when the corresponding transmission line is out of service. Thirdly, existing information services assume fixed point-to-point communication. For example, if a service uses digital cellular access as its transmission line, transmitting information to multiple destinations is difficult. Consider the problem of determining the most suitable hospital for a patient; in this situation, it is desirable that the patient's condition can be transmitted to all candidate hospitals, but this is difficult when the service uses fixed point-to-point communication lines, thus fixed point-to-point communication is inadequate for emergency medical care. The service ideally should provide data transmission to multiple destinations.

The essential reason for the drawbacks above is the lack of a way to integrate both information sources and communication lines. A communication platform is needed for emergency medical care that solves the above problems. In 2005, we started a joint project to develop an efficient communication platform for emergency medical care. This project is called *Ikoma119 project*[6], which is affiliated with the Nara Institute of Science and Technology and the Nara Prefectural Nara Hospital. The Ikoma119 project has developed communication platforms between doctors and EMTs for emergency medical care called *Mobile ER*, which is based on wireless internet technology. Our platform is approaching practical use. Three demonstration experiments have already been conducted. This section describes the design and the current implementation of the Mobile ER platform.

Services developed on the Mobile ER platform are also described in this section.

2.3.2 Design Requirements

For achieving desirable properties of extensibility, efficiency, availability, and robustness, the following requirements should be satisfied by the communication platform:

- *Flexibility to support diverse information*: There are diverse data formats exchanged between a doctor and an EMT: images, audio, video streams, text data, electrocardiogram (ECG) data, and a patient's clinical history. Since these data formats have different requirements for transmission, appropriate transmission methods depend on these requirements. Some formats (e.g., textual data) must be transmitted exactly, whereas other formats (e.g., video streams) must be transmitted quickly but can suffer a limited amount of data loss. For example, textual data or a document file could have a certain file format and it could be broken if the data is not transmitted exactly. However, video streams are typically tolerant of some data loss but they should be transmitted in real time. The communication platform must have flexible transmission methods to support appropriate data transmission for each data format.

- *Universal connectivity*: Since ambulances travel anywhere, the platform should provide connectivity from any place at any time. In particular, some important information such as ECG data might be useless if the platform lacks connectivity because these data represent patient's instantaneous conditions. When such instantaneous data cannot be delivered to doctors on time, they cannot give appropriate medical directions. Other information, which does not have to arrive immediately, should still be transmitted as soon as possible to provide the most up to date information to doctors.

- *Location independent service*: In general, a doctor is typically in a hospital and EMT typically takes place in an ambulance. However, this is not always the case; both sometimes might be located elsewhere. A doctor could be outside a hospital when an emergency call is made and the doctor often is unable to go to an operation terminal immediately. Likewise, an EMT could be in the neighborhood of an ambulance at an earlier stage of emergency medical care. Therefore, the platform must have the capability to transmit information anywhere to enhance the quality of emergency medical care.

- *Device and data link independent services*: Most emergency medical information systems currently in use depend on specific devices and data-links. For such information systems, devices and data-links are inseparable from the information transmission service, which is a potential drawback. When an ambulance leaves the service area of the data-link, a service using that data link will no longer be available. In addition, since wireless communication technology is constantly advancing, a data-link currently being used may soon become obsolete. This means that a data link may well become inappropriate over time. If a service and its data-link are inseparable, introducing a new medical service could well prove very expensive and/or require a vast amount of bandwidth at some future time. By avoiding such dependence between a service and its data-links, the platform can resolve these difficulties.

- *Easy installation of new services*: In an ambulance, data is generated by many different sources such as types of medical equipment, and Global Positioning System (GPS) devices. The platform should provide a common and easy way to establish a new service allowing these devices to communicate.

- *Dependability*: Dependability is a crucial property for the platform. Wireless data-links often lack connectivity and stability; however, the platform should provide reasonable service availability whenever this is possible. The impact of a lack of connectivity must be minimized and maximum possible stability of the data-link assured.

- *Security*: To a greater or lesser degree, any information pertaining to emergency medical care is related to a patient. Such data is considered private, and therefore the platform must protect this private data, especially when the platform acts on it or transmits it.

2.3.3 Design

Based on the above requirements, a communication platform for emergency medical care, called Mobile ER, has been designed. It employs wireless internet technology for the

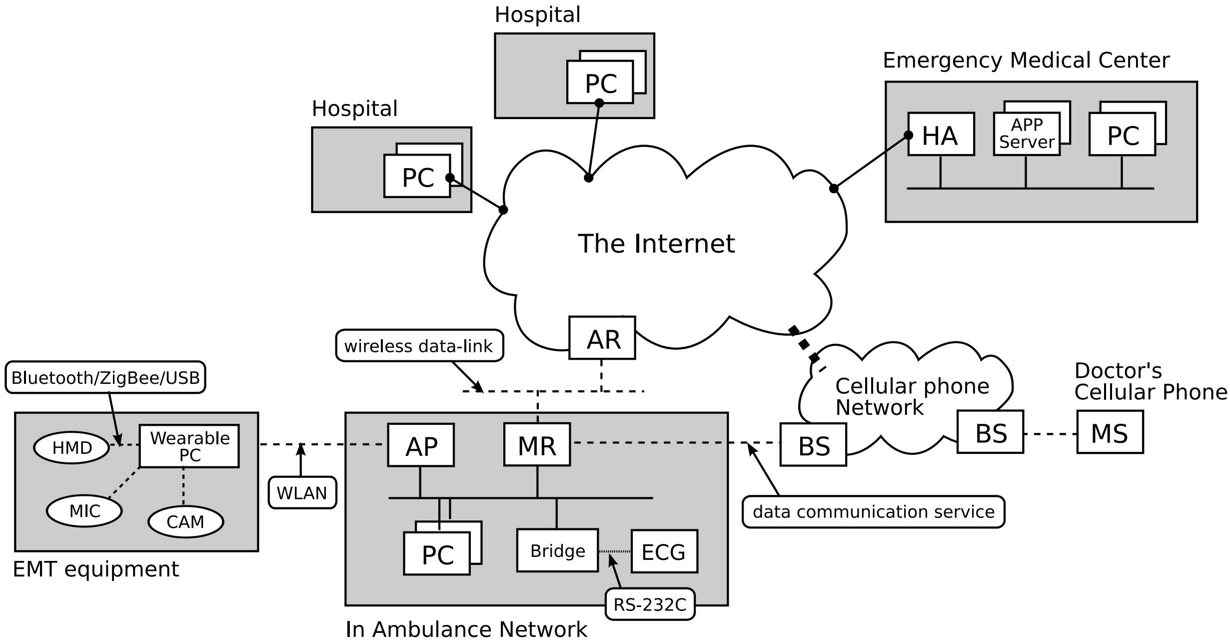


Figure 2.3: Conceptual architecture of the Mobile ER Platform

communication framework and wearable computing technology for the EMT equipment. This section provides an overview of the platform design and then describes it in detail.

Design Overview

Figure 2.3 illustrates the conceptual architecture of the Mobile ER platform. The platform is divided into four main components: the emergency medical center network, the in-ambulance network, the EMT equipment, and service clients. An emergency medical center is positioned by the organization that has responsibility for controlling ambulances, which, in Japan, is the fire defense division of local government units. In the platform, the organization must place a network; the network must contain an HA to provide mobility transparency for an in-ambulance network, and also should contain application servers to enhance the quality of services provided by the platform. An in-ambulance network is a network which is located in an ambulance and connects several devices and PCs within the ambulance. These devices and PCs could be components of certain services provided by the platform. To provide connectivity to the Internet, an MR is also connected to the in-ambulance network. EMT equipment consists of several wearable computing devices, such as a wearable PC, an HMD, a microphone, and a webcam. An EMT puts on the EMT equipment to communicate with doctors. Service clients are the clients of services provided by the platform. Since the platform is internet based, various kinds of devices which can connect to the Internet can be used as service clients. Typical such service clients include doctors ' PCs in hospitals, but may also, for example, include doctors ' cellular phones when the doctors are located outside of a hospital.

Each component of the platform is connected with the others through the Internet directly or indirectly. EMT equipment is connected to the in-ambulance network and is connected with others through it. Since an in-ambulance network is provided with mobility transparency by NEMO, all components can always communicate with each other regardless of the ambulance's location. This means that the platform can provide location-independent service as long as the ambulance retains connectivity to the Internet.

In the platform, services are deployed as internet applications and the platform has flexibility in developing services, which can assume various architectures because each component of the platform can communicate with each other.

Communication Technology and Services

As mentioned above, the platform employs the Internet, which is based on TCP/IP protocol suites, as its communication infrastructure, and services on the platform are therefore deployed as TCP/IP applications. Internet technology has appropriate properties to satisfy most of the requirements described in 2.3.2, with a few exceptions. In this section, we will describe appropriate properties first, and then describe adverse properties of the internet technology with regard to the requirements.

The first advantage of the internet technology is that it is open to the public, widely used, and therefore relatively inexpensive. Since many applications and libraries have been developed for the Internet, they can be applied to developing a new service on our platform. In addition, service clients do not need to construct extra devices and software. Most TCP/IP applications, especially web-based applications, will work with almost all service clients. In fact, even small mobile devices, such as cell phones, can browse web pages, and therefore they can be used as service clients for web-based applications. This advantage makes it possible to realize device services independently. Furthermore, commonly used methods of developing TCP/IP applications can be applied to developing emerging services. When it is decided that the platform requires a new service, it can simply be independently developed as a standard TCP/IP application.

The second advantage is that the TCP/IP protocol suite has flexibility with respect to replacement of individual technologies. The TCP/IP protocol suite is divided into four layers and has a hierarchical structure. Since there is a minimal relationship between layers, a protocol or technology in a particular layer can be easily switched. As an example, consider the physical layer. A data-link currently used to connect to the Internet can switch to a new data link without affecting the other layers. Since the change in the data-link does not affect services running on the platform, the platform can provide data-link independent services. Next, consider the transport layer. The TCP/IP protocol suite has different types of transport protocols such as TCP and UDP. A service can select an appropriate transport protocol depending on the characteristics of the information sent by the service. For example, a video streaming service will use RTP (Real-time Transport Protocol)[7] with UDP, while a text-based service will use TCP.

In contrast with the advantages above, internet technology is not that appropriate with regard to a few requirements at present. In particular, dependability is not entirely guaranteed in the Internet. However, the Internet is becoming more reliable over time in general and it is unlikely that the Internet would cease to function. Security is also an important consideration. Since our platform employs the Internet as a communication infrastructure, it must protect data privacy. The platform provides encrypted data transmission following the NEMO specifications; all traffic sent through the bidirectional tunnel between MR and HA is encrypted by IPSec [8]. However, a data communication between an HA and a CN is not encrypted under normal conditions. Therefore the service must use a security technology such as TLS [9].

Emergency Medical Center Network

As shown in Fig. 2.3, the HA, application servers, and PCs are located in the emergency medical center network, which has two roles: (1) as the home network of the MR in the ambulance and (2) as an intermediate service network.

The HA, which is the home agent of the ambulance, collaborates with the MR in the ambulance to provide mobility transparency to both the MR and the MNNs. The HA maintains the ambulance's current location. When the HA receives a packet intended for the MR or the MNNs in the ambulance, the HA forwards it to the MR.

Application servers are optional components; they may or may not be located in the network, depending on the demands of the service provided by the platform and their roles reflect these demands. Typically, application servers act as proxy nodes. A proxy server improves service availability and reliability. The connectivity and stability of the ambulance may deteriorate due to its movement. The service may not be available when the ambulance lacks connectivity by not having a proxy server. In contrast, when a proxy server is available, once the proxy server retrieves the information from the ambulance, the application server rather than the source device can provide information to clients. There are many other ways to use application servers, for example, they can act as cache servers, data conversion servers, and relay servers.

The In-Ambulance Network

Ambulances have diverse devices, sensors, and computers that have to be connected to the Internet to provide various data to clients as functions of services. However, these devices may not have enough processing capacity to individually support mobility transparency. The platform has therefore introduced an MR to the ambulance. The MR forms a network and provides mobility transparency to these devices with NEMO. The MR has multiple wireless interfaces and these interfaces can be selected to connect to the Internet depending on the situation because it has mobility transparency provided by NEMO.

Since some medical equipment does not have a network interface, the equipment cannot be connected directly to the network, however, fortunately, such equipment may have other interfaces such as RS-232C and USB for which one can construct bridge devices to connect them to the network. A bridge device has both a network interface and other input/output interfaces and it can work as a bridge between the device and the network.

An EMT often works outside of the ambulance; typically, he or she will be in the neighborhood of an accident scene to relieve the sufferers. To provide connectivity for an EMT who works outside the ambulance, an AP(Access Point) is located in the ambulance network. The AP acts as a bridge and provides wireless connectivity to the in-ambulance network for EMTs.

EMT Equipment

The main component of the EMT equipment is a wearable computer that provides several functions for an EMT. To provide connectivity to the Internet, the wearable computer has a WLAN interface and connects to the AP in the ambulance network. The wearable computer also has such interfaces as Bluetooth, ZigBee, and USB that are used to connect input/output devices. An HMD, a microphone (MIC), and camera (CAM) devices are connected by these interfaces that can be used for interaction with doctors.

Service Clients

The role of service clients are depends on the particular service which is being used by doctors or other medical staff, and they may require communication with any components of the platform. For example, a client of an ECG data transmission service will receive ECG data from an ECG device in an ambulance and will display the received ECG data graphically. One can also develop a group chatting service, so that clients can communicate with each other, to provide the function of remote consultation.

Since services on the platform are provided through the Internet, service clients can be anywhere as long as they have connectivity to the Internet and the client has the proper authority to use the service. In addition, no specific device is needed to use the services. Thus the platform provides a location/device independent service. A service client could be a commonly used PC in a hospital, a doctor's cellular phone, or a communication terminal in an emergency medical center.

2.3.4 Current Implementation and Services

Most of the components of the platform are replaceable because the platform was designed as a device/data-link independent platform. Therefore, the latest devices, data-links and technologies could be used as components of the platform. The main specifications of the current implementation are listed in Table 2.2. MIPv6 and NEMO are supported by SHISA [10], which is a MIPv6/NEMO implementation running on several BSD-based OSs developed by the WIDE project [11]. The MR can use the following wireless data links: IEEE802.11b/g, b-mobile [12], and emobile [13]. When the MR is in a WLAN coverage area, it uses a WLAN as the data link. Once the MR moves outside a WLAN coverage area, the MR switches its data link to b-mobile or emobile.

Figure 2.4 shows the EMT equipment. The EMT wears goggles with a built-in camera that always follows the direction of the EMT's eyes. The HMD, which is mounted on the goggles, displays various information to the EMT.

Table 2.2: Current implementation specifications

Mobile Router (MR)	
OS	NetBSD2.0 + SHISA
HW	custom made
Network IF	IEEE802.11b/g, b-mobile, e-mobile
Home Agent (HA)	
OS	FreeBSD 5.4-RELEASE + SHISA
HW	Dell PowerEdge
Wearable Computer	
OS	Windows XP SP2
HW	Sony Vaio type U
Network IF	IEEE802.11b/g
I/O IF	USB, Bluetooth
HMD	Micro Optical SV-6
Camera	custom made
Microphone	Bluetooth MIC

Mobile ER Services

Services on the platform can be developed as standard TCP/IP applications. At present, the following three services have been developed on the platform: a still image transmission service, an ECG information transmission service, and a video interaction service. This subsection describes their design and implementation. These services are related because they share common communication platforms and are based on extensively used Internet technology. In fact, the video interaction service uses the ECG information transmission service as a part of the service.

Still Image Transmission Service At an early stage of the project, the project developed a still image transmission service that periodically transmits still images of the EMT's viewpoint to medical institutions. This service works as follows:

- The wearable computer takes an image with the camera embedded in the goggles at regular (15 seconds) intervals and sends it to the image processing PC in the ambulance by WLAN.

Figure 2.4: EMT equipment

Figure 2.5: Screenshot of still image transmission service

- After receiving an image, the PC generates and stores a thumbnail image as well as the original image. The PC also notifies the application server in the emergency medical center network that a new image has arrived.

- The application server does not immediately acquire images from the PC after being notified; instead, the server makes a request message which is a message that indicates the server should acquire the thumbnail image from the PC later. Then, the server appends the request message to the request queue, which holds request messages created previously, to acquire the thumbnail image triggered by the notification. This behavior is designed with the bandwidth limitation of the ambulance's data link in mind.

- If a request message is found in the request queue, the application server opens it and acquires the image based on the message.
- A doctor can view these images on her PC by web browser from the application server.

Figure 2.5 shows a screenshot of the web interface of this service. The web interface lists the previously taken thumbnail images. If the doctor would like to scrutinize the image, she can request the server to acquire the original. When the server receives the request, it immediately sends the original image if the server has already acquired it; otherwise, the server acquires the original image from the image processing PC in the ambulance and sends it to the doctor's PC.

ECG Information Transmission Service One of the most helpful pieces of information in emergency medical care is an ECG. This service distributes ECG data to doctors. There are three desirable properties for distributing ECG data: immediacy, security, and availability.

An ECG describes a patient's momentary condition. If the service takes an overly long time to transmit the data, it may become meaningless. Therefore the data has to be received by doctors as soon as possible. In addition, the ECG data must be encrypted because it is considered personal. To obtain immediate and secure information transmission, the service uses the Secure Real-time Transport Protocol (SRTP) [14] as the transport protocol.

To enhance availability, the service sites the proxy server in the emergency medical center network. The proxy server is the sole client that receives the ECG data from the ambulance and it acts as a proxy server for other clients. Other clients (e.g., a doctor's PC) are connected to the proxy server to receive the ECG data.

The proxy server provides many ways to transmit the ECG data. It can transmit the data using SRTP, TCP, or HTTP to make it possible to develop different kinds of clients. In fact, some different kinds of client have been developed: a standalone application using SRTP, a Java applet using TCP and NTT DoCoMo's DoJa[15] application, using HTTP. This feature also increases the service's availability. Even if a doctor is outside the hospital, she can view the ECG information using the DoJa application on her cellular phone.

Figure 2.6: ECG equipment and bridge device

Because the ECG equipment currently in use does not have a network interface, we prepared a bridge device to connect the ambulance network. Fig. 2.6 shows the ECG equipment and the bridge device, which also acts as a SRTP server to transmit ECG data to the proxy server.

Video Interaction Service We have also developed a video interaction service that supports the cooperative operation of emergency medical activities. This service was developed with Adobe's Flash Video (FLV) technology for easy installation. Most modern web browsers support Adobe's Flash and FLV as a plug-in component, so service users only have to prepare a web browser and the Flash plug-in component for PCs to use the service.

The following describes typical processing flows of this service. The streaming server is in the emergency medical center network. Both a doctor and an EMT publish the live stream with a camera and headset to the streaming server, respectively. They subsequently connect to the streaming server as subscribers to get other live streams. Accordingly, the doctor and the EMT can communicate with video and voice. These live streams have some few, but non-negligible delays due to encoding and decoding videos. At the present time the delays are approximately less than one second. These delays somewhat complicate the communication between the doctor and the EMT, but this service is still useful; for example, the doctor can observe a patient's condition.

(a) Screenshot of doctor-side interface (b) Doctor using service

Figure 2.7: Video interaction service

Figure 2.7 shows an example of a doctor using the video interaction service. The left-hand image shows the doctor-side web-based interface. There are video streams published by an EMT and ECG information. A doctor can indicate to the EMT what she would like to check precisely. The still image, located on the right-hand side of the video stream, shows what the doctor is analyzing. The still image also displays the HMD of the EMT. The ECG information is provided by the ECG data transmission service described above.

2.4 Summary

This section has described the concept of heterogeneous wireless networks, together with benefits and issues that arise with their deployment. An empirical example of utilization of a heterogeneous wireless network, namely, a communication platform for emergency medical care, was also presented in this section. In heterogeneous wireless networks, a mobile node can utilize both high-speed communication and a wide communication range with an appropriate strategy for utilization of different kinds of wireless technology. To allow clients to receive the full benefits of heterogeneous wireless networks, mobility support is necessary. Fortunately, there are several technologies for providing mobility support that have been studied and some of these have been standardized by IETF. Therefore, this study focuses on the transport layer adaptation for mobile nodes with mobility support on heterogeneous wireless networks. The next Chapter will describe the process in TCP when a vertical handoff is conducted and why transport layer adaptation is crucial for effective utilization of heterogeneous wireless networks.

Chapter 3

TCP and Heterogeneous Wireless Networks

The Internet Protocol (IP) [16, 17] essentially provides an end-to-end packet transmission service in a "best-effort" manner, in which packets are not always delivered to the destination end node. TCP (Transmission Control Protocol) [18], which is a transport protocol established upon the IP, provides reliable, connection-oriented, data-stream communication to an application. To provide reliable data transmission to applications, TCP has several features such as error detection, retransmission control, flow control, and congestion control. Unfortunately, these features sometimes do not work effectively when an end node is within a wireless network because these features have been developed and improved in the context of a wired fixed network. As implied by the fact that dominant Internet-based services, such as WWW, email and FTP, employ TCP as their transport protocol, TCP is the fundamental transport protocol on the Internet. Therefore, it is crucial to adapt these TCP features to wireless networks to enhance the quality of Internet-based services in a mobile environment. This Chapter briefly describes the TCP features that offer reliable data transmission to applications and shows why they do not work as expected in wireless networks. Most of the illustrations of TCP features throughout this section are based on TCP NewReno[19] because it is used extensively and it is the basis of the following improved versions of TCP.

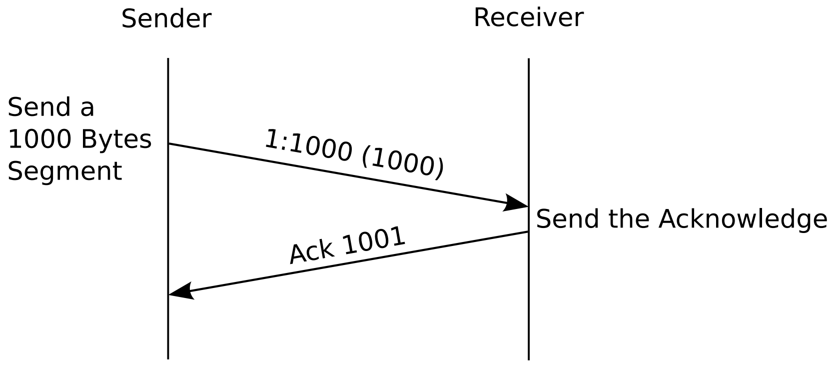
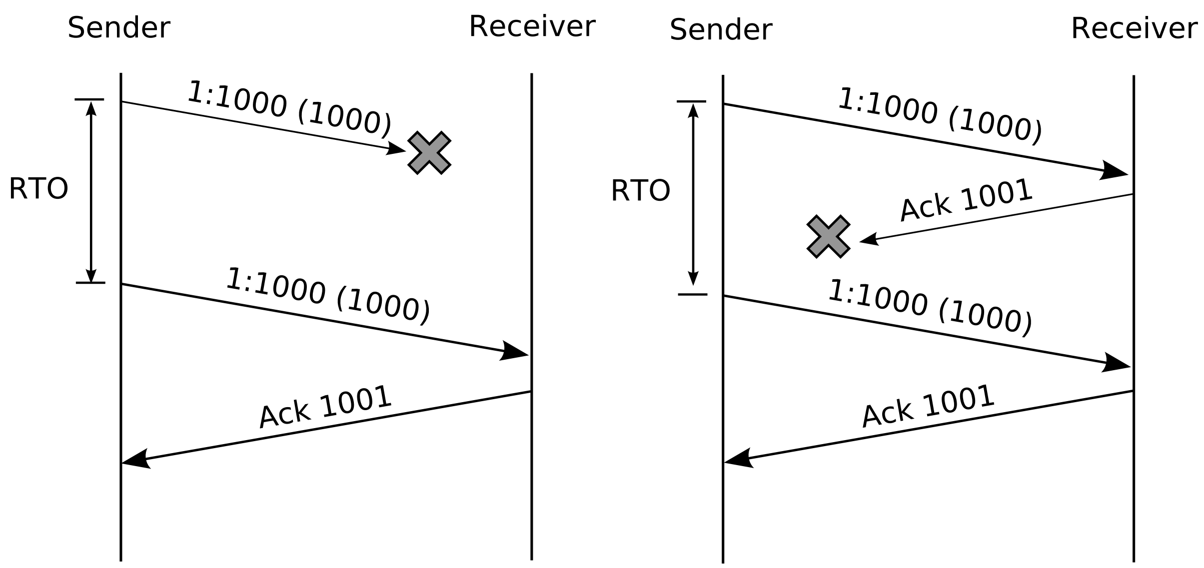


Figure 3.1: TCP segment transmission/acknowledgement sequence

3.1 TCP Retransmission Control Mechanism

As noted above, packets transmitted by IP are not always delivered to the destination node. To provide reliable data transmission, TCP has to detect an occurrence of *segment loss* and has to retransmit lost segments. A *segment* is a chunk of data and the unit of data exchanged between end nodes using TCP. When TCP transmits a segment, a receiver acknowledges reception of the segment by sending the *acknowledgement segment*. An acknowledgement segment indicates the position in the stream of the data that the receiver expects to receive the next time. The position is referred to as an *acknowledgement number (ACK)*. Figure 3.1 shows the interaction of TCP segment exchange. A TCP segment contains the *segment number* that indicates where the segment is located in the stream of data, and a *segment length* that indicates the length of the segment. Throughout this dissertation, the notation "n:m" is used for representing an $m - n + 1$ byte segment with the sequence number n. In Figure 3.1, a sender sends the segment 1:1000; then a receiver acknowledges with ACK 1001.

TCP uses acknowledgement segments to detect an occurrence of segment loss. Once TCP assumes that segments may have been lost, it retransmits these segments. TCP provides three segment loss detection methods: a timeout based method, a duplicate acknowledgement based method, and a selective acknowledgement based method. The following subsections describe these three methods.



(a) A case of a send segment lost

(b) A case of an acknowledgement lost

Figure 3.2: Example of retransmission caused by timeout

3.1.1 Retransmission with Timeout

When a sender sends a segment, the sender sets a *retransmission timeout (RTO)* and maintains a *retransmission timer* for waiting for an acknowledgement from a receiver. If the segment has not been acknowledged when the RTO expires, the sender assumes that the segment has been lost and retransmits it. Figure 3.2 shows two possible situations where an RTO expires. In the case of Figure 3.2(a), a sent segment has been lost, whereas in the case of Figure 3.2(b), although a sent segment arrives at the receiver, the acknowledgement is lost. In both cases, an RTO expires and a segment retransmission is invoked. When an acknowledgement is lost, a receiver receives the same segment twice due to segment retransmission. Even if a receiver receives duplicate segments, TCP still works correctly because a receiver always acknowledges reception of a segment with an acknowledgement number that indicates what the receiver expects to receive next time, even if the segment is a duplicate.

The measurement of the round-trip time (RTT) on a TCP connection is fundamental to the timeout-based retransmission. An RTO of a TCP connection should be calculated by using its RTT: the RTO must be greater than the RTT to avoid unnecessary segment retransmission and should be sufficiently small to detect segment loss as soon as possible.

