On the Transition to the Mobile Internet

Albert Cabellos-Aparicio

Advisor: Prof. Jordi Domingo-Pascual
Co-Advisor: Dr. Francisco J. Garcia

Department of Computer Architecture

Technical University of Catalonia

A Thesis presented to the Technical University of Catalonia in fulfillment of the requirements for the degree of

Doctor in Computer Science

December 2008

a Sonia y a mi Familia.

A cluster of stars, a rising nebula. Unvisible slopes of running matter and a glowing ring of ancient debris. Counter-clock-wise orbits, abnormal conduits linking remote parts of space, thats a laberyinth beyond view. Walls of cyphers stand. Unlatched. Those freshly emerged wrinkles in the surface of time have discredited the whole theoretical working basis we were used to. The cradle is suddenly on fire. No more spinning coils of substance. A black unexplained hole in the pattern. Shall I fill it with numbers? Yeah, lets deploy. He opened up the window and scanned.

Space-Internet..., the final frontier. These are the. Inner off-world terrains, in-world structures, so untouchable (you cant spill a jar of Internet, you cant store it on the shelf), widely sustained on math laws. He built a spaceship out from words and ideas. And departed right away.

Dove in it. We stayed at home and watched his rear light as he flew on and away. Received regular reports of worlds and regions never stomped, his face depicted on the long distance communication device screen seemed tired but his eyes were shining strong. Schemes and graphics stuffed the walls of his ship at the back. There was a little column of smoke coming out his laptop. Hey are you coming to the match tomorrow? Wow boys I dunno. Right. OK. Work on it bro, well be right here. Yeah. And he went on climbing dunes of equations and sliding down bi-polar axis of white death neutrons. Ahead through the large sinoidal derivative valley and up the hills of test proofs. He got stucked in some brownish desert coordinate-sands by April but made it through by considering all terms. The engine was working properly. Jettisoned some ballast pieces. Try. Re-think. Reflect. Go on. Expanding knowledge is a titanic task, every single milimiter the borders increase, a human pushing effort has been applied from behind. His hair got disheveled like a nest of parrots.

By the end of summer he landed back home. The ship had stains of mud and stellar dust all over. He was radiant happy and smiling as he stepped off. I have finally charted that portion of space! He was waving, a book clutched in his hand. We saw particles sparkling off the pages. The whole adventure is in here... Now. If you want to know. Read on.

by *Javier Terrisse*

Acknowledgements

I have lived the elaboration of this thesis as the work it contains along with a group of great people. I would like to express here my sincere gratitude to all of them.

First I would like to show my gratitude to my advisor, Prof. Jordi Domingo-Pascual. His time and patience and the innumerable amount of meetings that we have shared. His always-calm advices have helped to a great extent on the development of this thesis. *Gràcies Jordi!*

I would also like to acknowledge my co-advisor's help, Dr. Francisco J. García. He offered a unique opportunity by inviting me to Agilent Technologies. The final part of this document must be assumed as a result of the work and inspiration collected there during my research stay.

During the development of this work I have met some excellent researchers. I am proud that some parts of this thesis are the result of a joint collaboration with them. In particular I would like to express my gratitude to Marc Portolés (Centre Tecnològic de Telecomunicacions de Catalunya), Rubén Cuevas (Universidad Carlos III de Madrid) and Dr. John Thompson (University of Edinburgh). It has been a pleasure working with them. The joint research carried out, side by side, has been inspiring, moving and revealing. I hope to keep working with them in new exciting projects.

Parts of the experimentation carried out in the first stages of this thesis are the result of the effort and the capacity of some BSc students. Also to them I want to express my deep gratitude: José Núñez, Héctor Julian and Loránd Jakab. They have taught me what are tenacity, determination and taste for detail. It must not be forgotten my BSc co-advisor, Carlos Veciana and Dr. Josep Mangues. They accompanied me at my very first steps as a researcher. Thank you for your time Carlos and Josep.

I want also to acknowledge the great help provided by the entire research group and my colleagues at the Department of Computer Architecture. In particular I would like to show my gratitude to Pere Barlet and René Serral. The informal discussions that we shared, and the technical advices that were provided helped very much indeed to the elaboration of this thesis. Also to Marc Solé, it has been really enjoyable working with him.

I would like to express my gratitude to the internal and external reviewers and to my PhD committee (Prof. Josep Solé, Prof. Fernando Boavida, Prof. Giorgio

Ventre, Dr. Fabio Ricciato and Dr. Carmen Guerrero). Thank you for reading this manuscript and helping improving it through your reviews.

Lastly I would like to thank to the UPC (UPC per la recerca) for awarding me with a research grant, and to the European Doctoral School of Advanced Topics In Networking (SATIN) of the Network of Excellence CONTENT (IST-2006-NoE-0384239) for their support. I have also participated in several research projects that have played an important role in the development of this thesis: E-NEXT (FP6-2002-IST/1 506869), Laboratories over Next Generation Networks (UE IST-1999-20393) and the Spanish research projects contracts TSI 2005-07520-C03-02 and TIC2002-04531-C04-02.

A nivel personal quiero agradecer a mi padre y a mi madre por estar allí, siempre y apoyándome de manera incondicional. Así como a Jose y Javi (escritor y prologuista de este documento) por acompañarme durante este período, de hecho, casi desde que nací.

Y a Sonia, éste ha sido un viaje que hemos realizado de la mano. Sus ánimos, su apoyo, su infatigable serenidad y buen humor, su capacidad para escucharme y para entenderme están presentes en esta tesis. Sólo escribiendo durante una eternidad podría expresar la gratitud que siento hacia ella. Gracias nena!

Albert Cabellos-Aparicio

Abstract

The Internet is an evolving system. Recently wireless technologies have opened up the possibility of deploying mobility at the Internet. With mobility end hosts can change its point of attachment while maintaining its network connections. Deploying such functionality at the current Internet architecture is a complex task. Basically this is because with mobility end hosts require two identifiers, one related with its identity and another one related with its location. However the current Internet's end host identifiers (IP addresses) represent both the identity and the routing locator of the node. This issue has raised a debate among the research community: "Which is the optimal layer to deploy mobility?". This thesis analyzes such issue using a technical and a cost-effective point of view. Our analysis suggests that although technically the optimal solution would be a cross-layer deployment, network-layer mobility provides the most cost-effective solution. The proposed network-layer mobility protocol, defined by the IETF, is the Mobile IP technology. Taking this into consideration this thesis analyzes the transition to the Mobile Internet considering the deployment of this family of protocols. Specifically the main objectives of this thesis are to analyze the transition to the Mobile Internet, identify its potential issues and propose solutions.

The analysis is carried out at three different stages of the transition. First at present, by analyzing the Mobile IP technology. Our study shows that one of the key issues of Mobile IP is the performance of the handover. The thesis presents an analytical model and experimentation to study several metrics related with the performance of the handover of the main protocols of Mobile IP. Summarizing, the obtained results show that the Mobile IPv4's handover can support real time applications, but Mobile IPv6's not. A reason for that is the large amount of time that takes to reconfigure IPv6 (in the order of seconds). Furthermore the thesis shows that Fast Handovers for Mobile IPv6, an extension of Mobile IPv6 that improves the handover, effectively enhances the protocol to support delay-sensitive applications.

The second stage of the analysis is in the near-future, during the deployment phase of Mobile IP. As shown in this part of the thesis, the main issues during this phase of the transition are the lack of route optimization and the low reliability of Mobile IP-based networks. The first issue prevents mobile clients from communicating directly with its peers, and this has a significant impact on the performance of such networks. In order to solve this we propose the fP2P-HN architecture. A P2P-based network of distributed Home Agents that reduces significantly the delays of the paths of mobile clients. Our evaluation shows that our proposal is scalable ($O(1)$) and can improve significantly such networks. In order to solve the second issue we propose a novel Home Agent architecture that distributes its operations throughout the network. Our evaluation shows that this increases reliability and

reduces the load at the Home Agents.

Finally the third stage of our analysis is in the future, were the Internet is Mobile IP-enabled and new architectures can improve its functionalities. Particularly this thesis analyzes the advantages and the complexity of terminals equipped with multiple interfaces. These terminals can provide more aggregate bandwidth, increase the reliability and the area of coverage. However supporting multiple interfaces can be a complex task. Our analysis reveals that a generic architecture able to deal with these issues can be greatly enhanced if the available bandwidth of the different paths provided by the multiple interfaces can be estimated. Research on bandwidth estimation has focused mainly on periodic probing processes and wired networks, however very little research has been conducted considering wireless links (a typical scenario of mobility). Therefore this thesis focuses on analyzing the existing methodologies and tools in the presence of wireless links, taking the IEEE 802.11 standard as a reference. Our study shows that periodic probing processes target the achievable throughput instead of the available bandwidth. This metric is related with the fair share of the network. Additionally, measurements using periodic probing processes present a bias. This affects the first packets of the process that are served faster than the remaining ones. This bias appears due to the random nature of wireless networks and impacts the measurement processes. Taking this into consideration this thesis explores poisson-probing process to estimate the available bandwidth in wireless and wired scenarios. In particular our research has lead us to design three different tools that can operate in a wide range of scenarios. Furthermore, our study in poisson probing processes reveals that they are useful to estimate the available bandwidth in wireless links and can produce lighter (less intrusive) and faster tools.

Resumen

Internet es un sistema en plena evolución donde recientemente las nuevas tecnologías inalámbricas han abierto la posibilidad al despliegue de la movilidad. La movilidad se define como la habilidad de que un nodo pueda cambiar su punto de acceso a Internet sin interrumpir sus conexiones de red. Desplegar dicha funcionalidad en la arquitectura actual de Internet es una tarea compleja, básicamente ésto se debe a que la movilidad requiere de dos identificadores, uno relacionado con la posición del nodo y otro con su identidad. Sin embargo en la arquitectura actual los identificadores (direcciones IP) representan tanto la identidad del nodo así como su posición. Este problema ha generado un amplio debate en la comunidad científica: Cuál es la manera óptima de desplegar la movilidad en Internet?. En esta tesis analizamos dicha cuestión desde un punto de vista técnico y de costes. Nuestro análisis sugiere que, a pesar de que probablemente la mejor solución a nivel técnico sea un protocolo que abarque varios niveles de la arquitectura, desde un punto de vista de costes la solución óptima se encuentra a nivel de red. El protocolo de movilidad a nivel de red propuesto por la IETF es Mobile IP. Teniendo en cuenta las conclusiones de este análisis dicha tesis analiza la transición de Internet hacia la movilidad con Mobile IP. En particular los objetivos principales de esta tesis son analizar la transición a la Internet Móvil, identificar posibles problemas y plantear soluciones.

Desarrollamos el análisis en tres etapas de la transición. Primero en el presente donde la tesis analiza la familia de tecnologías de Mobile IP. Nuestro estudio señala que uno de los aspectos más importantes de dicha tecnología es el rendimiento del handover. Por lo tanto esta tesis presenta un modelo analítico y resultados empíricos con el objetivo de estudiar su rendimiento. Nuestros resultados muestran que mientras que el handover de Mobile IPv4 está preparado para soportar aplicaciones con requisitos de tiempo real, el de Mobile IPv6 no lo está. Esto se debe principalmente a la gran cantidad de tiempo que requiere re-configurar IPv6 (en el orden de segundos). Siguiendo con este análisis la tesis tambin muestra que, Fast Handovers for Mobile IPv6, un protocolo que tiene como objetivo mejorar el rendimiento del handover de IPv6, consigue sus objetivos y permite que Mobile IPv6 soporte aplicaciones con estrictos requisitos de latencia.

La segunda etapa del análisis es en el futuro cercano, particularmente en la fase de despliegue de Mobile IP. Tal y como se muestra en esta parte de la tesis los principales problemas que potencialmente pueden surgir son la falta de optimización de rutas y la baja fiabilidad de las redes basadas en Mobile IP. El primer problema fuerza a los clientes móviles a usar rutas no óptimas para comunicarse y esto tiene un impacto significativo en el rendimiento general de Mobile IP. La solución propuesta es una nueva arquitectura de movilidad denominada fP2P-HN. Dicha arquitectura consiste en un conjunto de Home Agents distribuidos por Internet que, a

su vez, forman una red P2P. La evaluación de nuestra arquitectura demuestra que la solución propuesta es escalable (O(1)) y que reduce efectivamente la latencia de las comunicaciones de los nodos. Respecto al segundo problema, la baja fiabilidad de redes basadas en Mobile IP, en esta tesis se propone una nueva arquitectura para Home Agents, denominada flexible Home Agents. Dicha arquitectura distribuye las operaciones de los Home Agents a través de la red, incrementando la fiabilidad y reduciendo la carga en dichas entidades.

Finalmente, la tercera etapa del análisis se centra en el futuro, donde Internet ya es móvil gracias a Mobile IP. En esta etapa nuevas arquitecturas pueden surgir para mejorar las prestaciones o funcionalidades de los nodos móviles. En particular se analizan las ventajas y la complejidad asociada a terminales equipados con múltiples interfaces de red. Dichos terminales pueden beneficiarse de un mayor ancho de banda agregado y una zona de cobertura más amplia. Sin embargo gestionar múltiples interfaces es una tarea compleja. Nuestro análisis revela que una arquitectura genérica para gestionar dichos terminales puede beneficiarse enormemente si se dispone de información acera del ancho de banda disponible en las conexiones ofrecidas por las múltiples interfaces. Las metodologías existentes para estimar el ancho de banda se han centrado básicamente en procesos de prueba periódicos y redes fijas. Existe muy poca investigación en el ámbito de redes inalámbricas (un escenario típico de movilidad). Por lo tanto esta tesis se centra en analizar dichas metodologías pero en redes inalámbricas, concretamente en redes IEEE 802.11. Nuestro estudio muestra que los procesos periódicos de prueba estiman una métrica diferente a la esperada: el rendimiento disponible. Esta métrica está relacionada con el reparto equitativo del ancho de banda. Así mismo nuestro estudio señala que las estimaciones tomadas mediante procesos periódicos de prueba tienen un error intrínseco, es decir, no son estimadores ergódicos. Dicho error se produce debido a la naturaleza aleatoria de las redes inalámbricas. Considerando estas conclusiones la tesis explora procesos de estimación mediante poisson. Particularmente nuestra exploración nos lleva a diseñar tres herramientas diferentes para estimar el ancho de banda disponible. Además nuestro estudio muestra poisson que puede ser útil para estimar el ancho de banda disponible en redes inalámbricas y que permite diseñar herramientas menos intrusivas y más rápidas.

Contents

List of Figures

List of Tables

Glossary

AB	Available Bandwidth
ABEST	Available Bandwidth Estimation Tool
AKBEST	Active Kalman-based ESTimation
AP	Access Point
AR	Access Router
ARP	Address Resolution Protocol
AS	Autonomous System
BA	Binding Acknowledgement
BGP	Border Gateway Protocol
BR	Border Router
BS	Base Station
BU	Binding Update
c.i	Confidence Interval
CCDF	Complementary Cumulative Distribution Function
cCoA	collocated Care-of Address
CDF	Cumulative Distribution Function
CN	Correspondent Node
CoA	Care-of Address
CR	Correspondent Router
CT	Cross Traffic
DAD	Duplicate Address Detection
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Service
eBGP	exterior Border Gateway Protocol
EDF	Empirical Distribution Function
ER	Exit Router
FA	Foreign Agent
faCoA	foreign agent Care-of Address
FHA	flexible Home Agent
FIFO	First in First Out
FMIPv6	Fast Handovers for Mobile IPv6
fP2P-HN	flexible Peer-to-Peer Home Network Architecture
FQ	Fair Queing
HA	Home Agent
HIP	Host Identity Protocol
HL	Home Link
HMIPv6	Hierarchical Mobile IPv6
HoA	Home Address
HTTP	Hyper-Text Transport Protocol
i.i.d	independent and identically distributed
iBGP	interior Border Gateway Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPDV	Inter-Packet-Delay-Variation
IPSec	Internet Protocol Secure
ISP	Internet Service Provider
KF	Kalman Filters
KS	Kolmogorov-Smirnov Test
LB	Load Balancer
MA	Mobility Agent
MAP	Mobility Anchor Point
MIPv4	Mobile IPv4
MIPv6	Mobile IPv6
MN	Mobile Node

MR	Mobile Router
NAR	Next Access Router
NAT	Network Address Translation
NEMO	Network Mobility
NTP	Network Time Protocol
NUD	Neighbor Unreachability Detection
oCoA	on-link Care-of Address
OWD	One-Way-Delay
PAR	Previous Access Router
PASTA	Poisson Arrivals See Time Averages
PDF	Probability Density Function
PGM	Probe Gap Model
PHM	Passive Handover Monitoring
PKBEST	Passive Kalman-based ESTimation
PL	Packet Losses
PLR	Packet Loss Ratio
PRM	Probe Rate Model
PT	Packet Train
PT	Probe Traffic
QoS	Quality of Service
rCoA	regional Care-of Address
RFC	Request For Comments
RR	Return Routability
RTT	Round Trip Time
SCTP	Stream Control Transmission Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
W-Path	Wireless end-to-end PATH estimation
WG	Working Group
WLAN	Wireless LAN (IEEE 802.11)

Part I

Introduction

1

Summary and Road Map

This initial chapter discusses the motivations and the objectives of this thesis. Additionally the chapter points out the main contributions and concludes with an overview of the structure of this manuscript.

1.1 Motivation

The Internet is an evolving system and since its creation, four decades ago, it has grown exponentially. This growth has been fuelled by governments, companies and, in general, common people that have connected new devices and deployed new services in the Internet. As a system, the Internet has supported an exponentially growing load showing scalability and flexibility. Mainly this is due to the Internet's design fundamentals: the end-to-end principle [1; 2].

The end-to-end principle states that, whenever possible, complexity must be put at the end points of a system. This implies that the required state of the communication between two applications must be maintained by the end points. Keeping this state into the network might not be scalable. This principle ensures openness, increases reliability and robustness preserving the properties of user choice and ease of new service deployment [3].