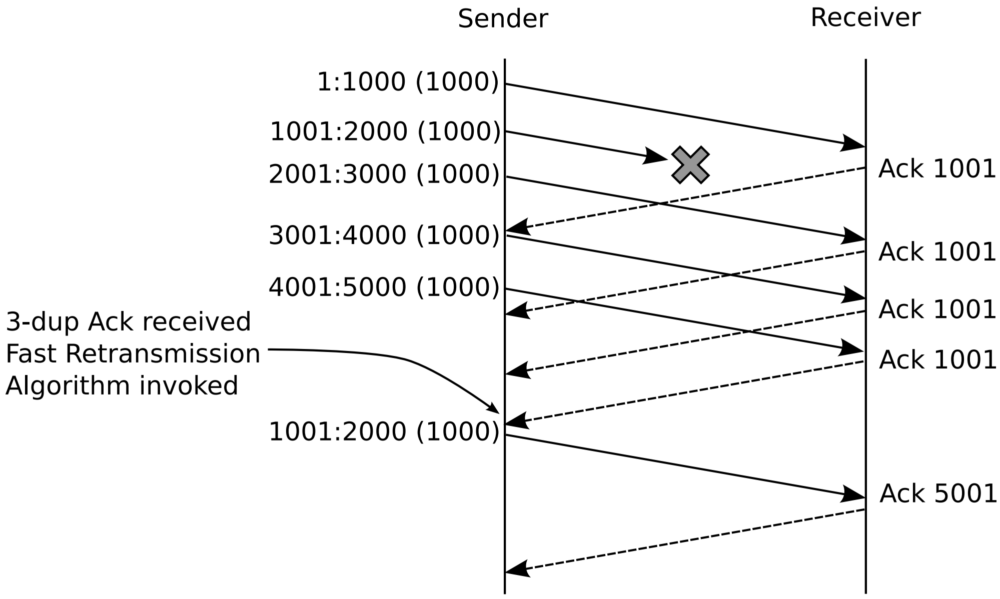


Figure 3.3: Example of fast retransmission

TCP dynamically measures the RTT on a TCP connection and modifies its RTO accordingly. From each RTT measurement rtt, TCP derives an RTO for a TCP connection by following Jacobson's equations [20].

$$delta = rtt - srtt \tag{3.1}$$

$$srtt \leftarrow srtt + g \times delta \tag{3.2}$$

$$rttvar \leftarrow rttvar + h(|delta| - rttvar) \tag{3.3}$$

$$RTO = srtt + 4 \times rttvar \tag{3.4}$$

where $srtt$ is the smoothed RTT and $rttvar$ is the smoothed mean deviation. $Delta$ is the difference between the measured value just obtained and the current RTT estimator ($srtt$). Both g and h are smoothing coefficients and many TCP implementations set $g = 0.125$ and $h = 0.25$, respectively.

3.1.2 Fast Retransmission Algorithm

An ACK always indicates the sequence number which is the position in the data flow that a receiver expects to receive next time. This behavior can be used for detecting segment loss. When a sender receives duplicate ACKs, which contains a sequence number that

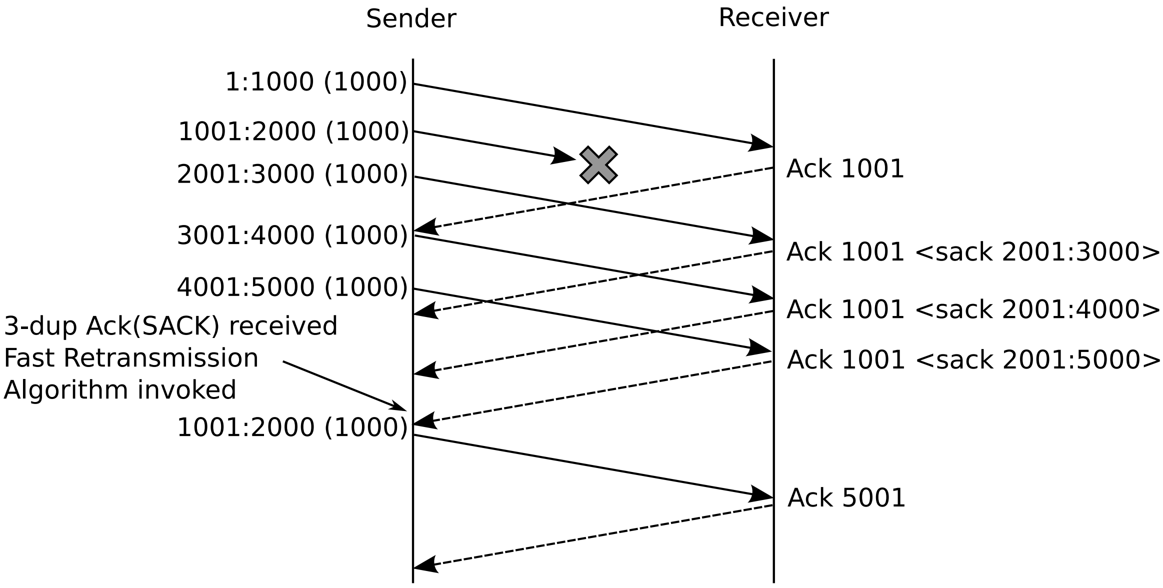


Figure 3.4: Example of fast retransmission with SACK

has already been sent by a sender, this implies an occurrence of segment loss. These duplicate ACKs are useful for detecting segment loss and TCP uses duplicate ACKs for this purpose. However, TCP segments can arrive out of order since TCP segments are transmitted as IP packets, and such *segment reordering* also involves duplicate ACKs. Avoiding false detection of segment loss and unnecessary retransmission, TCP retransmits segments when a sender has received three duplicate ACKs [18]. This retransmission mechanism is called the *fast retransmission algorithm* because this mechanism can detect segment loss more quickly than timeout-based mechanisms in general. Figure 3.3 explains the behavior of the fast retransmission algorithm. In Figure 3.3, since the segment 1001:2000 is lost, a receiver acknowledges with ACK 1001 for three successive segments because a receiver expects to receive a segment with the sequence number 1001 and has received segments that are out of order. At the point in time that a sender receives three duplicate ACKs, the sender assumes that the segment 1001:2000 has been lost and retransmits it.

3.1.3 Fast Retransmission Algorithm with SACK

In Figure 3.3, a receiver only indicates that it has received the segment 1:1000 by sending ACK 1001; even though it has received successive segments it is still expecting the segment 1001:2000. Therefore, a sender cannot determine whether successive sent segments have arrived at a receiver until the sender receives an acknowledgement without duplication.

Selective acknowledgement (SACK) [21, 22] lets a receiver inform a sender about successive segments that have successfully arrived. With SACK, an acknowledgement contains extended acknowledgement information, which indicates that non-contiguous blocks of data have been received, when segment loss has occurred. An example of SACK is shown in Figure 3.4. In this example, each selective acknowledgement indicates a block of data that has arrived at a receiver. Therefore, a sender can determine that it need retransmit only the lost segment, which was 1001:2000, by the information provided by SACKs.

A sender does not retransmit the lost segment immediately when it receives a single selective acknowledgement, in the same way that the fast retransmission algorithm does not. RFC 3517 defines a variable *DupThresh* that holds the number of duplicate acknowledgements required to trigger a retransmission [22]. RFC 2581 [18] currently sets this threshold at 3, but RFC 3517 suggests SACK implementations should consult any updates to determine the current value for *DupThresh*.

3.2 TCP Congestion and Flow Control

TCP controls the rate of segment transmission to support *congestion control* and *flow control*, which are required features for reliable communication between two end nodes in the Internet. The purposes of these features are to avoid network congestion and to avoid overflow of the receiver-side receive buffer, respectively. TCP uses the *sliding window* mechanism to seek a suitable rate of segment transmission to achieve such a result with high throughput. The mechanism defines the *window size*, which is the number of segments a sender can transmit at the same time without waiting for acknowledgements. A sender adjusts the window size dynamically depending on the context of the network and the receiver.

Figure 3.5 shows how the sliding window mechanism works. For convenience of illustration, assume that the size of segments here is 1 byte. In Figure 3.5, the *offered window*, which is advertised by a receiver, has a window size of 6. Since segments 1 through 3 are sent and acknowledged, the window has moved 3 segments and covers segments 4 through 9. A sender has sent segment 4 through 6 but they have not been acknowledged yet. Consequently, a sender computes the *usable window*, which is how

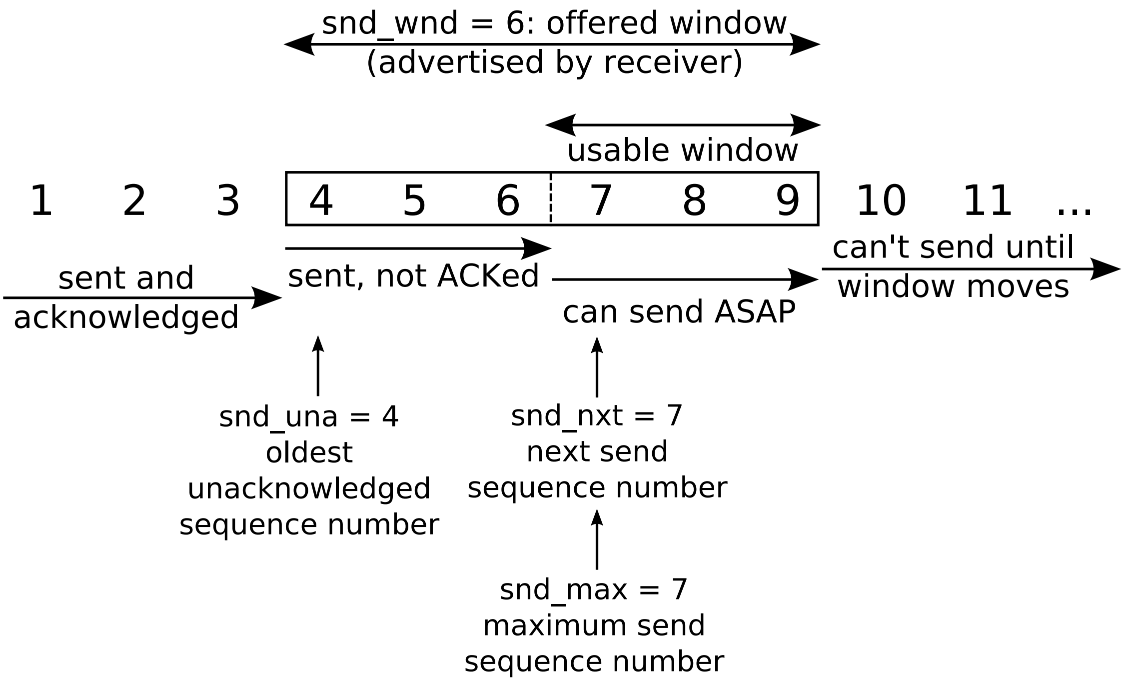


Figure 3.5: Example of sliding window*

*quoted from TCP/IP Illustrated Vol.2 [23]

many segments a sender can send without waiting for acknowledgements, to be 3.

TCP's congestion control and flow control are implemented in a manner that maintains respective windows and adjusts their sizes. For flow control, the offered window is used because it indicates how many segments a receiver can receive at one time. For congestion control, another window is used: the *congestion window*, called *cwnd*. A sender can transmit up to the minimum of the values of the offered window and the congestion window at the same time. Whereas the offered window is advertised by a receiver and the sender just uses it, the congestion window must be maintained by the sender. The following subsection describes how TCP obtains a suitable value for the size of the congestion window.

3.2.1 Adjusting the Congestion Window

TCP uses two strategies to increase the size of the congestion window.

- Initially, the size is increased quickly. In this phase, the congestion window is increased by one *maximum segment size (MSS)* each time an ACK is received. This phase called the *slow-start phase*.

- When the congestion window size reaches a certain threshold, called the *slow-start threshold (ssthresh)*, TCP reduces the rate of increasing the congestion window size to avoid network congestion. In this phase, the congestion window is increased by $MSS^2/cwnd$ each time an ACK is received. This phase called the *congestion avoidance phase*.

TCP uses both of these strategies based on whether *cwnd* is less than *ssthresh* or not. When a new TCP connection is established, the *ssthresh* is initialized at the maximum window size which TCP can handle, that is 65535 bytes, without the window scale option. TCP updates the *ssthresh* when it detects segment loss as well as the *cwnd* since TCP assumes that an occurrence of segment loss implies some network congestion. When TCP detects an occurrence of segment loss, TCP estimates that the current value of *cwnd* is too aggressive and tries to determine suitable values for *cwnd* and *ssthresh*.

When TCP detects segment loss, it estimates that a suitable value of the *cwnd* lies between the current value and one-half of it. Seeking to establish a larger value of *cwnd* to achieve high throughput and to avoid congestion, *ssthresh* is set to one-half of *cwnd* and the sender moves to the congestion avoidance phase.

At the same time as updating *ssthresh*, *cwnd* is also decreased to reduce the transmission rate of segments. How to calculate a new value for *cwnd* depends on how a sender detects the segment loss. If a sender detects the segment loss by the timeout-based method, *cwnd* is set to be 1 MSS. On the other hand, if a sender detects the segment loss by the fast retransmission algorithm, *cwnd* is set to be either $\min(ssthresh, FlightSize + 1MSS)$ or *ssthresh*. A TCP implementation can select either of these; however, RFC 3982 warns that implementations should avoid a transmission burst of segments after receiving acknowledgement of a retransmitted segment.

Although these assumptions and behavior have been appropriate and adequate for fixed wired networks, the assumption is not appropriate for heterogeneous wireless networks, as the following section 3.3 will argue.

3.3 Pitfalls of TCP for Heterogeneous Wireless Networks

Segment retransmission methods and the congestion control mechanism of traditional TCP have been developed and improved in fixed wired networks, in which optical fibers and copper cables are used as its data-links. Since these TCP features have been optimized for wired data-links, they work effectively in fixed networks. However, such TCP features, especially the congestion control mechanism, have less affinity to the mobile internet environment because wireless networking technologies, which are used as data-links in mobile networking environments, have different characteristics from wired data-links. This section discusses what makes traditional TCP features less appropriate for wireless data-links.

3.3.1 Characteristics of Wireless Data-links

A major difference between wired and wireless data-links is the quality of data transmission. Bit errors hardly ever occur in fixed wired networks. Therefore, a segment loss would be triggered only by dropping the segment on a router as the result of network congestion. This fact bears out the approach of a TCP congestion control mechanism being triggered by a data loss.

In contrast, a bit error is likely to occur in wireless networks. The quality of wireless data-links is vulnerable to the surrounding conditions and the movement of end nodes. Since bit errors involve dropping segments on a destination node as the result of error correction, segment loss is likely to occur even if wireless networks have sufficient capacity for the current network traffic. Therefore, it is hard to say that the assumption of TCP congestion control, in which a segment loss implies network congestion, fits in well with wireless data-links. TCP may decrease *cwnd* and *ssthresh* unnecessarily when it detects a segment loss. This behavior leads to less effective utilization of wireless data-links.

3.3.2 Connectivity Loss

A mobility-supported mobile node can keep TCP connections even if it moves over different wireless networks. However, a mobile node often loses connectivity to the Internet temporarily in the following situations:

- A mobile node moves around and leaves coverage of all available wireless data-links. During the period a mobile node is out of the coverage area, connectivity is totally lost.

- A mobile node conducts a horizontal handoff. Although a horizontal handoff is conducted quickly relative to a vertical handoff, it involves a slight connectivity loss.

- A mobile node conducts a vertical handoff. Conducting a vertical handoff may take a long time because it requires several processes, such as detecting data-link availability/unavailability, switching data-links, and binding information about the data-link the mobile node is currently using. A mobile node cannot transmit any segments until these processes have completely finished.

Lack of connectivity combined with mobility support has harmful effects on TCP performance. Although a mobile node cannot send any packets to the Internet since it is not connected, TCP is not aware of this because mobility support hides it. Consequently, TCP continues to send segments and this leads to several segment losses and retransmission timeouts of segments. However, as mentioned in section 3.2 concerning triggers for reducing segment transmission rates, such segment loss and retransmission timeouts do not necessarily imply any network congestion and reducing transmission rates would not be appropriate.

3.3.3 Changes in Link Characteristics

In heterogeneous wireless networks, the data-link characteristics of a mobile node, such as bandwidth, delay, and packet loss rate, sometimes change suddenly. Conducting vertical handoffs could be a major cause of such changes because characteristics of the before-handoff data-link and the after-handoff data-link are often significantly different. TCP connections on a mobility-supported mobile node cannot be aware of changes of link characteristics due to a vertical handoff since mobility support protocols hide occurrences of vertical handoffs.

Another cause of sudden changes in link characteristics is *rate adaptation* on IEEE802.11-based wireless networks. Rate adaptation is a link-layer mechanism of IEEE802.11-based wireless technology, which adapts the transmission rate at the physical layer level dynamically, depending on the link quality of a mobile node. Once rate adaptation is adopted, the link characteristics of a mobile node, especially the data transmission speed, might change suddenly. Since rate adaptation is an internal feature of IEEE802.11-based wireless technology, TCP connections on mobile nodes cannot be aware of such changes.

These sudden changes in link characteristics will lead to a mismatch of TCP congestion control parameters and thus TCP congestion control will not work as expected. In particular, vertical handoffs can change link characteristics drastically. Therefore, this dissertation mainly focuses on drastic changes in link characteristics. The following section 3.4 investigates the impact of drastic changes in link characteristics on TCP congestion control.

3.4 Mismatch of TCP Congestion Control Parameters

The TCP congestion control mechanism assumes that link characteristics do not change drastically in a short period of time. However, a vertical handoff will cause such drastic changes, which will have negative impacts on the TCP congestion control mechanism. By conducting exploratory experiments, this section examines what occurs with the TCP congestion control mechanism when a vertical handoff ensues.

3.4.1 Exploratory Experiments

Exploratory experiments are conducted in an emulation environment, which is shown in Figure 3.6. It contains two data-links and these data-link characteristics are configurable. Dummynet [24] boxes, which enforce bandwidth limitations, delay, and packet losses, are placed on each network path and they emulate a certain kind of wireless data-link. Software used in the exploratory experiments is shown in Table 3.1. The kernel software of the server is modified to track congestion control parameters. The modified kernel dumps congestion control parameters to log files for each TCP segment ' s transmission and reception. Tcpdump [25], which is a packet monitoring software package, is also used in the experiments for tracking the sequence numbers of transmitted segments.

In the following experiments, the server transmits a 1Mbyte file to the client using HTTP. During file transmission, the routers switch over the route of segment transmission from one to another to emulate vertical handoffs. The following experiments are conducted to track the congestion control parameters, specifically *cwnd*, *ssthresh* and *RTO*. In all graphs in this section, the horizontal axis is the number of received segments (they are ACKs of transmitted segments by the server) on the server and the vertical axis is a

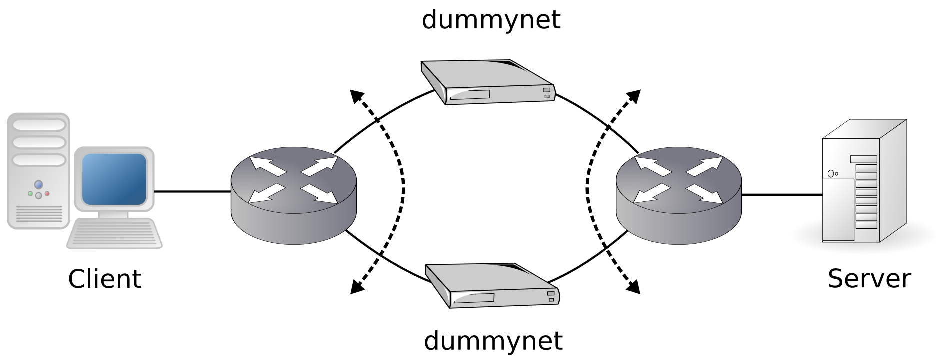


Figure 3.6: Exploratory experiments environment

Table 3.1: Software used in exploratory experiments

Node	Item	Description
Server	OS	NetBSD 3.1 (Modified)
	TCP version	TCP NewReno (SACK enabled)
	HTTP server	lighttpd-1.4.11 (Modified)
Client	OS	NetBSD 3.1
	TCP version	TCP NewReno (SACK enabled)
	HTTP client	Wget 1.10.2
Routers	OS	NetBSD-CURRENT (2006.12.02 build)
dummynet	OS	FreeBSD 6.2-PRERELEASE

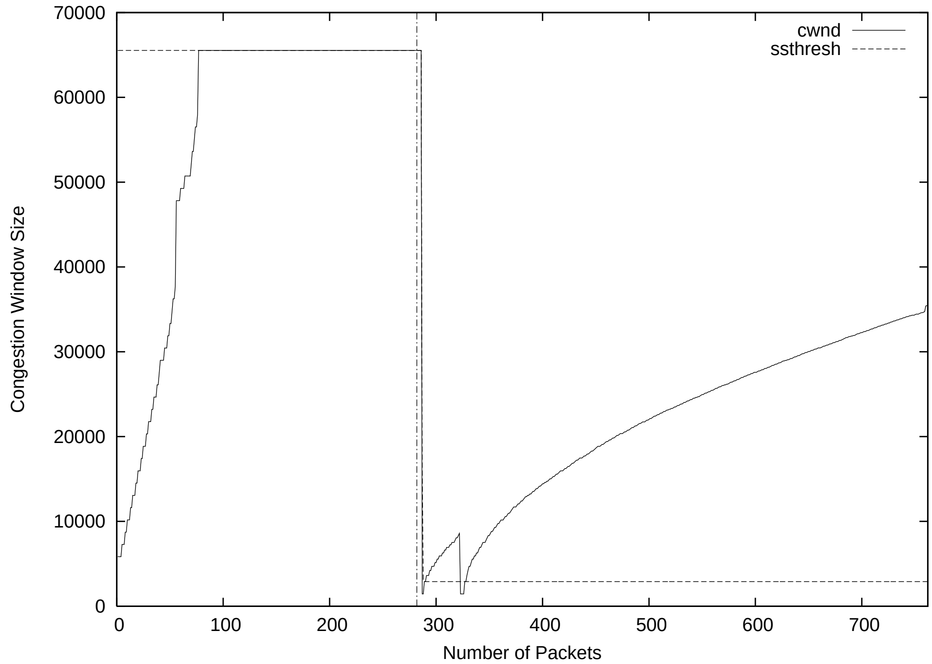


Figure 3.7: *Cwnd* mismatch with a drastic bandwidth change

byte scale (both *cwnd* and *ssthresh* are maintained in a byte scale by NetBSD). In addition, the vertical dashed line in each graph indicates when the route switches over from one router to another.

Congestion Window

Cwnd is a fundamental parameter of the TCP congestion control because it determines how many segments a sender can transmit simultaneously. In general, when the current data-link has a high bandwidth, *cwnd* grows larger in order to yield a high throughput. Therefore, *cwnd* can be too large when a mobile node switches over its data-link from a relatively high bandwidth data-link to a relatively low bandwidth data-link. With an overestimated value for *cwnd*, TCP will send segments aggressively and would trigger network congestion. This behavior detracts from the reliability of TCP since it causes performance degradation and interferes with the transmission of other traffic on the data-link that has been moved to.

Figure 3.7 plots values of *cwnd* and *ssthresh* for each received segment on the server during a file transmission with a drastic bandwidth change. In this experiment, bandwidths of paths are limited to 11Mbps and 64 Kbps respectively, and the routers switch over from 11 Mbps to 64 Kbps at the point of the vertical dashed line on the graph. The plots show that *cwnd* is reduced to 1MSS after the bandwidth change. This behavior is caused by congestion control involving a retransmission timeout. The retransmission timeout is triggered by network congestion, which arises from too much transmission of segments with an oversized *cwnd*.

Slow Start Threshold

Since *ssthresh* is updated for each congestion control invocation, TCP adjusts *ssthresh* to the current data-link characteristics over long transmission of segments. In other words, *ssthresh* is going to decrease if the data-link has a low bandwidth, while it is going to increase if the data-link has a high bandwidth over time. Therefore, a mismatch of *ssthresh* will be caused by a drastic change in link characteristics and it produces harmful effects on congestion control.

Consider the situation when a mobile node conducts a vertical handoff from a relatively low bandwidth data-link to a relatively high bandwidth data-link. Following the low bandwidth link, *ssthresh* will be small at the point before the vertical handoff is conducted. After the vertical handoff is conducted, in spite of the bandwidth having increased, *sshtresh* will retain the same value because there is no opportunity to update the value. Such a value is too low for the after-handoff data-link. The inappropriately low value for *ssthresh* holds TCP in the congestion avoidance phase, in which TCP increases the rate of segment transmission slowly, despite the after-handoff data-link still having enough bandwidth to transmit much heavier traffic. Figure 3.8 shows an example of such a situation. In this case, the bandwidth changes from 64 Kbps to 11 Mbps. As Figure 3.8 shows, once *ssthresh* is decreased, it requires considerable segment transmission to make *cwnd* larger, even if the data-link has excess capacity.

Next, consider the opposite situation where a mobile node conducts a vertical handoff from a relatively high bandwidth data-link to a relatively low bandwidth data-link. Figure 3.9 shows the result of an experiment which is conducted under the same conditions as Figure 3.7. In Figure 3.9, at the point the vertical handoff is conducted, *ssthresh* is reduced by half as the result of a segment retransmission caused by an RTO expire, the same as for Figure 3.7. However, *ssthresh* is still too large for the after-handoff data-link, as implied by decreasing *ssthresh* and *cwnd* once again due to network congestion, which is caused by aggressive segment transmission with an overestimated value for *ssthresh*.

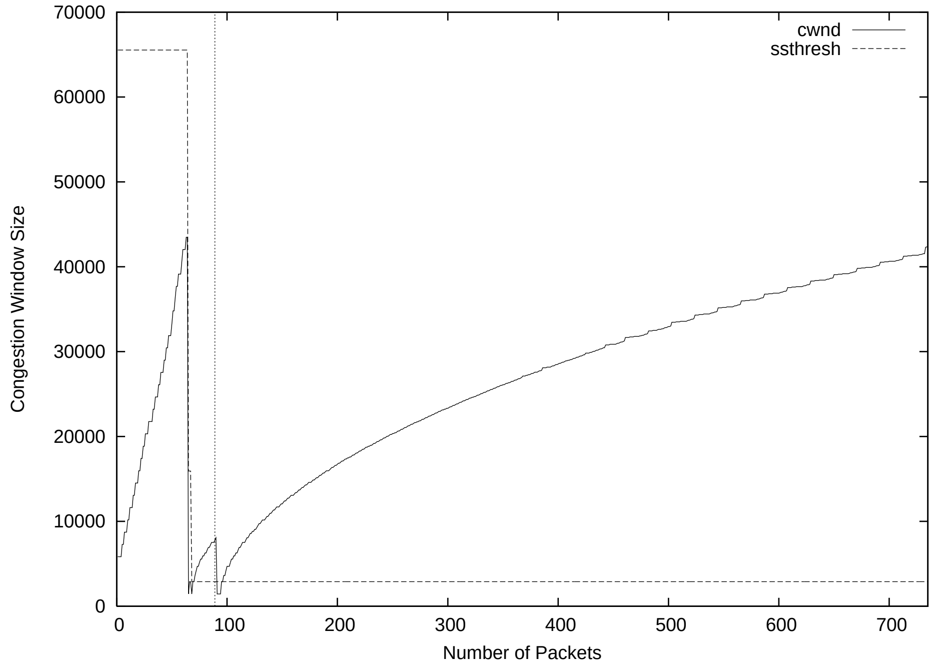


Figure 3.8: *Ssthresh* mismatch with a drastic bandwidth change (1)

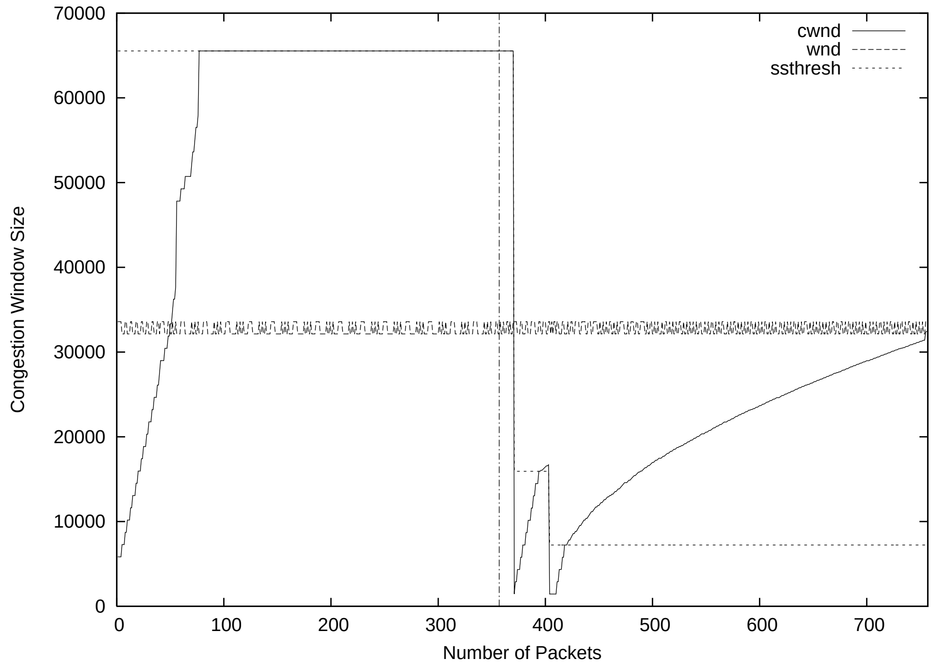


Figure 3.9: *Ssthresh* mismatch with a drastic bandwidth change (2)

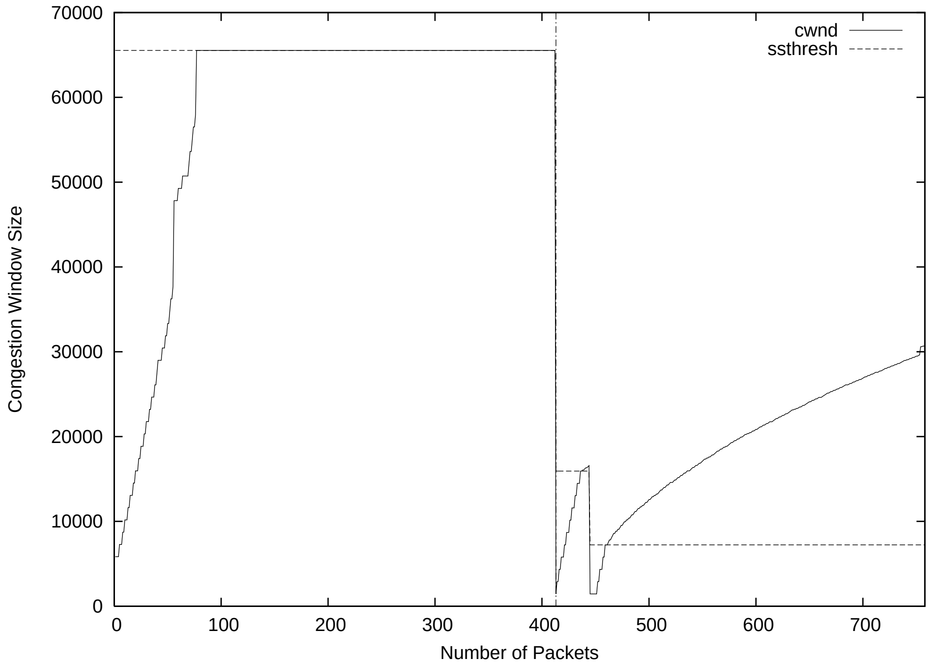


Figure 3.10: Needless congestion control with a drastic delay change

As mentioned above, cross traffic on the after-handoff data-link will suffer from such network congestion.

Retransmission Timeout

As described in section 3.1, when an RTO expires, in other words, a sent segment is not acknowledged in a certain period of time, TCP assumes that the segment is lost as a result of network congestion, and TCP invokes congestion control to adapt the segment transmission rate. Since the method for RTO estimation is based on the measured RTT of segments, RTO will be mismatched when a vertical handoff occurs in which the transmission speed is changed drastically.

Consider a situation where a mobile node conducts a vertical handoff from a relatively fast data-link to a relatively slow data-link. Since the transmission speed of the before-handoff data-link is fast, RTO will be underestimated for the after-handoff data-link, which has a relatively slow transmission speed, just after the vertical handoff is conducted. Consequently, an RTO may expire even if segments are successfully received and acknowledged. Such *spurious timeouts*[26] trigger unnecessary segment retransmis-

Table 3.2: Summary of exploratory experiments

Relation	Impact on TCP
$cwnd \gg optimal\ value$	causes network congestion
$ssthresh \gg optimal\ value$	causes network congestion
$ssthresh \ll optimal\ value$	less utilization of data-link capacity
$RTO \ll optimal\ value$	causes spurious segment retransmissions and needless congestion control

sion and degradation of the transmission rate. Figure 3.10 shows an example of such a situation. At the point indicated by the vertical dashed line on the graph, the transmission speed is changed from 10 ms to 2000 ms. In this case, a spurious timeout occurred when the transmission speed was changed, and consequently *cwnd* and *ssthresh* are decreased by unnecessary congestion control.

3.4.2 Findings from the Experiments

The previous subsection described several situations in which TCP congestion control parameters are mismatched as the result of drastic changes in link characteristics and such mismatches have harmful effects on TCP congestion control. Table 3.2 summarizes the findings of exploratory experiments described in that subsection.

As shown in Table 3.2, mismatches of TCP congestion control parameters cause network congestion, less utilization of link capacity, spurious segment retransmissions, and needless congestion control. On an actual TCP communication, these harmful effects would be triggered simultaneously by a compound mismatch of parameters caused by drastic changes in link characteristics. Since such drastic changes in link characteristics occur often as a result of conducting vertical handoffs in heterogeneous wireless networks, comprehensive solutions for such problems are needed to adapt the TCP congestion control mechanism for use in heterogeneous wireless networks.

3.5 TCP Adaptation Methodologies

The previous section described how drastic changes in link characteristics, which will be incurred by vertical handoffs, can have harmful effects on the TCP congestion control mechanism. The research objective of this dissertation is to develop schemes that can reduce such harmful effects by adapting TCP to different characteristics of data-links on heterogeneous wireless networks. There are some development methodologies for the purpose and several schemes have been developed along with each methodology. This section describes such methodologies and existing schemes.

3.5.1 Classification of the Methodologies

Methodologies for developing TCP adaptation schemes on heterogeneous wireless networks can be classified roughly into the following three types.

Developing a wireless network friendly version of TCP As has been explained in section 3.3.1, the traditional TCP congestion control mechanism may not work as expected for wireless data-links. The assumption of the traditional TCP congestion control mechanism, in which an occurrence of packet loss indicates an occurrence of network congestion, is not appropriate for wireless data-links. One radical solution for this problem has been to develop wireless network friendly version of TCP. Developing a specialized TCP for wireless networks aims at conducting appropriate congestion control and achieving high throughput on wireless networks, in which the link characteristics are fragile and sometimes change suddenly. Since these specialized TCPs have tolerance for changes in link characteristics, they can be a solution for the problem involved by vertical handoffs.

There are two approaches to developing such specialized TCPs. One is the end-to-end approach. This approach changes the algorithm and behavior of the TCP congestion control mechanism itself on end nodes. TCP Westwood [27], WTCP [28], and TCP-J [29] take this approach. The other approach is to place an intermediate node on the network path between a mobile node and a correspondent node. The intermediate node is typically placed on the edge of a wired network and supports communication between the mobile node and the correspondent node in some way. For example, the intermediate node may divide TCP connections into two types, namely connections to nodes in the wired network and connections to nodes in the wireless network, and adapts connections of each type to their link characteristics by using different congestion control algorithms. M-TCP [30] and I-TCP [31] take this approach.

Enhancing existing TCP adaptability to changes in link characteristics A more conservative approach to adapting TCP for drastic changes in link characteristics is to make minor modifications of the traditional congestion control mechanism. Adjusting the internal state of congestion control, for example by adjusting the congestion control parameters after an occurrence of a vertical handoff is one of the possible solutions using this approach. The bandwidth-aware method [32] takes this approach.

Using existing TCP features is also a possible way of adapting TCP to drastic changes in link characteristics with minor modifications of the behavior of TCP. Some solutions have been studied that take this approach. The TCP time stamp option [26] and the zero window advertisement [33] are examples of this approach.

Using multiple data-links simultaneously Using multiple data-links to connect to the Internet simultaneously is commonly referred to as *multi-homing*. Introducing multi-homing to mobile nodes, which are then called *multi-homed mobile nodes*[34], can be a solution for adapting TCP to drastic changes in link characteristics caused by vertical handoffs since they can receive the benefit of vertical handoffs without actually conducting them. There are several ways to provide multi-homing to mobile nodes. They can be classified by which layer provides the functionality of multi-homing. At the Application layer, ARMS [35] and a socket level aggregation mechanism [36] have been proposed. At the Transport layer, TCP based mechanisms [37, 38, 39, 40], Stream Control Transmission Protocol (SCTP) based mechanisms [41, 42, 43], and original transport protocols for supporting multi-homing [44, 45] have been proposed. At the Internet layer, multiple care-of address registration for MIPv6 [46] has been standardized.

Although multi-homed mobile hosts need not conduct vertical handoffs and consequently need not consider TCP adaptation, they can receive the benefit of vertical handoffs indirectly. However, multi-homed mobile nodes have another difficulty. One of the major issues for multi-homed mobile nodes is how they determine the best data-link for sending packets. In heterogeneous wireless networks, link characteristics vary widely between wireless data-links. If a multi-homing mechanism does not select the best data-link to send packets, this will cause significant throughput degradation. Therefore, studies on multi-homing mechanisms which liberate mobile nodes from conducting vertical handoffs must still take into account the significant differences in link characteristics that exist on heterogeneous wireless networks.

3.5.2 Which Methodology Should be Adopted?

The above subsection has described three methodologies for developing TCP adaptation schemes on heterogeneous wireless networks, which is the research objective of this dissertation. This subsection discusses which methodology best fits the design criteria of Chapter 1.

First, consider developing a wireless network friendly version of TCP. It is unclear whether this methodology distributes the network capacity fairly, since it changes the algorithm of the congestion control mechanism and it will therefore change the manner of packet transmission of mobile nodes. This may violate fairness with respect to acquiring network resources, which is one of the design criteria. This methodology will also violate another one of the design criteria, which is to retain the end-to-end principle, when a scheme is considered that inserts an intermediate node. A typical purpose of placing an intermediate node is to support communication between mobile nodes and correspondent nodes within the network path. When operations related to communication protocols are conducted within the network, such a network is no longer a network that obeys the end-to-end principle. For these reasons, this methodology does not follow the design criteria and should not be adopted.

The remaining two methodologies, enhancing existing TCP adaptability to changes in link characteristics and using multiple data-links simultaneously, are conceptually matched to the design criteria. Therefore, this dissertation adopts these two methodologies as basic approaches to develop schemes for TCP adaptation to heterogeneous wireless networks. The following two Chapters, Chapter 4 and Chapter 5, will propose different schemes which employ these two methodologies. Of course, even if the proposed schemes are developed according to these methodologies, it is still unclear whether the proposed schemes will actually satisfy the design criteria. Chapter 6 will discuss whether the schemes actually do satisfy the design criteria.

3.6 Summary

This Chapter has presented a brief description of the TCP congestion control mechanism and why it may not work as expected. These are the background and the research issues of this dissertation. A major problem of TCP on heterogeneous wireless networks is the mismatch of TCP congestion control parameters triggered by vertical handoffs, which can involve drastic changes in link characteristics. Since TCP is a dominant transport protocol of the Internet, it is a crucial issue to ensure that TCP functions work properly in heterogeneous wireless networks. This dissertation consequently considers how TCP should cope with vertical handoffs. The following two Chapters will present different approaches to that aim of avoiding harmful effects of vertical handoffs on TCP.

Chapter 4

Teppi: A Scheme for Adapting TCP to Drastic Changes in Link Characteristics

This Chapter presents a method, called "*Teppi*", to adapt the TCP congestion control mechanism to drastic changes in link characteristics. As described in section 3.3, drastic changes in link characteristics may occur during a vertical handoff and such changes have adverse affects on the after-handoff network. This problem can be avoided by resetting the congestion control parameters after the vertical handoff. However, it is not trivial to determine when the parameters should be reset. To determine when the parameters should be reset, the proposed method introduces a certain period, which is called the *link transition period*, and the method resets congestion control parameters after the link transition period has expired. Some experiments are conducted and results show that the method can be used to adapt the TCP congestion control mechanism to drastic changes in link characteristics.

4.1 Background and Motivation

A simple approach to adapting the TCP congestion control mechanism to drastic changes in link characteristics caused by a vertical handoff is to reset the congestion control parameters after the vertical handoff occurs. Since this does not change the congestion control mechanism itself, it is an approach that is compatible with extensibility, multiplicity, and TCP fairness. However, even if these parameters are reset after the vertical handoff, TCP is likely to override them with values that are calculated by needless congestion control, which also results from the vertical handoff.

The method presented in this Chapter addresses this problem. The proposed method determines when TCP should reset the congestion control parameters after the vertical handoff. To determine the point at which it is appropriate for TCP to reset the parameters, the method defines a certain period of time, called the *link transition period*. The link transition period is defined in the following manner: it starts at the point that a vertical handoff begins and ends at the point when there is no further occurrence of spurious congestion control caused by the vertical handoff. The method ensures that TCP resets parameters after the link transition period has ended; thus the reset parameters are not inappropriately adjusted by congestion control.

The primary goals of this research are to find ways to reduce needless congestion control and to adapt the TCP congestion control mechanism to the after-handoff data-link characteristics as soon as possible. There are some prospective benefits in achieving the goals such as avoiding too aggressive segment transmission and improving the utilization of the after-handoff data-link.

4.2 Impact of Vertical Handoffs on TCP

As previously shown in sections 3.3 and 3.4, once a vertical handoff is conducted in heterogeneous wireless networks, the TCP congestion control mechanism may not work as expected on the after-handoff data-link. This section reviews and specifies the harmful effects of vertical handoff on TCP, which are addressed by the method proposed in this Chapter.

Drastic changes of link characteristics Data-link characteristics, such as bandwidth, delays and loss rates, will change drastically with a vertical handoff. For example, when a mobile node conducts a vertical handoff from WLAN to 3G, the bandwidth will decrease drastically and the delay will increase drastically. In this situation, *cwnd* and *ssthresh* will be overestimated for 3G since, prior to the handoff, TCP has been optimizing its congestion control parameters for WLAN, which has relatively large bandwidth and fast transmission speed compared with 3G. Both overestimated *cwnd* and *ssthresh* may lead to network congestion, which triggers performance degradation of the network.

Consider the inverse situation in which a mobile node conducts a vertical handoff from 3G to WLAN. In this situation, *cwnd* and *ssthresh* will be considerably less than the

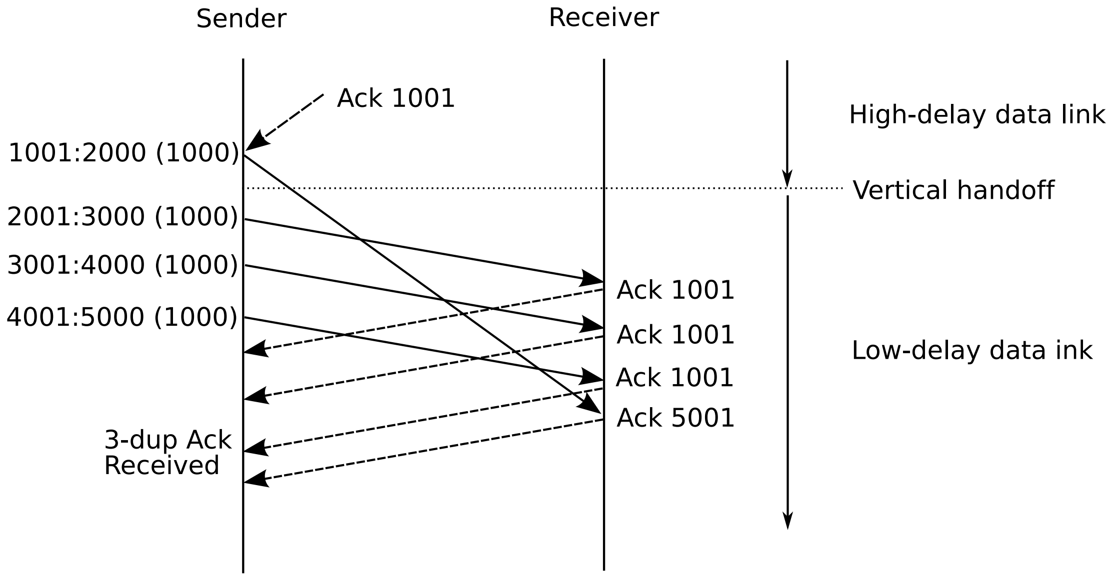


Figure 4.1: Segment reordering

optimal values. Although small *cwnd* and *ssthresh* have no adverse effect on the network, the throughput of the mobile node will increase very slowly because the mobile node will remain in the congestion avoidance phase even if the after-handoff data-link has considerable network capacity available.

Connectivity loss When a mobile node changes its upstream data-link, connectivity to the Internet may be lost for a certain period of time. This triggers some problems such as segment loss and retransmission timeout. These problems also lead to performance degradation of TCP.

Segment reordering TCP segment reordering occurs when the before-handoff data-link has relatively high latency and the after-handoff data-link has relatively lower latency. Figure 4.1 shows an example of segment reordering caused by a vertical handoff from a relatively slow data-link to a relatively fast data-link. In this figure, the segment 1001:2000 arrives at a receiver following the successive three segments. Since the successive three segments are out-of-order segments, the receiver acknowledges with duplicate ACKs for each reordered segment. When a sender receives more than three duplicate ACKs, TCP invokes the fast retransmission algorithm and reduces its segment transmission rate. However, the 3-duplicate ACKs in this situation do not mean that a segment loss has occurred and therefore such congestion control should not be conducted.

Spurious timeouts and inaccurate RTT measurements An RTO could be too short for the after-handoff data-link when the after-handoff data-link has significantly slower transmission speed than that of the before-handoff data-link. Consequently, an RTO may expire when there is a significant difference between the transmission speeds before and after a vertical handoff, even if no segment loss occurred during the handoff. As described in section 3.1.1, such spurious timeouts trigger needless segment retransmission and reduction of the segment transmission rate.

A vertical handoff which results in a drastic change in transmission speed also has another harmful effect; it makes RTT measurements become inaccurate. Suppose a segment is sent through a before-handoff data-link and its acknowledgement is received through the after-handoff data-link. In this case, the measurement of RTT consists of both one-way delays of these two data-links. Since the before-handoff data-link is no longer used after a vertical handoff, such a measurement does not reflect accurate RTT, which should be measured for only the after-handoff data-link. An RTO derived from an inaccurate RTT will be inappropriate for the TCP congestion control mechanism. If the RTO is too small, it may involve spurious timeouts, and if the RTO is too large, TCP will take a relatively long time to detect an actual segment loss.

Delayed Acknowledgement To enhance TCP performance, a receiver often holds up an acknowledgement and aggregates acknowledgements of some received segments. This aggregated acknowledgement is referred to as a *delayed acknowledgement*. If a delayed acknowledgement is sent during a vertical handoff, it might be transmitted through the after-handoff data-link in spite of the segments being transmitted through the before-handoff data-link. In this case, the RTT of a delayed acknowledgement contains one way delays of these data-links, therefore, delayed acknowledgements transmitted during vertical handoffs obstruct an accurate RTT measurement, which is the same problem as before.

4.3 Related Work

The Eifel algorithm [26] handles spurious timeouts and spurious fast retransmits. Spurious timeouts are timeouts that would not have occurred had the sender waited "long enough". The Eifel algorithm is simple and effective for the problems of vertical handoff because a vertical handoff often causes segment reordering as a result of sudden increases in RTT. Freeze-TCP [33] uses a *Zero Window Advertisement (ZWA)* to avoid performance degradation due to a handoff. TCP stops sending segments and enters a special persist mode when a sender receives a ZWA. In this mode, the sender freezes all retransmission timers and does not decrease its *cwnd*. Freeze-TCP utilizes this behavior; if a receiver senses an impending handoff, the receiver sends a ZWA to the sender, to prevent time-outs and reduction of *cwnd*. After the handoff, the receiver sends three copies of the ACK for the last data segment with a non-zero window size. The sender restarts sending after it receives this ACK. However, these schemes are not intended to handle drastic changes in link characteristics. As we mentioned in section 4.2, drastic changes in link characteristics may involve performance degradation.

The Bandwidth-aware method[32] restarts data transmission with a slow start phase when vertical handoff occurs, and then estimates *ssthresh* to allow *cwnd* to increase rapidly. This method uses the first three data packets after the vertical handoff to estimate *ssthresh*. However, connectivity of a mobile node often lacks stability during vertical handoff, as mentioned in section 4.2. Therefore, the *ssthresh* estimation may be inaccurate and this method might be difficult to apply in real environments.

A seamless vertical handoff scheme was proposed in [47], in which the authors defined a Handoff (HO) optional field in the TCP header. The receiver uses the HO optional field to allow the sender to recognize an impending handoff and a completing handoff. After the vertical handoff, the sender tries to adjust *cwnd* and *ssthresh* to the new data-link. However, this scheme requires both the sender and the receiver to modify TCP implementation.

4.4 Goals and Basic Approach

This section specifies the requirements to achieve the primary goal of this research, that is, to adapt the TCP congestion control mechanism after a vertical handoff as soon as possible with the least modification of the TCP congestion control semantics. This section also discusses which approach is most applicable for satisfying the requirements.

4.4.1 Requirements

Since TCP is the most widely used transport protocol in the Internet and it must provide reliable communication for end nodes with fair use of shared network resources, there are several requirements that have to be considered when one wants to modify the TCP congestion control mechanism. The following requirements for the modification need to be satisfied to achieve the goal of this research.

- Independence from link characteristics
- Fairness
- Keeping the semantics of the TCP congestion control mechanism

In a typical mobile internet environment, a mobile node generally uses a wireless technology as its data-link. However, when the mobile node is kept indoors, it may use a fixed wired technology as its data-link. Therefore, a modification of the congestion control mechanism itself for adapting to particular link characteristics (e.g. adapting to lossy data-links) or behavior (e.g. adapting a vertical handoff) is not appropriate for heterogeneous wireless networks. Modifying the congestion control mechanism itself creates a further problem: it is unclear whether such modifications will violate TCP fairness. TCP is regarded as fair when there are N TCP connections on a shared network and each connection takes up approximately $1/N$ of the total network capacity. In other words, if each TCP connection can achieve roughly the same maximum throughput, the *TCP fairness* condition holds. Since the throughput of each TCP connection depends on the behavior of the congestion control mechanism and different congestion control mechanisms involve different throughput, modifying the congestion control mechanism may violate TCP fairness. Therefore, the adaptation method should retain the semantics of the TCP congestion control mechanism to ensure that the extensibility and fairness properties continue to hold.

4.4.2 Basic Strategy

The primary reason for the problems mentioned in section 4.2 is the mismatch of the TCP congestion control parameters to the after-handoff data-link. Therefore, a strategy that resets the TCP congestion control parameters after a vertical handoff would be a simple but effective strategy to cope with the problems. Consequently, such a parameter resetting strategy is taken as the basis of this research.

One can take other strategies for adapting to drastic changes in link characteristics such as modifying the TCP congestion control mechanism itself, placing an intermediate node to cover an occurrence of a vertical handoff for senders, and using multiple interfaces simultaneously. However, the parameter resetting strategy has the following advantages. Firstly, since the strategy does not modify the congestion control mechanism itself, it is easy to retain the TCP fairness property and TCP semantics. Secondly, while the strategy that employs an intermediate node has the drawback that the intermediate node could be a single point of failure, the parameter resetting strategy does not have this problem since it works on an end-to-end basis. Finally, the strategy of using multiple interfaces simultaneously has the limitation that at least two interfaces must be available at the point where vertical handoff occurs. The parameter resetting strategy does not have such a limitation.

4.5 Congestion Control Adaptation Scheme

Although the basic strategy, which resets the TCP congestion control parameters after a vertical handoff, is quite simple, it is non-trivial to determine *when* the parameters should be reset. One naive way to carry out this reset would be just to execute it immediately after a vertical handoff. However, as described in section 4.2, several spurious retransmissions and timeouts would occur immediately after a vertical handoff, and these could result in needless congestion control. As a result, the parameters will be overridden even if the parameters were reset just after the vertical handoff.

Figure 4.2 depicts an example where parameters are overridden by spurious fast retransmission. The experiment is conducted in the emulation environment described in section 3.4.1, and the following scenario is used: a vertical handoff is conducted from a slow data-link (64 Kbps) to a fast data-link (11 Mbps) and *cwnd* and *ssthresh* are reset at the point where the vertical handoff has been completed. The vertical dotted line in the figure indicates the point where the vertical handoff has finished and values of *cwnd* and *ssthresh* are assigned to initial values at that point. As the plots shows, by resetting parameters, TCP goes to a slow start phase after the vertical handoff has finished. Although TCP stays in the slow start phase as expected for a while, *cwnd* and *ssthresh* are eventually overridden by spurious fast retransmissions. These spurious fast retransmissions are

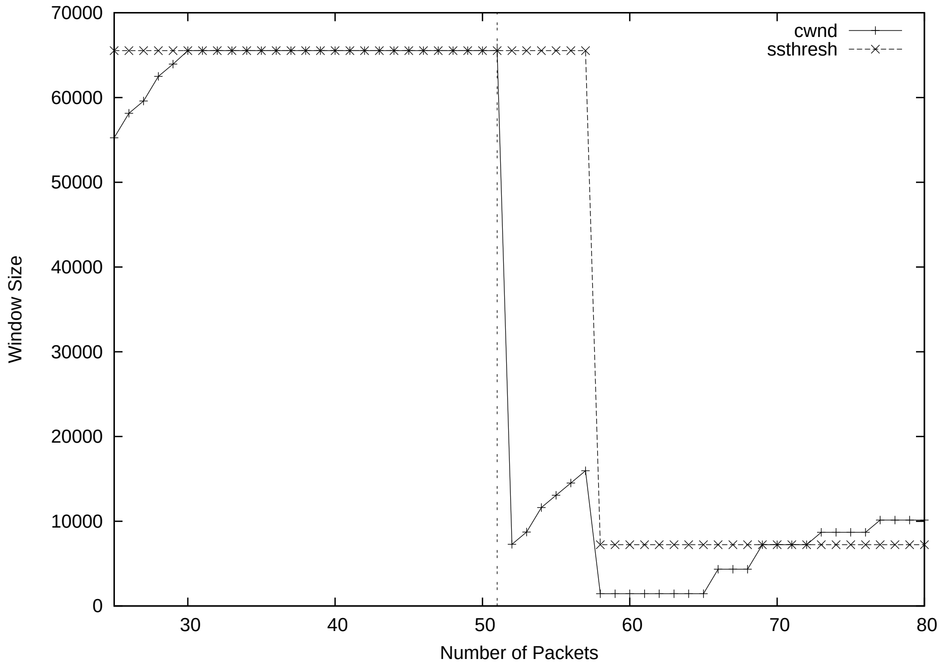


Figure 4.2: A case for which resetting parameters does not work

invoked by segment reordering caused by the vertical handoff.