The research community designed the OSI reference model [4], which is also based on this principle. This model depicts a layered-architecture that it is widely accepted as being a good abstraction for network device functionality. The model assigns the different functionalities of a network into layers. Thus each layer provides functionalities to its upper layer by using the functionalities offered by its lower layer. The Internet is based on the TCP/IP model that, in turn, is *roughly* based on the OSI reference model. Specifically it includes the first four layers of the OSI stack, namely: Physical layer, Link Layer, Network Layer and Transport Layer.

The Internet, based on the TCP/IP model and the end-to-end principle, has grown since its creation without a centralized control. However during the last decade new requirements and technologies have appeared. One of the new technologies that has significantly changed the initial requirements of the Internet are the wireless access networks. This technology has evolved and become cheaper on the last years and currently it is widely available. As a consequence a new requirement has risen: a host can be wirelessly attached to the Internet. This opens up the possibility of adding a new feature to the Internet architecture: mobility.

The ITU (International Telecommunication Union) [224] defines mobility as:

"The ability for a user to access services while in motion, and the capability of the network to identify and locate the user's terminal."

Mobility has two desirable properties [5; 6], first changing the point of attachment should be done seamlessly. This means that the transition should not result in unacceptable loss of application data and should minimize the disconnection time, hence transitions should be transparent to applications. Second the node should remain reachable regardless of its point of attachment.

Therefore in order to implement mobility three basic requirements arise [7; 8]. First, each node needs two identifiers, one constant and related with to identity of the node (WHO) and another one dynamic and related with its topological position (WHERE). Second mobility requires an authentication mechanism for both identifiers otherwise impersonation security risks arise. This is a complex task since authentication must be done quickly to ensure seamless transitions. Finally, in order to ensure reachability, a mobility implementation requires a location management system that maintains bindings between the WHO and the WHERE identifiers.

Then, a question arises, which is the best way to implement mobility at the Internet? According to the design principles of the Internet mobility should be managed by end hosts rather than by the network. In fact reviewing the OSI reference model, the fifth layer of the stack, the session layer [4], provides mobility-related functionalities to the end hosts. This layer is responsible of establishing and maintaining end-to-end connections. This means that this layer could establish a new connection after each transition. Since this layer manages sessions, this can be done transparently. In this case the session layer should provide a constant static identifier regardless of the topological position while the network layer provides a topologically related identifier. The OSI reference model does not define a location management system, however this could be easily implemented by using an external infrastructure. Unfortunately the TCP/IP model lacks of a session layer. In fact the Internet architecture uses just one identifier: the IP address. This address identifies both the node and its topological position. That is why deploying mobility in the current Internet architecture is a challenging task and requires a different approach.

The most obvious option is to include mobility at the Internet-suite transport layer. This basically means adding OSI-like session layer functionalities to the existing transport layer: including a new constant identifier and using IP addresses just as topological identifiers. Then the transport connections can be resumed after the node changes its point of attachment. In fact SCTP (Stream Control Transmission Protocol) [9] already defines these functionalities. Since TCP [226] only supports bindings to single IP addresses it needs and extension for chaining the bound address of a connection (see [10; 11]). Regarding the location management it cannot be deployed at this layer because it is a global functionality, not and end-to-end one. Thus it requires an external infrastructure such as dynamic DNS. The main drawback is that it depends on other layers and that the convergence time may be slow. For instance dynamic DNS updates take tens of seconds to be propagated over the Internet [12]. In addition Internet applications use directly the transport layer and updating it may require also updating the applications. Finally it has been shown as very difficult to provide efficient authentication mechanisms at this layer [34; 35; 36; 37; 38; 39] and this issue remains unsolved. In any case it is technically and philosophically mandatory that the transport layer is aware of mobility because this layer is responsible of congestion control [5], and good congestion control requires differentiating between packets lost due to congestion from those lost due to mobility disruptions.

Another option, which is taken by the Mobile IP [13; 14] standard, is to deploy mobility at the IP layer. In this case each node has two IP addresses, the first one

(WHO) related to its home network and the second one (WHERE) related to its visiting network. Location management can be implemented also in this layer by deploying a special entity located at the node's home network, the Home Agent. Finally authentication can be implemented relaying in the Internet routing system [15]. This approach is transparent to the applications and has location management built-in, however the interface between IP and transport protocols is not rich enough for the upper layers to be notified when mobility is taking place, or when the transition has finished.

Finally a completely different approach is to replace the current Internet architecture by a new one based, possibly, on different principles [19; 20]. In fact some researchers argue that the end-to-end principle should be revisited [16; 17; 18]. Mainly because the Internet was originally developed among a community of like-minded technical professionals who trusted each other, and was administered by academic and government institutions who enforced a policy of no commercial use. However the major stakeholders in the today's Internet are quite different. Revisiting this principle, and rethinking the Internet architecture is currently an active research topic lead by the New Arch project [21]. Several proposals have been made (see [22; 23] and the references therein) and these new architectures incorporates mobility as a central functionality.

There has been some discussion among the research community regarding at which layer does mobility belong. Besides from the technical discussion, there is a practical one. Which is the deployment cost in terms of time and resources of the different approaches? At the transport and at the network layer the solutions require updating the kernel of all the hosts and the applications, in addition they require deploying a location management infrastructure. Regarding new architectures they are still in its infancy and it will take years, or even decades, to roughly agree on a new architecture, create a prototype and deploy it. That is why we disregard new architectures as an option for mobility deployment for the current Internet. It is worth to note that we believe that rethinking the Internet principles is a necessary task that potentially can lead to much more advanced networks.

The deployment cost of transport and network layer mobility is roughly similar, however the IP layer has other urgent issues. The exponential growth of the Internet is depleting the IPv4 address space. In fact, a recent study predicts an exhaustion of the unallocated IANA pool in January 2011 [24]. This has forced the users to deploy

NAT devices [25] that, in turn, break the end-to-end principle. That is why the IETF (Internet Engineering Task Force) has designed IPv6 [26], a replacement for IPv4. IPv6 has a huge address space and has built-it network mobility through the Mobile IPv6 [14] protocol. At the current status IPv6 is a standard protocol with mature implementations tested both by the research community and by the industry [27; 28]. However IPv6 has not taken off as hoped (see [32] for further details), since no single party has applied enough pressure to the market or taken the role of a leader in using the new technology.

Therefore the most cost-effective solution to deploy mobility in the Internet is at the network layer. This is mainly due because IPv4 has to be replaced and IPv6 already incorporates mobility. The deployment of Mobile IP leads to a challenging process: the transition from the current Internet to a mobility-enabled Internet. This process involves many actors such as ISPs, vendors and end-users. At present, Mobile IP is a standardized technology ready for its deployment. Some implementations and experimental networks have started to appear [29; 30]. It is expected that in the near future this technology will be widely deployed. In the future new architectures may appear that improve and extend the functionalities of Mobile IP. And this is what this thesis is about: the Transition to the Mobile Internet.

1.2 Objectives

The subject of this thesis is to explore the transition process from the current Internet to the Mobile Internet. This process has already started and finishes with an Internet able to provide global mobility, within the geographical areas of coverage, to any mobility-aware node. The process is expected to last years, but not decades, otherwise the transition could lead to a new architecture. In more detail, the objectives in this thesis are:

1. Analyze the transition from the current Internet to the Mobile Internet and foresee and identify potential issues. The analysis is done at three different stages of deployment. First at present, analyzing the current status of the Mobile IP technology. Second in the near-future, during the deployment phase of the Mobile IP family of protocols. Finally in the future, where new architectures and

new wireless technologies can improve the performance and functionalities of the Mobile Internet.

2. Propose solutions to the identified potential issues.

1.3 Contributions

1.3.1 Present: Analysis of the handover

The IETF is leading the standardization of network layer mobility protocols. The solution proposed by the IETF is Mobile IP, which comes into two flavours, Mobile IPv4 and Mobile IPv6. Mobile IP uses two identifiers, the Home Address (HoA), which identifies the mobile node and the Care-of Address (CoA), which is related to the mobile node's topological position. The Home Agent provides location management services and it is placed at the node's home network. It intercepts packets addressed to the HoA and forwards them to the mobile node, ensuring reachability. In the basic protocol operation the mobile node sends its data-packets through the Home Agent. However Mobile IP also includes authentication methods that allow end-to-end communications between the mobile node and its peers. Additionally the IETF is standardizing several extensions to Mobile IP that include new functionalities and that improve its performance. As we have mentioned previously, Mobile IP is a standardized technology with mature implementations and some experimental deployments [29; 30].

The main concern at the pre-deployment phase of any technology is regarding its performance. In Mobile IP the performance depends basically on the handover process. This is the sequence of actions that a mobile node and/or the network have to perform in order to regain connectivity after a change in the attachment point. This occurs either at the link layer, when the mobile node changes its base station or at the network layer, when it changes its default router. The first case involves just the physical and the link layers while the second one, the worst case, involves also the network layer.

In a full handover, first the wireless interface must regain connectivity, then the IP layer must obtain a new IP address (CoA) and finally Mobile IP must inform the Home Agent, and possibly its peers, about its new location. During the time that all these actions are executed, the handover latency, the mobile node is unable to send or receive data packets. On the one hand incoming data packets are addressed to the old location and thus, lost. On the other hand outgoing data packets are buffered by the network layer until the node regains connectivity, hence increasing the delay. Since the wireless interface must reattach to a new base station, the radio signal may suffer some disruptions. In Mobile IPv6 the IETF has standardized Fast Handovers for Mobile IPv6 [31] (FMIPv6) a protocol extension that speeds up the handover. This is achieved by adding complexity at the network, specifically at the access routers. In FMIPv6 access routers buffer packets addressed to the old location of the node and when the node regains connectivity these packets are forwarded. This reduces considerably the amount of packet lost. In addition the access routers prepare the network layer reconfiguration before the handover takes place. This reduces the handover latency and the impact on the delay and on the jitter.

Contributions

This thesis studies the impact of the handover on the QoS of the mobile node's communications. More precisely:

1. The thesis presents a general methodology intended to evaluate handover-related metrics of layer-3 mobility protocols. This methodology is based on a mixture of active and passive measurements and it is able to differentiate between the different parts of the handover.

2. Provide an empirical analysis of the handover of the layer-3 mobility protocols.

3. Provide the first public implementation of the FMIPv6 protocol and evaluate its performance using the developed methodology. The implementation can be found at [222].

Publications

The research conducted in this part of this thesis has been published in the following papers:

- Albert Cabellos-Aparicio, Jordi Domingo-Pascual, ”Enhanced Fast Handovers Using a Multihomed Mobile IPv6 Node” in Proceedings of IEEE New2AN 2005, St. Petersburgh, Russia

- Albert Cabellos Aparicio, Jose Núñez-Martínez, Hector Julian-Bertomeu, Loránd Jakab, René Serral-Graciá, Jordi Domingo-Pascual "Evaluation of the Fast Handover Implementation for Mobile IPv6 in a Real Testbed" in Proceedings of IEEE IPOM 2005, Barcelona, Spain

- Albert Cabellos Aparicio, Hector Julian-Bertomeu, Jose Núñez-Martínez, Loránd Jakab, René Serral-Graciá, Jordi Domingo-Pascual "Measurement-Based Comparison of IPv4/IPv6 Mobility Protocols on a WLAN Scenario" in Proceedings of Networks UK HET-NET Ilkley, UK 2005

- Albert Cabellos Aparicio, René Serral-Graciá Loránd Jakab, and Jordi Domingo-Pascual "Measurement Based Analysis of the Handover in a WLAN MIPv6 Scenario" in Proceedings of Passive and Active Measurements 2005, Boston, USA.

- Albert Cabellos Aparicio, Lluís Calafell Liebanas, Jose Núñez-Martínez, Jordi Domingo-Pascual "Desarrollo y Evaluacin del Protocolo Fast Handovers for Mobile IPv6 en un Entorno Virtual". Jornadas Técnicas Rediris, Logroño Octubre 2005 (only available in spanish)

- Josep Mangues Bafalluy, Albert Cabellos Aparicio, René Serral-Graciá, Jordi Domingo Pascual, Antonio Gómez Skarmenta, Tomás P. de Miguel, Marcelo Bagnulo, Alberto García Martnez, "IP Mobility: Macromobility, Micromobility, Quality of Service and Security". in Novatica journal, no 165, May/June 2004

1.3.2 Near-Future: Deployment phase of Mobile IP

As it has been mentioned above it is expected that the Mobile IP technology is deployed in the Internet in the near future. At present IPv4 is the predominant protocol, therefore it is reasonable that first Mobile IPv4 is deployed, and whenever IPv6 is available, migrate to Mobile IPv6. However Mobile IPv6 has many advantages in front of Mobile IPv4. First it has the inherent advantages of IPv6 in front of IPv4, a huge address space that allows real end-to-end communications, built-in security and extensibility. Second and most important, Mobile IPv4 lacks of route optimization, that is, mobile nodes communicating with their peers must forward their data packets through the Home Agent. This leads to a sub-optimal path that increases the delay, the infrastructure load and the load at the home link that may become the bottleneck of the whole system. The lack of route optimization in Mobile IPv4 is due to the fact that it requires some support at the mobile node's peers. Since standard IPv4 does not provide such support, Mobile IPv4 lacks of route optimization.

Regarding Mobile IPv6 it incorporates route optimization and it is built-in in IPv6. It is worth to note that, Mobile IPv6 should be included in any implementation. However existing IPv6 implementation do not include mobility [27; 28]. This is because Mobile IPv6 is a recent standard and the implementations have not been tested enough, thus they are not mature yet.

Given the above considerations, we identify three deployment scenarios:

- *Gradual deployment*: In this deployment scenario the Internet evolves gradually from the current status: IPv4 to a full-featured Mobile IPv6 Internet. First Mobile IPv4 is deployed, then IPv6, and whenever the Mobile IPv6 implementations are ready the protocol is deployed. Note that in this deployment scenario, during a certain period of time there are IPv6 nodes without mobility support.

- *Hybrid deployment*: In this case first Mobile IPv4 is deployed and then, IPv6 along with Mobile IPv6 are deployed.

- *Straight deployment*: In this deployment scenario mobility is not deployed until Mobile IPv6's implementations are mature enough, and the Internet becomes Mobile IPv6-eabled directly.

It is worth noting here that these deployment scenarios are not considered as strict cases. The Internet is not controlled by a single entity and different parts of the Internet may evolve separately. For instance the lack of IPv4 address affects more Asia than Europe or America and some parts of the Internet can begin to deploy IPv6 sooner than others.

The first scenario implies deploying a Mobile IPv4 Home Agent at the networks serving mobile nodes and Foreign Agents at the visiting networks. The main issue in this case is the lack of route optimization. In fact this may prevent Mobile IPv4 from being deployed since this indirect routing affects significantly the performance of communications, especially for real time applications.

In the second case the main concern is the deployment cost of Mobile IPv6. This cost includes deploying a Home Agent in each network servicing mobile nodes, and modifying the kernel of the mobile node's peers to include route optimization support.

Finally, in the third case the Internet is Mobile IPv6-aware and mobile nodes benefit from built-in route optimization and, potentially, the extra-functionalities and improvements of the Mobile IPv6's extensions. However we have identified two potential issues. Firstly Mobile IPv6's route optimization is not compatible with some Load Balancing techniques. This is because many load balancing techniques changes the IP header using NAT-like techniques. Secondly mobile node's connectivity depends on its Home Agent that represents a single point of failure in Mobile IPv6-based networks.

Contributions

Given the above considerations this thesis identifies and analyzes the following potential issues and explores its solution space.

1. *Lack of Route Optimization in Mobile IPv4*: This issue reduces significantly the performance and may be an obstacle when considering the deployment of Mobile IPv4. This thesis explores a mechanism to provide route optimization to Mobile IPv4 nodes. The mechanism should be Internet-scalable and should not require deploying new entities or modifying end hosts. Since NEMO (mobility for networks [63]) also suffers from the same problem the mechanism should be also applicable to mobile routers.

2. *High deployment cost of Mobile IPv6*: In case that IPv6 and Mobile IPv6 are deployed separately all the end hosts of the Internet should later be upgraded to support mobility. This thesis explores mechanisms to deploy route optimization support with a low deployment cost and without modifying the end hosts.

3. *Incompatibility between Mobile IPv6's route optimization and Load Balancing techniques*: As it has been stated previously and discussed in further chapters route optimization is not compatible with load balancing techniques. This thesis explores solutions to this issue.

4. *Home Agent represents single-point of failure*: In Mobile IP-based networks the Home Agent represents a single point of failure. This reduces the overall robustness of the Internet and increases the deployment and maintance cost. This thesis explores mechanisms to solve this issue by adding some complexity at the edges of the network.

Publications

The research conducted in this part of this thesis has been published in the following papers:

- Rubén Cuevas, Albert Cabellos-Aparicio, Ángel Cuevas, Jordi Domingo-Pascual, Arturo Azcorra "fP2P-HN: A P2P-based Route Optimization Architecture for Mobile IP-based Community Networks" to appear in Computer Networks, Elsevier, special issue in Content Distribution Infrastructures for Community Networks (January 2009)
- Albert Cabellos-Aparicio, Jordi Domingo-Pascual, "Mobility Agents: Avoiding the Signaling of Route Optimization on Large Servers" in Proceedings of IEEE PIM-RC 2007, Athens, Greece
- Albert Cabellos-Aparicio, Jordi Domingo-Pascual, "A Flexible and Distributed Home Agent Architecture for Mobile IPv6-based Networks" in Proceedings of IFIP Networking 2007, Atlanta, USA.
- Albert Cabellos-Aparicio, Jordi Domingo-Pascual, "Load Balancing in Mobile IPv6's Correspondent Networks with Mobility Agents" in Proceedings of IEEE ICC 2007, Glasgow, UK

1.3.3 Future: New Architectures

After the deployment phase of the Mobile Internet all the nodes are Mobile IPv6-enabled and can potentially change its point of attachment seamlessly. At the same time wireless technologies are widely available and it is reasonable to think that this trend will hold. The properties of the economy of scale reduce the cost of wireless interfaces. Hence, in the future terminals can be equipped with multiple wireless interfaces. These multiple interfaces can increase the overall performance of the mobile nodes. For instance an unreliable and high-bandwidth link can be used for file transfer while low-latency links can be used for real-time communications.