To avoid this problem, the proposed method resets the parameters twice. The first point when parameters are reset is immediately after a vertical handoff has finished. This reset aims to prevent segments being sent too aggressively. The second point is when spurious retransmissions and timeouts caused by the vertical handoff no longer occur. To determine this point, the method introduces a certain period of time, called the *link transition period*. The link transition period defines the period during which spurious retransmissions and timeouts resulting from the vertical handoff could be occurring. The following subsection will define the link transition period more precisely.

4.5.1 The Link Transition Period

The conceptual definition of the link transition period is as follows: it starts at the point where the vertical handoff has started and ends at the point where there is no occurrence of spurious retransmission and timeouts caused by the vertical handoff. Following the definition, once the TCP congestion control parameters are reset at the point where the link transition period terminates, these parameters cannot be overridden by needless congestion control. The start of the link transition period, which is the point a sender recognizes the occurrence of a vertical handoff, is clear. However, it is unclear and difficult to determine when the link transition period terminates. The rest of this subsection considers how the method identifies the end of the link transition period.

Since needless congestion control occurs as a result of undesirable events such as segment reordering and drastic changes in transmission speed, the link transition period should be continued until such events are no longer registered. In addition, the link transition period should also be continued as long as the measurement of RTT could be inaccurate. Two kinds of acknowledgement arise from an inaccurate RTT measurement. One is a duplicate acknowledgement, and the other is an acknowledgement transmitted through the after-handoff data-link that is a response to a segment transmitted through the before-handoff data-link. Therefore, the link transition period should not terminate as long as the sender could receive such acknowledgements.

In this research, the following idea is used for finding the point where such undesirable events and inaccurate measurements of RTT are no longer occurring: determine the first acknowledgement which responds to a segment transmitted through the after-handoff data-link. Spurious retransmissions and timeouts are associated with segments transmitted through the before-handoff data-link. Therefore, such events no longer occur when the first segment transmitted after the handoff has been acknowledged. In addition, inaccurate RTT measurements would not occur after the first segment transmitted subsequent to the handoff has been acknowledged because all segments transmitted before the handoff have already been acknowledged at that point. There is consequently no opportunity for receiving an acknowledgement that causes an inaccurate RTT measurement after that point. Summarizing the discussion, we define the termination of the link transition period as follows. The link transition period ends when the following two conditions are satisfied: (1) all acknowledgements responding to segments transmitted before the handoff have been received by the sender, and (2) at least one acknowledgement responding to a segment transmitted after the handoff has been received by the sender. In this research, the first acknowledgement that satisfies condition (2) is referred to as the *WedgeACK*.

Now the problem to be solved for achieving the ultimate goal of this research is reduced to how the proposed method should detect the WedgeACK. The next subsection will identify the way to detect the WedgeACK.

4.5.2 Detecting the WedgeACK

For determining the WedgeACK, the proposed method introduces a new parameter called *tsmax*. This parameter is set at the beginning of the link transition period and used for holding the first sequence number that will be contained in the segment sent next time. In other words, when a vertical handoff has begun, *tsmax* holds the sequence number of the segment which will be first sent through the after-handoff data-link. Since each acknowledgement contains an acknowledgement number (*ACK*), the proposed method could examine each acknowledgement received to determine whether or not it could be the WedgeACK, by comparing *tsmax* and *ACK*. Basically, when $tsmax < ACK$, the acknowledgement could be the WedgeACK. In addition to this condition, another condition should be satisfied to solve the problems caused by duplicate acknowledgements and segment reordering. The WedgeACK will not be detected correctly when a duplicate acknowledgement is transmitted during a vertical handoff since such a duplicate acknowledgement satisfies the above condition. Fortunately, such delayed acknowledgements and segment reordering can be identified by using the *snd_una* parameter, which is a TCP parameter that holds a sequence number that has already been sent but is not yet acknowledged. By comparing *tsmax* and *snd_una* for each acknowledgement received, the proposed method can identify whether the received acknowledgement responds to a segment transmitted before the vertical handoff. If $tsmax \leq snd\_una$, the acknowledgement is a response to a segment transmitted before the vertical handoff.

As a consequence of the above discussion, the following condition is obtained to detect the WedgeACK.

$$(tsmax < ACK) \wedge (tsmax \leq snd\_una) \tag{4.1}$$

The proposed method treats the first acknowledgement that satisfies condition (4.1) as the WedgeACK and it terminates the link transition period when this WedgeACK is received.

To summarize the discussion of this subsection, the following describes the sequence of events for detecting the WedgeACK. When a vertical handoff has occurred, a sender stores the sequence number which will be sent next time in *tsmax* and starts the link transition period. The link transition period continues until the sender receives an acknowledgement which satisfies condition (4.1). During the link transition period, for each acknowledgement received, the sender updates the value of *tsmax* to *ACK* of the received acknowledgement when $tsmax < ACK$. The aim of this update is keeping the magnitude relation of sequence numbers since its value could overflow. Once the sender receives an acknowledgement which satisfies the condition (4.1), the sender terminates the link transition period and resets the congestion control parameters.

4.5.3 Validation of the Termination Condition

This subsection verifies whether the condition (4.1) can detect the WedgeACK appropriately for some possible segment sequences. There are four possible segment sequences before and after a vertical handoff.

- In-order sequence
- Segment reordered sequence
- Acknowledgement reordered sequence
- Delayed acknowledgement sequence

The following describes how the proposed method detects WedgeACK for each sequence by using example sequences. Other sequences, such as a sequence in which a segment loss has occurred, are also verifiable by applying the verification described in this subsection. An example sequence will be verified after these four typical sequences are verified.

Detecting the $WedgeACK$ with an In-Order Sequence

Figure 4.3 shows an example of detecting the WedgeACK with an in-order segment sequence. In Figure 4.3, a vertical handoff occurs after the segment 1001:2000 is transmitted. Since the last sequence number already sent at the beginning of the vertical handoff was 2000, *tsmax* is assigned the value 2001. The sender subsequently transmits the segment 2001:3000, and then receives the ACK 2001. At that point, *tsmax* and *ACK* hold the same value (2001) and they do not satisfy the condition. Therefore, the sender does not consider this acknowledgement to be the WedgeACK and continues the link transition period as well as updating *tsmax* to the value of *ACK*. In this case, the value of *tsmax* does not change since the values of *tsmax* and *ACK* are the same. Note that *snd_una* is also updated to the value of *ACK* since the acknowledgement contains the largest *ACK* at the point. Finally, the sender receives *ACK* 3001. This acknowledgement is a response to the segment 2001:3000. Since the segment was transmitted through the after-handoff data-link, the acknowledgement should be considered as the WedgeACK, and indeed the condition is satisfied in this case: $tsmax(= 2001) < ACK(= 3001)$ and $tsmax(= 2001) \leq snd\_una(= 2001)$.

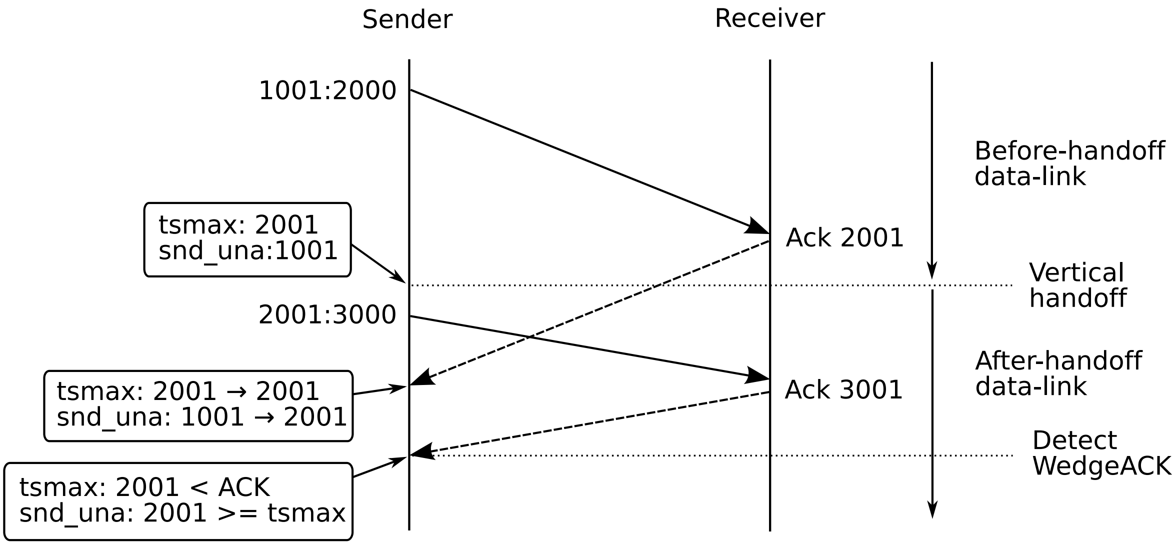


Figure 4.3: Detecting the $WedgeACK$ without reordering and a delayed acknowledgement

Therefore, the sender considers this acknowledgement to be the WedgeACK and terminates the link transition period.

Detecting the $WedgeACK$ with Segment Reordering

Figure 4.4 depicts an example of detecting the WedgeACK with a sequence that contains a segment reordering. In this sequence, the preceding segment 1001:2000 arrives at the receiver after the succeeding segment 2001:3000. When the segment 2001:3000 arrives at the receiver, the receiver acknowledges with a duplicate *ACK* 1001 because the receiver has not received the segment 1001:2000 at that point. Even if the segment 2001:3000 and the corresponding duplicate acknowledgement are transmitted through the after-handoff data-link, such a duplicate acknowledgement should not be considered as the WedgeACK because it contributes to an occurrence of spurious fast retransmission. The condition delivers the expected judgment: the duplicate acknowledgement could not satisfy the condition because $tsmax(= 2001) > ACK(= 1001)$ at that point. After this, the segment 1001:2000 arrives at the receiver and the receiver acknowledges with *ACK* 3001. Neither should this acknowledgement be considered the WedgeACK because it is associated with the segment 1001:2000, which was transmitted before the handoff. The condition once again rejects the acknowledgement as the WedgeACK since $tsmax(=$

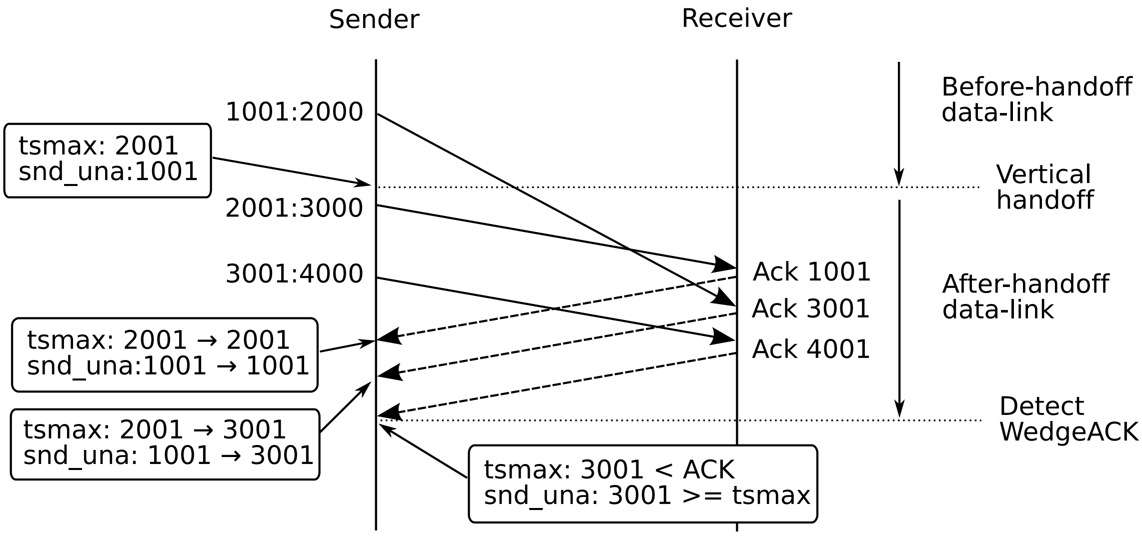


Figure 4.4: Detecting the $WedgeACK$ with segment reordering

$2001) < snd\_una(= 1001)$ at that point. The next acknowledgement with *ACK* 4001, which acknowledges the segment 3001:4000, satisfies the condition because $tsmax(= 3001) < ACK(= 4001)$ and $tsmax(= 3001) \leq snd\_una(= 3001)$. This judgment is correct because the acknowledgement is associated with the segment transmitted through the after-handoff data-link and there is no segment reordering at that point.

Detecting the $WedgeACK$ with ACK reordering

Figure 4.5 depicts an example of detecting the WedgeACK with a sequence containing an acknowledgement reordering. In Figure 4.5, the acknowledgement of the segment 1001:2000 arrives at the sender after acknowledgement of the segment 2001:3000. At the point where the sender receives acknowledgement of the segment 2001:3000, the condition is not satisfied. Although the sender subsequently receives acknowledgement of the segment 1001:2000, the sender does not care about that acknowledgement since it is an out-of-order acknowledgement. Therefore, such acknowledgement reordering does not always have to be avoided. The next acknowledgement with *ACK* 4001, which acknowledges the segment 3001:4000, is received at the sender, and it satisfies the condition because $tsmax(= 3001) < ACK(= 4001)$ and $tsmax(= 3001) \leq snd\_una(= 3001)$ at that point. As a result, this acknowledgement is considered to be the WedgeACK and the sender terminates the link transition period at this point.

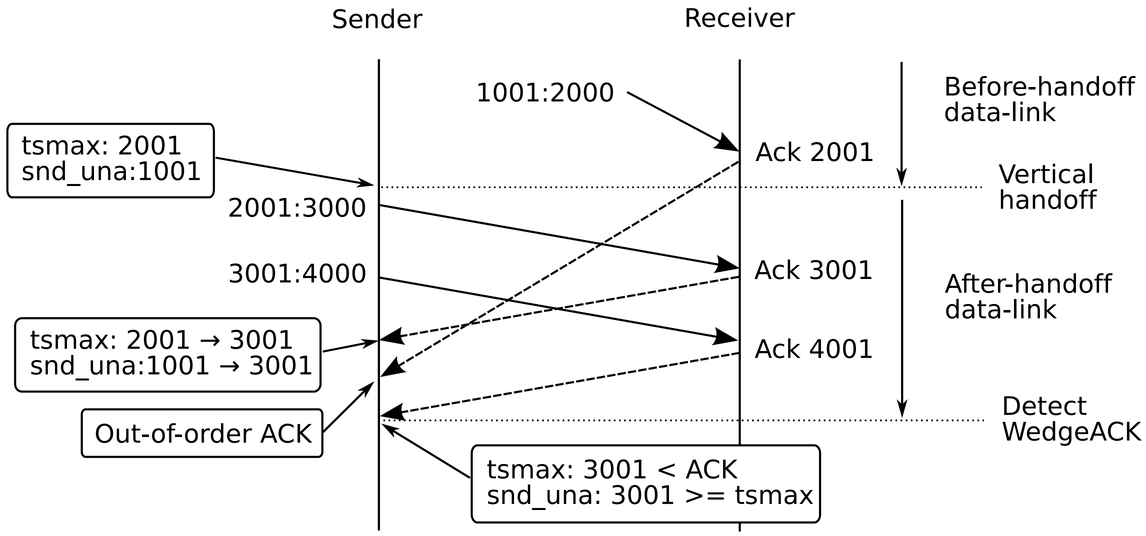


Figure 4.5: Detecting the $WedgeACK$ with ACK reordering

In this sequence, at least two acknowledgements transmitted after the handoff are needed to detect the WedgeACK. The reason that the first acknowledgement cannot be taken as the WedgeACK is that it may be delayed, and therefore should not be considered as the WedgeACK. The next example describes such a sequence.

Detecting the $WedgeACK$ with a Delayed ACK

Figure 4.6 depicts an example of detecting the WedgeACK with a sequence containing a duplicate acknowledgement over a vertical handoff. In this sequence, *tsmax* is assigned the value 2001 when the vertical handoff has started. After the vertical handoff has finished, the sender receives a delayed acknowledgement with *ACK* 3001. This acknowledgement is associated with both the segments 1001:2000 and 2001:3000. Since the segment 1001:2000 was transmitted before the handoff, the acknowledgement should not be considered as the WedgeACK. For this acknowledgement, $tsmax(= 2001) < ACK(= 3001)$ but $tsmax(= 2001) \leq snd\_una(= 1001)$, so the acknowledgement is not considered as the WedgeACK, as required. The next acknowledgement with *ACK* 4001, which acknowledges the segment 3001:4000, is received by the sender, and satisfies the condition because $tsmax(= 3001) < ACK(= 4001)$ and $tsmax(= 3001) \leq snd\_una(= 3001)$ at that point. Therefore, the condition determines the WedgeACK appropriately.

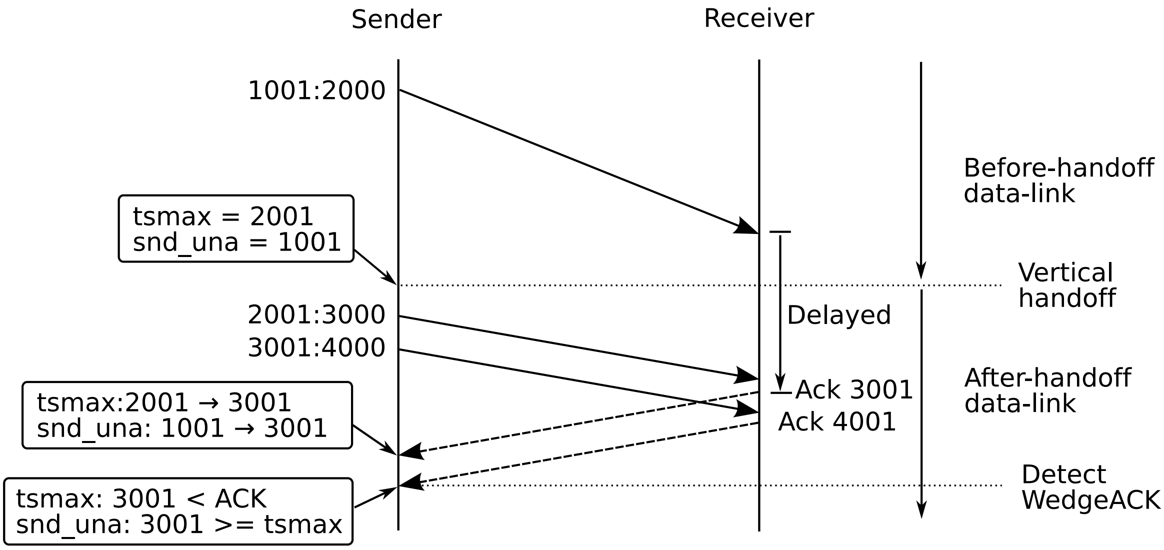


Figure 4.6: Detecting the $WedgeACK$ with a delayed ACK

Detecting $WedgeACK$ with Segment Loss

Figure 4.7 depicts an example of detecting the WedgeACK with a sequence in which a segment loss has occurred. After a vertical handoff has been conducted, the sender transmits the segment 2001:3000, but it is lost along the way. Since *ACKs* are never greater than or equal to *tsmax* until the lost segment is retransmitted and the segment is acknowledged, the condition is not satisfied until the retransmission has been completed. In general, the link transition period is thereby continued while such segment losses are being incurred. As Figure 4.7 shows, the acknowledgement which acknowledges the retransmitted segment does satisfy the condition and the sender considers it as the WedgeACK. This retransmission uses only the after-handoff data-link. Therefore this judgment is appropriate in this situation.

4.5.4 Managing RTT Measurement

As stated in section 3.3, measurements of RTT over a vertical handoff could be inaccurate. Since an RTO is calculated depending on measurements of RTT, inaccurate RTT measurement has a harmful effect on the segment loss detection mechanism. An overestimated RTO leads to a long time to detect segment losses, while an underestimated RTO could trigger spurious timeouts.

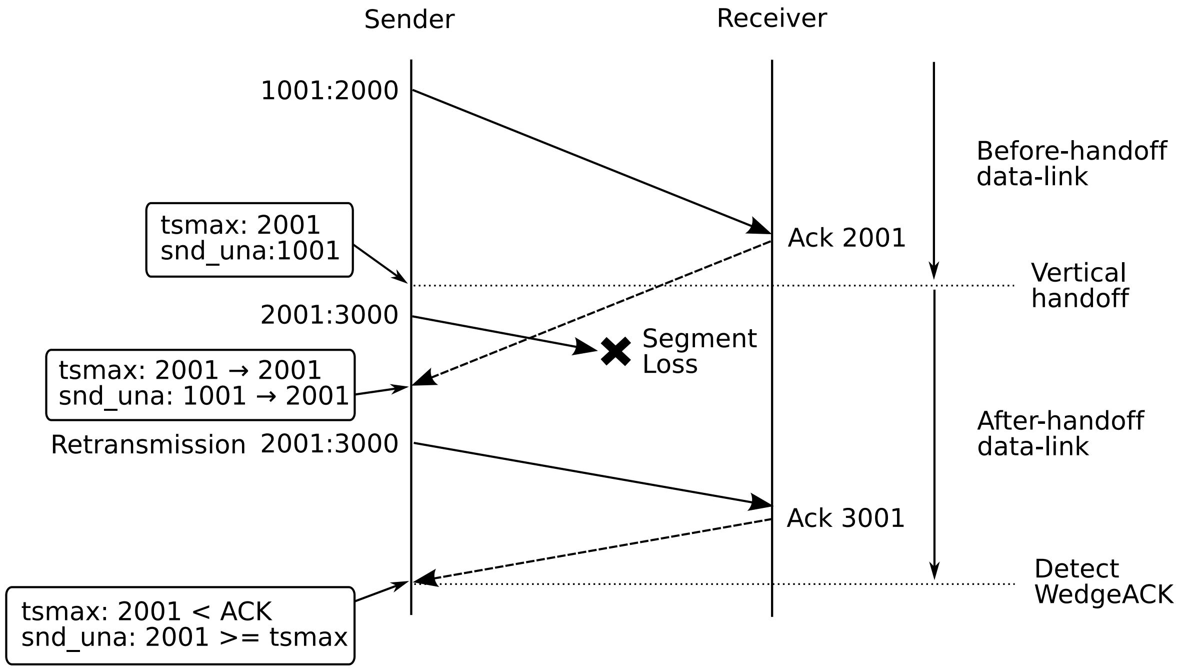


Figure 4.7: Detecting $WedgeACK$ with segment loss

To avoid inaccurate RTT measurements, the proposed method makes TCP stop the measurement of RTT during the link transition period. The proposed method stops the measurement of RTT when a vertical handoff has started and restarts the measurement when the WedgeACK is received at the sender. At the same time as stopping the measurement, the proposed method resets the value of the parameters related to the measurement, such as RTT, the average deviation of RTT (*rttvar*), and RTO, to adjust these values to the after-handoff data-link situation as soon as possible.

TCP uses two different schemes to measure RTTs: one tracks the time that it takes a certain segment to be acknowledged by using a timer, another appends the *TCP time-stamp option* to each transmitted segment and retrieves the elapsed time of the segment from its corresponding acknowledgement. The timer-based scheme is only employed when the TCP time-stamp option cannot be enabled. Since these two schemes measure RTT in two different ways, the proposed method needs to take into account each scheme. For the timer-based scheme, the proposed method stops the timer when the link transition period starts. During the link transition period, the sender does not measure RTT even if the sender receives acknowledgements. After the WedgeACK is detected by the sender, which means that the link transition period has ended, the sender restarts mea-

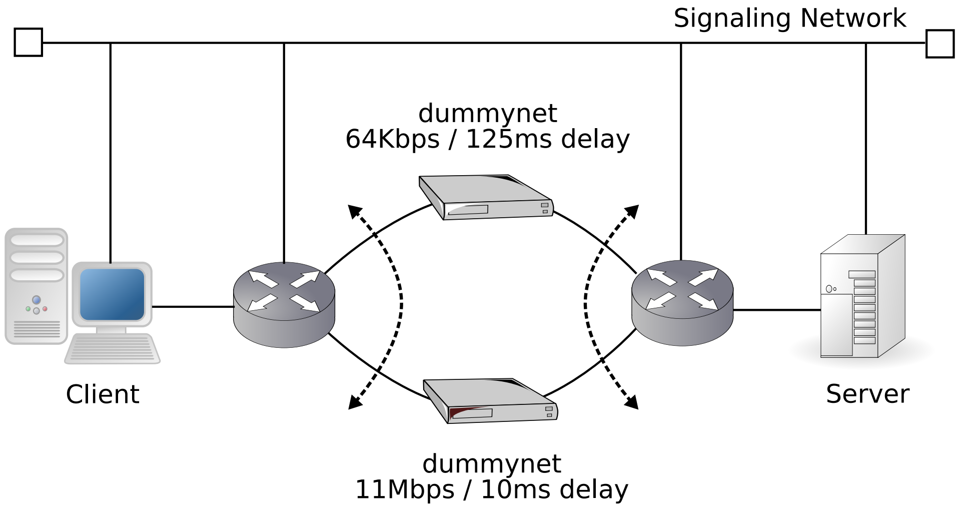


Figure 4.8: Experimental Environment

surement of RTT with the first segment transmitted after reception of the WedgeACK. For the time-stamp based scheme, the proposed method just ignores the time-stamp which is included in acknowledgements during the link transition period. In addition, the proposed method also makes TCP stop calculating RTT for each acknowledgement receipt. Once the sender receives the WedgeACK, the proposed method again takes note of the time-stamp and restarts calculation of RTT for each acknowledgement, including the WedgeACK.

4.6 Evaluation

The proposed method is evaluated in an emulation environment, which is shown in Figure 4.8. The emulation environment is almost the same as the environment employed in section 3.4.1. In experiments conducted here, one network path is configured as a relatively fast link (bandwidth 11 Mbps, one-way delay 10 ms) and the other as a relatively slow link (bandwidth 64 Kbps, one-way delay 125 ms). This configuration aims to emulate the performance of a wireless LAN (WLAN) and a Personal Handy-phone System (PHS). In the rest of this section, these two network paths are referred to as WLAN and PHS, respectively. In this environment, a vertical handoff is emulated by switching over the network from one path to the other. To exchange information about the occurrence of a vertical handoff, the server, client, and routers are connected to a secondary network,

Table 4.1: Software specification of the experimental environment

Component	Item	Software Version
Server	OS	NetBSD 4.0
	TCP	TCP NewReno (SACK enabled)
	HTTP Server	lighttpd-1.4.18
Client	OS	NetBSD 3.1
	TCP	TCP NewReno
	HTTP Client	Wget 1.10.2
Router	OS	NetBSD-current (2006.12.02 build)
dummynet	OS	FreeBSD 6.2-PRERELEASE

called the signaling network. The server is notified of the occurrence of a vertical handoff through the signaling network.

Table 4.1 describes the software specification of the experimental environment. There are slight differences between this environment and the environment in section 3.4.1, however they are essentially equivalent.

4.6.1 Experimental Settings

In each experiment conducted in this section, the server transmits a 1Mbyte file to the client, using HTTP. During the transmission, the network is switched over from one path to another to emulate a vertical handoff. To see the effect of the proposed method and to compare the proposed method to a naive parameter resetting method, the following two methods are examined for each experiment:

- *The naive method:* Resetting the congestion control parameters immediately after a vertical handoff occurs.

- *The proposed method:* Resetting the congestion control parameters according to the proposed method.

Throughout the file transmission, the congestion control parameters are tracked to examine whether the proposed method successfully adapts TCP to the after-handoff data-link.

Table 4.2 shows the initial values of the congestion control parameters, which are used as the values for resetting these parameters, in this environment. These values are derived from NetBSD, which is the operating system employed in this experiments.

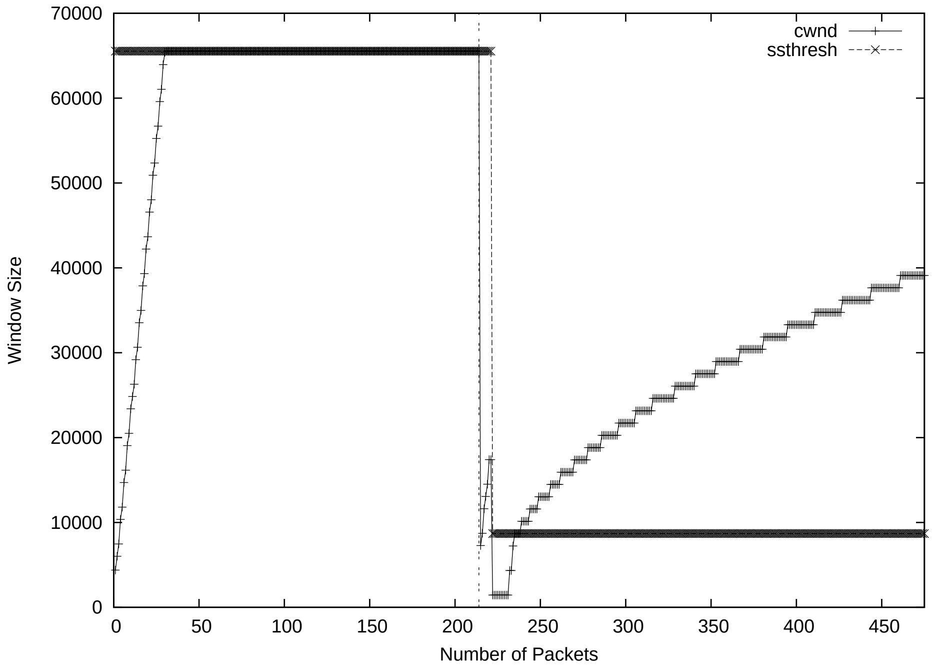
Table 4.2: Initial values of TCP congestion control parameters

parameter	initial value
cwnd	1MSS
ssthresh	65535 octets
RTT	6 seconds
srtt	0 seconds
rttvar	3 seconds

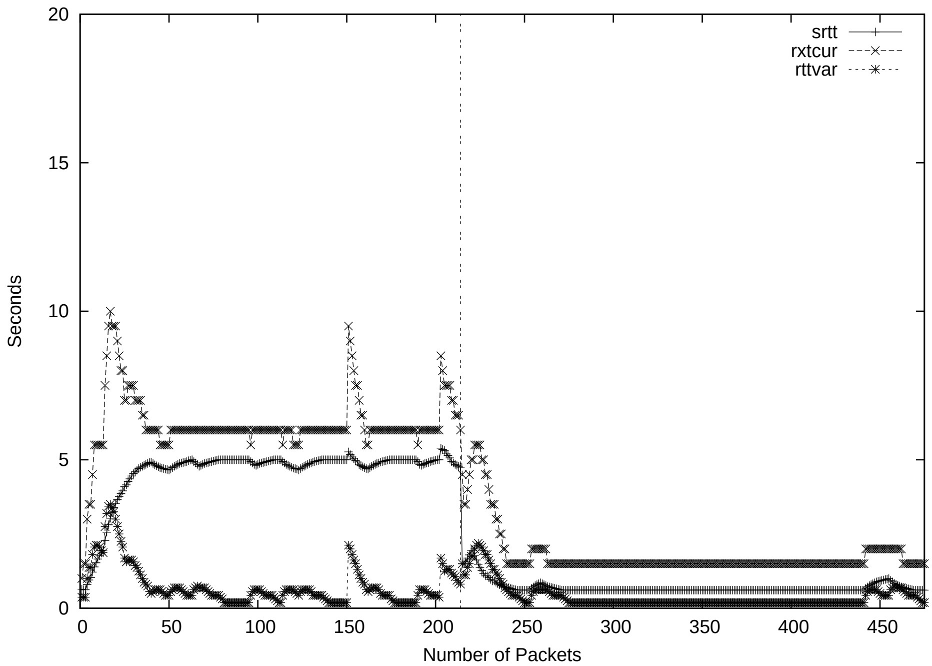
4.6.2 Handoff from Slow-link to Fast-link

The first experiment supposes that a vertical handoff is conducted from PHS to WLAN. Figure 4.9 and Figure 4.10 plot the congestion control parameters for each segment with the naive method and the proposed method, respectively. Figure 4.9(a) and Figure 4.10(a) plot *cwnd* and *ssthresh*. In these figures, the vertical axis is the window size, and the horizontal axis is the number of segments. Figure 4.9(a) and Figure 4.10(a) plot RTO (*rxtcur*), smoothed RTT (*srtt*) and the average deviation of RTT (*rttvar*). In these figures, the vertical axis is time, in seconds, and the horizontal axis is the number of segments. The vertical dashed line in Figure 4.9 denotes the point where an vertical handoff occurred. Likewise, the vertical dashed lines in Figure 4.10 denote the start and end points of the link transition period.

As shown in figure 4.9(a), although the congestion control parameters are reset immediately after the vertical handoff by the naive method, *cwnd* and *ssthresh* decreased in a short period of time. This reduction of *cwnd* and *ssthresh* is caused by a spurious fast retransmission. In the case of a vertical handoff from PHS to WLAN, the transmission rate increases drastically. Therefore, sets of segments sent after the vertical handoff could arrive at the receiver before segments sent prior to the vertical handoff. Although the TCP SACK option was enabled in this experiment and the SACK option provides tolerance for a small amount of segment reordering, spurious fast retransmission occurred and was observed in this experiment since a lot of segment reordering occurred as a result of drastic changes in transmission speeds. After the spurious fast retransmission, TCP enters the congestion avoidance phase and then the growth of *cwnd* becomes quite slow. This behavior is undesirable because the bandwidth becomes significantly larger after the vertical handoff.

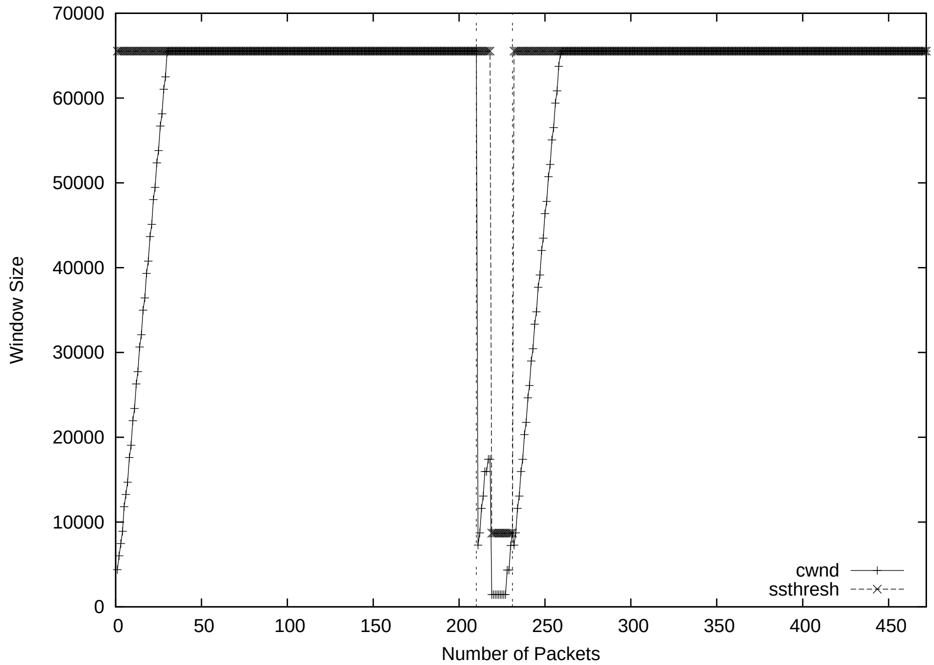


(a) *cwnd* and *ssthresh*

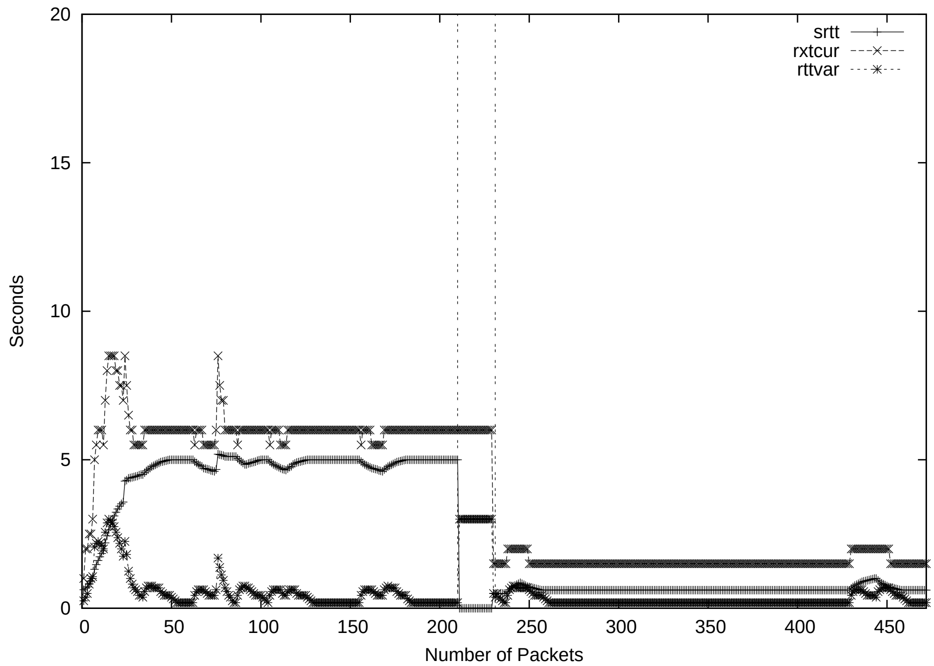


(b) *srtt*, *rxtcur* and *rttvar*

Figure 4.9: A vertical handoff from a slow-link to a fast-link without the proposed method



(a) *cwnd* and *ssthresh*



(b) *srtt*, *rxtcur* and *rttvar*

Figure 4.10: A vertical handoff from a slow-link to a fast-link with the proposed method

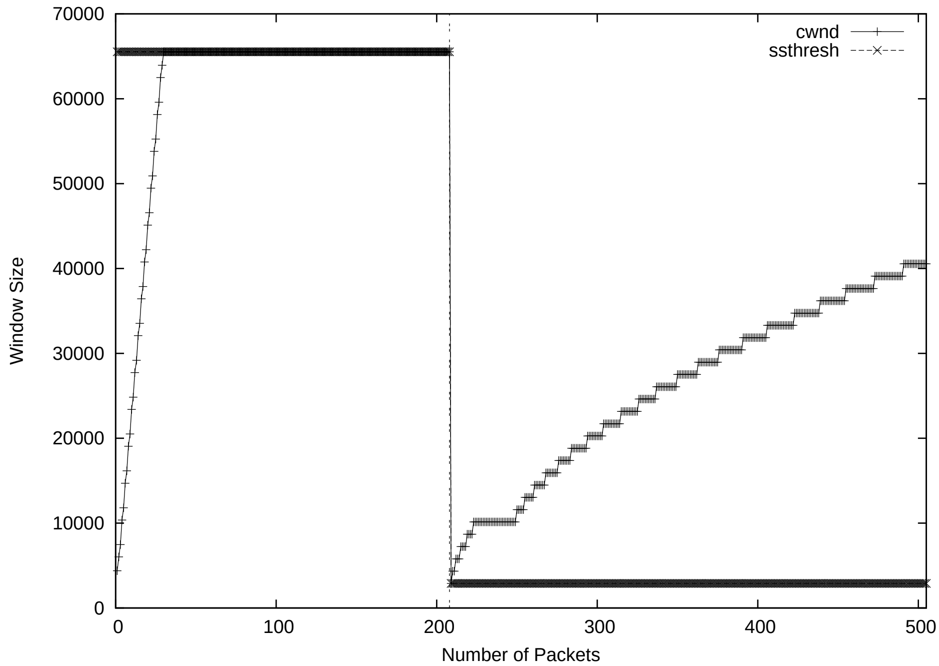
In Figure 4.10(a), decreases in *cwnd* and *ssthresh* are also observed for a short period of time after the vertical handoff has started, in spite of these parameters being reset when the vertical handoff started. This reduction is for the same reason as above. However, the proposed method resets the parameters once again at the point where the link transition period ends. As the result, TCP stays in the slow-start phase and *cwnd* increases quickly after the link transition period has finished. Since the bandwidth gets larger after the vertical handoff, this behavior implies that TCP adapts to the change of link characteristics appropriately.

Next, let us consider the behavior of the RTO related parameters. Before and after a vertical handoff, a temporary connectivity loss and segment reordering in general is likely to occur. These could lengthen RTTs for segments and such inaccurate RTT values have a harmful effect on the segment loss detection mechanism, as described in section 4.5.4. As Figure 4.9(b) shows, with the naive method, *srtt*, *rxtcur*, and *rttvar* are greater than their steady-state values for a short period of time after the vertical handoff. Even though this period is short, it is undesirable to use these overestimated values for detecting segment losses. On the other hand, with the proposed method, these parameters do not change during the link transition period since the proposed method stops the measurement of RTT over that period. Moreover, the proposed method restarts measurement after the end of the link transition period. As the result, these parameters can immediately adjust to their steady-state values.

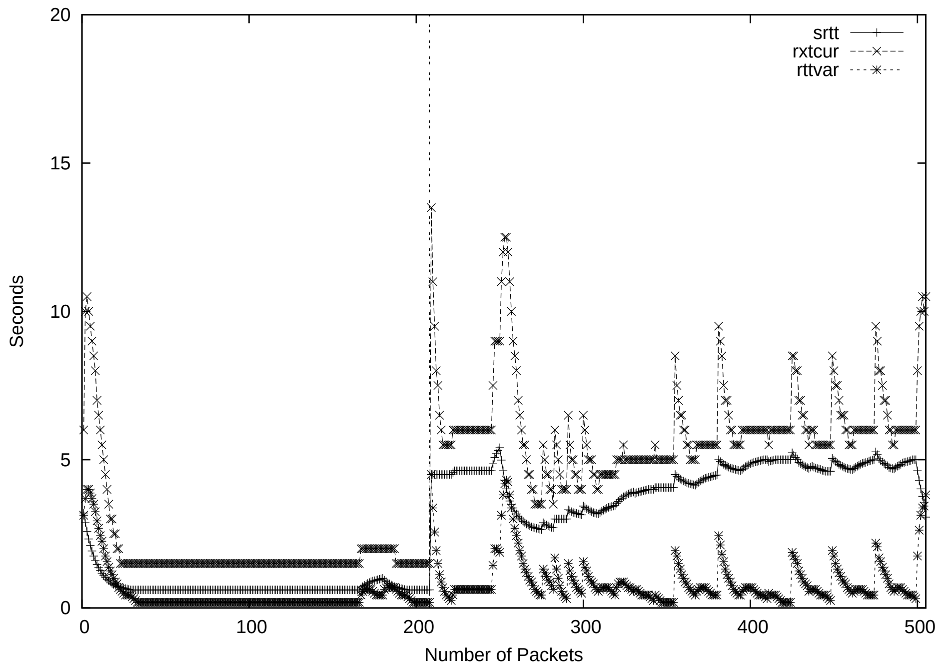
4.6.3 Handoff from a Fast-link to a Slow-link

The second experiment supposes that a vertical handoff is conducted from WLAN to PHS. Figure 4.11 plots the congestion control parameters for each segment for the naive method and Figure 4.12 plots the same values for the proposed method. The vertical and the horizontal axes of these figures correspond with those of Figure 4.9 and Figure 4.10.

In Figure 4.11(a), the naive method actually resets the congestion control parameters immediately after the vertical handoff. However, the effect of resetting parameters is hard to observe from the figure because the congestion control parameters are immediately overridden as a result of needless retransmission caused by a spurious timeout. In the case that a vertical handoff is conducted from WLAN to PHS, the transmission speed is drastically slowed. Since an RTO optimized for WLAN is too short for PHS, the RTO has expired even if all segments successfully arrive at the receiver. This behavior implies the

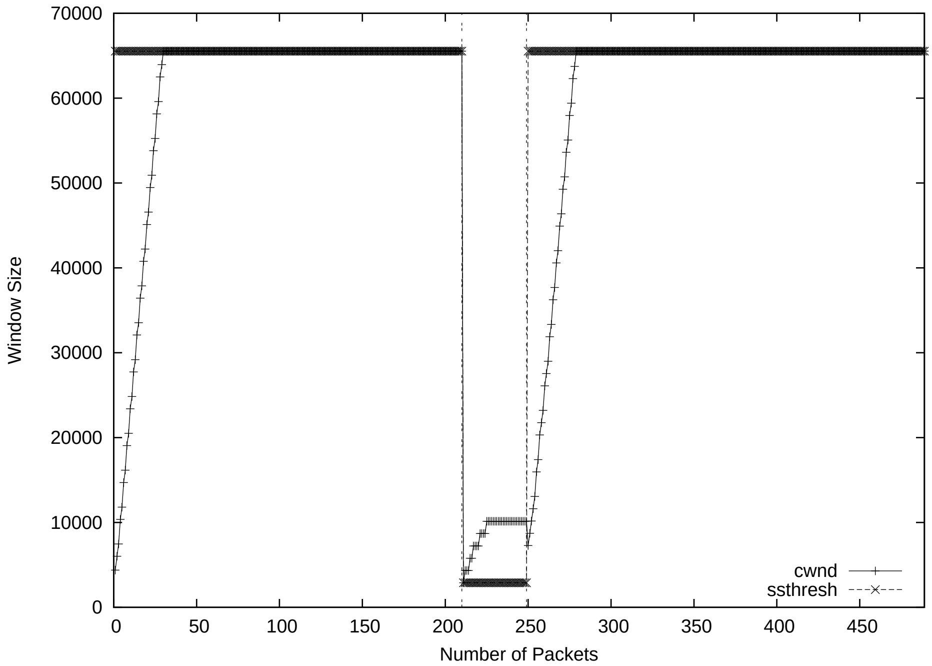


(a) *cwnd* and *ssthresh*

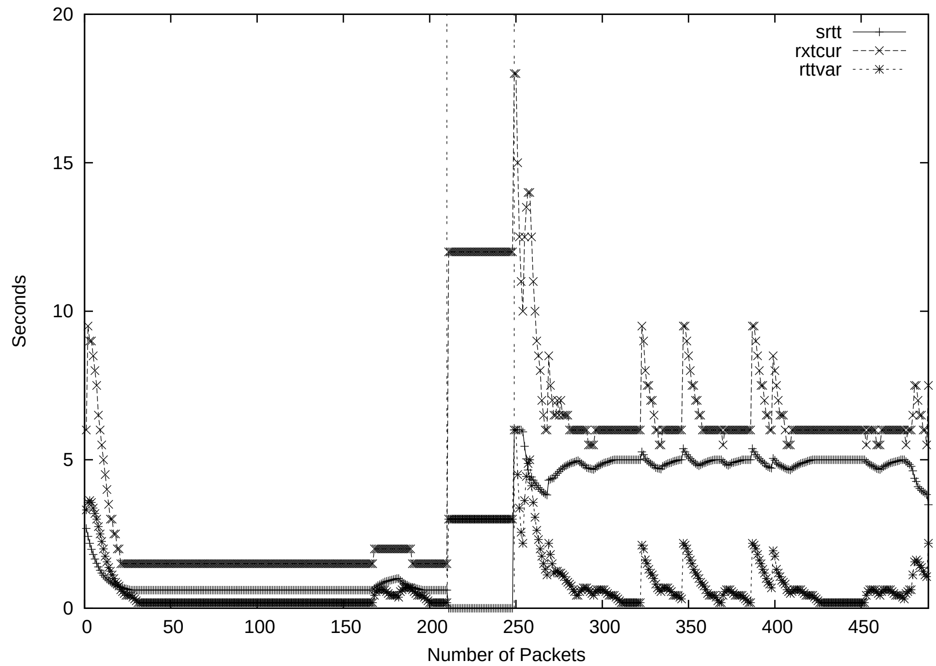


(b) *srtt*, *rxtcur* and *rttvar*

Figure 4.11: A vertical handoff from a fast-link to a slow-link without the proposed method



(a) *cwnd* and *ssthresh*



(b) *srtt*, *rxtcur* and *rttvar*

Figure 4.12: A vertical handoff from a fast-link to a slow-link with the proposed method

naive method could not successfully adapt TCP to the after-handoff data-link.

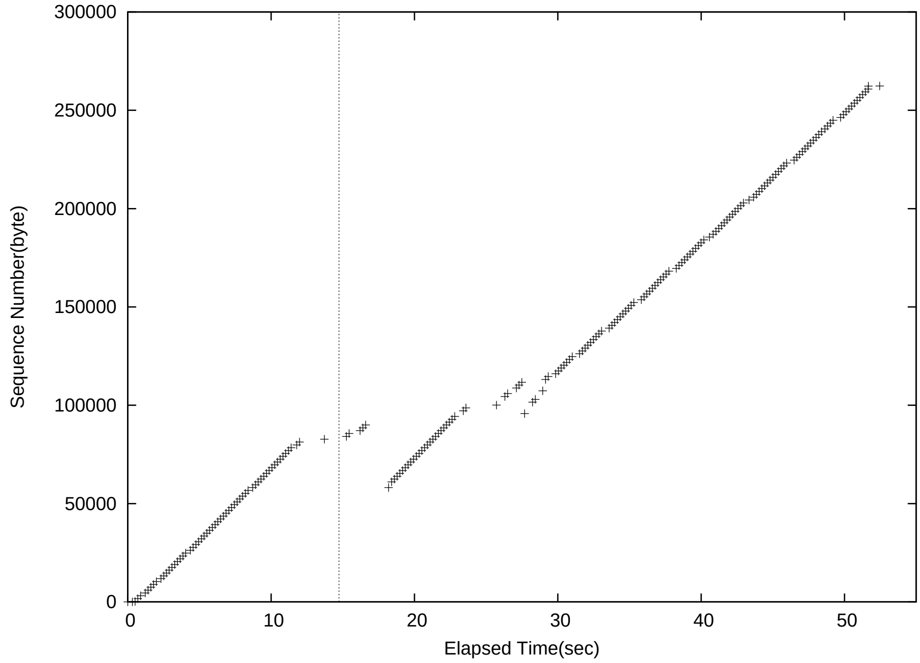
In contrast, the proposed method successfully resets the congestion control parameters after the end of the link transition period, as shown in Figure 4.12(a). With the proposed method, TCP stays in the slow-start phase after the end of the link transition period. TCP uses the *self clocking algorithm*, in which receipt of an acknowledgement triggers the next segment transmission, to adapt to the bandwidth bottleneck. Therefore, the sender can successfully transmit segments with no network congestion. As a result, the sender can stay in the slow-start phase after the link transition period has finished.

In Figure 4.11(b) and Figure 4.12(b), at both the point where the vertical handoff occurred and during the link transition period, the value assigned to *rxtcur* is nearly 12 seconds, which is twice the initial value. This behavior comes from the effect of the *exponential back-off algorithm*. TCP uses this algorithm, which increases RTO exponentially when TCP conducts retransmission control. Since a spurious timeout is involved at the point where the vertical handoff has started in both cases, *rxtcur* is assigned twice the initial value temporarily.

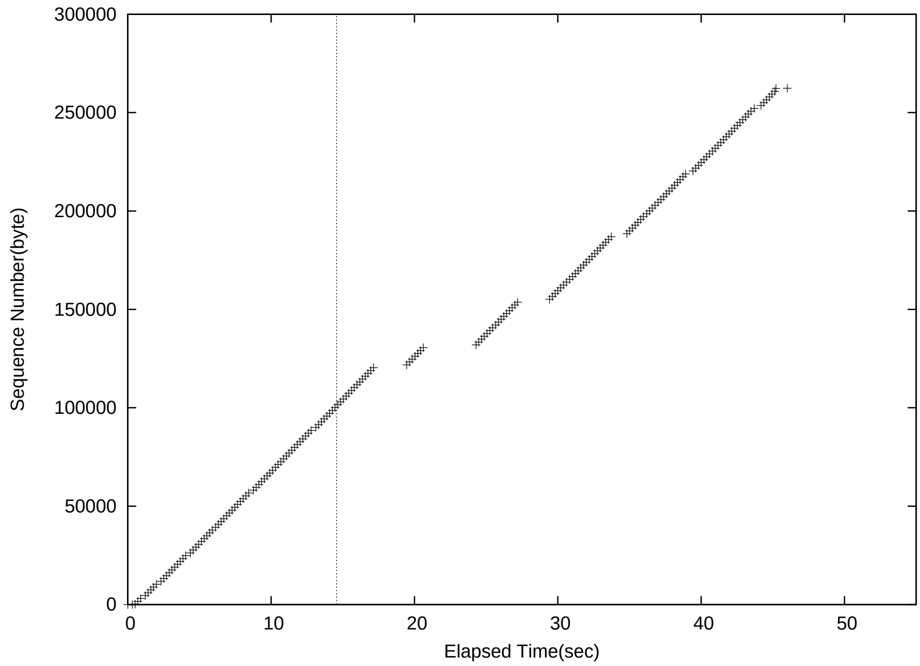
4.6.4 Impact on Cross Traffic

To retain the TCP fairness property is one of the requirements of this study. As section 4.2 has described, when a mobile node conducts a vertical handoff from a relatively fast data-link to a relatively slow data-link, the mobile node may send packets too aggressively and may cause network congestion. This behavior negatively impacts TCP fairness. To investigate whether the proposed method is consistent with TCP fairness, or that it avoids network congestion in other words, the following experiment was conducted. The experiment supposes that a vertical handoff is conducted from WLAN to PHS. Before the vertical handoff occurs, while the mobile node is on the before-handoff data-link, a preliminary TCP connection is established on the after-handoff data-link. This connection is referred to as the *existing connection*. Given this situation, the vertical handoff was conducted in both cases, namely, when the proposed method was enabled and when it was disabled.

Figure 4.13 plots the sequence numbers of the existing connection in both cases. The vertical dashed line in the figures denotes the point when the vertical handoff occurred. Figure 4.13(a) plots the sequence numbers of the existing connection without the proposed method. Figure 4.13(a) implies an occurrence of network congestion; the sequence



(a) without the proposed method



(b) with the proposed method

Figure 4.13: Sequence Numbers of cross traffic on the after-handoff data-link

Table 4.3: TCP throughput for transfer of a 1Mbyte file(unit: Kbps)

Method	PHS to WLAN	WLAN to PHS
proposed method	253.0	65.7
naive method	250.3	61.0
no reset of parameters	251.3	60.2

number is rewound twice after the vertical handoff. On the other hand, when the proposed method was enabled, there is no occurrence of network congestion. This fact can be seen in Figure 4.13(b); the sequence number did not rewind at any time. These results indicate that the proposed method is consistent with the TCP fairness property.