Nodes equipped with multiple interfaces are known as multihomed [71]. This is a well-known technique whose main objectives are to increase reliability and provide more aggregate bandwidth. Many researchers have proposed mechanisms to exploit multihoming in static nodes or networks [72; 73; 74], however multihoming in mobile environments is a relatively recent research topic. This presents both a challenge and an opportunity for the Mobile Internet. Terminals equipped with multiple interfaces can increase the overall performance and provide extended functionalities and features. On the downside managing multiple paths may be a complex task.

In order to efficiently accommodate terminals with multiple interfaces the Mobile Internet requires a new mobile architecture. Basically this architecture should be aware of the QoS requirements of the data flows and assign them to the most suitable path. To achieve this goal the architecture must know the QoS flow requirements and the characteristics of the available paths. The first issue has already been addressed by the QoS research community. For instance the QoS API [85] extends the socket interface so that applications can specify their requirements.

Using the methodologies and tools proposed by the measurement research community can solve the second issue. In fact it is reasonable to consider that estimating QoS metrics (delay, jitter or packet loss) end-to-end is an already solved problem. Unfortunately estimating the available bandwidth is a much more complex task. The available bandwidth is the remaining capacity of an end-to-end path. This is a key metric when considering the performance of this architecture since it is a basic parameter to efficiently schedule flows to paths.

Researchers have proposed sound methodologies and tools to estimate this metric [93; 94; 95; 96; 97; 98; 99; 100; 101]. However these mechanisms are designed for FIFO-based wired networks. In fact it has been shown empirically [102] and theoretically [103] that these techniques are not suited to work in scenarios where wireless networks are present. A reason for that is that wireless networks such as IEEE 802.11 [104] operate using distributed MAC algorithms with random access and non-FIFO scheduling disciplines.

At the best of the author's knowledge there are very few tools able to estimate this metric in the presence of wireless links [102; 105]. However their accuracy is very low when operating in realistic conditions, that is wireless networks with multiple contending nodes and variable packet sizes.

Contributions

As we have seen there is a lack of fundamentals, methodologies, models and tools to estimate the available bandwidth in wireless networks (i.e. random access networks). This can potentially limit the performance of multihomed mobile architecture since this is a key parameter when we consider efficient scheduling algorithms. This thesis explores solutions to these issues with the following contributions:

1. Review the implications of wireless networks in the estimation of bandwidth related metrics. This thesis provides an analytical model that describes the relation between the input and the output gap of periodic probe traffic in a wireless links. Our main conclusion is that existing methodologies, based on periodic probing processes, target the achievable throughput instead of the available bandwidth This is a different metric related with the fair-share of the link.

2. Explore methodologies based on poisson probing processes with the goal of estimating the available bandwidth in such networks. This analysis has lead to the design of a tool: W-Path. This is an heuristic based-tool able to estimate this metric in the presence of wireless links.

3. Explore the benefits of poisson-based methodologies to the available bandwidth estimation in wired scenarios. Our findings show that these methodologies can provide accurate tools that are less intrusive than the existing ones (based on periodic probing). We contribute with two different tools: AKBest & PKBest. The first one is an active tool while the second is a passive one.

Publications

The research conducted in this part of this thesis has been published in the following papers[1]:

[1] Several papers are, at the writing of this thesis, under review.

- Albert Cabellos-Aparicio, Francisco J. Garcia, Jordi Domingo-Pascual, ”A Novel Available Bandwidth Estimation and Tracking Algorithm” in Proceedings of 6th IEEE Workshop on End-to-End Monitoring Techniques and Services (E2EMon), Salvador de Bahia, April 2008

- Albert Cabellos-Aparicio, Francisco J. Garcia, Jordi Domingo-Pascual ”A non-congesting available bandwidth estimation and tracking algorithm”, PDNo:20080257, 25-Jan-2008.

1.4 Road Map

The thesis is organized in five parts. In the sequel, we overview the contents of each part and the topics addressed in its corresponding chapters.

Part I

After a brief summary of the aims and contributions in Chapter 1, the subsequent chapter in this part analyzes the transition to the Mobile Internet. That is the technical evolution of the Internet's architecture to fully support mobility. The main objective of this analysis is to identify potential issues of the transition. Specifically the analysis is carried out at three different stages of the transition. First at present, where we analyze the different mobility solutions (at different layers). This analysis is focused on the main advantages, drawbacks and cost of deploying mobility at the network, transport and session layer. Second in the Near-Future, during the deployment phase of mobility. In this case the analysis helps us to identify potential issues in the performance and the deployment of the Mobile IP family of protocols. Finally, in the Future, where new architectures can enhance the functionalities and the performance of the Mobile Internet.

Part II

This part deals with the potential issues identified in the previous part. The main conclusions of the above-mentioned analysis is that at present, the major issue when considering the Mobile IP technology is the performance of the handover. Therefore this thesis analyzes the performance of the handover both analytically and by experimentation.

Chapter 3 presents an analytical model of the handover of Mobile IPv4, Mobile IPv6, Hierarchical Mobile IPv6, Fast Handovers for Mobile IPv6 and Network Mobility.

Chapter 4 provides a sound methodology to measure several metrics related with the handover of the above-mentioned protocols. Further the methodology is applied in a testbed to provide results of the performance of these protocols.

Part III

This part of the thesis provides solutions to the potential issues identified in the deployment phase of the transition to the Mobile Internet. In particular we address the lack of Route Optimization in several scenarios and enhance the performance of the Home Agent for the Mobile IP technology.

Chapter 5 proposes the Mobility Agents to reduce the deployment cost of Mobile IPv6's route optimization. This is a transparent proxy that handles mobility signaling and that provides compatibility between the return routability procedure and several load balancing techniques.

Chapter 6 presents the flexible Home Agent architecture. This novel entity reduces the load of Home Agents in Mobile IP-based networks and increases reliability.

Chapter 7 details the fP2P-HN architecture. This is a P2P-based mechanism that provides route optimization to Mobile IPv4 and NEMO clients. As we detail in chapter 2 this is one of the major issues of the Mobile Internet.

Part IV

This part of this thesis explores solutions for estimating bandwidth-related metrics. This is a key issue when considering the performance of multihomed mobile architectures. These architectures can enhance the functionalities and performance of the Mobile Internet.

Chapter 8 establishes the fundamentals of bandwidth estimation in wireless networks. In particular we show that traditional methodologies and tools target the achievable throughput rather than the available bandwidth. Additionally our study shows that periodic probing processes have bias.

Chapter 9 explores poisson probing processes to design methodologies able to estimate the available bandwidth in the presence of wireless links. This leads us to design W-Path, a tool able to infer congestion in wired and wireless scenarios. Further the chapter applies such probing processes to design tools for wired scenarios. We show that poisson probing can provide accurate tools with low intrusiveness and that do not impact the performance of the path under measurement. In particular we present two different tools and active and a passive one: AKBest and PKBest respectively.

Part V

This last part presents the conclusions that can be drawn from this thesis and analyzes future lines of work.

Chapter 10 highlights the main conclusions of this thesis.

Chapter 11 proposes several ways for extending the reach of the work done in this thesis

2

The Transition to the Mobile Internet

This chapter analyzes the transition to the Mobile Internet. The first section discusses the benefits and drawbacks of implementing mobility at different layers of the Internet stack. Then the second section analyzes the different deployment scenarios. Finally in the last section we detail future new architectures and directions of the Mobile Internet.

2.1 Present: State-of-the-Art

In the current Internet status, a node's IP address identifies the node's point of attachment, that is its topological position at the Internet. Therefore the node is located by its IP address in order to receive datagrams. For a node changing its point of attachment without losing its ability to communicate, two straightforward mechanisms can be employed [13]:

1. The node must change its IP address whenever it changes its point of attachment, or

2. Host-specific routes must be propagated throughout the Internet for each end host movement.

Both of these alternatives are often unacceptable. The first makes it impossible for a node to maintain transport and higher-layer connections when the node changes location. The second one has obvious and severe scaling problems, especially relevant considering the explosive growth of the Internet. Therefore a new scalable mechanism is required for accommodating node mobility within the Internet. In order to analyze the transition to the Mobile Internet the first requirement is to study how mobility can be accommodated in the current Internet architecture.

This issue has raised a philosophical discussion among the research community [5; 6]. Which is the optimal way to implement mobility? According to the end-to-end principle mobility should be managed by end hosts, and by reviewing the OSI reference model, mobility should be located at the session Layer. But the TCP/IP model has always lacked of a proper session layer. Hence there are only two options, either we develop a complete new architecture that incorporates mobility [21] or we modify the current TCP/IP model to incorporate this functionality. New architectures are already being researched [22; 23], but this is considered as long-term research and they will not be ready in a reasonable period of time. However users are demanding this functionality, and both the research community and the industry must design a mid-term solution to this issue. That is why this thesis is focused in the mid-term transition to the Mobile Internet. In this section we analyze what is mobility and which are its requirements and then we review the proposed solutions to analyze their feasibility.

2.1.1 What is Mobility?

Mobility is the ability of a mobile node to change its point of attachment to the Internet without breaking its network connections. Mobility has two desirable properties [5][6]:

1. *Seamless Transition*: Changing the point of attachment to the Internet should not result in unacceptable loss of application data and should minimize the disconnection time. In the case that the connections are broken, this should be transparent to applications. This is especially important for long-lived connection-oriented communications such as telnet sessions.

2. *Location Management*: The moving device must always be reachable via some static identifier regardless of its physical location.

Given the above requirements, the design principles of a mobility solution are:

1. *2-Dimension Identifiers*: A mobile node must include two different identifiers. The first one, generically known as WHO, must be related to the identity of

the node and must be static to ensure reachability. The second one, known as WHERE, must be related to the location of the node and must change according to its movements.

2. *Location Management Mechanism*: A mechanism (host-centric or network-centric) is required in order to maintain bindings between the WHO and the WHERE addresses. This mechanism redirects datagrams addressed to the WHO identifier to the current location of the node (WHERE). Additionally, to ensure seamless transitions, when the WHERE identifier changes bindings must be updated quickly.

3. *Authentication*: Each time the mobile node changes its WHERE identifier it must inform the location management system and/or its peers. This must be done securely otherwise an attacker could impersonate the mobile node and hijack its connections. Therefore a mobility implementation must include some authentication mechanism.

Considering the requirements and the design principles of mobility the research community has proposed several solutions, at different layers, to provide mobility to the current Internet architecture. In the following subsections we review these solutions.

2.2 Analysis of Mobility Solutions at different layers

2.2.1 Session Layer

Internet applications are usually engineered using the concept of sessions, that is a long-term relationship that may span multiple transport connections [225]. An example of this are interactive sessions, such as telnet or ssh, web sessions or file transfer sessions. Since sessions consist of one or more transport connections they provide a convenient abstraction with which to manage coordinated application state between hosts.

As it has been previously mentioned the OSI reference model defines an explicit session layer that provides synchronized message exchange. This layer establishes and terminates sessions, exchanges tokens and negotiates communications fundamentals such as full or half-duplex. In addition the OSI session layer allows the definition of synchronization points within the session, the interruption of a session and session resumption from a previously agreed synchronization point.

Transport protocols are unable to deal with changes in attachment points, that is why applications running in the Internet have been forced to develop their own application-specific mechanisms for managing mobile end points. In fact these applications include session features to effectively aggregate multiple connections from different network attachment points into one single relationship. HTTP cookies for Web-based applications is a good example of this approach.

Mobility could easily fit into the session layer. The mobile node should have 2-Dimensional identifiers. Clearly the network layer can provide WHERE identifiers (IP addresses). After a new WHERE identifier is obtained (when the mobile node changes its point of attachment) the session layer may simply initiate new transport connections to replace the existing ones. Alternatively if there were transport layer protocols that had mobility support in the form of dynamic address rebinding (the case of SCTP [9; 40]), the session layer can simply monitor movement and trigger binding updates. The WHO identifiers cannot be easily included into the session layer since it operates end-to-end, not network-wide. Session-layer mobility must use some type of external identifiers and a location management infrastructure that translates WHO identifiers to WHERE identifiers is necessary. Finally it requires authenticating mechanisms to provide trustness.

Even that this layer is the most appropriate to include mobility very few proposals have been made. This may be due to the fact that the idea of using session layers has not been popular amongst application developers who may just as easily (with a similar amount of code) create and use transport layer connections. The different proposals of session-layer mobility [33; 34; 35; 36; 37; 39] use IP addresses as WHERE identifiers and external ones, such as e-mail addresses [38; 39] or hostnames [33; 34; 35; 36; 37]. Additionally they use specific rendez-vous servers to ensure rechability. In all the proposals new transport connections are established after each change of point of attachment and the main challenge among all of them is to incorporate authentication. This problem would be relatively easy to solve if IP Security (IPSec) [138] were deployed, however IPSec has not found widespread deployment. The approach followed by these solutions is either using cryptographic tokens exchanged during the connection establishment [33; 34; 35; 36; 37] or using user-based authentication methods [38; 39]. It is worth to note that the lack of real deployment of session layers among common Internet applications, prevent these proposals from having much impact.

Summarizing session layer mobility provides many advantages; the transitions can be made very smooth if both sides can preset some state before the transition takes place. This can allow them to resume the communications very quickly. In addition the session layer hides mobility related issues to the applications and some complexity might be removed from the already over-featured transport and network layer. The main drawbacks are that it provides a new interface to applications and thus, they must be updated. In addition it requires an external infrastructure (location management) and external WHERE identifiers operating outside the session layer, this may impact the overall performance. Finally it has been proved as very difficult to provide efficient authentication mechanisms to session-layer mobility solutions.

2.2.2 Transport Layer

The transport protocol is responsible of delivering data packets between two end points. Connection-oriented transport protocols establish a communication channel, or a connection between two end points and exchange packets over the connection. One of the main tasks of a transport protocol is to multiplex communication channels between end points. Some protocols provide additional services such as reliability and packet reordering.

Due to the constraints imposed by the layered nature of the TCP/IP model, the transport protocol use the network layer thus, IP addresses. Hence connections in the Internet are communication channels between two network attachment points, not end points. Several connections are multiplexed by using a special tag: the port number. Thus an Internet transport connection end point is specified in terms of the well-known tuple <*IP Address, port*>. It is easy to see that this proves problematic if an end point changes its network attachment point. Hence transport protocols defined in the TCP/IP model do not support mobility.

Connection Migration

Once a connection is established between two attachments points, both end points only accept packets addressed from the other point of attachment. Hence connection established by an end point at one attachment point cannot be used from another one.

In order to resume this communication a new connection must be established. This raises two complications. First each end point must discover a new attachment point and somehow communicate it to the remote end point. Second after both end points agree on the new attachment point, the end points must abort the old connection and establish new ones. If reliability is required this may result in the loss of packets that were not yet successfully transmitted on the initial connections. This is known as connection migration. In order to implement mobility at the transport layer the first requirement is to allow connection migration. Since TCP and UDP [227] do not implement this feature the research community has proposed solutions at the transport layer to provide this feature. Mainly three different approaches have been used.

First some researchers have focused on TCP and have introduced higher-level mechanisms to manage multiple separate TCP connections into one virtualized connection [41; 42; 43; 44]. This is similar to including session-layer functionalities at the transport layer. These proposals define a new WHERE identifier and connection reestablishment is managed by a software library interpose between applications and the operating system. Connection semantics are not modified and some proposals [43] are even able to interact with non-modified TCP hosts. On the contrary MSOCKS [10] proposes using a SOCKS proxy [45] to forward transport connections to a mobile node. Mobile peers establish regular TCP connections with the proxy, in turn the proxy establishes a separate TCP connection with the mobile node. If the mobile node changes its point of attachment a new TCP connection is established with the SOCKS proxy. As before, the SOCKS library conceals the ”virtual” connection from applications on the mobile end point.

Second, some authors propose modifying TCP itself to support changes in attachment points. For instance ETCP [46], an extension to the standard TCP, includes a flow identifier (acting as the WHO identifier). By assigning each TCP connection a unique connection identifier end points can associate packets with the appropriate end point, regardless of WHERE identifier. An alternative TCP extension, TCP-R [11], defines a special redirection message (similarly to SCTP) that contains the IP address of the new point of attachment.

Third, the IETF has defined new transport protocols aimed to replace TCP and UDP. None of both protocols define a proper WHERE identifier, instead they use a similar approximation to that used by TCP-R, a connection identifier. The Datagram

Congestion Control Protocol (DCCP) [47], a replacement for UDP, allows end points to notify to its peers a new network attachment point by sending a special signalling message. The Stream Control Transport Protocol (SCTP) [9; 40], intended to replace TCP, also allows end points to change attachment points and it even supports using a set of attachment points simultaneously (multihoming). A special extension called mobile SCTP (mSCTP) [48] provides seamless transitions for mobile nodes.

Location Management

The second requirement is to incorporate a location management mechanism. None of the above mentioned solutions propose a specific solution, but a simply rendez-vous server suffices. In this case mobile nodes should signal its current location to this server and peers willing to establish communication should direct its packets to the rendez-vous server. Once the connection has been established the mobile node can include its current attachment point to the transport connection.

Authentication

The last requirement is to provide authentication methods for the mobile node. Again, none of the previous solutions propose any specific method. In fact, the SCTP standard states that authentication of points of attachment must be offered by the network layer [9; 40]. However, at is has been mentioned before, IPSec is not a widely available technology.

As a summary we can conclude that the main advantages of transport layer mobility are that communications are always route-optimized, this means that connections are not forwarded through the rendez-vous server. Additionally, if more than one interface is available, a multihomed transport protocol, such as SCTP, can achieve seamless transition without losing packets. Transport-layer mobility can be transparent to the applications, or a more rich interface can provide mobility-related information to the applications that can react if a disruption will or has occurred.

Regarding the drawbacks they are quite similar to session-layer mobility: dependence on other layers for location management, applications must be updated (if the transport interface changes) and lack of authentication methods. In any case it is mandatory that the transport layer is aware of mobility. A mobility scheme that hides mobility to the transport layer is problematic because this layer is responsible of congestion control. Good congestion control requires optimizing throughput at the end-to-end path. If the path changes due to mobility, then the transport layer needs to be aware so that it may adjust the rate. In fact this problem has been identified in TCP operating over WLAN networks [49]. In this scenario TCP is unable to differentiate if a packet was lost due to congestion or to a link-layer error.

2.2.3 Layer 3.5

A different approach is to incorporate a new layer into the Internet TCP/IP model between the network and the transport layer. This is the solution proposed by the IETF with HIP (Host Identity Protocol) [50; 51]. HIP basically introduces a new 3.5-layer to avoid that sockets are bind to IP addresses forcing them to act both as the WHO and the WHERE identifiers. In HIP, upper layer sockets are bound dynamically to Host Identities (HI) instead of IP addresses.

The operation of HIP is as follows: hosts are identified with a HI that, in fact, is a public key of an asymmetric key-pair. Each host has at least one HI that can either be public or anonymous. Since public keys may have different sizes depending on the public key method HIs are represented via its 128 (SHA-1) hash, called Host Identity Tag (HIT) or via 32 bit Local Scope Identity (LSI). The HIT and LSI identify the public key that can be used for authentication purposes and are unique. HIT are mainly used for IPv6 while LSIs for IPv4. This way HIP is compatible with both versions of IP and does not require updating them.

During connection establishment HIP must be used. In this case the transport layer protocol (e.g. TCP) must be enclosed with a HIP header, which contains either the HIT or the LSI (figure 2.1). The transport-layer end hosts are bind to this identifier, instead of the IP address. Since HITs are public keys it uses the Diffie-Hellman [205] key exchange to provide authentication. Additionally it can also provide confidentiality and message integrity. During the authenticated connection, mobility in HIP is quite straightforward. As HIs are used to identify the mobile node instead of IP addresses, the location of the node is not bound to the identifier. Therefore only a simple signalling protocol is needed to take care of the dynamic binding between the node's IP address and its HI. When the mobile node changes its point of attachment it simply sends a special signalling message (HIP REAddress) through the already authenticated channel.

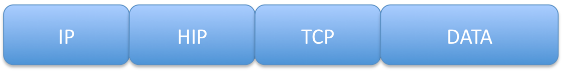


Figure 2.1: HIP Connection establishment

Again, since sockets are bound to HITs and not IP addresses, the connection can continue uninterrupted.

Finally a simple rendez-vous server is required to ensure reachability. This rendez-vous server is aware, through the use of some simple signalling, of the current location (IP address) of its serving HIP nodes. When another HIP-enabled node wants to establish a communication it retrieves its HIT identifier in some public address directory, such as the DNS. This directory stores both the HIT and the rendez-vous IP address. Then the node sends the initial HIP connection establishment method to the rendez-vous server, which in turn, forwards it to the actual location of the node. The remainder datagrams can be sent directly (route optimization).

The main advantages in HIP are that it does not change the architectural principles of the socket interface and that is transparent to applications. In addition since it is based in public key identifiers it relies on well-known and proven security mechanisms that provide authentication, confidentiality and message integrity. However this has also a downside, cryptographic algorithms, especially those based on asymmetric key pairs, are very costly in terms of CPU. Mobile devices have limited CPU power and HIP may impact its performance.

2.2.4 Network Layer

According to the OSI reference model, the network layer provides the functional and procedural means of transferring variable length data sequences from a source to a destination via one or more networks. The network layer performs network routing functions, and might also perform fragmentation, reassembly, and report delivery errors. Routers operate at this layer forwarding data throughout the extended network, and making the Internet possible. The TCP/IP model uses one particular network-layer protocol: the Internet Protocol (IP).

The Internet Protocol provides IP addresses that are hierarchical identifiers that reflect the topological location of an attachment point in the Internet and enable the Internet routing infrastructure to deliver packets destined to network point of attachments. Hence IP addresses identify both the end point (WHO) and its location (WHERE). In order to incorporate mobility at the network layer there are two different approaches. Either use host-specific routes, updating them as each host moves, or use indirection agents to the architecture. These agents forward mobile node's packets from the home network to the mobile node's current location. Although there are some proposals that follow the first approach [52; 53; 54] they are clearly unscalable to the number of hosts on the Internet and can be disregarded. The second approach has received much attention among the research community and many similar solutions have been proposed [55; 56]. Among all of them, the one that stands out is Mobile IP [13; 14], proposed by the IETF.

In Mobile IPv4 [13] a mobile node has two IP addresses. The first one identifies the mobile node's identity (Home Address) while the second one identifies the node's current location (Care-of Address). The mobile node is always reachable through its HoA while it changes its CoA according to its movements. Reachability is achieved through the Home Agent (a rendez-vous server) located at the node's Home Network, that is, its usual network. The Home Agent intercepts packets addressed to the HoA and redirects them to the CoA. Hence the mobile node must inform to its Home Agent about its location. Finally, authentication is not necessary in basic Mobile IP operations since the node's peers always communicate through the Home Agent. This forces packets to follow a sub-optimal route. This increases delay, the infrastructure load and the Home Agent itself may become the bottleneck of the whole system. To solve this Mobile IP for IPv6 (Mobile IPv6 [14]) incorporates authentication, that is, the mobile node can prove that it owns a given HoA and a given CoA. This is achieved by the Return Routability procedure that relies on the routing system. Basically the Return Routability procedure operates as follows: the node's peer sends two challenges, one addressed to the CoA and another one addressed to the HoA. If the mobile node's effectively owns both addresses receives both challenges. By combining both challenges the mobile node obtains a token that allows it to communicate directly. Note that this method does not require the deployment of IPSec in the Internet. The major upside to network-layer mobility is that since it is at the waist of the protocol stack model mobility can benefit every higher layer. This is not only beneficial from the standpoint of minimizing reproduction of effort, but also in limiting potential bugs or security concerns. In addition it has built-in location management while seamless transitions can only be achieved by cooperating with the transport protocol. Unfortunately the IP interface layer is not rich enough to provide such information to the transport layer. In addition Mobile IPv6 incorporates routing-based authentication mechanisms and does not depend on an external infrastructure such as Certification Authorities. Note that this Mobile IPv4 does not include route optimization and that this impacts significantly the performance of the communications.

2.2.5 Comparison of Mobility at different layers

The host mobility problem can be attacked from many layers. Link layer support is mandatory in any case, but we can do little to either preserve higher layer connections or provide location management when movement is across administrative domains. Table 2.1 summarizes how the mobility requirements are met considering the different layers at which mobility can be implemented.

Table 2.1: Mobility Requirements

Layer	**Seamless Transitions**	**Reachability**
Session-layer	Included	Requires external infrastructure
Transport-layer	Included	Requires external infrastructure
Layer 3.5	Included	Requires external infrastructure
Network-Layer	Transport layer must deal with disruption	Built-in

As the table 2.1 shows true seamless transition can only be achieved at the transport or the session layer. These layers interact directly with the applications, and can deal with communication disruptions. Regarding reachability, all the layers achieve it, although the network-layer, responsible of routing, implements it with a lower convergence time.

Table 2.2 presents how the different functionalities, inherent to mobility, can be implemented at the different layers. The most complex functionality is authentication, higher layers are not well-suited to provide authentication, and this can only be achieved using IPsec. Unfortunately this technology is not widely available, and its performance in terms of delay/latency is low. Obviously this impacts the performance of mobility. Regarding lower layers, they have built-in mechanisms to provide authentication and they do not require any external infrastructure.

Table 2.2: Mobility functionalities

Layer	**Location Management**	**2-Dimensional Identifiers**	**Authentication**
Session-layer	Requires external rendez-vous server	Service identifier and location identifier	Requires IPSec
Transport-layer	Requires external rendez-vous server	Connection identifier and location identifier	Requires IPSec
Layer 3.5	Requires external rendez-vous server	Identity (HIP) and IP address (location)	Built-in
Network-Layer	Buit-in	Two IP addresses: location and identity	Built-in

Summarizing, lower layers provide higher performance in terms of convergence time and authentication, while higher layers are best suited to provide true seamless transitions. Basically this is because lower layers operate into the network while higher layers operate closer to the application. Although the question of what layer mobility should properly be provided at is largely an open question we extract the following conclusions from our analysis:

- The most complex task is providing authentication mechanisms to mobile nodes. Taking this argument into consideration, the most complete mobility solution is provided by the network layer because it implements an effective routing-based authentication mechanism.

- Network-layer mobility is unable to provide true seamless transitions while higher layers have a lower performance in terms of location management.
- Possibly, the optimal mobility solution is a cross-layer approach. A cross-layer solution involving both the network and the transport layer can provide good performance and seamless transition.

2.2.6 Deployment of Mobility at different layers

In the previous sections we have analyzed which is the best layer to accommodate mobility from a technical point of view. In this section we have a practical point of view and we analyze the deployment cost of mobility at different layers considering the current status of the Internet.

Incorporate a new session layer into the existing Internet infrastructure is a challenging approach due to the cultural unacceptance of such layer in the Internet community. In this case all the end hosts should be updated incorporating the new layer and all the applications should be re-programmed to use the new session interface. Finally very little infrastructure is required, since only rendez-vous servers have to be deployed. The authentication problem has not been yet solved, and possibly session-layer mobility would require widely available public-key infrastructure.

Deploying transport-layer mobility is similar to deploying session-layer mobility. Again applications must be recompiled to link with the new transport layer. If the new transport protocol incorporates a new interface, then applications should also be reprogrammed.

HIP has a reasonable low deployment cost, since it just requires applications to be recompiled. They do not need to be updated since the TCP or UDP interface remain unchanged.

Finally regarding network-layer mobility, two different cases are considered. Mobile IPv4 has the lowest deployment cost, the only requirement is to deploy the Home Agents. However, Mobile IPv6 has a higher deployment cost. Basically because all network-layer communication equipments, such as routers and firewalls, must incorporate IPv6. It is worth to note that Mobile IPv6 is the only solution that incorporates an effective authentication mechanism.

Table 2.3: Deploying mobility at different layers

Layer	Applications	End host	Infrastructure
Session-layer	Re-programed	Updated	Rendez-vous servers and Authentication infrastructure
Transport-layer	Re-compiled and possibly re-programed	Updated	Rendez-vous servers and Authentication infrastructure
Layer 3.5	Re-compiled	Updated	Rendez-vous servers
Network-Layer	Re-compiled	Updated	Rendez-vous servers and network-layer communications equipment

As summary, the question of "At which layer mobility belongs?" is an open question, however, when we consider the deployment cost (table 2.3) and the practical issues that each solution entails we can clearly see that network-layer mobility is the most cost-effective. Mobile IPv4 has the lowest deployment cost, and in fact is supported by the current Internet infrastructure, and IPv6 (with built-in mobility) must be deployed to solve the urgent issues suffered by IPv4. That is why in this thesis we consider the deployment of Mobile IPv6 as the mid-term solution to support mobility in the Internet.

2.2.7 Mobile IP (Background)

In order to identify the potential issues of the Mobile IP family of protocols we present an overview in this section. The reader familiar with these protocols can skip this section safely.

The Mobile IP family of protocols has been defined by the IETF through several RFCs. Basically there are three IETF Working Groups (WG) standardizing and maintaining the Mobile IP technology:

- mip4 (Mobility for IPv4) [57]: This WG is responsible of the Mobile IPv4 standard.
- mext (Mobility EXTensions for IPv6) [58]: This WG has standardized Mobile IPv6, NEMO (Network Mobility) and support for multiple interfaces.
- mipshop (Mobility for IP: Performance, Signaling and Handoff Optimization) [59]: This WG is standardizing extensions for Mobile IPv6 that solve the main issue of this protocol: Signaling Overhead and Handoff Optimization.

Mobile IPv4

Mobile IPv4 is an extension of IPv4 to support mobility. Mobile IPv4 can be thought of as the cooperation of three major subsystems. First there is a discovery mechanism defined so that mobile nodes can determine their new attachment points (new IP addresses) as they move from place to place within the Internet. Second, once the mobile node knows the IP address at its new attachment point, it registers with an agent representing it at its Home Network. Lastly, Mobile IPv4 defines simple mechanisms to deliver datagrams to the mobile node when it is away from its home network.

Formally the Mobile IPv4 standard defines four functional entities:

- *Mobile Node* (MN): an autonomous device equipped with at least one wireless interface.
- *Home Agent* (HA): An entity on a mobile node's home network which delivers datagram's to departed mobile nodes, and maintains current location information for each.
- *Correspondent Node* (CN): A peer with which a mobile node is communicating, can be either mobile or stationary.
- *Foreign Agent* (FA): Foreign agents advertise their presence by using a special message, which is constructed by attaching a special extension to a router advertisement.

Each MN is configured with two IP addresses, the Home Address (HoA) and the Care-of Address (CoA). The first one, represents the WHO identifier, and it is an IP address that is assigned for an extended period of time to a MN and remains unchanged regardless of where the node is attached to the Internet. Usually this address belongs to the prefix of the MN's home network. The Care-of Address, the WHERE identifier, is the termination point of a tunnel towards a mobile node, for datagrams forwarded to the mobile node while it is away from home. There are two different types of Care-of Address: a foreign agent Care-of Address is an address of a Foreign Agent with which the mobile node is registered; a collocated Care-of Address is an externally obtained local address which the mobile node has associated with one of its own network interfaces.

The protocol operations define three different phases. First, in the *Agent Advertisement* phase, Home and Foreign Agents advertise themselves by sending agent advertisements messages. An impatient mobile node may optionally solicit an agent advertisement message. After receiving this message, a MN determines whether it is on its home network or a foreign one. When the MN is at home works like any other node. When it is away it obtains a Care-of Address on the foreign network, for instance by soliciting or listening for agent advertisements or contacting Dynamic Host Configuration Protocol (DHCP) [228].

Then, in the *Registration* phase, the MN registers the Care-of Address with its Home Agent. The registration is done through a special signalling message, called Binding Update. Since the MN and its HA belong to the same administrative domain they have pre-configured keys. This enables the MN to protect the Binding Update message using common cryptographic mechanisms.

The next phase is the *Routing and Tunnelling* where the MN resumes its communications with the CNs. The datagrams sent to the MN's HoA are intercepted by its HA and tunnelled and forwarded to the CoA, received by the tunnel end point and finally delivered to the MN. In the reverse direction datagrams sent by the MN are delivered to their destination tunnelled through the HA. When the HA tunnels a datagram to the Care-of Address, the inner IP header destination (i.e. the MN HoA) is effectively shielded from intervening routers between the home network and its current location. At the CoA the original datagram exits from the tunnel and it is delivered to the MN.

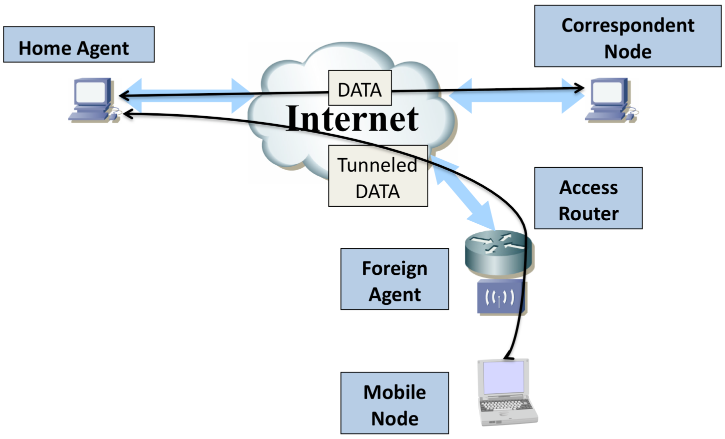


Figure 2.2: Mobile IPv4 Overview

Finally, every HA attracts and intercepts, using the proxy ARP (Address Resolution Protocol) technique, packets destined to the MN sent from the home link.

The last phase is the *Handover* phase. The handover is the process where the MN changes its point of attachment and must regain connectivity. In Mobile IPv4, the MN must discover the new point of attachment and obtain a new CoA (Agent Discovery phase), register its new address with its Home Agent (Registration) and resume its connections (Routing and Forwarding). The handover disrupts communications and some packets may be lost or delayed due to incorrect MN location.

Figure 2.2 illustrates the routing of datagrams to and from the MN away from home, once the MN has registered with its HA. The mobile node is presumed to be using a CoA provided by the Foreign Agent.

Note that Mobile IPv4 lacks of route optimization and communications must be forwarded through the Home Agent. This is because it does not include an authentication mechanism. As we will see later IPv6 has a routing-based authentication mechanism that cannot be migrated to Mobile IPv4 since it requires specific support at the CNs. Including this support would mean modifying all the nodes attached to the Internet, or at least, all the servers.