4.6.5 Performance

The previous subsections showed that the proposed method can adapt TCP to the after-handoff data-link reasonably well. As a secondary effect, this behavior brings about the prospect of a short-term improvement in the throughput of TCP. This subsection investigates the effect of the proposed method on improving throughput characteristics.

Throughput Evaluation

To investigate how much of a performance increase can be provided by the proposed method, the following experiments were conducted. In the emulation environment, the server transmitted a 1Mbyte file to the client and the throughput was measured. During transmission, the routers switched network paths to emulate vertical handoffs. Emulated vertical handoffs were conducted in both directions; from PHS to WLAN and from WLAN to PHS. The experiments were conducted for the following three scenarios; (1) using the proposed method, (2) using the naive method, and (3) doing nothing even if a vertical handoff occurred. Table 4.3 shows throughput of each measurement.

The proposed method contributes slightly to improving throughput for both handoffs. This improvement comes about from the effect of adapting the congestion control parameters to the after-handoff data-link. When the server employs the proposed method, TCP can transmit packets in the slow start phase, even after the vertical handoff. As a result, *cwnd* increases quickly and contributes to a throughput improvement.

Table 4.4: TCP throughput for transfer of files of various sizes(unit: Kbps)

Method	File size		
	512Kbytes	1Mbytes	2Mbytes
proposed method	78.4	65.7	66.3
naive method	65.7	61.0	66.3
no reset of parameters	64.2	60.2	66.3

Throughput Improvement Limit

Since the proposed method just adapts the TCP congestion control mechanism to the after-handoff data-link, the effect of throughput improvement is limited to short data transmissions. To investigate the limitation of the effect of throughput improvement due to the proposed mechanism, the following experiments were conducted. In each experiment, the server transmitted files of different data lengths to the client and the throughput was measured. Data lengths of files were 512Kbytes, 1Mbytes, and 2Mbytes, respectively. During transmission, an emulated vertical handoff was conducted from WLAN to PHS. Table 4.4 shows the throughput measurements.

For the case of a 512Kbyte file transmission, a significant throughput improvement was observed. On the other hand, no effect on throughput improvement was observed for the 2Mbyte file transmission. The effect on throughput improvement can be seen for files up to 1Mbyte in length. These results imply that the proposed method contributes to throughput improvement only for a short period after the handoff.

4.7 Discussion

Although our proposed scheme can adapt TCP to drastic changes in link characteristics, it has some issues and limitations. This section discusses the following issues and limitations.

Handoff notification mechanism. The proposed method assumes that a sender can obtain information about occurrences of vertical handoffs. However, as yet, there is no standardized handoff notification mechanism. Developing an appropriate handoff notification mechanism for heterogeneous wireless networks is an important research topic. Chapter 6 will discuss how such a mechanism should be developed.

Limitation of adaptation. The adaptation capability of the proposed method depends on how fast a sender can sense the occurrence of a vertical handoff. In particular, the apparent capability of avoiding network congestion on the after-handoff data-link may not work as expected when the sender cannot recognize the occurrence of a vertical handoff quickly. The proposed method adapts TCP to drastic changes in link characteristics by resetting the congestion control parameters twice: firstly, it resets the parameters immediately after a vertical handoff and secondly, it resets the parameters after the end of the link transition period. When the first parameter resetting is delayed, the sender may send packets too aggressively on the after-handoff data-link. As a result, network congestion might occur. To cope with this limitation, an appropriate handoff notification mechanism, which can notify occurrences of vertical handoffs as early as possible, is required.

Less performance enhancement . Another limitation of the proposed method is its reduced performance enhancement. As section 4.6.5 has shown, the proposed method does not provide significant improvement of throughput. This fact indicates that the effect of enhancing performance by adapting TCP to the after-handoff data-link is slight. However, this does not mean that TCP adaptation is worthless. It is still important because adapting TCP to the after-handoff data-link also contributes to avoiding network congestion.

4.8 Summary

An occurrence of a vertical handoff in a heterogeneous wireless network will involve drastic changes in link characteristics and traditional TCP does not adapt to such drastic changes. This Chapter has focused on the method by which the congestion control parameters can be reset after the vertical handoff to adapt TCP to the after-handoff data-link characteristics as soon as possible. During vertical handoffs, spurious retransmissions and spurious timeouts are likely to occur and consequently the naive way of resetting the parameters does not work as expected. The proposed method avoids this problem by introducing the link transition period. The experiments conducted in this Chapter show that the proposed method successfully adapts TCP to drastic changes in link characteristics.

Chapter 5

Luct: A Practical Multi-homing Mechanism for Seamless Handoff on Heterogeneous Wireless Networks

The primary benefit of introducing multi-homing in heterogeneous wireless networks is to provide continuous network connectivity in an effective manner, rather than achieving a large throughput by aggregating bandwidth. For practical use, a multi-homing mechanism for mobile hosts should work with fewer configurations and no modifications of existing applications. This Chapter proposes a practical multi-homing mechanism, called "*Luct*", which provides continuous network connectivity to mobile nodes on heterogeneous wireless networks. The mechanism has the following features: (1) existing applications can run on the mechanism without any modification, and (2) it does not require configurations for data striping. The mechanism is implemented using the Linux operating system and some empirical experiments are conducted to evaluate the mechanism. The results show that the mechanism has less communication overhead and quick adaptation of network situations.

5.1 Background

In current mobile environments, mobile nodes often have multiple wireless interfaces and employ them to connect to the Internet. Coordination between multiple wireless interfaces offers stability, tolerance to network disconnections, availability and bandwidth aggregation to mobile nodes due to their complementary coverage area, bandwidth and service cost. As we will discuss later, mobile nodes should act as *multi-homed mobile hosts*[37], which employ multiple interfaces simultaneously, to receive the full benefits of coordination between these wireless interfaces.

There are several ways to create multi-homed mobile hosts, and many schemes have been proposed for multi-homing in mobile environments in the past. However, most of these schemes are difficult to apply to actual mobile environments. Some of these, such as IP layer aggregation based multi-homing schemes, require characteristics of interfaces and policy-based strategies to stripe data across multiple interfaces in an effective manner. Other schemes, such as most existing transport layer multi-homing schemes, also have a difficulty in that they impose modification on existing applications. Moreover, although most of these schemes focus on achieving a large bandwidth by aggregating multiple interfaces, the effect of bandwidth aggregation is only slight since the link characteristics of each wireless technology are significantly different. Instead, providing sustained network connectivity and mobility transparency is the primary advantage of multi-homed mobile hosts. Therefore, we consider that multi-homing is an appropriate way to provide seamless network connectivity to applications running on mobile hosts, whereas increasing throughput by bandwidth aggregation is a secondary benefit.

This Chapter proposes a practical multi-homing mechanism for heterogeneous wireless networks. The key features of the mechanism are: (1) existing applications require no modification, and (2) no configuration for data striping. Since the mechanism acts as a virtual interface, existing network applications can run using the mechanism without modification. Packets sent through the virtual interface are transmitted via TCP connections, which are established for each real interface. TCP connections play a crucial role in providing zero configuration data striping. TCP congestion control parameters are applied to select the best connection to send packets by the proposed data striping strategy.

The mechanism was implemented using the Linux operating system and we investigated whether the mechanism is applicable to heterogeneous wireless networks by conducting experiments in an actual mobile environment. Experimental results show that our mechanism has lower performance overhead and that the data striping strategy is more appropriate than a rate-based strategy in a practical environment.

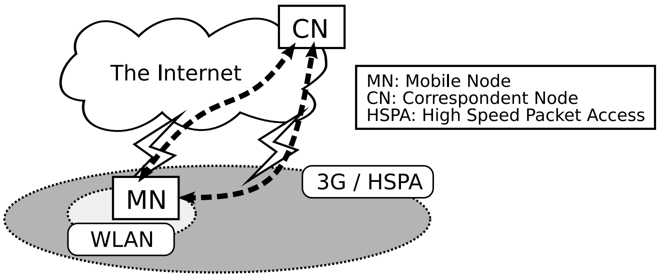


Figure 5.1: Example of using multiple wireless technologies

5.2 Goals

The goal of this research is to provide sustained network connectivity to mobile nodes on heterogeneous wireless networks in order to achieve optimized performance. This section describes the characteristics of heterogeneous wireless networks, the target environment of this study, and presents an example environment that indicates clearly the benefits of coordinating multiple wireless technologies. This section also discusses the benefits of introducing multi-homing in the target environment and how we should construct a multi-homing mechanism that provides a solution to our problems.

5.2.1 The Target Environment

A wide variety of wireless technologies are available in the current mobile environment. These technologies, such as wireless LAN (WLAN), third-generation cell phones (3G), High Speed Packet Access (HSPA) and Mobile WiMAX all have distinct characteristics. Their complementary bandwidth, delay, jitters, and service availability motivates mobile nodes to employ multiple wireless technologies to enhance their communication quality. In a nutshell, obtaining high-speed transmission involves losing service availability, and vice versa.

Figure 5.1 depicts a typical example of using multiple interfaces on heterogeneous wireless networks. A mobile node (MN) has two wireless interfaces: WLAN, which has a relatively high transmission speed and limited coverage, and 3G/HSPA, which has a relatively slow transmission speed and wide coverage. The MN employs both of these to communicate with a correspondent node (CN). In the rest of this section, we refer to this example to investigate how we might create a mechanism that provides continuous network connectivity to mobile nodes in an effective manner.

A simple strategy for coordinating multiple wireless interfaces is to conduct vertical handoffs according to the situation of the MN. The MN uses WLAN to obtain high-speed transmission when the MN is within the coverage area of a WLAN, whereas the MN changes its primary interface to 3G/HSPA when the MN is away from the coverage area of the WLAN, maintaining network connectivity while sacrificing transmission speed. Although this strategy in itself is quite simple, it generally requires the following additional features for practical use. (1) *Mobility transparency support:* a vertical handoff involves a change of IP address, which causes disconnection of transport layer connections. To retain transport layer connections, mobile nodes require mobility transparency. Mobile IP/IPv6 [48][3] are standardized technologies to support mobility transparency for mobile nodes, and Network Mobility Basic Support [4] is standardized for mobile networks. (2) *Congestion control adaptation:* the MN should adapt the congestion control context of existing TCP connections after a vertical handoff to share bandwidth of a post-handoff wireless network fairly with other cross-network traffic. A discrepancy in the congestion control context involves performance degradation and network congestion in the post-handoff wireless network. To cope with this problem, several schemes have been proposed to modify the congestion control context after a vertical handoff [49, 32, 50, 26]. (3) *Monitoring and handling network states:* since the MN moves around wireless networks, the MN must check periodically whether wireless interfaces are active or not. If a wireless interface becomes active or inactive, the MN should be able to react to provide continuous network connectivity to achieve high performance.

Supporting these features simultaneously complicates the implementation of this strategy. Thus, this research does not employ this strategy and instead takes an alternative route: introducing multi-homed hosts in heterogeneous wireless networks, which are referred to as multi-homed mobile hosts. The next subsection will discuss the benefits and issues of multi-homed mobile hosts.

5.2.2 Multi-homed Mobile Hosts

Introducing multi-homing into mobile hosts provides the same benefits as the above strategy; it provides a high transmission speed and wide network availability in a natural fashion because multi-homing mechanisms involve the features mentioned in the previous subsection. In addition, multi-homed mobile hosts can assume flexible data transmission strategies since they can stripe data across multiple interfaces simultaneously. For example, if a mobile node knows the transmission speeds of each interface, it can stripe data across multiple interfaces based on these speeds to optimize throughput.

Ideally, multi-homed mobile hosts can utilize the total bandwidth across multiple interfaces by bandwidth aggregation. However, as briefly described in section 5.1, the effect on throughput by aggregating interfaces is slight. There are two primary factors working against achieving a large throughput by interface aggregation. (1) *Disparities in bandwidth.* Consider two wireless technologies that mobile nodes can use at present, IEEE802.11n and HSPA, which can be adapted for the example shown in Figure 5.1. While the ideal maximum data rate of IEEE802.11n is 450 Mbps, emobile [13], an HSPA service that is available in Japan, has just 7.2 Mbps as its ideal maximum data rate. Aggregating these interfaces yields an increase of only 1.6%, compared with the maximum data rate of IEEE802.11n. (2) *Vulnerable link quality.* The unsteady link quality of wireless networks involves high jitters, which have a negative impact on application-level throughput. When the MN stripes packets across multiple interfaces, a certain amount of *packet reordering* will be caused due to high jitters even if the MN employs an appropriate data striping strategy based on the transmission speeds of each interface. Packet reordering has a significant impact on the throughput of TCP, which is used as the transport protocol by dominant network applications such as HTTP, FTP, and SMTP. There are two problems caused by packet reordering. Firstly, packet reordering triggers a spurious fast retransmission[26], which causes an irrelevant packet retransmission and reduces the transmission rate of the connection unnecessarily. Second, packet reordering also causes a *head-of-line blocking* problem. When a head-of-line blocking problem occurs, TCP cannot push succeeding received packets to the destination application until the MN receives the head packet since TCP has the responsibility for reassembling received packets. As a result, application-level throughput will decrease. Since TCP is the dominant traffic protocol on the Internet [2], this performance degradation is unacceptable. For these reasons, aggregating multiple interfaces does not contribute to achieving a higher throughput in heterogeneous wireless networks.

5.2.3 Multi-homing Methodologies

As we will describe in section 5.2.5, one of our primary aims is to provide a multi-homing mechanism with no modification of existing applications. A typical way to achieve this is as follows: to aggregate multiple network entities on a certain abstraction layer of the

TCP/IP model and then to provide the aggregated entities as a single entity, such as a virtual interface. From the viewpoint of an application, such aggregation methods can be viewed as some sort of tunneling method. We can classify the aggregation methods according to which layer is aggregated:

- *aggregation of internet layer entities.* This involves aggregating multiple IP addresses as a single network entity. From the viewpoint of an application, this method can be considered as an *IP tunneling* method. This method aggregates multiple IP tunnels and provides a single network entity. We refer to this method as *IP aggregation based multi-homing*.

- *aggregation of transport layer entities.* This involves aggregating transport layer entities as a single network entity. In this paper, we focus particularly on aggregating TCP connections when we consider aggregating transport layer entities since aggregating UDP entities is almost equivalent to aggregating IP addresses. From the viewpoint of an application, this method can be considered as a *TCP tunneling* method. This method aggregates multiple TCP tunnels and provides a single network entity. We refer to this method as *TCP aggregation based multi-homing*.

Which aggregation method is better for mobile nodes on heterogeneous wireless networks? We put weight on compatibility with TCP according to the fact that most network applications employ TCP as their transport protocol, and so our answer is TCP aggregation based multi-homing. The brief reason is that TCP aggregation based multi-homing can maintain the context of TCP separately on each network path. The following three factors describe the reason in detail. Firstly, a flexible data striping strategy can be achieved in a natural fashion since TCP aggregation based multi-homing can measure the statistics of each network path separately. TCP measures RTT of segments and adjusts the congestion window(*cwnd*) to perform congestion control. These statistics are useful for achieving an effective data striping strategy. Secondly, TCP aggregation based multi-homing does not require congestion control modification. Multi-homed mobile hosts often use the active/backup data striping strategy, which transmits packets through only the primary wireless network and other wireless networks are held in reserve, since it is hard to achieve a large throughput by interface aggregation, as discussed in section 5.2.2. If multi-homed mobile hosts employ IP aggregation based multi-homing with the active/backup data striping strategy, they need congestion control adaptation to share bandwidths of wireless networks fairly, as mentioned in section 5.2.1. In contrast, TCP aggregation based multi-homing does not require congestion control adaptation since it can maintain congestion control contexts for each network path separately. Finally, it is difficult to perform proper congestion control when a multi-homing mechanism aggregates multiple network paths on the internet layer, which is what happens with IP aggregation based multi-homing. When a TCP connection is established on such a multi-homing mechanism, characteristics of each network path, such as the RTT of segments, jitters, and rates of packet loss, are mixed in a single TCP context. Therefore, the TCP congestion control mechanism does not work properly for IP aggregation based multi-homing when the characteristics of each network path are significantly different. Since the target environment has such diverse characteristics, TCP aggregation based multi-homing is a better way to ensure that the congestion control mechanism works properly.

5.2.4 Related Work

There have been many mechanisms proposed to introduce multi-homing for mobile nodes. In this section, we discuss transport layer multi-homing mechanisms that have been proposed in the past since our mechanism takes similar approach. Some transport layer multi-homing mechanisms are based on TCP [37, 38, 39, 40], while others are based on the Stream Control Transmission Protocol (SCTP) [41, 42, 43]. A socket level approach [36] has also been proposed. In addition, several researchers have developed their original transport protocol to support multi-homing [44, 45]. The main purpose of these mechanisms is to achieve a large throughput by aggregating multiple connections, but as discussed in section 5.2.2, the effect of bandwidth aggregation is often slight in heterogeneous wireless networks. Additionally, these mechanisms require the modification of existing applications. From the perspective of practical use, this requirement is undesirable.

5.2.5 Goals

To provide sustained network connectivity to multi-homed mobile hosts in an effective manner, the following goals should be achieved by the multi-homing mechanism that will be proposed in section 5.3.

- *Capability of handling diverse wireless technologies.* Data link characteristics, such as bandwidth, delay, jitter and packet loss rate, differ among wireless technologies in heterogeneous wireless networks. The mechanism should ensure that these wireless technologies can be coordinated with others to provide both high-speed transmission speed broad network connectivity without significant performance degradation.

- *Zero configuration Required for Data Striping.* Some multi-homing mechanisms employ policy-based data striping strategies. For example, they often take a source/destination address-based strategy or a flow-based data striping strategy. However, these policy-based data striping strategies are not suitable for heterogeneous wireless networks in practical use because it is difficult to define an appropriate policy, due to their erratic connectivity and link characteristics. The mechanism should thus not require any configuration for data striping.

- *No application modifications required.* As mentioned in section 5.2.4, transport layer multi-homing mechanisms typically require modifications of existing applications, but it is generally difficult and sometimes impossible to modify existing applications. Therefore, the mechanism should not require any changes to existing applications to support multi-homing.

- *Support of Mobility Transparency.* Since a mobile node moves around different wireless networks, it is crucial to support mobility transparency to provide sustained network connectivity to mobile nodes.

- *Capability of handling dynamic address assignment.* In heterogeneous wireless networks, network interfaces of mobile nodes frequently become active or inactive in accordance with the context of the mobile node. The mechanism should handle these *network events*, such as interface up/down, IP address assignment/removal, in order to provide optimized performance of wireless networks.

- *Fairness.* The mechanism should work without any harmful effect on cross traffic on wireless networks. In other words, the mechanism should perform appropriate congestion control on each wireless network. This requirement motivates us to employ transport layer multi-homing for our mechanism. Note that we assume that wireless networks are the bottleneck links between a mobile node and a correspondent node since the backbone network of the Internet generally has a large bandwidth relative to wireless networks. In other words, we assume that there are no shared bottleneck links when a mobile node communicates with a correspondent node with multiple connections and this does not violate fairness of bandwidth utilization.

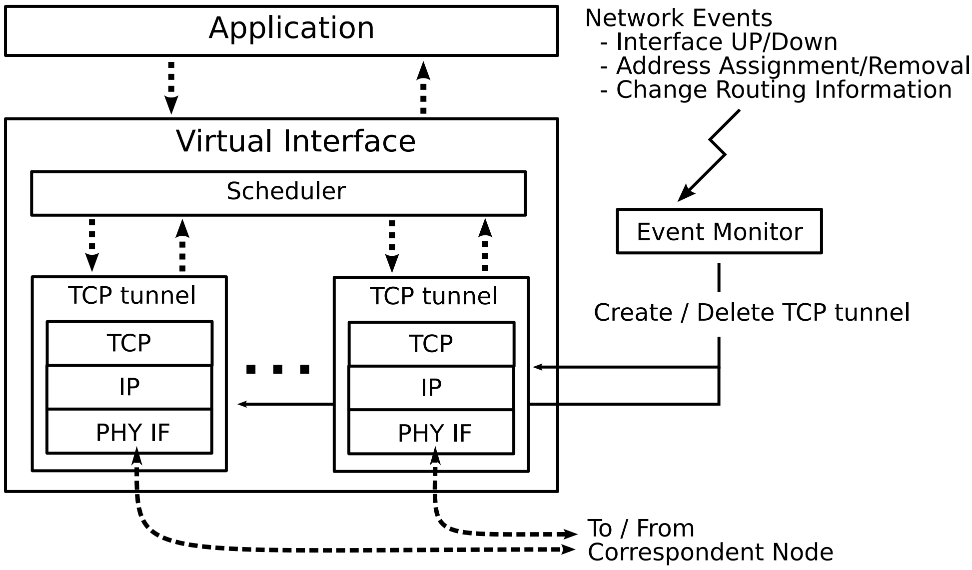


Figure 5.2: A conceptual design of a multi-homing mechanism

5.3 Design

This section describes the proposed multi-homing mechanism designed for achieving the goals discussed in section 5.2.5. We firstly present an overview of the design and subsequently describe details of each component.

5.3.1 Overview

Figure 5.2 depicts a conceptual design of the proposed multi-homing mechanism. The mechanism behaves as a *virtual interface*. Since the mechanism takes an end-to-end approach, both a mobile node and a correspondent node must install this virtual interface. The virtual interface contains multiple TCP connections, which are established on each active real interface. These TCP connections act as TCP tunnels, which are used for communicating with a correspondent node. The mechanism must create a TCP tunnel on a physical interface when the interface becomes *active*, which means at least one IP address is assigned to the interface and the interface can be used to connect to the Internet. The mechanism must also destroy a TCP tunnel on a physical interface when the interface becomes *inactive*, which means the interface is down, or there is no assigned IP address on the interface, or there is no routing information for the interface. The *event monitor* component creates and destroys TCP tunnels in accordance with occurrences of network events, such as interfaces up/down, an address assignment to an interface, or changing routing information. Section 5.3.3 will describe the role of the event monitor component.

When an application delivers a packet to the virtual interface, the scheduler component selects the best connection to transmit the packet in accordance with the strategy which section 5.3.2 will describe; then the packet is encapsulated and transmitted via the selected connection to a correspondent node. At the receiver side, the virtual interface decapsulates the packet and then delivers it to the destination application according to the packet header.

5.3.2 Data Striping Strategy

As discussed in section 5.2.2, a naive data striping strategy does not work effectively in heterogeneous wireless networks due to differences in transmission speeds and jitters. Since packet reordering reduces application-level throughput, the mechanism should employ some kind of jitter-tolerant data striping strategy. In addition, from the viewpoint of practical use, a data striping strategy should work without any configuration. Consequently, we propose a jitter-resistant and non-policy-based data striping strategy. The basic behavior of the strategy is similar to that of the active/backup strategy. It usually selects the fastest connection to send packets, while other relatively slow connections are kept in reserve. So that no configuration is required, a TCP congestion control mechanism is applied to the strategy; the strategy dynamically calculates a *priority metric* for each connection by using RTT and cwnd. The strategy uses calculated values to determine the priority of each connection and the highest priority connection will be used for packet transmission.

To show how to determine the connection for sending packets and how the metric values are calculated, we introduce some variables. Let $\mathcal{C}$ be the set of TCP connections established for each active interface. A member of $\mathcal{C}$ is denoted by c. At any time t, the RTT and the cwnd of c are denoted by $rtt_c(t)$ and $cwnd_c(t)$, respectively. The metric for c is calculated recursively and denoted by $m_c(i_c)$, where i_c is the index of the metric for c. The index i_c is set to be zero when c is established and the index is incremented each time the metric is updated.

It is important to note that low values for the metric mean that connections have desirable characteristics, i.e., their priority is high. The strategy calculates a low value for the metric for a connection when it has short RTT and large cwnd. Therefore, at any time

t, the best connection for sending a packet, which is denoted by $sc(t)$, can be determined by:

$$sc(t) = {}^{\forall}m \in \mathcal{M} \ where \ \mathcal{M} = \{c \in \mathcal{C} | \min_{c \in \mathcal{C}}(m_c(i_c))\} \tag{5.1}$$

The metric for c is initialized and updated as follows. Once a connection c has been established at time t, the metric is initialized as:

$$m_c(i_c = 0) = \frac{rtt_c(t)}{cwnd_c(t)} \tag{5.2}$$

At this point, $rtt_c(t)$ and $cwnd_c(t)$ are set to the values immediately after the three-way handshake. Thus, formula (5.2) initializes the metric to a low value when c has been established with desirable characteristics. The metric for c is updated when the strategy selects c as the connection to send packets. It is increased temporarily when c sends a packet and restored when the packet is successfully acknowledged. Therefore, as long as c can send packets and can receive acknowledgements, the metric for c will remain at the value calculated by the RTT and the cwnd of c. On the other hand, when c is no longer able to receive acknowledgements, the metric will continue to increase. By using this approach, the strategy can calculate priorities based on both connectivity and communication speed of the connection. The amount by which the metric changes is also determined by the RTT and the cwnd of the connection. When c sends a packet at time t, the metric for c is increased temporarily using formula (5.3).

$$m_c(i_c + 1) = m_c(i_c) + \frac{rtt_c(t)}{cwnd_c(t)} \tag{5.3}$$

When c successfully receives the acknowledgement of a packet transmitted at time t, the metric for c is restored using formula (5.4).

$$m_c(i_c + 1) = m_c(i_c) - \frac{rtt_c(t)}{cwnd_c(t)} \tag{5.4}$$

In this manner, as long as the fastest connection can transmit packets normally, that is, the connection continuously receives acknowledgements, the strategy continues to select the fastest connection to transmit packets. However, when the connection no longer receives acknowledgements, the priority of the connection continues to decrease, and eventually the strategy selects another connection to transmit packets.

5.3.3 Event Monitor

The role of the event monitor is to monitor occurrences of network events. A *network event* is an event that changes the network configuration of a mobile node. For example, interface up/down, IP address allocation/deallocation, and changing of routing information are typical network events. The event monitor must detect the occurrence of a network event and handle the event to create and to destroy TCP tunnels. When an interface becomes active and obtains connectivity to the Internet, the event monitor gives a direction to the virtual interface to establish a TCP connection for the activated interface. Likewise, when an interface becomes inactive, the event monitor gives a direction to the virtual interface to close the TCP connection for the inactivated interface. Note that the event monitor must detect not only IP address assignment to an interface but also changes of routing information since IP address assignment does not always mean the interface has connectivity to the Internet.

Applications using the virtual interface need not recognize the creation and destruction of TCP tunnels. TCP tunnels are automatically created and destroyed by the virtual interface depending on the occurrence of network events. These TCP tunnels are automatically used by the virtual interface in the manner of the data striping strategy, which was described in section 5.3.2. An application simply sends and receives packets through the virtual interface to use the proposed mechanism. Therefore, creation and destruction of TCP tunnels are hidden inside the virtual interface.

Because a virtual interface is constructed for each correspondent node, when multiple applications communicate with a correspondent node through the virtual interface, a TCP tunnel established on each physical interface is shared among the applications. The virtual interface could establish an individual TCP tunnel for each application to avoid sharing. However, it does not do so, because such individual TCP tunnels will still share the same network path. Therefore, even if the virtual interface were to establish individual TCP tunnels for each application, if one such TCP tunnel were to become slow, other TCP connections would also become slow. The virtual interface consequently does not establish TCP tunnels for each application.