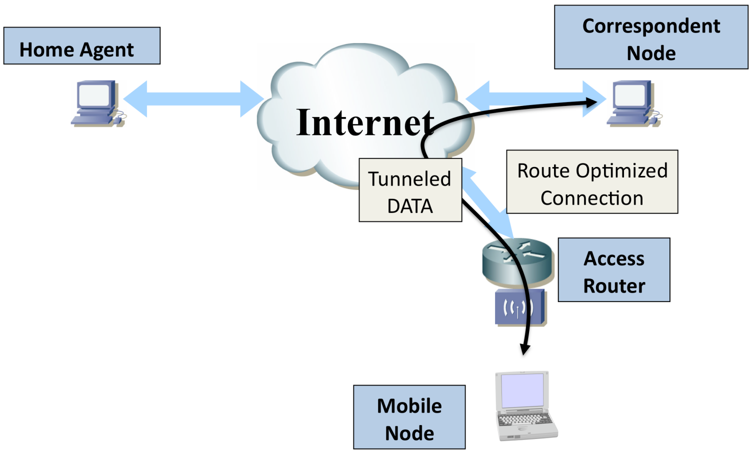


Figure 2.3: Mobile IPv6 Route Optimization

Mobile IPv6

Mobile IPv6 is the IPv6 version of Mobile IPv4. Mobile IPv6 operates similarly to Mobile IPv4 with two important differences.

First Mobile IPv6 is included into a standard IPv6 implementation and thus, all the hosts are potentially mobile nodes. However it is worth noting that existing IPv6 implementations such as [27; 28] do not include mobility. This is because mobility is a much recent standard, at the writing of this thesis several experimental Mobile IPv6 implementations are available [60; 61], and probably they will be included into the IPv6 implementations in a short period of time.

The second main difference is that Mobile IPv6 includes route optimization (see figure 2.3), mainly because all IPv6-enabled nodes support mobility. In Mobile IPv6 routes can be optimized using the Return Routability procedure. This procedure provides a basic authentication mechanism and enables the mobile node to prove that it owns both the CoA and the HoA. This mechanism is based on routing and it is largely discussed in section 5.2.

Mobile IPv6's Issues

The IETF has identified two major issues in Mobile IPv6:

1. *Signalling Overhead*: There is a non-negligible amount of signalling between the

MN, its HA and CNs. This signalling overhead may impact the performance of the Internet if the number of MNs moving around is very large.

2. *Handover Latency*: The lag between when the mobile node has link layer connectivity and when it begins sending and receiving packets on the new link may be substantial. During this interval the MN is not able to send or receive packets.

In order to solve these issues the IETF, through the mipshop WG, has desiged the following Mobile IPv6 extensions:

1. *Hierarchical Mobile IPv6 mobility management* (HMIPv6) [62]: HMIPv6 deals with reducing the amount and latency of signalling between a MN, its HA and one or more CNs by introducing the Mobility Anchor Point (MAP) (a special node located in the foreign network). The MAP acts somewhat like a local home agent for the visiting mobile node by limiting the amount of signalling required outside the MAP's domain.

2. *Fast Handovers for Mobile IPv6* (FMIPv6) [31]: FMIPv6 reduces packet loss by providing fast IP connectivity as soon as a new link is established. It does so by fixing up the routing during link configuration and binding update, so that packets delivered to the old CoA are forwarded to the new one. In addition, FMIPv6 provides support for preconfiguration of link information (such as the subnet prefix) in the new subnet while the mobile node is still attached to the old subnet. This reduces the amount of preconfiguration time in the new subnet.

In this thesis we focus on the handover of network-layer mobility protocols, that is why we provide an overview of FMIPv6 in the following subsection.

Fast Handovers for Mobile IPv6

FMIPv6 is a Mobile IPv6 extension that reduces the handover latency and stores packets delaying them instead of losing them. This is accomplished by allowing the MN to send packets as soon as it detects a new subnet link, such as the IEEE 802.11 link [104], and delivering packets to the MN as soon as the new access router detects its attachment. FMIPv6 has two different operational procedures. In the "Predictive Handover" the MN discovers nearby link layer Base Stations and then requests all the important

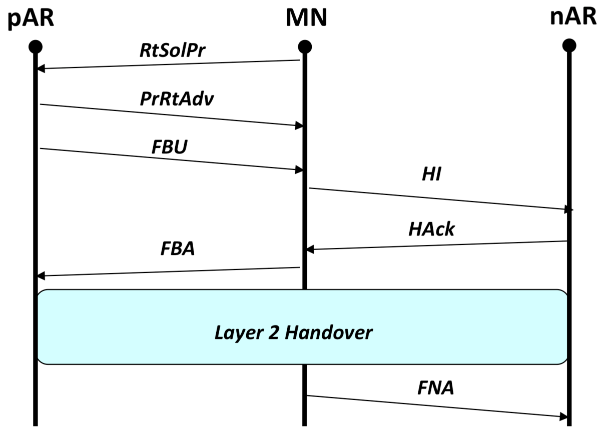


Figure 2.4: FMIPv6 Signaling Interaction

information related to the corresponding new access router. When attachment to a new Base Station takes place, the MN knows the corresponding new router's coordinates including its prex, IP address and MAC address. Through special "Fast Binding Update" and "Fast Binding Acknowledgment" messages the MN is able to formulate a prospective new CoA (without changing its point of attachment), this CoA must be accepted by the new access router prior to the MN movement. Once the MN has changed its point of attachment and it is connected to the new access router link, it can use its new CoA without having to discover the subnet prex, it also knows the new access router MAC and IPv6 address, and hence this latency is eliminated. As soon as it is attached the MN sends a "Fast Neighbour Advertisement" announcing its presence. Moreover, the previous access router will tunnel and forward packets to the new care of address until the MN sends a "Binding Update" registering its new CoA to HA and CNs, hence, no packet is lost. Figure 2.4 shows the messages exchanged during the Predictive Handover operational procedure. NAR denotes the next Access Routers, that is the destination access router while PAR denotes the previous Access Router, that is the original access router of the MN.

The other FMIPv6 operational procedure is the "Reactive Handover" where the network initiates and manages the handover. In this case the PAR decides when the handover is imminent and sends information about the MN to the NAR. Then, when the MN regains connectivity at the new access network, the NAR has already performed the required network-layer operations and the MN can resume its communications immediately.

Additionally the IETF's mext WG is working mainly in extending mobility support to networks. In the following subsection we detail the protocol operations of NEMO.

Network Mobility (NEMO)

Network Mobility [63] (NEMO) been designed as an extension of Mobile IPv6[1], hence its procedural operations are very similar to that of Mobile IPv6. Currently the NEMO Basic Supports ensures session continuity for all the nodes in a mobile network, even as the Mobile Router (MR) changes its point of attachment. It also provides connectivity and reachability for all nodes in the mobile network as it moves. The solution supports both mobile nodes and hosts that do not support mobility in the mobile network. In NEMO Basic Support a MR acts as a Mobile IPv6 node obtaining CoAs and registering them on its HA. The MR configures a tunnel with its HA and all the packets from the mobile network are routed through this tunnel. NEMO Basic Support does not implement any sort of Route Optimization. Basically this is due because with the Return Routability procedure a Mobile IPv6 node can prove that it owns two addresses, the CoA and the HoA. However a MR must prove that it owns the CoA and the Home Prefix. Routing-based authentication mechanisms are difficult to adapt to this situation. Figure 2.5 presents a schema of the operations of NEMO.

Nevertheless several proposals have been published [65; 66; 67; 68] to provide Route Optimizations for NEMO. Most of them require deploying a new entity at the Correspondent Network. These proposals have the same disadvantages as Mobile IPv4 Route Optimization solutions. Since regular nodes do not include explicit support for NEMO a new entity must be deployed, and doing this at an Internet scale may be unfeasible.

2.2.8 Summary and potential issues

As we have seen throughout this section we have analyzed the different solutions and protocols that can provide mobility to the Internet. In our analysis we have seen that the Mobile IP family of protocols is the most cost-effective solution in the mid-term and Mobile IPv6 the only solution that includes effective authentication mechanisms.

[1] There is also an IPv4 version for NEMO (see [223] for details).

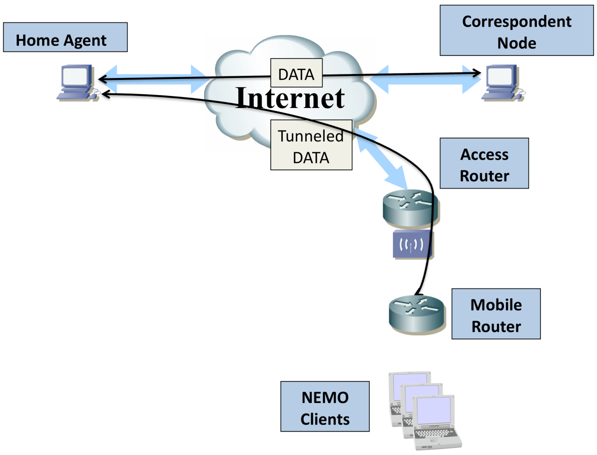


Figure 2.5: NEMO Basic Support

If we consider a long-term solution for the mobile Internet then we have to take into consideration cross-layer solutions based on new architectures.

By analyzing the Mobile IP technology (Mobile IP, Mobile IPv6 and its extensions) we have seen that it has two major issues: the signalling overhead and the handover latency. In this thesis we focus only on the handover latency. The handover (or handoff in other related documents) disrupts communications and can potentially impact the performance of the communications at different layers:

- *Link Layer*: Before, during and after the handover process the quality of the radio signal may be low, for instance 802.11 wireless interfaces trigger the handover when they detect that the signal/noise ratio received by the current Base Station is below a given threshold. This can produce errors during the transmission/reception and impact the delay and/or jitter of the communications.

- *Network Layer*: During the handover the mobile node is not able to send or receive data packets, thus packets are lost.

- *Transport Layer*: Packets lost may trigger congestion control mechanisms of transport protocols such as TCP. In addition the new path may have different QoS parameters such as delay or bandwidth. The current congestion control configuration of the transport protocol may not be the optimal for the new path.

- *Application*: Losses and more generally, communication's disruption, can force the application to terminate the current session or even the service.

We identify the first potential issue of the transition to the Mobile Internet.

Potential Issue: Performance of the handover (I.A)

Clearly the performance of the mobile Internet depends, at least, on the performance of the handovers of the Mobile IP family of protocols. This is one of the major issues of the Mobile IP family of protocols and affects Mobile IPv4, Mobile IPv6 and NEMO clients. In fact we believe that this issue may be an obstacle when considering the deployment of such protocols.

Contributions

1. The Part II of this thesis presents a structured methodology to measure and assess the performance of the handover of the Mobile IP family of protocols. Specifically we propose a methodology able to measure the handover latency, the impact on the QoS parameters and on the applications. Additionally the methodology is able to differentiate between the different parts of the handover, that is the link layer handover, the network handover and the handover of the mobility protocol.

2. The second contribution is to apply the designed methodology and evaluate the handover of the Mobile IP family of protocols. In more detail we evaluate the existing experimental implementations of the different Mobile IP-related protocols.

3. The last contribution is to provide a public experimental implementation of the Fast Handovers for Mobile IPv6 protocol [222]. On the time that we carried out the research no public implementation of this protocol was available.

2.3 Near-Future: Deployment phase of Mobile IP

In the previous section we have concluded that the Mobile IP technology is the most cost-effective solution to provide mobility to the Internet. It is expected that this technology is deployed in the mid-term. In this section we analyze the deployment of the Mobile IP family of protocols in order to identify potential issues. In the following subsections we detail the deployment process of Mobile IPv4 and Mobile IPv6.

Deployment of Mobile IPv4

Since the current Internet uses IPv4 the deployment of Mobile IPv4 is quite straightforward. First each sub-network servicing mobile nodes needs a Home Agent, this entity can be placed either as a separate server or as a piece of software into the access router. Second each visiting sub-network, typically wireless hot-spots, should include a Foreign Agent. Again this can be placed into the access router or as a separate server. It is worth to note that Mobile IPv4 can operate without Foreign Agents, in this case the mobile node obtains an IP address from a DHCP server and manages by itself mobility related issues.

Deployment of Mobile IPv6

Deploying Mobile IPv6 is a costly process. Obviously it requires IPv6 and in fact, IPv6's implementations should include mobility support. However existing implementations do not include it, this is because Mobile IPv6 is a much recent standard and its implementations are already in a experimental status. Therefore Mobile IPv6 can be deployed along with IPv6, or separately (i.e. after IPv6 is deployed).

In the first case all the nodes of the Internet are mobile-enabled and can act as mobile nodes or correspondent nodes supporting route optimization. Then the only remaining step is to deploy a Home Agent at each sub-network servicing mobile nodes. Note that foreign agents are not required in Mobile IPv6 because all IPv6 routers announce its presence through special router advertisement messages.

In the second case the Internet is IPv6-enabled but without mobility support. Then the deployment of Mobile IPv6 is a costly process. First Home Agents must be deployed at each sub-network and, in order to provide route optimization, all the end hosts of the Internet must be updated to include mobility support.

Summary

Table 2.4 summarizes the deployment process of the Mobile IP technology:

2.3.1 Deployment Scenarios

As we have seen Mobile IP can be deployed in very different ways, therefore we identify three deployment scenarios (figure 2.6):

Table 2.4: Deployment Cost

Protocol	Infrastructure	End hosts
Mobile IPv4	Home Agents and optionally Foreign Agents	Update mobile nodes
Mobile IPv6 along with IPv6	Home Agents	Nothing
Mobile IPv6 separately	Home Agents	Update all the nodes

- *Gradual deployment*: In this deployment scenario the Internet evolves gradually from the current status: IPv4 to a full-featured Mobile IPv6 Internet. First Mobile IPv4 is deployed, then IPv6, and whenever the Mobile IPv6 implementations are ready the protocol is deployed. Note that in this deployment scenario, during a certain period of time, there are IPv6 nodes without mobility support and thus, without route optimization.
- *Hybrid deployment*: In this case first Mobile IPv4 is deployed and then, IPv6 along with Mobile IPv6 are deployed.
- *Straight deployment*: In this deployment scenario mobility is not deployed until Mobile IPv6's implementations are mature enough, and the Internet becomes Mobile IPv6-eabled directly.

Clearly these deployment scenarios are not strict cases. The Internet is not under the management of a single administrative domain and thus different parts of it may evolve independently. Some parts of the Internet may have different deployment speeds due to different requirements. For instance the lack of IPv4 address affects more some countries such as China or India than others (e.g. North America).

Considering the above-mentioned deployment scenarios, during the transition to the Mobile Internet the network can be at three different states:

Mobile IPv4-only

Both in the Gradual and in the Hybrid deployment the Internet will be, during a certain amount of time, Mobile IPv4-enabled. In this case the major issue is the lack of route optimization of this protocol. This has the following disadvantages [69]:

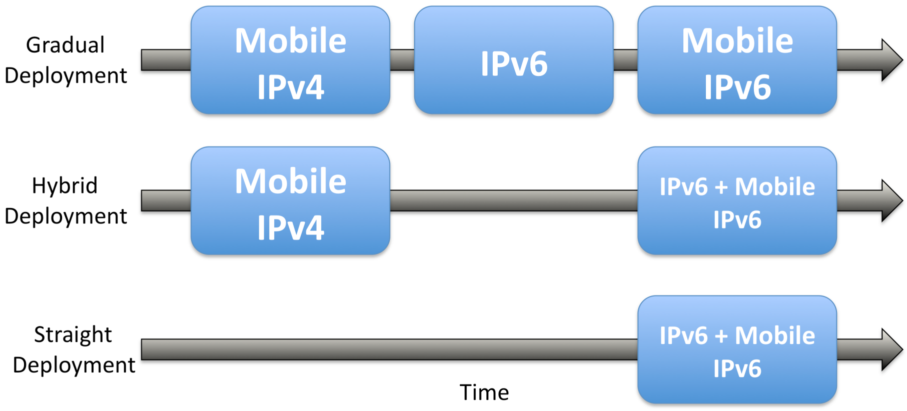


Figure 2.6: Deployment Scenarios

- *Increased delay*: Lack of route optimization involves the selection and utilization of a larger route between the mobile node and its peers, hence it increases the delay. This may in turn impact the overall QoS characteristics such as increased jitter and packet loss.

- *Increased Consumption of Overall Network Resources*: Through the selection of a larger router, the total link utilization for all links used between two end nodes should be much higher than when route optimization is used. This would result in a higher network load with increased congestion.

- *Increased susceptibility to link failure*: If a link along the path is disrupted, all traffic to and from the mobile node will be affected until IP routing recovers from the failure. A non-optimized route utilizes more links and thus, the probability of a loss of connectivity due to a single point of failure at a link may be higher.

- *Bottleneck in the Home Network*: The lack of route optimization forces all the packets to be forwarded by the Home Agents. This may lead them to be the potential bottleneck of a Mobile IPv4-based network.

Therefore the lack of route optimization of Mobile IPv4 is a potential issue for the transition to the Mobile Internet.

IPv6 nodes without mobility support

During the hybrid deployment there are some IPv6 nodes that do not support mobility, hence, in such cases, Mobile IPv6 nodes cannot use route optimized connections. The main concern is that deploying mobility support to these nodes may be a very costly process since the kernel of all these end hosts must be updated. Therefore the high deployment cost of route optimization for Mobile IPv6 in case of an hybrid deployment is another issue of the transition.

Full featured Mobile IPv6 Internet

In all three deployment scenarios the Internet becomes Mobile IPv6-enabled. In this subsection we detail the issues that we have identified by analyzing this scenario. NEMO clients (i.e. mobile routers) also lack of route optimization. As we have seen in the previous section there are no effective mechanisms to authenticate NEMO's Home Prefixes and thus, NEMO clients are forced to forward its data packets through the Home Agent. Again we identify another potential issue: the lack of route optimization of NEMO clients.

Regarding the Mobile IPv6 protocol, the Home Agents are responsible of multiple mobile nodes on a Home Link. This means that the failure of a single Home Agent may then result on the loss of connectivity of numerous MNs. Thus Home Agents represent a single point of failure of Mobile IPv6-based networks. Moreover mobile node's communications through the Home Agent may also lead to either the Home Agent or the Home Link become the bottleneck of the whole system. In addition, the Home Agent's operations such as security check, packet interception and tunnelling might not be as optimized in the Home Agent software as plain packet forwarding.

It is worth noting that the Mobile IPv6 standard allows the deployment of multiple Home Agents on the Home Link to provide reliability and load balancing. This is done so that upon the failure of the serving Home Agent another Home Agent can take over the functions of the failed one. This provides continuous service to the mobile nodes registered with the failed Home Agent. However the transfer of the service is problematic [70]. The solution is node-driven and forces the mobile node to detect the failure and select a new Home Agent. This causes delayed failure detection, service interruption in the upper layers applications, increased workload on the mobile node, message overhead over the air interface and IPsec Security Associations re-establishment. Therefore, considering the above-mentioned limitations we identify another potential issue, Home Agents are single point of failures in Mobile IPv6-based networks.

Finally the Mobile IPv6 Internet should be compatible with the technologies, already deployed. Unfortunately Mobile IPv6's Return Routability procedure, which provides route optimization, is incompatible with some load balancing techniques. This incompatibility prevents Mobile IPv6 nodes from communicating through optimal routes. This is a serious issue since mobile nodes are usually clients and correspondent nodes are usually servers managed by a load balancing mechanism. Therefore we identify another potential issue, that is the incompatibility between Mobile IPv6's return routability and load balancing techniques. This incompatibility is detailed and discussed in chapter 5.2.

2.3.2 Summary, potential issues and contributions

In this subsection we summarize the identified potential issues and the main contributions.

Potential Issue: Lack of Route Optimization (II.A)

This is one of the major issues of the Mobile IP family of protocols and affects Mobile IPv4 and NEMO clients. In fact we believe that this issue may be an obstacle when considering the deployment of Mobile IPv4 or NEMO.

Potential Issue: High Deployment Cost of Mobile IPv6's Route Optimization (II.B)

In case of an hybrid deployment some IPv6 nodes do not include Mobile IPv6 and, if they act as correspondent nodes, they do not support route optimization. Deploying such support may be very expensive since it requires updating the kernel of all these nodes.

Potential Issue: Home Agents as single point of failures (II.C)

The connectivity of Mobile IP clients is highly dependent on the Home Agents. In addition, even with route optimization support, many packets must be forwarded through this special entity. Therefore Home Agents represents a single point of failure in Mobile IP-based networks and may act as bottlenecks. This issue affects the whole Mobile IP family of protocols.

Potential Issue: Incompatibility between Mobile IPv6 and some Load Balancing techniques (II.D)

Mobile IPv6's route optimization is achieved through the Return Routability procedure. As we discuss in chapter 5 this procedure is incompatible with some load balancing techniques. Hence in the Mobile Internet mobile clients cannot use this desirable feature with servers operating behind such devices.

Contributions

Having identified these issues this thesis depicts the following contributions:

- *Lack of Route Optimization*: This thesis explores the solution space of the lack of route optimization problem for Mobile IPv4 and NEMO clients. We present fP2P-HN, a Peer-to-Peer based architecture that allows deploying several HAs throughout the Internet. With this architecture, a mobile node can select a closer HA to its topological position in order to reduce the delay of the paths towards its peers.

- *High deployment cost of Mobile IPv6's Route Optimization*: The thesis provides a solution to IPv6 nodes that do not include Route Optimization support. The solution is intended for servers since mobile nodes are clients, and usually clients communicate with servers. We present the Mobility Agents, a novel entity placed at the Correspondent Networks that manages mobility related issues on behalf the CNs. The main advantages of our Mobility Agents is that it does not require modifying the servers and hence, reduces the deployment cost.

- *Incompatibility between Mobile IPv6's route optimization and Load Balancing techniques*: As it has been stated previously, and discussed in further chapters route optimization is not compatible with load balancing techniques. This thesis extends the functionalities of the Mobility Agents to incorporate a load balancing mechanism compatible with the Return Routability procedure. This allows mobile clients to communicate directly with servers operating behind our Mobility Agent.

- *Home Agent as single point of failures*: This thesis explores new Home Agent architectures and provides a new one that increases reliability and load balancing. We present the flexible Home Agent architecture; our solution only requires a set of HAs for the whole network. Our basic idea is that the mobile node's location can be announced to exit routers, this way re-directing packets can be done without involving the Home Agent.

2.4 Future: New Architectures

In this section we analyze the potential issues that may appear after the deployment phase of the Mobile Internet. In such scenario the Mobile Internet is Mobile IPv6-enabled and potentially all the nodes can change its point of attachment. In addition route optimization is widely supported and users benefit from these functionalities. At the same time wireless technologies are widely deployed and it is reasonable to think that in the future this growth will continue. In fact the cost of a wireless interface has constantly decreased and it is expected that this trend will hold. This enables the Mobile Internet to equip terminals with more than one wireless interface and use them simultaneously. For instance an unreliable high-bandwidth access network might be used for file transfer while a reliable low-bandwidth radio access might be used for voice calls. This technique is known as multihoming.

Multihoming [71] is a well-known technique whose main objective is to increase reliability of the Internet connection of networks or single nodes. This technique uses multiple interfaces connected to different ISPs. This way the multihomed node has different paths to communicate with its peers. In addition, Multihoming may provide load balancing, load sharing and overall performance improvement. Many research papers [72; 73; 74] have been published exploiting the benefits of multihoming in static nodes or networks, however multihoming in mobile nodes or networks is a relatively recent research topic.

Multihomed mobile nodes present both a challenge and an opportunity for the Mobile Internet. On the downside managing such terminals is a complex task. However terminals equipped with multiple interfaces can greatly improve the performance and the functionalities of the Mobile Internet. The benefits of these multihomed mobile architectures have been analyzed by the MONAMI6 WG [75] and are the following:

1. *Permanent and Ubiquitous Access*: These architectures can provide an extended area of coverage via distinct access technologies.

2. *Reliability*: These terminals are able to react upon a failure of an access network or an end-to-end path. Connectivity can be guaranteed as long as at least one connection is maintained. In addition "Bi-Casting" may be supported, with this mechanisms a flow can be sent duplicated through two different interfaces in order to reduce latency and increase reliability.

3. *Flow Redirection*: Re-allocate flows from one path to another can increase the overall performance.

4. *Load Sharing*: The load can be distributed across a set of interfaces.

5. *Load Balancing/Flow Distribution*: To assign a set of flows between multiple interfaces (simultaneously active or not) of a node. Usually the less loaded interface is selected.

6. *Aggregate Bandwidth*: Multiple interfaces can provide more bandwidth.

7. *Preference Settings*: To enable the user, the applications or the ISP to choose the preferred interface based on a policies, cost, flow requirements, etc

In the following section we discuss how these interfaces can be accommodated in the Mobile Internet's architecture and which potential issues arise.

2.4.1 A Generic Multihomed Mobile Architecture

The generic architecture that we present in this section aims to provide multihoming to Mobile IPv6 nodes. Since this is a complex task we focus this architecture in multihomed mobile routers. Mobile nodes are autonomous devices with limited CPU and energy constraints while mobile routers are usually installed in vehicles and may

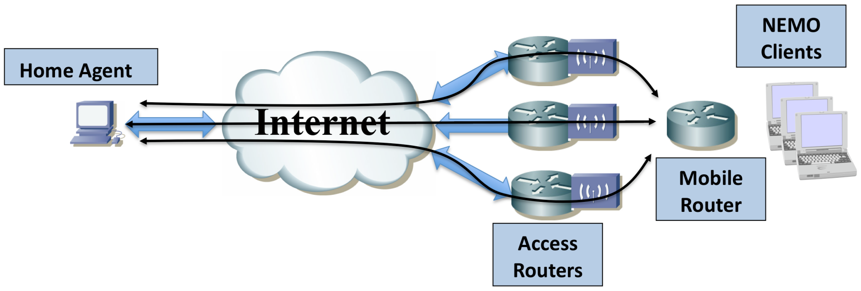


Figure 2.7: NEMO Basic Support

have a more capable CPU and can be connected to an external power source. That is why we focus our architecture in multihomed mobile routers, nevertheless it is directly applicable to mobile nodes.

Figure 2.7 shows the applicability of this architecture. In this scenario presented in the figure the multihomed mobile router is under the area of coverage of several wireless links and can use a set of them to transmit and receive datagrams. It is worth noting that the characteristics of these wireless access networks may change abruptly and that the mobile router must deal with this erratic behaviour efficiently.

The generic architecture is engineered by reviewing the existing literature [76; 77; 78; 79] in this research topic. The architecture that we present is generic enough to accommodate the existing proposals. In fact the main goal of the architecture is to be a general framework to accommodate any multihomed mobile node/router design or implementation.

Signalling

The first requirement to support multihoming it is to extend the Mobile IPv6 protocol to allow bindings of multiple Care-of Addresses simultaneously. The IETF's MEXT WG is standardizing such features through the documents [80; 81]. The first document describes how a mobile node/router can register multiple Care-of Addresses for a single Home Address and create multiple binding cache entries. This can also be applied to bindings to Correspondent Nodes.

The second document extends Mobile IPv6 to support bindings a particular flow to a Care-of-Address without affecting other flows using the same Home Address. Flow bindings are useful to direct certain flows to certain addresses. This may be done because some flows are better suited to certain link layers or simply to load balance flows between different interfaces. These specifications introduce the flow identifier option, which is included in the binding update message and used to distribute policies to the recipient of the binding update.

Architecture

Figure 2.8 depicts the internal details of the generic architecture. As the figure shows the clients send their flows towards the mobile router[1]. In turn the mobile router must decide how to assign these flows to the different available paths. This can be done either statically or dynamically. Clearly a dynamic assignment can potentially increase the overall performance. In our architecture the *Scheduler* is responsible of such tasks. In order to enable this module to assign flows efficiently two basic requirements arise: the *Scheduler* must have information about the characteristics of the paths and the requirements of the flows. Hence the generic architecture must include a *Measurements* module able to monitor the characteristics of the available paths and a *Flow Requirements* module that contains information about flows. Recently the *Multi-Path Concurrent Transfer* area of research is exploring mechanisms to split a flow and send it simultaneously through multiple paths. The main objectives of such mechanisms is to increase the overall available bandwidth and hence, the throughput. The generic architecture includes an optional *Split* module that implements one of these mechanisms.

Measurement Module

The measurement module is responsible of monitoring the characteristics of the available paths and informing the *Scheduler*. This can be done at two different layers: the link layer (i.e. the wireless access network) and at the network layer (i.e. the end-to-end path). Specifically this module can monitor the following metrics related with the wireless access network.

- *Capacity*: The capacity of a wireless access network represents the maximum theoretical available bandwidth of the link.

[1] If this architecture is applied to a mobile node then clients can be seen as applications.

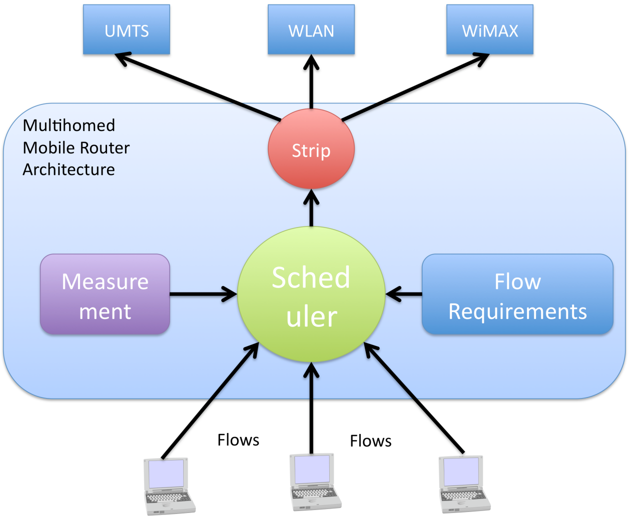


Figure 2.8: Multihomed Mobile Router Architecture

- *Connectivity*: The mobile router must be aware of which interfaces are connected and which not in order to provide seamless transitions.

- *Losses*: Packet losses at the link layer must be monitored in order to provide good congestion control.

At the network layer the measurement module can monitor the following metrics:

- *Available bandwidth*: The available bandwidth is the remaining capacity of and end-to-end path and it is directly related with the maximum theoretical throughput that a flow can achieve. This is a key parameter when considering efficient *Scheduling* or *Multi-Path Concurrent Transfer*.

- *Losses*: The *Measurements* module should monitor network and link layer losses to differentiate congestion from transmission errors. This information can be used by the transport-layer to efficiently implement a congestion control mechanism.

- *Delay/Jitter*: These metrics provide important information for delay-sensitive flows.

- *Path Failure*: This module should monitor the path in order to detect path-failures and react re-allocating flows. Since it is also monitoring the link layer it can differentiate between path and link failures.

Flow Requirements

This module is responsible of providing updated information about the flows that the *Scheduler* has to allocate. The *Scheduler* must assign flows to paths efficiently. This means that it should know their specific requirements. For instance if a flow belongs to a delay-sensitive application the *Scheduler* should assign this flow to a low-latency path.

Strip

The *Strip* module implements a *Multi-Path Concurrent Transfer* mechanism and it is able to split a flow and send it (or receive it) through different interfaces to increase the overall bandwidth. Only flows with hard bandwidth/rate requirements should benefit from this technique.

Finally it is worth noting here that the different modules conforming the architecture can be deployed at the mobile router/node or at the Home Agent. This is especially useful when considering NEMO clients since all the flows must be sent thought the Home Agent. For instance the *Measurements* module could be deployed at the Home Agent and this entity would be responsible of sending probe packets to monitor the characteristics of the end-to-end paths.

2.4.2 State-of-the-Art

In this section we detail the state-of-the-art of the different techniques and mechanisms required for implementing the modules that compose the generic multihomed mobile router architecture.

First the flow-requirements module can use well-known techniques and mechanisms [82; 83; 84; 85] developed by the QoS research community. In QoS-enabled networks flows must also be assigned to different classes of service in order to receive the appropriate treatment by the QoS enforcing mechanism. As an example of a solution designed by this research community that can be applied to the *Flow Requirements* module consider the QoS API [85]. This API extends the socket API so that applications can specify their flow requirements.

Regarding the *Strip* module it uses *Multi-Path Concurrent Transfer* mechanisms. As we have mentioned earlier this is an active area of research and many solutions have been published [86; 87; 88; 89; 90; 91]. Researchers are mainly following two approaches. First they are extending existing transport protocols such as TCP [87; 88] and SCTP [90] to support multiple interfaces simultaneously. Secondly they are proposing new transport layer protocols [89; 90; 91] with native *Multi-Path Concurrent Transfer* support.

Finally the measurement module can benefit from the active and passive measurements area of research. Researchers have been proposing methodologies to estimate and collect network related metrics during the last decades. In fact it is reasonable to consider that estimating QoS metrics end-to-end is an already solved issue. For instance the *Measurements* module can rely on In-Line Measurements [92]. This technique defines IPv6 extension headers that can be dynamically included and removed to existing data packets and that carry a timestamp. By using this information existing traffic can be used to measure QoS network-layer metrics. Regarding Link layer metrics such as the Capacity or the Losses they can be simply provided by the driver of the wireless interface. Finally the available bandwidth is a key metric for assigning flows efficiently and especially when we consider *Multi-Path Concurrent Transfer* and *Scheduling* mechanisms (see [86; 87; 88; 89; 90; 91] for further details).

The estimation of the available bandwidth is also an active area of research. Researchers have been proposing and designing tools and mechanisms to estimate this metric during the last years. Most of the proposed tools designed to estimate the AB fall into two categories: the Probe Rate Model (PRM) [95] and the Probe Gap Model (PGM) [98]. The first model uses packet trains and it is based on the concept of self-induced congestion. Informally, if one sends a packet train at a rate lower than the AB along the path, then the arrival rate of the packet train at the receiver will match the rate at the sender. However if the sending rate is greater or equal than the AB then the packet train will congest the queues along the path and the receiving rate will be lower than the sending rate. Tools such as Delphy [93], TOPP [94], PathLoad [95], IGI/PTR [96], pathChirp [97], BART [99] and Forecaster [100] use this model. The second model (PGM) uses packet pairs and bases its estimation on the differences of input and output time gaps of the packet pairs [98].

Both the PRM and the PGM models are designed for FIFO-based wired networks and may not operate accurately in wireless networks. Usually wireless networks, such as IEEE 802.11, are random access networks that use Fair Queuing mechanisms. In fact it has been shown empirically [102] and theoretically [103] that the above-mentioned tools are not suited to work in scenarios where wireless networks are present. As the authors [102] explain, this is manly due to three factors:

- Random Access Links do not have a fixed or well-defined raw bandwidth (capacity). For instance dynamic multi-rate schemes in 802.11 change the transmission rate of the stations.
- The scheduling in such networks may not be FIFO because of a fully distributed contention-based MAC as in 802.11.
- Multi-rate 802.11 links interfere to create highly bursty cross-traffic patterns that may result in significant deviation from the assumed fluid model of cross-traffic.

At the best of the author's knowledge there are very few tools proposed to estimate the available bandwidth in the presence of random access networks (i.e. IEEE 802.11). First ProbeGap [102] is intended for single-hop wireless links and fails when applied to multi-hop paths where multiple wired links are present. A reason for this is that the technique used to infer congestion does not take into account the interference of the wired cross-traffic. A case when the most congested link is a wired one would not be identified correctly. Also ProbeGap requires the a priori knowledge of the transmission rate of the bottleneck link (assumed in the wireless hop) to infer the available bandwidth. This rate might not be always available and varies over time in IEEE 802.11 links. Finally as we will show empirically in further chapters ProbeGap is not accurate when multiple contending stations are present or when cross-traffic flows use different packet sizes.

Second Dietopp [105] is a simple modification of the TOPP algorithm whose main goal is to measure the available bandwidth in single-hop IEEE 802.11 links. Again Dietopp has low accuracy when evaluated in links with multiple contending nodes or variable packet sizes.