5.3.4 Dealing with TCP over TCP Problems

The virtual interface behaves as a TCP over TCP tunnel when an application employs TCP as its transport protocol. The mechanism should take account of the problems involved in TCP over TCP tunneling since they will result in poor performance unless countermeasures are taken [51]. A serious problem is the interference in congestion control and retransmission control between upper and lower levels of TCP. This causes significant reduction in application-level throughput. To handle the problem, the mechanism will try to disable congestion control and retransmission control at the upper TCP level. The semantics of the upper level TCP are not violated even if these features are disabled since the lower level TCP provides them.

Another problem is the overhead of packet encapsulation. At least an extra 40 bytes (sum of the IP header size and the TCP header size) are contained in the outer packets. However, it is difficult to reduce this overhead since it is essential for constructing TCP tunnels. Hence, we do not try to reduce this overhead in this research. We will investigate the impact of this overhead in section 5.5.3.

5.3.5 Validity of Design

Here we discuss whether our mechanism satisfies the goals discussed in section 5.2.5. The mechanism does not assume any characteristics of data links. Therefore, the mechanism can be applied to any wireless technology. As described in section 5.3.2, the proposed data striping strategy does not require any configuration since it automatically calculates priorities of each TCP connection and selects the best connection to transmit packets. The mechanism also does not require any modifications of existing applications since the mechanism behaves as a virtual interface. An application just sends or receives packets through the virtual interface in the usual way, such as using BSD socket APIs. The event monitor collaborates with the virtual interface to provide mobility transparency and to handle dynamic address assignment. Since an application uses the virtual interface instead of using real interfaces directly, the application need not consider whether real interfaces are active or not. The responsibility for handling the condition of real interfaces is taken by the event monitor. Finally, transport layer multi-homing, which is employed by our mechanism, shares network capability fairly, as described in section 5.2.3.

5.4 Implementation

The primary goal of our research is to create an appropriate multi-homing mechanism for actual mobile environments. Therefore, to implement the proposed mechanism in an actual operating system is an essential part of our research.

Virtual Interface The proposed mechanism has been implemented as a virtual interface in the Linux kernel 2.6.28.9. Since the virtual interface is implemented as a kernel module, we need not modify the Linux kernel itself to install the mechanism. This provides users with easy installation. The virtual interface can be created or deleted by using the command interface, which we will describe below. To deal with the TCP over TCP problems described in section 5.3.4, the virtual interface hooks the establishment of a TCP connection to disable congestion control and retransmission control of the connection. This feature is implemented by sniffing the context of the connection and taking over the congestion control functions of the connection.

Command Interface The *ioctl* system call is employed as the interface for userland programs to communicate with our mechanism. We defined some new ioctl request codes for our mechanism. They are used for: (1) creating and deleting the virtual interface, and (2) communicating network events to the virtual interface. This command interface is generally used by the event monitor.

TCP Tunnels In the current implementation, TCP tunnels employ a modified version of CUBIC [52], which includes functions of the data striping strategy described in section 5.3.2 as its congestion control, in accordance with the default congestion control of the Linux operating system. The congestion control scheme of TCP tunnels could be changed to other schemes such as Reno by using the ioctl system call if this was desired. Fortunately, the Linux kernel has many congestion control schemes. We just need to add our data striping strategy functions to them if we wish to use them.

Event Monitor The event monitor component runs as a userland daemon program. It employs the Linux Netlink interface[53] to observe network events. The Linux Netlink has sufficient capabilities for the event monitor: it can detect interface up/down, IP address assignment/removal, change of routing information and other such events. When the monitor component detects a network event, it requests the virtual interface to handle the event appropriately by using an ioctl system call.

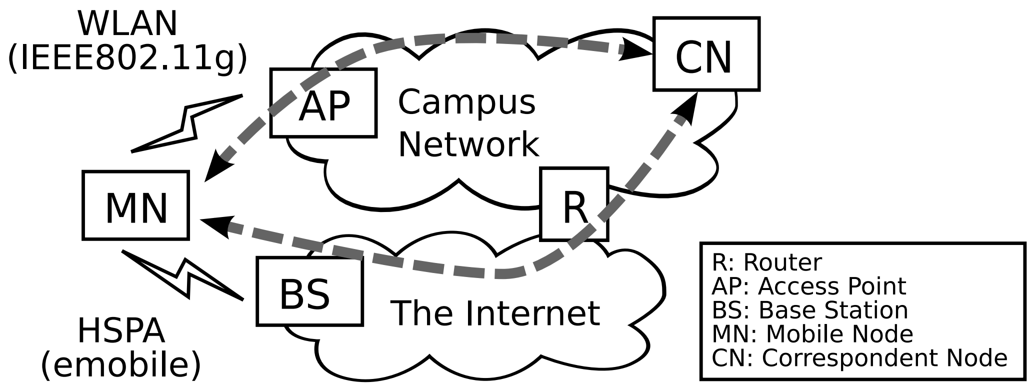


Figure 5.3: Experimental Environment

5.5 Evaluation

Since the mechanism is designed for practical use, it should be evaluated in an actual environment. Evaluating the mechanism in an actual environment is one of the contributions of our research. Most previous proposed mechanisms have been evaluated by simulations or in emulation environments. Figure 5.3 shows the evaluation environment that we used to conduct experiments. A mobile node (MN) has two wireless interfaces: WLAN (IEEE802.11g) and HSPA (emobile). To provide WLAN access to the MN, an access point (AP) was placed in our campus network. A correspondent node (CN) was also placed in our campus network. As a result, packets are sent via WLAN only through our campus network, while packets are sent via HSPA through both the Internet and our campus network.

5.5.1 Handling Network Events

One of the significant features of our mechanism is providing zero configuration data striping, which automatically handles network events to obtain optimized performance. To make sure the mechanism could handle network events in an effective manner, we conducted the following experiments.

Handling Interface Down/Up Events We conducted the following experiment to investigate whether the mechanism could choose the fastest data-link automatically. We set both WLAN and HSPA to be active at the beginning of the experiment. The MN sent 100 ping packets to the CN at intervals of 1 second. Each ping packet contained a 1000-

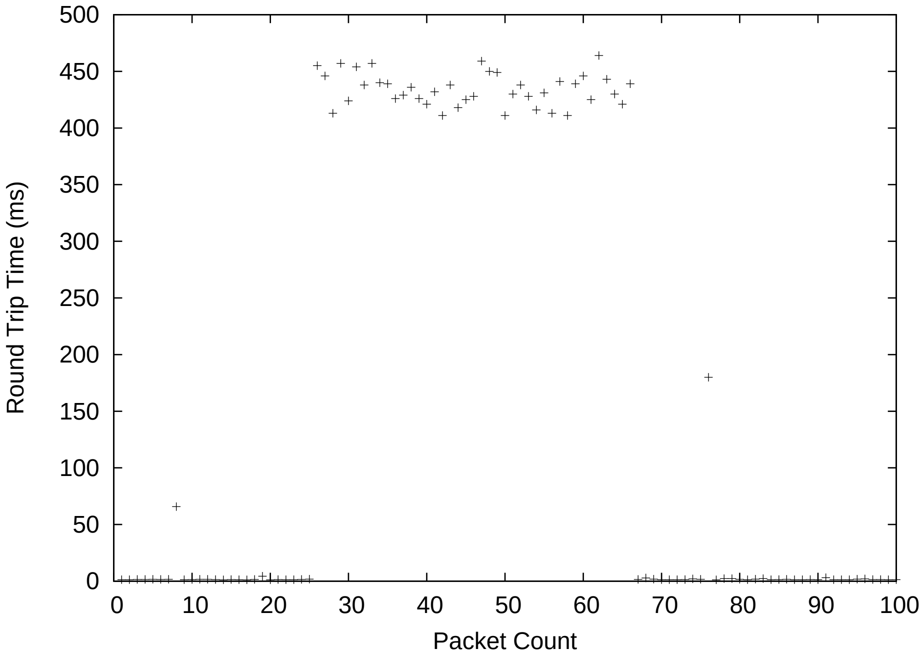


Figure 5.4: RTT of 100 ping packets with WLAN down/up

byte payload. During the transmission, the hardware switch for WLAN on the MN was turned off to make WLAN inactive, and after a certain period of time, the switch was turned on to reactivate WLAN.

Figure 5.4 shows the observed RTT of each ping packet. RTT for each packet indicates that the mechanism selected WLAN as the primary interface to send packets as long as WLAN was active. Figure 5.4 also shows that once WLAN became inactive, the mechanism immediately switched the primary interface to HSPA with no packet loss. This behavior is appropriate in this environment since the RTT of WLAN is significantly shorter than the RTT of HSPA. Note that this behavior was achieved automatically. The mechanism does not require any configuration and does not need preliminary interface information.

Interface Switching Speed To investigate how fast the mechanism was able to handle network events, we conducted almost the same experiments as above. However, in these experiments, we did not reactivate WLAN. Transmission intervals were also changed; the MN sent ping packets at intervals of 10^{-1}, 10^{-2}, 10^{-3}, and 10^{-4} seconds in each experiment. To ensure enough time to make WLAN inactive, the MN sent 1,000 ping packets

Table 5.1: Packet loss rates with WLAN down

Interval (sec)	Received packets/Sent packets	Loss rate
10^{-1}	100/100	0%
10^{-2}	999/1000	0.1%
10^{-3}	992/1000	0.8%
10^{-4}	990/1000	1%

to the CN when the transmission interval was greater than 10^{-1} seconds. In these experiments, we measured packet loss when the primary interface became inactive. Table 5.1 shows packet loss rates for each transmission interval. No packet loss was observed when the transmission interval was 10^{-1} seconds. Even when the transmission interval was 10^{-4} seconds, the packet loss rate was only 1%. These results indicate that the mechanism was able to handle network events very quickly.

Handling Signal Attenuation When a mobile node moves away from the communication range of the WLAN, the communication speed of the WLAN typically becomes progressively worse, and eventually the WLAN becomes inactive. The following experiment was conducted to investigate how fast the mechanism switched over the primary data-link from WLAN to HSPA in such a situation. At the beginning of the experiment, we let both WLAN and HSPA be active. As with the above experiment, 100 ping packets were transmitted from the MN to the CN. During transition, the MN moved out of communication range of the WLAN, and then returned within range. Figure 5.5 shows the observed RTT of each ping packet. As the MN left the WLAN's communication range, a total of 20 packets were lost because the mechanism took some time to detect the loss of WLAN's connectivity in this situation. However, the mechanism was eventually able to switch the primary interface to HSPA in the way described in section 5.3.2. Section 5.6 will discuss how this packet loss could be reduced. Although not a few packet losses were observed when the MN moved out of range of the WLAN, no packet loss was observed when the MN turned back within the range of the WLAN. This behavior indicates that the mechanism was able to switch its primary interface to the fastest data-link with no packet loss once a faster data-link became active.

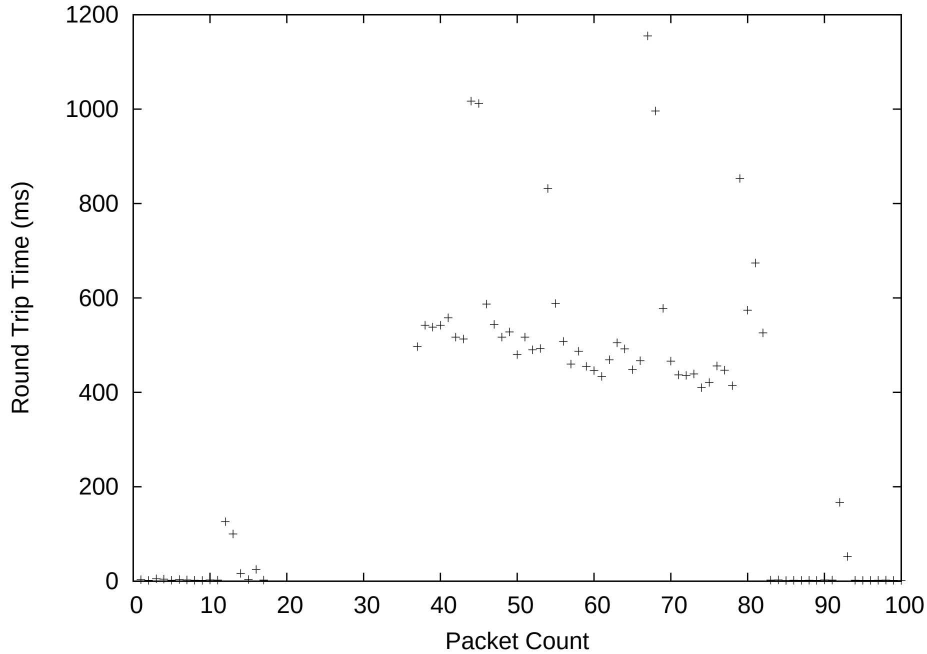


Figure 5.5: RTT of 100 ping packets with signal attenuation

5.5.2 Performance

To examine the performance of our mechanism, we implemented a rate-based data striping strategy as a competing strategy for the proposed data striping strategy. The rate-based strategy stripes packets over multiple TCP connections in proportion to their cwnd values. This strategy is almost the same as the strategy employed by ARMS [35]. We measured the throughput of each strategy in the evaluation environment using Iperf [54]. Throughput measurements were conducted in the following two cases: (1) both WLAN and HSPA were active during the measurement, (2) only HSPA was active at the beginning of a measurement and WLAN became active during the measurement.

First, we measured throughput in the case that both WLAN and HSPA were active during measurement. The measurements were conducted 50 times for each strategy and the average throughput calculated. Table 5.2 shows the average throughput, maximum throughput, and minimum throughput for each strategy. The throughput for the cwnd-based strategy was significantly lower than that for the proposed strategy. This performance degradation was caused by the head-of-line blocking problem, mentioned in section 5.2.2. The cwnd-based strategy sends a certain amount of packets via the connection

Table 5.2: Throughput when WLAN became active during measurements

Strategy	Average	Max	Min
proposed	12.6 Mbps	14.0 Mbps	10.8 Mbps
cwnd-based	567.7 Kbps	661.0 Kbps	381.0 Kbps

established on HSPA. However, almost none of these packets can arrive at the CN in sequence since later packets, which are sent via the connection established on WLAN, will arrive at the CN before them due to a significant difference in the RTT for WLAN and HSPA. As a result, an appreciable amount of packet reordering occurs and application-level throughput is drastically decreased.

Next, we measured throughput in the case that only HSPA was active at the beginning of a measurement and WLAN became active during the measurement. The measurements were conducted 10 times for each strategy. Table 5.2 shows the average throughput, maximum throughput, and minimum throughput for each strategy. For the proposed strategy, overall throughput in this scenario was lower than for the previous case since WLAN was inactive at early stages of the measurement. However, the proposed strategy was able to achieve significantly larger throughput than the cwnd-based strategy since the proposed strategy used WLAN as the primary data-link once WLAN became active. On the other hand, similar results were observed as for the previous case for the cwnd-based strategy. Such significantly lower throughput of the cwnd-based strategy was also caused by the head-of-line blocking problem in this scenario also.

5.5.3 Tunneling Overhead

Since the mechanism employs TCP tunneling, a packet sent through the mechanism has an extra payload, namely, an inner packet header. To investigate the impact of this overhead, we compared throughput of a non-tunneling connection (referred to as a *raw connection*) with throughput of a tunneling connection (referred to as a *tunneling connection*). We used Iperf again to measure throughput and conducted measurements 50 times. For these measurements, we made only HSPA active to eliminate other factors that affect measurements of throughput. Figure 5.6 shows the average throughput of a raw connection and a tunneling connection. The average throughputs of raw and tunneling connections were 359.7 Kbps and 346.4 Kbps, respectively. The degradation of throughput is 3.7%. It seems that this is a reasonable overhead since the mechanism provides mobility transparency,

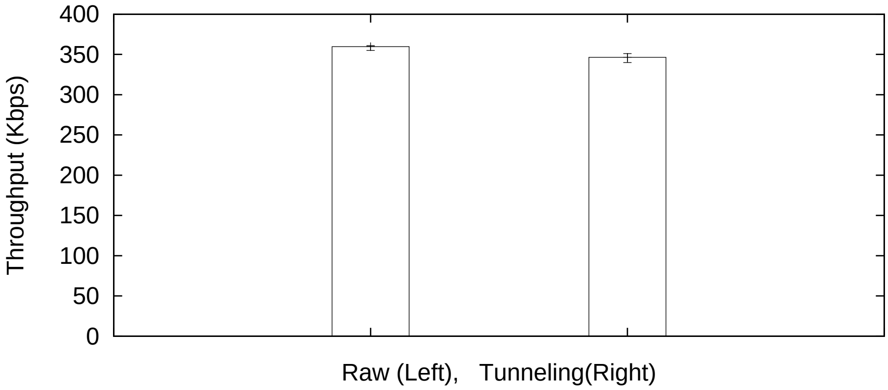


Figure 5.6: Throughput overhead of TCP tunneling

Table 5.3: Effects of disabling upper TCP features

Upper TCP features	Average	Max	Min
disabled	18.3 Mbps	19.1 Mbps	16.3 Mbps
enabled	17.9 Mbps	18.7 Mbps	15.6 Mbps

and it is impossible to provide mobility transparency without any extra information.

As we described in section 5.3.4, when an application employs TCP as its transport layer protocol, the mechanism tries to disable congestion control and retransmission control at the upper TCP level to prevent performance degradation caused by TCP over TCP problems. We investigated the effect of disabling such upper TCP features by measuring throughput in each case of the upper TCP features being disabled or not. The measurements were conducted 10 times for each case. In these measurements, we made only WLAN active to eliminate other factors that affect measurements of throughput. Table 5.3 shows the average, the maximum, and the minimum throughput for each case. The throughput for the case in which upper TCP features were enabled is slightly lower than for the case in which the features were disabled. It seems that interference between the upper TCP features and the lower TCP features involved lowered the throughput when the upper TCP features were enabled.

5.5.4 Impact of different MTU Sizes

In the proposed data striping strategy, most of the packets are transmitted through the fastest TCP tunnel, but a few packets may be transmitted through slower TCP tunnels.

Table 5.4: Throughput comparison with different HSPA MTU sizes

MTU size	Average	Max	Min
1500	23.2 Mbps	24.5 Mbps	18.3 Mbps
500	18.6 Mbps	23.9 Mbps	12.2 Mbps

In this situation, a head-of-line blocking problem may occur when the MTU size of each physical interface is different. If a large packet is transmitted through a TCP connection which was established on a physical interface that had a small MTU size, the packet may arrive at a correspondent node after successive packets which were transmitted through another TCP connections. To investigate the impact of different MTU sizes of each physical interface, we measured throughput in both cases that the MTU size of HSPA was set to 500 and 1500. Throughout the measurement, both WLAN and HSPA were active and the measurements were conducted 10 times for each case. Table 5.4 shows the average, the maximum, and the minimum throughput for each case. The throughput for the case where the HSPA MTU size was set to 500 was relatively lower than for the case of an HSPA MTU size of 1500. This performance degradation indicated that a head-of-line blocking problem can occur when the MTU size of each physical interface is different. We will discuss how we might cope with this problem in section 5.6.

5.6 Discussion

The proposed mechanism is able to provide an applicable multi-homing strategy for heterogeneous wireless networks, as shown in the previous section. However, there are several drawbacks of the proposed mechanism and problems to be solved. In this section, we discuss the drawbacks and problems of the proposed mechanism.

Scalability Although the virtual interface provides a multi-homing mechanism for existing applications without any modification, the virtual interface must be placed in each correspondent node. This is an undesirable limitation with regard to deployment and scalability.

However, we can ease the limitation by adding mapping information about correspondent nodes and introducing an automatic TCP tunnel generation mechanism into the virtual interface. Suppose a mobile node equipped with the proposed mechanism transmits a packet whose destination is *CNAddr*. The virtual interface checks whether mapping information associated with *CNAddr* exists or not. If the mapping information does not exist at that time, the virtual interface establishes TCP tunnels for *CNAddr* on each active physical interface and creates the mapping information associated with *CNAddr*. Otherwise, the mapping information already exists, TCP tunnels for *CNAddr* have been established, therefore the virtual interface will be ready to transmit the packet in this case. Once TCP tunnels for *CNAddr* have been established, the virtual interface can transmit the packet through the tunnel which is determined by the packet striping strategy described in section 5.3.2. When a network event (e.g. interface up/down) occurs, the virtual interface establishes or destroys TCP tunnels depending on the event. In addition, when there is no packet associated with *CNAddr* transmits or receives for a certain period of time, TCP tunnels for *CNAddr* will be destroyed. By introducing these features into the virtual interface, the limitation can be avoided.

Tunneling Overhead TCP tunneling, which is employed by the virtual interface, also has a drawback. It involves a certain per-packet overhead, as mentioned in section 5.3.4. In this research, we took advantage of the virtual interface at a sacrifice of scalability and performance since providing multi-homing for existing applications without any modification was a primary purpose of our research. It is difficult to reduce the per-packet overhead since tunneling involves such overhead essentially. However, we can take alternative solutions to reduce packet length such as packet compression.

Reducing packet loss during data-link switching As the experimental result revealed, when the fastest data-link gradually becomes unavailable, the mechanism would take some time to switch the primary data-link and this would result in some packet loss. There are two ways to reduce such packet loss. One is to retransmit packets which are sent but not acknowledged through the second-best data-link when the fastest data-link seems to be unavailable. Another is to improve the behavior of the data striping strategy when acknowledgements are no longer received. A promising idea for improving the strategy is to decrease the priority of a TCP tunnel exponentially for each segment transmission when the tunnel does not receive acknowledgements.

Bandwidth aggregation Generally, the proposed data striping strategy sends packets through only the fastest connection in a stable state. Consequently, the strategy may not achieve an optimized aggregate throughput. Even if the effect of bandwidth aggregation is a side benefit in heterogeneous wireless networks, as discussed in section 5.2.2, the strategy should be improved to generate the full benefit of multi-homing.

Supporting shared bottlenecks The proposed data striping strategy assumes that there is no shared bottleneck link in the backbone network. This assumption is typically considered reasonable for heterogeneous wireless networks. However, if a shared bottleneck link exists, the strategy may violate fair sharing of the bottleneck link since the strategy transmits data over multiple connections and such transmission is not compatible with TCP friendliness [55]. The strategy should be refined to be compatible with TCP friendliness even if a shared bottleneck link exists.

Reducing impact of different MTU sizes The experimental result described in section 5.5.4 showed that the proposed data striping strategy may cause performance degradation when the MTU sizes of each physical interface are different. Since the degradation is caused by a head-of-line blocking problem, we should consider a way to reduce the occurrence of the problem. One of the ways to cope with the problem is enhancing the data striping strategy to make the strategy unlikely to select a TCP connection which is established on a physical interface that has a small MTU size. For this purpose, the MTU size of each physical interface should be as a parameter of the calculation of the priority metric for TCP connections.

5.7 Summary

This Chapter has proposed a practical multi-homing mechanism for heterogeneous wireless networks. The mechanism behaves as a virtual interface and aggregates each real interface by using TCP connections. The contributions of our research are: (1) providing multi-homing support to existing applications without any modification, and (2) developing a non-policy-based jitter-resistant data striping strategy. These things are important for providing multi-homing support to mobile nodes in practical settings. We implemented the mechanism in the Linux operating system and conducted some experiments in an actual environment. The results show that our mechanism has less performance overhead and can handle network events very quickly.

Chapter 6

Towards Appropriate Utilization of Heterogeneous Wireless Networks

This dissertation has tackled improving TCP adaptability for different link characteristics to enhance the communication quality for heterogeneous wireless networks. The last two Chapters presented different approaches to enhance TCP adaptability for heterogeneous wireless networks. This Chapter considers whether the proposed schemes satisfy the design criteria for the research objective of this dissertation. This Chapter also discusses the open issues towards the appropriate utilization of heterogeneous wireless networks.

6.1 Reviewing the Research Contributions

This section reviews whether the proposed methods successfully achieve the research objectives and provide solutions for the issues discussed in section 1.4.

First, we review the method presented in Chapter 4, called Teppi. As section 1.4 mentioned, the motivation for TCP adaptation is to provide solutions to both avoiding network congestion and improving transitional behavior of the TCP congestion control mechanism in heterogeneous wireless networks. For the evaluation of avoiding network congestion, an experiment which investigated the impact of a vertical handoff of an existing TCP connection on the after-handoff data-link was conducted. When Teppi was disabled, which means that the mobile node does not do anything at the occurrence of the vertical handoff, some segment retransmissions were observed on the existing TCP connection. This behavior implies some network congestion. In contrast, when Teppi was enabled, there were no segment retransmissions observed on the existing TCP connection. This fact implies that Teppi successfully avoided generating network congestion. With regard to the improvement of transitional behavior, Teppi was evaluated by throughput measurements, which were described in section 4.6.5. The results showed that Teppi can contribute to short-term throughput improvement after vertical handoffs. These facts together imply that Teppi can adapt the TCP congestion control mechanism to the after-handoff data-link immediately and can improve transitional behavior of the TCP congestion control mechanism. The results also showed that Teppi does not contribute to throughput improvement for a long-term data transmission. This behavior comes about as a natural result because the TCP congestion control mechanism will eventually adapt its behavior to the after-handoff data-link even if the TCP congestion control parameters are not set to suitable values when vertical handoffs occur. However, TCP adaptation is still important because it prevents the occurrence of network congestion and contributes to a short-term throughput improvement.

Next, we review the method presented in Chapter 5, called Luct. Luct provides a multi-homing mechanism to eliminate the necessity of conducting vertical handoffs. Since the primary factor of drastic changes in link characteristics is conducting vertical handoffs, eliminating the necessity of conducting such vertical handoffs is one of the ways to adapt TCP for heterogeneous wireless networks. However, introducing a multi-homing mechanism in itself does not provide the function of TCP adaptation. To provide a TCP adaptation function by introducing a multi-homing mechanism, the multi-homing mechanism has to manage the TCP congestion control context separately for each data-link. As described in section 5.2.3, when the multi-homing mechanism does not manage the TCP congestion control context separately, further mechanisms for TCP adaptation have to be constructed. For managing the TCP congestion control context separately, Luct employs TCP tunnels. The TCP congestion control context for each data-link is managed by each TCP tunnel that is established for the data-link. Therefore, Luct presented in Chapter 5 also contributes to some extent in enhancing TCP adaptability for heterogeneous wireless networks.

6.2 Reviewing the Design Criteria

Design criteria for developing TCP adaptation schemes for heterogeneous wireless network were specified in the beginning of this dissertation. This section investigates whether the proposed methods satisfy the four design principles: keeping the design concept of the Internet, ease of integration with other technologies, fairness, and applicability to real environments.

Keeping the Design Concept of the Internet Teppi resets the congestion control parameters after vertical handoffs. Since this behavior can be achieved by modifying only the internal state of the sender side TCP, it does not violate the end-to-end principle. In addition, since such modification does not change the behavior of the congestion control mechanism in the stationary state, the semantics of TCP will be kept. Therefore, the independence property of TCP is not changed even if Teppi is adopted.

Luct provides a multi-homing mechanism as a virtual interface. The virtual interface must be installed in both end nodes to use Luct, but nothing further is required. The modification required by the proposed method is enclosed within end nodes and therefore the end-to-end principle still obtains when Luct is employed. In addition, since Luct employs TCP tunnels to provide a multi-homing mechanism, the independence property of Luct depends on which congestion control algorithm is adopted for TCP tunnels. Therefore, the independence property of Luct can be retained by adopting a generic congestion control algorithm for the TCP tunnels.