Given the above-mentioned considerations we identify a potential issue. There is a lack of fundamentals, methodologies and tools to estimate the available bandwidth in wireless links. This can prevent the measurement module to estimate the available bandwidth of the paths and hence impact the effectiveness of the *Multi-Path Concurrent*

Transfer and the *Scheduler* mechanism. Potentially this can also significantly impact the overall performance of multihomed mobile router architectures.

2.4.3 Summary, Potential Issues and Contributions

In this section we have analyzed the potential issues that may appear in the post-deployment phase of the Mobile Internet. As we have seen wireless technologies are constantly evolving and the cost of the interfaces is constantly decreasing. Therefore it is reasonable to think that terminals can be equipped with multiple wireless interfaces. This can increase the reliability, provide more aggregate bandwidth and, in general, increase the overall performance of the communications. In the downside managing multiple paths may be a complex task. By following this assumption we have presented a generic architecture to deal with multiple interfaces (and the multiple paths that they provide).

By analyzing the different modules that form the generic architecture we conclude that, in most of the cases, the issues that they arise are either solved or under active research. However the architecture depends on the estimation of the available bandwidth and, at the best of the authors' knowledge, very few tools have been proposed to estimate this metric in the presence of random access networks (i.e. 802.11). In fact the fundamentals of available bandwidth estimation in random access networks have not been explored. Research in available bandwidth estimation has mainly focused in wired and FIFO-based networks.

Potential Issue Lack of fundamentals in available bandwidth estimation in random access networks (III.A)

Lack of tools, methodologies and fundamentals to estimate the available bandwidth in random access wireless networks can limit the performance of multihomed mobile router/nodes architectures. This metric is of great importance for *Multi-Path Concurrent Transfer* and, in general, to allocate flows to paths efficiently.

Contributions

This thesis contributes to this issue with the following contributions:

1. Review the implications of wireless link for the estimation of bandwidth related metrics. Our finding shows that, under fluid assumptions classical tools and models, which are based on periodic probing[1], target the achievable throughput rather than the available bandwidth. This is a different metric and related with the fair-share of the link. Further we extend our analysis relaxing the fluid assumption and we show that periodic probing processes suffer a bias in its measurement process. Additionally we provide a simple mechanism to reduce the effect of this bias.

2. Explore poisson probing processes to design a novel methodology able to estimate the available bandwidth in wireless links. Further we present W-Path, an heuristic-based tool able to estimate this metric with very low convergence time.

3. Explore the applications of poisson probing for wired scenarios. Existing methodologies exploit periodic probing process and, although they are very accurate they are also very intrusive and may potentially impact the performance of the path under measurement. Our study shows that poisson methodologies can also benefit these scenarios proving accurate tools with low intrusiveness. Specifically we design AKBest, an active tool able to estimate the available bandwidth of a wired end-to-end path by sending packet trains at a rate lower than the available bandwidth. Even more, we present PKBest, the first passive available bandwidth estimation that exploit existing traffic between to nodes to produce accurate estimations. Clearly the main advantage of PKBest is that it does not require injecting probe traffic into the network.

2.5 Summary and Conclusions

In this chapter we have provided a detailed analysis of the transition of the Mobile Internet in order to identify potential issues. The analysis has focused on three stages of the transition: Present, Near-Future and Future. As a summary the main conclusions of this study are:

- *Present*: At present, our analysis has focused on the main advantages and drawbacks of deploying mobility at different layers. We have shown that the most

[1] Both the PRM and the PGM are based on such probing process

cost-effective solution is Mobile IP, that is, deploying mobility at the network layer. The main reason for that is because IPv4 has urgent issues that must be addressed in a short period of time, therefore deploying IPv6 (that incorporates mobility) is the most cost-effective solution. By analyzing the Mobile IP family of protocols we conclude that its major issue is the performance of the handover.

- *Near-Future*: It is expected that in the mid-term Mobile IP will be deployed. We have analyzed this process and identified several potential issues. First there is a lack of route optimization for Mobile IPv4 and NEMO clients, this reduces the performance of the Mobile Internet. In addition the Home Agent represents a single-point-of-failure in such networks and may become the bottleneck of the whole system. Lastly an incompatibility exists between Mobile IPv6's Route Optimization and certain load balancing techniques.

- *Future*: In the future, once Mobile IP has been deployed new architectures can be deployed to enhance the performance of the Mobile Internet. In particular mobile nodes can be equipped with multiple interfaces. This increases the overall bandwidth, reliability and provides permanent and ubiquitous access. Assigning different flows to the different available paths provided by the wireless interfaces is a complex task and, as we have shown, estimating the available bandwidth of such paths is one of the basic requirements. Unfortunately there is very few research of the estimation of this metric in wireless links and hence, we consider the lack of tools to estimate the available bandwidth in such links as a potential issue of the post-deployment phase of mobility.

The remaining of the thesis presents solutions for the potential issues identified in this chapter.

Part II

Present: Analysis of the Handover Process

3

The Handover Process

3.1 Introduction

In this chapter we analyze the handover process of the Mobile IP family of protocols. The reader can find a list of abbreviations in the Glossary. The handover is the sequence of actions that a MN (or the network) has to perform to regain connectivity after a change on the point of attachment, either at the link or at the network layer. Therefore some handovers only affect the link layer, in these cases the MN has only changed its wireless Base Station and the network and the mobility layers are unaffected. In a full handover the MN changes its default access router. In this case the wireless interface must re-associate to a new Base Station, the IP layer must obtain a new address (the Care-of Address) and the mobility protocol must inform to its HA (and optionally its CNs) about its new location.

In the following sections we detail the different parts of the handover process, from the link layer, assuming IEEE 802.11 [104], to the mobility protocol. In order to analyze the handover we provide a simple analytical model of the handover latency. This metric is the total duration of the handover, that is the amount of time that the MN is unable to send or receive data packets. This metric is able to capture the performance of a handover since it is directly related with the disruption suffered by the MN.

3.2 Analysis of the Handover Latency

3.2.1 IEEE 802.11

The Wireless LAN protocol (IEEE 802.11) [104] is based on a cellular architecture, where each cell is managed by a Base Station (BS, commonly known as Access Point or AP). Such a cell with the BS and the MNs is called a Basic Service Set (BSS) and can be connected via a backbone (called Distribution System or DS) to other cells, forming an Extended Service Set (ESS). All these elements together are one single link layer entity from the upper OSI layers' point of view. APs announce their presence using periodic Beacon Frames containing synchronization information. If a MN desires to join a cell, it can use passive scanning, where it waits to receive a Beacon Frame or active scanning, where it sends Probe Request frames and receives a Probe Response frame from all available APs. Scanning is followed by the Authentication Process, and if that is successful, the Association Process. Only after this phase the MN is capable of sending and receiving data frames. MNs are capable of roaming, i.e. moving from one cell to another without losing connectivity but the standard does not define how it should be performed, it only provides the basic tools for that: active/passive scanning, re-authentication and re-association.

Figure 3.1 presents a schema of the messages exchanged between the AP and the MN in a wireless handover. The AP sends periodically Beacon messages. The MN listens to those messages and sends a Probe Request message to ensure that the AP is reachable. Then, after the Authentication Phase and the Association Phase, the MN regains connectivity.

3.2.2 Mobile IPv4

The handover latency of Mobile IPv4 has several components. When a MN changes from one access router to another first it has to associate to a new AP, which involves Scanning, Authentication and Association (D_{L2}), this is the wireless part of the handover. Then the network part of the handover starts. First IP must obtain a new Care-of Address. Foreign Agents advertise its presence through periodic Agent Advertisement messages. These messages include a pool of addresses. Whenever a MN receives such message it checks if it was sent from its actual access router or from a new one. If it was sent from a new one it configures a new IP address. If no Foreign Agents

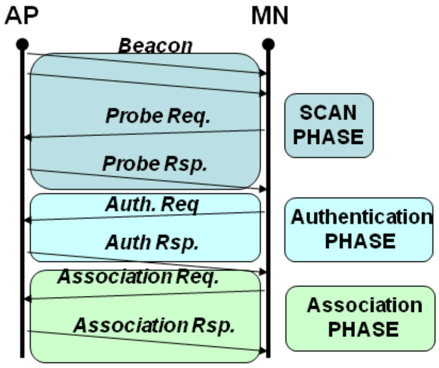


Figure 3.1: IEEE 802.11 handover schema

are present MNs can use other procedures such as DHCP. We define the delay of this operation as (D_{IP}). Then Mobile IPv4 must register its new address with its Home Agent through a Binding Update message and wait its acknowledgment counterpart. Equation 3.1 models the handover latency for this protocol[1]. Note that $RTT(X, Y)$ accounts for the average Round-Trip time between the two end-hosts (X and Y).

$$HL_{MIPv4} = D_{L2} + D_{IP} + RTT(MN, HA) \tag{3.1}$$

3.2.3 Mobile IPv6

The Mobile IPv6 handover is similar to its IPv4 counterpart. As before, first the MN has to Scan, Authenticate and Associate to the new AP. Then, in the Agent Discovery phase the MN needs to check if the old access router is unreachable using the Neighbor Unreachability Detection (D_{NUD}) [107] algorithm and obtain a new temporal IP address (Care-of-Address). It also needs to ensure that the recently obtained CoA is unique on the new link using the Duplicate Address Detection algorithm (D_{DAD}) [108]. Both operations are required for IPv6 reconfiguration. Finally, the MN must send a Binding Update message to its HA. Optionally if the MN is using Route Optimization with its CNs it has to send two signalling messages (see section 5.2 for further details)

[1] Note that some implementations may start sending packets right after the MN has sent the Binding Update message.

to each of them, wait for an acknowledgment and finally send a Binding Update. We do not consider this process as part of the handover latency.

$$HL_{MIPv6} = D_{L2} + D_{NUD} + D_{DAD} + RTT(MN, HA) \tag{3.2}$$

As we can see from equation 3.2 the handover depends on the constants D_{L2}, D_{NUD} and D_{DAD}. The last terms depends exclusively on the Round Trip Time (i.e. distance) between the MN and the HA.

3.2.4 Fast Handovers for Mobile IPv6

FMIPv6 is a Mobile IPv6 extension that reduces the handover latency and, during the handover it buffers packets delaying them instead of losing them. This is accomplished by allowing the MN to send packets as soon as it detects a new subnet link (IEEE 802.11 in our case) and delivering packets to the MN as soon as its attachment is detected by the new access router.

FMIPv6 has two different operational procedures. In the Predictive Handover the MN discovers nearby APs using the IEEE 802.11 Scan and then requests all the important information related to the corresponding new access router. When attachment to an AP takes place, the MN knows the corresponding new router's coordinates including its prefix, IP address and MAC address. Through special Fast Binding Update and Fast Binding Acknowledgment messages the MN is able to formulate a prospective new CoA (without changing its point of attachment). This CoA must be accepted by the new access router prior to the MN's movement. Once the MN has changed its point of attachment and it is connected to the new access router link, it can use its new CoA without having to discover the subnet prefix, it also knows the new access router MAC and IP address and hence, this latency is eliminated. As soon as it is attached, the MN sends a Fast Neighbor Advertisement message announcing its presence. Moreover, the previous access router will tunnel and forward packets to the new CoA until the MN sends a Binding Update registering its new CoA to the HA and to the CNs hence, no packet is lost. The other FMIPv6 operational procedure is the Reactive Handover which is very similar to the previous one. We only provide an analytical model for the first mode of operation, it is worth noting here that the second mode has a similar handover latency.

FMIPv6 reduces the handover latency roughly to the link layer handover latency. Once the MN has received the Fast Binding Acknowledgement message the communications between the MN and the CN are interrupted. Packets destined to the MN are buffered at the new access router (and thus delayed) until the MN regains connectivity. When the link layer handover is finished, and the MN is connected on the new link it announces its presence through a Fast Neighbour Advertisement (FNA) message and the new access router forwards the buffered packets. We define D_{FNA} as the time between the MN regains connectivity and the FNA message has been received by the new access router. Equation 3.3 models the handover latency for the FMIPv6 protocol operating in Predictive mode, which is the fastest one.

$$HL_{FMIPv6} = D_{L2} + D_{FNA} \tag{3.3}$$

From eq. 3.3 we can see that the handover latency for FMIPv6 when using the Predictive mode is very similar to the link layer handover because D_{FNA} is usually a small amount of time. It is worth noting that during this time packets are being buffered at the new access router and the maximum extra delay suffered by those packets is, in fact, the handover latency. This may impact real-time applications.

3.2.5 Hierarchical Mobile IPv6

HMIPv6 is another extension of Mobile IPv6. Its main goal is to reduce the signalling overhead and to improve the handover latency. HMIPv6 introduces a new entity called Mobile Anchor Point (MAP) placed at any point of the hierarchy of a network. The MAP is essentially a Home Agent. The MAP limits the amount of Mobile IPv6 signalling outside the local domain.

When a MN using HMIPv6 enters a MAP domain it receives Router Advertisements containing auto-configuration information and the IP address of the local MAP. The MN configures two Care-of-Addresses. The first one is called the regional Care-of-Address (RCoA) and belongs to the MAP's subnet. It is auto-configured by the MN when it receives the MAP option. The second one is called the on-link Care-of-Address (oCoA) and it is a regular CoA [14]. The MN can bind its oCoA with an address on the MAP's subnet (RCoA). The MAP acts as a local HA, it receives all packets destined to the MN and it forwards them to the MN's oCoA using a tunnel. If the MN changes its point of attachment within the MAP boundaries only the Regional CoA (RCoA) needs to be registered with CNs and the HA. The RCoA does not change until the MN moves outside the MAP domain. This makes the MN's mobility transparent to the CNs it is communicating with.

The HMIPv6 handover latency is very similar to the Mobile IPv6 handover latency. If an HMIPv6-aware MN moves from one access router to another one outside the MAP boundaries it has the same handover latency than Mobile IPv6. However, when the MN moves to another AR which is inside the MAP domain it does not require to send Binding Updates. The MN just sends one Binding Update to the MAP, the CNs have the RCoA registered and this address does not change. Equation 3.4 models this handover latency.

$$HL_{HMIPv6} = D_{L2} + D_{NUD} + D_{DAD} + RTT(MN, MAP) \tag{3.4}$$

The handover latency for HMIPv6 is similar to the MIPv6 handover latency. The only difference is during the Registration phase: in MIPv6 the Binding Update messages must be sent to the HA and CNs while in HMIPv6 those messages must be sent to the MAP, which is located in the same domain than the MN (i.e. with high probability $RTT(MN, MAP) < RTT(MN, HA)$.

3.2.6 NEMO

In NEMO the MR uses exactly the same signalling than Mobile IPv6. The only difference is that NEMO does not support Route Optimization and thus, only a Binding Update to the HA is required. The following equation models the handover latency of NEMO.

$$HL_{NEMO} = D_{L2} + D_{DAD} + D_{NUD} + RTT(MR, HA) \tag{3.5}$$

3.3 Summary and Conclusions

In this chapter we have detailed the handover process of the main protcols of the Mobile IP family of technology. Additionally we have provided a simple analytical model for the handover latency. By comparing the terms that appear in all the equations we conclude:

- All the handovers latencies depend linearly on the distance between the MN and its HA. Optionally if Route Optimization is used also depends on the distance to its CNs.
- Reconfiguring IPv6 requires more actions than reconfiguring IPv4. In IPv6 the MN must execute two algorithms (DAD and NUD) while in IPv4 the MN must only wait for an Agent Advertisement Message.

In the following chapter we provide empirical values for the handover latency of Mobile IPv4, Mobile IPv6 and Fast Handovers for Mobile IPv6. Additionally we analyze the impact of the handover in the QoS perceived by the MN.

4

Measurement-based Analysis of the Handover

4.1 Introduction

This chapter presents an empirical evaluation of the handover of the Mobile IP family of protocols. We aim to study these handovers in terms of performance and impact on the QoS perceived by the MN. As we have seen in the first part of this thesis the handover is one of the main issues when considering the performance of these protocols, and in general, of the Mobile Internet.

We evaluate the performance of the handovers by measuring its handover latency, and differentiating the different parts of such latency. In order to achieve these goals we rely on passive measurements, by intercepting signalling-related messages we can measure the duration of the link-layer, network layer and mobile layer parts of the handover.

Regarding the impact on the QoS perceived by the MN we rely on the metrics designed by the IPPM (IP Performance Metrics) WG [117]. This WG has defined the following metrics to assess the QoS. More specifically we rely on the OWD (One-Way-Delay also known as delay) [118], Inter-Packet Delay Variation (jitter) [119] and Packet Loss Ratio [120]. Delay and jitter are measured before and after the handover while losses are monitored during the handover. By using this information we can identify bottlenecks and help application/transport protocols developers.

To achieve these goals we present a methodology able to account for all these metrics. The methodology uses a mix of active and passive measurements and it is applicable to any network-mobility protocol. Active measurements are used to measure QoS metrics while passive ones for the handover latency. Further we apply this methodology in a testbed to provide statistically representative numerical results. The testbed uses IEEE 802.11 wireless interfaces and is configured with Mobile IPv4, Mobile IPv6 and Fast Handovers for Mobile IPv6. Since the performance of NEMO and Hierarchical Mobile IPv6 is similar to that of Mobile IPv6 (see chapter 3 for further details) we disregard these protocols.

In the testbed we used experimental implementations of Mobile IPv6 [111] and Mobile IPv4 [29], however no public implementation of Fast Handovers for Mobile IPv6 is available. That is why we develop our own implementation of Fast Handovers of Mobile IPv6. This implementation helps providing an empirical evaluation of its performance. The implementation was developed on 2005 and, on that time, FMIPv6 was in a draft status. IETF requires experimental implementations of the protocols in order to rise them to RFCs (the implementation can be found at [222]).

This chapter is divided as follows, first we present our testbed and the methodology. Then we provide results from applying the methodology to the tesbed. Finally we provide some details about the FMIPv6 public implementation.