Ease of Integration with Other Technologies The contribution of the proposed methods is to provide TCP adaptation features to mobile nodes on heterogeneous wireless networks. A solution to TCP adaptation is a research result that will lead towards the effective utilization of heterogeneous wireless networks. Therefore, ease of integration with technologies which provide solutions to other research targets is an essential design criterion. If a method is independent of any other technologies and does not require any modification of other technologies, the scheme can be integrated with other technologies.

Teppi requires information about the occurrence of vertical handoffs. This information can be provided by the Internet layer or the Link layer technologies and therefore it seems that Teppi has moderate dependence on vertical handoff notification mechanisms. However, once Teppi obtains information about occurrences of vertical handoffs, Teppi executes the TCP adaptation process internally. Teppi does not require any modification to other technologies. Therefore, the proposed method can be integrated with other technologies easily once vertical handoff notification mechanisms are appropriately abstracted.

Luct acts as a virtual interface and can be used in a general way, for example, by using Berkeley sockets APIs. Existing network applications can use Luct with no modification. In addition, the event monitor of Luct automatically detects network events such as interface up/down and handles them in an appropriate way. Luct does not require any support to be provided by other technologies to work. These features of Luct contribute to ease of integration with other technologies.

Fairness Since Teppi resets the congestion control parameters when a vertical handoff occurs, Teppi can be considered as based on the approach that it changes the internal state of TCP for adapting TCP to drastic changes in link characteristics. This approach does not modify the congestion control mechanism itself and consequently it retains the same behavior from the viewpoint of fairness in the stationary state. Even in a transient state, in other words, immediately after conducting a vertical handoff, Teppi can be considered to use network resources fairly. Teppi sets the congestion control parameters to their initial values after occurrences of vertical handoffs. This behavior can be considered as the creation of a new TCP connection on the after-handoff data-link. Creating new TCP connections are ordinary events and the TCP congestion control mechanism is designed to use network resources fairly among all connections on the network even if new TCP connections are established afterward.

Luct establishes TCP tunnels on each available network interface and stripes packets over these TCP tunnels to provide a multi-homing mechanism. From the viewpoint of the Link layer, Luct just establishes a TCP connection on each wireless data-link. Therefore, Luct can use network resources fairly whenever it adapts a generic congestion control algorithm to the TCP tunnels.

Applicability to a Real Environment Even if a method provides elegant, sophisticated, and novel solutions for TCP adaptation on heterogeneous wireless networks, it does not make sense if the method is hard to apply in a real environment. Simplicity and fewer assumptions are desirable characteristics from the viewpoint of application to real environments. The design concept and ideas of the proposed two methods described in this dissertation are reasonably simple to apply to real environments. Actually, these two methods have been implemented in actual operating systems and evaluated in an emulation environment and also real environments. As for Teppi, it requires the following two minor modifications of the TCP congestion control mechanism: introducing a new congestion control parameter called *tsmax* and adding a conditional expression which determines the point where TCP should reset the congestion control parameters. These modifications are easily applied to actual TCP/IP implementations. However, an assumption of Teppi, namely, that a sender can detect the occurrence of vertical handoffs immediately, is somewhat hard to satisfy in the real environment. The following section 6.3 discusses the difficulty and possible schemes for developing handoff notification mechanisms. As for Luct, it can be implemented by using functions which are provided by existing TCP/IP implementations and no modification of the TCP/IP stack is needed internally. In fact, Luct has been implemented as a Linux kernel module and it can be installed with no modification of the Linux kernel itself.

6.3 Open Issues

Although the contributions of this study are an essential part of a comprehensive study on the effective utilization of heterogeneous wireless networks, they are still just a part of such a comprehensive study. The ultimate goal of a comprehensive study on enhancing the communication quality of mobile nodes on heterogeneous wireless networks is to establish an integrated communication system in which technologies on each network layer cooperate with others to optimize the communication of mobile nodes. This section discusses the open issues to be resolved to accomplish the ultimate goal of the study.

Integration of Individual Technologies As mentioned above, not only the development and improvement of individual technologies of each network layer is required; it is necessary also to integrate all sorts of such technologies to optimize the overall performance of the communication system of mobile nodes. This dissertation has grappled with transport layer adaptation, which is one of the key technologies to achieve the ultimate goal of a comprehensive study on heterogeneous wireless networks. The contributions of this dissertation should be integrated with other technologies, such as physical layer enhancement technologies, mobility support protocols, handoff strategies, handoff notification mechanisms, high-speed handoff mechanisms, and application adaptation mechanisms.

It should be noted that the integration should not interfere with each constituent technology. Even though each individual technology may achieve a significant performance benefit considered by itself, it may not always lead to overall optimization, especially when the technology involves side effects. For example, consider a physical layer enhancement technology which caches transmitted data-frames and retransmits a cached data-frame secretly when the data-frame is lost. This technology guards against the fragile wireless link quality and would provide high throughput for mobile nodes. This technology works well as long as it is integrated with connectionless transport protocols, such as UDP. However, once the technology is integrated with the connection-oriented transport protocol, that is TCP for instance, side effects of the technology interfere with TCP. When frame retransmission is invoked in the physical layer secretly, it appears as large jitter characteristics or an occurrence of segment reordering to TCP. This side-effect could injure TCP retransmission control and congestion control, as section 4.2 described.

Handoff Notification Mechanism Teppi assumes that a sender can obtain information about the occurrences of vertical handoffs. The method consequently requires a mechanism which notifies the occurrence of a vertical handoff to a sender. The evaluation experiments conducted in section 4.6 added a signaling network for this purpose. However, such a workaround is obviously unacceptable for actual heterogeneous wireless networks. Therefore, a handoff notification mechanism is required, which observes occurrences of vertical handoffs and notifies network nodes interested in such occurrences. Since any network node can be a sender, the mechanism should cover the following cases:

1. A node which conducts the vertical handoff is also a node interested in it. In this case, the node has to acquire the information about vertical handoffs from its physical layer components. This is the minimum requirement of a handoff notification mechanism.

2. A correspondent node of a node that conducts a vertical handoff is also a node interested in it. In this case, a kind of method to transmit information about vertical handoffs to correspondent nodes is required, in addition to the requirement of Case 1. The notification transmission method should take account of the transmission delay of the notification.

3. The node conducting a vertical handoff is an intermediate node of nodes interested in the vertical handoff. This case could occur in a NEMO environment. When the in initiator of vertical handoffs is a mobile router, both mobile network nodes and correspondent nodes cannot recognize the occurrence of the vertical handoff in spite of the fact that they will be affected by the vertical handoff. In this case, a notification

transmission method is also required to notify occurrences of vertical handoffs to mobile network nodes and correspondent nodes. In addition, the intermediate node has to know which nodes are interested in the vertical handoff. The intermediate node should inform nodes interested in vertical handoffs about such occurrences in an appropriate manner.

It is difficult to develop a handoff notification mechanism which can cover all these cases in an effective manner. Some efforts have been made to develop handoff notification mechanisms [56, 57, 58]. However, most of these mechanisms do not consider cases 2 and 3. In cases 2 and 3, a crucial issue to establish an effective mechanism is how the mechanism should transmit notifications. When a notification is delayed, the notification may not make sense in some situations in which immediacy of notification is essential. Avoiding occurrences of network congestion is one such situation, as described in section 4.7. Since the transmission delay of the notification cannot be reduced to zero, some kind of suboptimal solutions should be considered for such situations. A possible approach to provide a suboptimal solution is to coordinate both the initiator of a vertical handoff and nodes which are interested in vertical handoffs. Before the initiator conducts a vertical handoff, the initiator can send a preliminary notification of the vertical handoff to another nodes. In such a case, nodes interested in vertical handoffs can take some action to prepare for the actual occurrence of the vertical handoff.

Generalizing Changes in Link Characteristics This dissertation mainly deals with the issues arising from vertical handoffs, especially drastic changes in link characteristics. However, conducting vertical handoffs is just an obvious example of occurrences of changes in link characteristics on heterogeneous wireless networks. Dynamic rate adaptation mechanisms on IEEE802.11-based wireless networks will also change their link characteristics. Since the proposed methods aim to handle changes in link characteristics appropriately, they should also handle changes in link characteristics involved by events other than vertical handoffs, for example rate adaptation mechanisms. For this purpose, additional features such as detecting events which cause changes in link characteristics and notifying these events must be developed and standardized.

Bandwidth Aggregation Luct usually transmits packets through only the fastest data-link. Therefore, the multi-homing mechanism does not provide aggregated bandwidth of all data-links that a mobile node can use simultaneously. The ideal aggregated bandwidth is the sum of bandwidths of all data-links that a mobile node can use simultaneously. It is almost impossible to achieve the ideal aggregated bandwidth on heterogeneous wireless networks, as shown in section 5.2.2. However, aggregating bandwidths contributes to improvement of the communication quality of mobile nodes and consequently it is worth considering.

The essence of the performance of bandwidth aggregation for TCP/IP protocols is how packets are striped over each data-link. A simple rate-based packet striping strategy may not achieve even the fastest data-link bandwidth. If a data striping strategy lacks the consideration of easy fluctuation of data-link characteristics on heterogeneous wireless networks, a head-of-line blocking problem is likely to occur. The head-of-line problem will limit the aggregated bandwidth to the slowest data-link bandwidth. Therefore, fragile characteristics of wireless data-links should be considered in developing a packet striping strategy for bandwidth aggregation.

Handoff Strategy This dissertation has not probed deeply into the handoff strategy, namely, when a mobile node should switch over the upstream data-link. Teppi assumes that the timing of vertical handoffs could be determined by obvious events such as network interface down, and disruption of radio signals. However, this assumption seems to be simplistic. In general, the communication quality of the wireless data-link gradually declines as a mobile node moves away from the source of radio signals. Luct takes the strategy of using the fastest data-link as the primary data-link to transmit packets as long as the data-link acknowledges. Although this strategy is simple and effective, there is room for improvement. One possible approach to an effective handoff strategy is to observe the radio signal condition and to switch over the upstream data-link depending on the observed information. Integrating effective handoff strategies with the contributions of this dissertation will improve the quality of communication for mobile nodes on heterogeneous wireless networks.

Overhead of Mobility Support Protocols Most mobility support protocols have some kind of communication overhead. For example, MIPv6 and NEMO basic support forms a triangular communication path, in which packets are encapsulated and go through a home agent, between a mobile node and a correspondent node without route optimization. This overhead somewhat reduces the communication quality of mobile nodes. Since Teppi depends on such mobility support protocols, reducing overhead of mobility support protocols is one of the open issues to resolve to achieve the ultimate goal of a comprehensive study.

Chapter 7

Conclusion

The Internet is still on the way to becoming a universal communication platform, by which any information humans want to acquire can be provided at anytime anywhere in an appropriate manner. Providing fast, widely available, and sustainable connectivity to the Internet in mobile environments is what it takes to make the Internet such a universal communication platform. As a promising approach to providing such features, this dissertation has focused on the effective utilization of heterogeneous wireless networks. In particular, adapting the transport layer to different link characteristics is the primary issue that was studied in this dissertation. To conclude the dissertation, this Chapter reviews the background, goals, and contributions of this study.

Chapter 2 provided the concept, benefits and issues of heterogeneous wireless networks. There are diverse kinds of wireless data-links to connect to the Internet. Their complementary link characteristics motivate the construction of heterogeneous wireless networks, in which mobile nodes can select the best data-link and use it as the upstream data-link depending on their situation. Several issues arise when mobile nodes want to gain the full benefit of heterogeneous wireless networks, irrespective of mobility support, fragile link characteristics, and drastic changes in link characteristics caused by vertical handoffs. This dissertation has mainly addressed the issue of adapting the transport layer to deal with drastic changes in link characteristics. An empirical application of heterogeneous wireless networks, namely, a communication platform for emergency medical care, was also presented in Chapter 2 to demonstrate the benefits of heterogeneous wireless networks.

Chapter 3 investigated the problems of TCP on heterogeneous wireless networks. Since the traditional TCP/IP protocols have evolved for stationary networks, some characteristics of heterogeneous wireless networks, such as fragile link characteristics and drastic changes in link characteristics caused by vertical handoffs, could inhibit the function of the protocols, especially the TCP congestion control mechanism. Exploratory experiments were conducted to investigate what happens to the TCP congestion control mechanism when a vertical handoff occurs. The results of exploratory experiments demonstrated the need for transport layer adaptation on heterogeneous wireless networks.

Two different approaches to transport layer adaptation on heterogeneous wireless networks, were presented in this dissertation. Chapter 4 described a parameter resetting method, called Teppi, to adapt the TCP congestion control mechanism to drastic changes in link characteristics caused by vertical handoffs. The primary reasons that the TCP congestion control mechanism did not work as expected were the inconsistent values of the congestion control parameters after a vertical handoff. Although this problem can be solved by resetting parameters after the occurrence of vertical handoffs, it is difficult to determine when the parameters should be reset. Teppi resets the parameters at the point when harmful effects of vertical handoff no longer occurred. Teppi was implemented into the NetBSD operating system and evaluated in the emulation environment. Experimental results showed that Teppi was able to adapt the TCP congestion control mechanism to the after-handoff data-link appropriately.

Another approach described in Chapter 5 introduced a TCP based multi-homing mechanism, called Luct, to mobile nodes to allow mobile nodes to manage TCP congestion control context separately for each data-link. The idea of this approach to avoiding the harmful effects of vertical handoffs is to use all available wireless data-links simultaneously with the multi-homing mechanism and eliminate the need to conduct vertical handoffs entirely. Since a TCP based multi-homing mechanism is an effective approach from the perspective of the need of transport layer adaptation, Luct adopts a TCP based multi-homing mechanism. Luct was implemented into the Linux operating system. Evaluations of Luct were conducted on an actual heterogeneous wireless network and the results showed that Luct was able to select a suitable data-link automatically and then fall back to the second-best data-link quickly when the best data-link was no longer available.

Finally, Chapter 6 discussed the open issues. The author believes that the contributions of this dissertation are essential elements for improving the communication quality of mobile nodes. However, at the moment, the proposed methods and contributions of this dissertation are only a part of a comprehensive study for improving the communication quality of mobile nodes, which is the ultimate goal of this dissertation. Towards the ultimate goal of the study, it is crucial that not only individual technologies are developed and enhanced but also that these technologies are able to be integrated with strictly limited mutual interference.

References

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

[2] Wolfgang John and Sven Tafvelin. Analysis of internet backbone traffic and header anomalies observed. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 111–116, New York, NY, USA, 2007. ACM.

[3] D. Johnson, C. Perkins, and J. Arkko. Mobility Support in IPv6. RFC 3775 (Proposed Standard), June 2004.

[4] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert. Network Mobility (NEMO) Basic Support Protocol. RFC 3963 (Proposed Standard), January 2005.

[5] Thierry Ernst Keisuke, Thierry Ernst, Keisuke Uehara, and Koshiro Mitsuya. Network mobility from the InternetCAR perspective. *Advanced Information Networking and Applications, 2003. AINA 2003.*, pages 19–25, March 2003.

[6] IKOMA119 Project. `http://www.ikoma119.jp/`.

[7] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (Standard), July 2003. Updated by RFC 5506.

[8] S. Kent and K. Seo. Security Architecture for the Internet Protocol. RFC 4301 (Proposed Standard), December 2005.

[9] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), August 2008.

[10] SHISA. `http://www.mobileip.jp/`.

[11] WIDE Project. `http://www.wide.ad.jp`.

[12] b-mobile. `http://www.bmobile.ne.jp/`.

[13] EMOBILE. `http://emobile.jp/`.

[14] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. The Secure Real-time Transport Protocol (SRTP). RFC 3711 (Proposed Standard), March 2004. Updated by RFC 5506.

[15] Doja Developer Network. `http://www.doja-developer.net/`.

[16] J. Postel. Internet Protocol. RFC 791 (Standard), September 1981. Updated by RFC 1349.

[17] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460 (Draft Standard), December 1998. Updated by RFCs 5095, 5722.

[18] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. RFC 2581 (Proposed Standard), April 1999. Obsoleted by RFC 5681, updated by RFC 3390.

[19] S. Floyd, T. Henderson, and A. Gurtov. The NewReno Modification to TCP's Fast Recovery Algorithm. RFC 3782 (Proposed Standard), April 2004.

[20] Van Jacobson. Berkeley TCP Evolution from 4.3-Tahoe to 4.3-Reno. In *In Proceedings of the 18th Internet Engineering Task Force*, page 365, Sep 1990.

[21] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP Selective Acknowledgment Options. RFC 2018 (Proposed Standard), October 1996.

[22] E. Blanton, M. Allman, K. Fall, and L. Wang. A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP. RFC 3517 (Proposed Standard), April 2003.

[23] G. R. Wright and W. R. Stevens. *TCP/IP Illustrated, Volume 2*. Addision Wesley, 1995.

[24] L. Rizzo. dummynet. `http://info.iet.unipi.it/~luigi/ip_dummynet/`.

[25] tcpdump. `http://tcpdump.org/`.

[26] Reiner Ludwig and Randy H. Katz. The eifel algorithm: making tcp robust against spurious retransmissions. *SIGCOMM Comput. Commun. Rev.*, 30(1):30–36, 2000.

[27] Saverio Mascolo, Claudio Casetti, Mario Gerla, M. Y. Sanadidi, and Ren Wang. Tcp westwood: Bandwidth estimation for enhanced transport over wireless links. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 287–297, New York, NY, USA, 2001. ACM.

[28] Prasun Sinha, Thyagarajan Nandagopal, Narayanan Venkitaraman, Raghupathy Sivakumar, and Vaduvur Bharghavan. Wtcp: a reliable transport protocol for wireless wide-area networks. *Wirel. Netw.*, 8(2/3):301–316, 2002.

[29] Shunsuke Saito and Fumio Teraoka. Implementation, analysis, and evaluation of TCP-J: A new version of TCP for wireless networks. *Syst. Comput. Japan*, 36(7):23–30, 2005.

[30] Kevin Brown and Suresh Singh. M-tcp: Tcp for mobile cellular networks. *SIGCOMM Comput. Commun. Rev.*, 27(5):19–43, 1997.

[31] Ajay Bakre and B.R. Badrinath. I-TCP: Indirect TCP for Mobile Hosts. In *International Conference on Distributed Computing Systems*, page 136, Vancouver, Can, 1995. IEEE, Piscataway, NJ, USA.

[32] "Kazuya Tsukamoto, Yutaka Fukuda, Yoshiaki Hori, and Yuji Oie". "new tcp congestion control schemes for multimodal mobile hosts(terrestrial radio communications)". *IEICE transactions on communications*, 89(6):1825–1836, Jun 2006.

[33] Tom Goff, James Moronski, and D. S. Phatak. Freeze-tcp: A true end-to-end tcp enhancement mechanism for mobile environments. In *In Proceedings of IEEE INFOCOM'2000, Tel Aviv*, pages 1537–1545, 2000.

[34] Jukka Ylitalo, Tony Jokikyyny, Tero Kauppinen, Antti J. Tuominen, and Jaakko Laine. Dynamic network interface selection in multihomed mobile hosts. In *HICSS '03: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03)*, page 315, Washington, DC, USA, 2003. IEEE Computer Society.

[35] Takahiro Nozawa, Michio Honda, Hiroshi Sakakibara, Jin Nakazawa, and Hideyuki Tokuda. ARMS: Application-level Concurrent Multipath Utilization on Reliable Communication. *Proc. of Internet Conference2008 (in Japanese)*, 2008.

[36] Hiroshi Sakakibara, Masato Saito, and Hideyuki Tokuda. Design and Implementation of Socket-level Bandwidth Aggregation Mechanism for Mobile Networking Environments. *IPSJ Journal*, 48(2):471–482, 2007.

[37] Shunsuke Saito, Yasuyuki Tanaka, Mitsunobu Kunishi, Yoshifumi Nishida, and Fumio Teraoka. Ams: An adaptive tcp bandwidth aggregation mechanism for multihomed mobile hosts. *IEICE TRANSACTIONS on Information and Systems*, E89-D(12):2838–2847, 2006.

[38] Hung-Yun Hsieh and Raghupathy Sivakumar. A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts. pages 83–94, 2002.

[39] Ming Zhang, Junwen Lai, Arvind Krishnamurthy, Larry Peterson, and Randolph Wang. A transport layer approach for improving end-to-end performance and robustness using redundant paths. In *ATEC '04: Proceedings of the annual conference on USENIX Annual Technical Conference*, pages 8–8, Berkeley, CA, USA, 2004. USENIX Association.

[40] Yu Dong, Niki Pissinou, and Jian Wang. Concurrency handling in tcp. *Communication Networks and Services Research*, 0:255–262, 2007.

[41] Antonios Argyriou and Vijay Madisetti. Bandwidth aggregation with SCTP. *IEEE Global Telecommunications Conference, 2003. GLOBECOM'03*, 7, December 2003.

[42] Ahmed Abd El Al, Tarek Saadawi, and Myung Lee. LS-SCTP: A bandwidth aggregation technique for stream control transmission protocol. *Computer Communications*, 27(10):1012–1024, February 2004.

[43] Janardhan R. Iyengar, Paul D. Amer, and Randall Stewart. Concurrent multipath transfer using sctp multihoming over independent end-to-end paths. *IEEE/ACM Trans. Netw.*, 14(5):951–964, 2006.

[44] Luiz Magalhaes and Robin Kravets. Transport Level Mechanisms for Bandwidth Aggregation on Mobile Hosts. page 165, 2001.

[45] Kyu-Han Kim, Yujie Zhu, Raghupathy Sivakumar, and Hung-Yun Hsieh. A receiver-centric transport protocol for mobile hosts with heterogeneous wireless interfaces. *Wirel. Netw.*, 11(4):363–382, 2005.

[46] R. Wakikawa, V. Devarapalli, G. Tsirtsis, T. Ernst, and K. Nagami. Multiple Care-of Addresses Registration. RFC 5648 (Proposed Standard), October 2009.

[47] S.E. Kim and JA Copeland. TCP for seamless vertical handoff in hybrid mobile data networks. *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*, 2, 2003.

[48] C. Perkins. IP Mobility Support for IPv4. RFC 3344 (Proposed Standard), August 2002. Updated by RFC 4721.

[49] Kenichi Ishibashi, Naoto Morishima, and Hideki Sunahara. A Scheme for Adapting TCP to Drastic Changes in Link Characteristics. In *International Conference on Mobile Ubiquitous Computing Systems Services and Technologies*, pages 183–188, Nov 2007.

[50] ”Kazuya Tsukamoto, Yoshiaki Hori, and Yuji Oie”. Mobility Management of Transport Protocol Supporting multiple Connections. In *Proc. of Second International Workshop on Mobility Management and Wireless Access Protocols (ACM MobiWac 2004)*, pages 83–87, October 2004.

[51] Olaf Titz. Why TCP Over TCP Is A Bad Idea. `http://sites.inka.de/~W1011/devel/tcp-tcp.html`.

[52] Sangtae Ha, Injong Rhee, and Lisong Xu. Cubic: a new tcp-friendly high-speed tcp variant. *ACM SIGOPS Operating Systems Review*, 42(5):64–74, 2008.

[53] J. Salim, H. Khosravi, A. Kleen, and A. Kuznetsov. Linux Netlink as an IP Services Protocol. RFC 3549 (Informational), July 2003.

[54] Iperf. `http://iperf.sourceforge.net/`.

[55] Michio Honda, Yoshifumi Nishida, Lars Eggert, Pasi Sarolahti, and Hideyuki Tokuda. Multipath Congestion Control for Shared Bottleneck. In *Proc. International Workshop on Protocols for Future, Large-Scale and Diverse Network Transports (PFLDNeT)*, May 2009.

[56] Ryuji Wakikawa, Keisuke Uehara, Fumio Teraoka, and Jun Murai. MIBsocket: An Integrated Mechanism to Manipulate General Network Information in Mobile Communications. *IEICE transactions on communications*, 84(8):2001–2010, 20010801.

[57] Shibui Rie, Mitani Hiroki, and Teraoka Fumio. LIES: A Inter Layer Information Exchanges System for Mobile Communication. In *The Sixth Workshop on Internet Technology (WIT2004, In Japanese)*, 2004.

[58] F. Teraoka, K. Gogo, K. Mitsuya, R. Shibui, and K. Mitani. Unified Layer 2 (L2) Abstractions for Layer 3 (L3)-Driven Fast Handover. RFC 5184 (Experimental), May 2008.

Achievements

Papers marked with asterisk (*) are closely related papers of this doctoral dissertation.

Papers in Refereed Journals

- * 石橋 賢一, 森島 直人, 砂原 秀樹. リンク特性変化追従のための TCP 輻輳制御パラメータ設定タイミングの提案. 情報処理学会論文誌, Vol.49, No.10, pp.3631–3644, (related with Chapter 4), Oct 2008.

- * Kenichi Ishibashi, Naoto Morishima, Masayuki Kanbara, Hideki Sunahara, Masami Imanishi, "Toward Ubiquitous Communication Platform for Emergency Medical Care," IEICE TRANSACTIONS on Communications, Vol.E92-B, No.04, pp.1077–1085 (*Invited Paper*, related with Chapter 2), Apr 2009.

Paper in International Conference

- * Kenichi Ishibashi, Naoto Morishima, Hideki Sunahara. "A Scheme for Adapting TCP to Drastic Changes in Link Characteristics," International Conference on Mobile Ubiquitous Computing Systems Services and Technologies, pp.183–188, (related with Chapter 4), Nov 2007.

Papers in Domestic Conference and Workshops

- * 石橋 賢一, 砂原 秀樹. トランスポート層を利用した移動体マルチホーム向けのスケジューリングアルゴリズム評価ツール. マルチメディア，分散，協調とモバイル (DICOMO2009) シンポジウム, (related with Chapter 5), Jul 2009.

- 石橋 賢一, 砂原 秀樹. トランスポートプロトコル解析のための多様なビューを持つ可視化ツールの開発. マルチメディア，分散，協調とモバイル (DICOMO2008) シンポジ

ウム, Jul 2008.

- * 石橋 賢一, 森島 直人, 砂原 秀樹. リンク特性変化追従のための TCP 輻輳制御パラメータ設定タイミングの提案. マルチメディア，分散，協調とモバイル (DICOMO2007) シンポジウム, pp.888–897, (related with Chapter 4), Jul 2007.

- * 石橋 賢一, 森島 直人, 砂原 秀樹. TCP の輻輳制御パラメータの設定によるリンク特性の変化に対する通信性能改善の可能性. 第 14 回マルチメディア通信と分散処理ワークショップ, pp.61–66, (related with Chapter 3), Nov 2006.

- 石橋 賢一, 米田 孝弘, 松浦 知史, 宇夫 陽次朗, 砂原 秀樹. イベント参加者の交流を促進するシステム「タグ・マッチ」の開発と評価. マルチメディア，分散，協調とモバイル (DICOMO2006) シンポジウム, pp.581–584, Jul 2006.

Awards

- マルチメディア，分散，協調とモバイル (DICOMO2009) シンポジウム ヤングリサーチャ賞, 情報処理学会, Jul 2009.

- マルチメディア，分散，協調とモバイル (DICOMO2008) シンポジウム 優秀プレゼンテーション賞, 情報処理学会, Jul 2008.

- マルチメディア，分散，協調とモバイル (DICOMO2007) シンポジウム 優秀論文賞, 情報処理学会, Sep 2007.