4.2 Measurement's Methodology

This section presents the measurement scenario and the methodology used to measure the handover-related metrics. First we present the testbed that we use to perform our experiments, second we detail our methodology.

4.2.1 Testbed

The testbed's main goal is to study the handover using active and passive measurements. The testbed can be seen in detail in Figure 4.1.

To avoid external interferences, this testbed is isolated from outside networks, all the input and output traffic on the testbed's network interfaces is controlled. Having this isolation, but without the lack of external access, the scenario has two parallel networks, the control network and the actual testing network, as highlighted on the

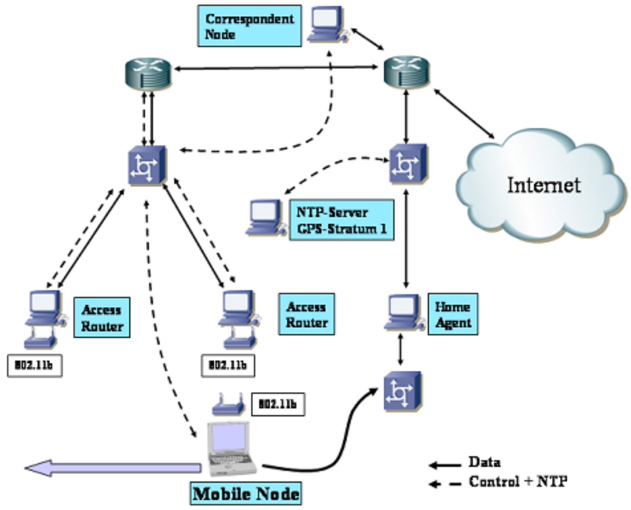


Figure 4.1: Testbed configuration

figure. This is important because with uncontrolled sources of traffic all the delays will be miscalculated. This testbed gives the tests all the privacy needed, this way, once the tests are ready, no foreign agents are able to interfere with them. At the same time, the path followed by the packets is long enough to consider the possible clock skew too small to have any negative impact on the results.

Regarding synchronization the testbed is configured to use four NTP (Network Time Protocol) sources [109], two of them belonging to a private network, Stratum 1 servers connected to a GPS source each. The other two sources are on the outside network and are as far as 3 hops away from the testbed. The NTP statistics show that, with this setup, we obtain less than 1ms of measurement accuracy. In order to mitigate the harmful effects of the jitter on the NTP algorithm running in our tetsbed, we use the Pulse-per-Second (PPS) Clock Discipline driver with a PPSAPI interface, which is a proposed IETF Standard [110].

In order to confirm our synchronization accuracy, we made a simple test; we sent several ARP broadcast packets in our measurement network, those packets were captured on all the machines involved in our tests, and the timestamps were compared. The maximum difference among those timestamps agreed with the threshold stated by NTP.

4.2.2 Hardware

All the machines involved on the tests are using the GNU/Linux Debian Sid distribution. Depending on the role of each computer, the hardware and the kernel varies accordingly:

- Access Points/Routers (AP): This testbed has two access points, each one with two wireless cards, one for communicating with the MN and the other one to monitor (capture frames). Those cards have the Atheros Chipset (802.11g) in 802.11b compatibility mode. The configured kernel is the 2.4.26.
- Mobile Node: The MN uses a Cisco Aironet 350 card (802.11b) for wireless connectivity, here the kernel is 2.4.26.
- Home Agent/Correspondent Node: The last two important hosts on the scenario have similar configuration with the 2.4.26 kernel

4.2.3 Software

The software tools that we use are:

1. MGen/DRec, NetMeter [113]: for the active measurement part.
2. Ethereal [114] and PHM Tool (Passive Handover Measurement) [115]: for passive measurements.

Both applications depicted here: NetMeter's and the PHM Tool are developed under the same code base. Their main goal is to analyze the Ethereal les and obtain for PHM Tool the handover latencies and for the NetMeter's part the packet losses and delays at application level.

The same capture is used for both solutions, the monitoring infrastructure is set up on the Access Points, given that is the only way of detecting all the handover latencies. Both captures (each on one access point) are merged (as they really represent the same traffc ow) and the data is analyzed.

Regarding the mobility protocols we use the MIPL 1.1 Mobile IPv6 [111] and the Dynamics HUT Mobile IPv4 [29] implementation. We have developed a Fast Handovers implementation written for the MIPL 1.1 protocol that complies with the draft-ietf-mipshop-fast-mipv6-03.txt and that supports any wireless card (with Linux Support) through the Wireless Toolkit for Linux [112]. The reader can find more details about the Fast Handovers implementation in the last section of this chapter.

4.2.4 Methodology

In this section we detail the methodology used to measure the handover latency and the QoS parameters.

Handover Latency

To compute the handover latency of the different mobility protocols we use passive measurements. All the mobility protocols have different handover parts as explained before. First goes the wireless layer, then the network layer (either IPv4 or IPv6) and finally the mobility protocol. All those protocols send their corresponding signaling messages to perform the handover. We capture all these signaling messages (sent or received by the MN) using a monitoring wireless card. Using the developed application PHM tool we compute the different parts of the handover latency of the different protocols offline. PHM tool compares the timestamps of the signaling messages providing numerical results of the handover latency.

During a handover, first the IEEE 802.11 card detects that the signal quality received by the current AP is becoming poor and scans for nearby access points sending Probe Request messages. This IEEE 802.11 signaling message denotes the handover start. After the wireless card has found a new access point it must authenticate and associate to it, the last message sent by the AP to the MN is the (Re)Association Reply that points the end of this part of the handover latency.

With Mobile IPv6, the MN must obtain a new CoA. IPv6 routers send periodically Router Advertisement messages which include autoconfiguration information. First the IPv6 layer must check, using the Neighbor Unreachability Detection algorithm (NUD) [107] that its previous access router is no longer reachable, then it will listen for Router Advertisement messages and it will configure a new CoA. Finally, using the Duplicate Address Detection algorithm (DAD) [108] it will ensure that its new CoA is unique on that link and then it will be ready to send and receive IP packets.

Computing the Mobile IPv6 handover latency is a straightforward problem. Mobile IPv6 sends a Binding Update to HAs and CNs indicating its new CoA (its new location) and receives a Binding Acknowledgement as response. PHM Tool computes the Mobile IPv6 handover latency starting at the Binding Update message to the HA and ending with the Binding Acknowledgment received by the CN.

The Mobile IPv4 implementation has been tuned to speed up the handover as much as possible. The implementation is constantly polling the wireless card and as soon as it detects that the AP has changed it sends a Router Solicitation triggering an Agent Advertisement. This message has all the related information to configure a new CoA. The MIPv4 implementation is also configured to accept this new Agent Advertisement without waiting until the last one expires (from the old access router). Next, the MN registers its new CoA sending a Registration Request to its HA. The PHM tool computes the MIPv4 handover latency from the Agent Advertisement to the Registration Reply.

FMIPv6 enhances the Mobile IPv6 handover reducing the IPv6 handover latency. As explained before, when FMIPv6 is used, the MN prepares the handover to the new access router when it is still connected to the old one. The whole IEEE 802.11/IPv6/FMIPv6 handover latency is reduced to the IEEE 802.11 handover latency. However we must check that our implementation works as expected and PHM tool computes the handover latency from the end of the IEEE 802.11 handover until the Fast Neighbour Advertisement message.

QoS metrics

We use active measurements to compute the packet losses, delay and jitter. The basis of active measurements is to generate a synthetic flow traveling through the network under test. NetMeter along with MGEN are the application used for such tests.

To measure the packet losses of the different mobility protocols we send the active flow (either from the CN or from the MN) and force a handover. The number of packets received is computed and the difference between the packets sent and received is the number of losses.

The other important QoS parameters to study the handover are the delay (OWD) and jitter (IPDV) before and after the handover. We use again active measurements to compute them. Having the active flow traveling from the CN to the MN or vice versa we force a handover and we compute the OWD and IPDV for the packets before and after the handover.

4.2.5 Tests

For a good analysis of the handover is necessary to build a good set of tests. We ran the following set of tests for each mobility protocol: Half of the tests had the generated traffic from the CN to the MN while the other half was on the opposite direction. Moreover each direction of the tests was split as follows:

- VoIP Traffic: This flow simulates with UDP the properties of VoIP traffic (34 packets per second with 252 bytes of payload.
- Data Traffic: In order to compare a different bandwidth the other tests are done on a higher packet rate. This flow has 84 packets per second with a payload size of 762 bytes per packet.

We ran a set of 16 tests, each 5 minutes long from where we extracted a set of 63 valid handovers for Mobile IPv6. For Mobile IPv4 we extracted 60 valid handovers. Finally for Fast Handovers we ran a set of 10 tests each 5 minutes long extracting 40 valid handovers.

The FMIPv6 tests are done only from the CN to MN. When the packets flow in this direction, the access routers must tunnel and buffer packets showing an interesting behavior. However when the traffic source is the MN, there is no need to tunnel packets, just to buffer them on the MN (the FMIPv6 handover latency remains constant for both directions), that's why we focus on the CN to MN direction.

For Mobile IPv4 and Mobile IPv6 the handovers are forced attenuating the signal sent by the AP. The MN realizes this (it detects that the signal quality is becoming poor) and tries to search for a new AP. In our testbed we do not have external interferences and thus, the MN changes to the other AP. This procedure tries to simulate regular user movement.

Our Fast Handovers implementation behaves as stated in [31][1]. When the MN receives the Fast Binding Acknowledgment message it is ready to move to the new access router. At that point we force the wireless card to change from the old AP to

[1] In particular the implementation behaves as stated by: draft-ietf-mipshop-fast-mipv6-03.txt

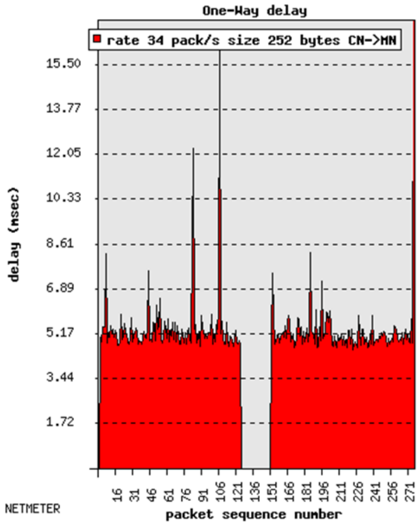


Figure 4.2: MIPv4 handover (instantaneous OWD)

the new one. As soon as our implementation detects the new link (using [112]) we send the Fast Neighbor Advertisement to announce the MN's presence. This method provides a faster handover because the MN does not need to wait until the wireless signal quality becomes poor, in fact, it knows to which AP will move when it is still on the previous link.

4.3 Results

This section describes the results obtained from the tests discussed on the previous section. The main goal of the results is to analyze the handover and to show trends, they do not must be taken as absolute numbers since they depend on the protocol implementation, the hardware and the distance between the entities, although the results are statistically representative.

4.3.1 Overview

This section presents an overview of the obtained results, Figures 4.2, 4.3 and 4.4 show three instantaneous One-Way-Delay (obtained with NetMeter) for the three protocols under test. All the figures have the packet sequence number on the x-axis and the OWD (in milliseconds) on the y-axis. Note that the figures are not in the same scale.

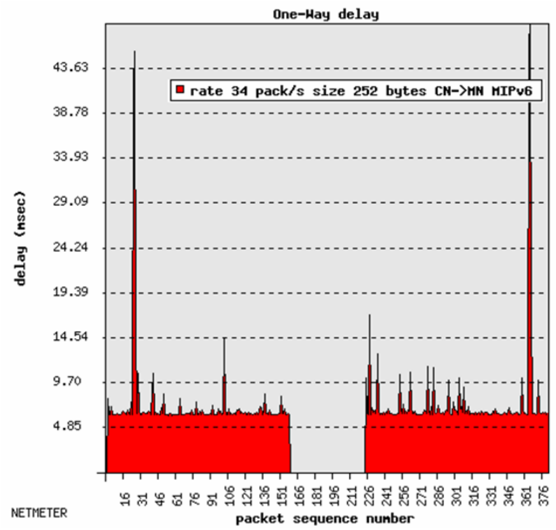


Figure 4.3: MIPv6 handover (instantaneous OWD)

Table 4.1: Handover Latency comparison for mobility protocols (ms)

	MIPv4		**MIPv6**		**FMIPv6**	
	Mean	**Std. Dev**	**Mean**	**Std. Dev**	**Mean**	**Std. Dev**
Handover Latency	104.5	6.3	1848.2	440.1	83.4	5.5

Figure 4.4 shows a sample for the FMIPv6 handover. We can clearly see that no packet is lost, regarding the OWD we see a spike. This behavior is due to the protocol operational procedures. While the MN is changing its point of attachment (from one AP to the other) the old access router is tunneling and forwarding packets to the new one and the new access router, at the same time, it is buffering packets until the MN regains connectivity. Therefore FMIPv6 delays (buffers) packets instead of losing them. The packets will be stored in a buffer while the MN's IEEE 802.11 layer is disconnected; hence, the packet's maximum delay is equal to the IEEE 802.11 handover latency.

4.3.2 Handover Latency

The whole system was tested doing a set of handovers, capturing all the signaling messages and processing them off line using PHM Tool. Table 4.1 (results in milliseconds) shows the results of the handover latency for the different mobility protocols without taking into account the wireless part.

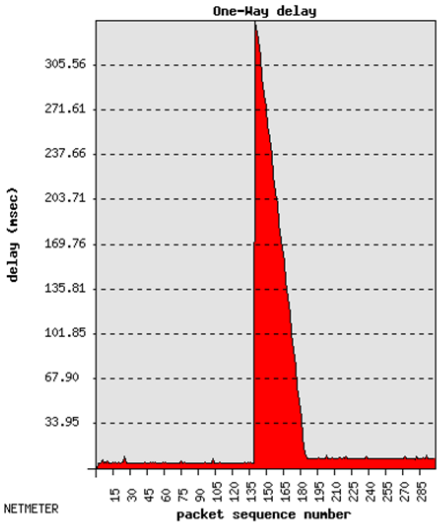


Figure 4.4: FMIPv6 handover (instantaneous OWD)

MIPv6 has the slower handover due to the IPv6 part. The MN has to perform Neighbor Unreachability Detection and Duplicate Address Detection to realize that its previous access router is no longer reachable and to check that its new CoA is unique on that link. Both algorithms have timeouts and they were set to their minimum value.

In order to further explore this issue we use the PHM tool to compute the different parts of the handover and reveal the specific duration of each part of the handover (table 4.2). The wireless handover is detailed and as we can see the scan phase is the longest one, the MN uses in average 257ms to find a new AP. The whole 802.11 handover (Scan, Authentication and Association) represents 12% of the total handover latency. The second phase is the time sent by IPv6, this is the longest part (87% of the total time) and is related to the timeouts of the NUD and DAD algorithms. Finally the mobility part of the handover (from Binding Update to Binding Acknowledgment) is only related with the distance from the MN to the HA and its CNs.

The Mobile IPv4 standard allows tuning the implementation and, as soon as the MN realizes that a new access router is present (upon reception of an Agent Advertisement) obtains a new CoA and forgets about the old access router. IPv4 does not perform DAD, the CoA is the FA address. The Mobile IPv6 protocol relies on Neighbor Discovery autoconfiguration which, due to security reasons, is difficult to speed up [106].

FMIPv6 has very low handover latency as expected. Note that the IEEE 802.11

Table 4.2: Numerical results obtained using PHM (ms)

	Mean	**Std. Dev.**
Scan	257.224	108.007
Authentication	2.733	1.183
Association	1.268	0.311
IPv6	1836.413	430.196
Registration (HA)	3.914	1.017
Registration (CN)	9.262	4.881
Total time	2107.82	450.619

part of the handover latency has not been taking into account (for any of the protocols) because it's a common part for all of them.

4.3.3 Packet Losses

We compute the packet losses for the different mobility protocols using active measurements. Table 4.3 shows a summary of the obtained results detailed for the different active tests performed (different flow direction and different traffic types).

During a handover, the packets are lost while the MN is changing its access point (IEEE 802.11), obtaining a new CoA and registering it. For MIPv4/MIPv6 as higher is the packet rate higher is the packet losses. In general, the packet losses for both protocols are the rate multiplied by the handover latency. Regarding FMIPv6 the results show that no packet is lost as expected. As explained above when the traffic source is the MN, there is no need to tunnel packets, just to buffer them on the MN (the FMIPv6 handover latency remains constant for both directions). That's why we did not perform these tests.

4.3.4 QoS Metrics

Tables 4.4 and 4.5 summarizes all the results regarding the provided QoS level of the mobility protocols under test.

Table 4.3: Packet Losses comparison for mobility protocols (ms)

	MIPv4		**MIPv6**		**FMIPv6**	
	Mean	**Std. Dev**	**Mean**	**Std. Dev**	**Mean**	**Std. Dev**
CN to MN (VoIP)	22	12.46	61.71	17.54	0	0
CN to MN (Data)	42.9	13.64	207.21	65.90	0	0
MN to CN (VoIP)	10.58	1.98	65.80	9.78		
MN to CN (Data)	30.27	6.22	162	16.97		

MIPv4 and MIPv6 show higher delay (with also a higher value for the Standard Deviation) before than after the handover, especially for longer packets. These important QoS fluctuations before the handover are because of the wireless card. For both protocols the wireless card decides to switch to a new access point when it detects that the signal quality becomes poor, hence, the provided QoS is severely affected.

FMIPv6 presents a different behavior; it has low OWD fluctuations before and after the handover; however after it the OWD is slightly higher. In FMIPv6 the wireless card is forced (by the above layers) to switch from one AP to another one without having to wait until the signal quality becomes low, thus the OWD is not affected after and before the handover. Note that during the handover the packets are severely delayed (instead of lost). After the handover the OWD is higher because the packets must be routed to the old access router and tunneled to the new access router, introducing an extra hop. As soon as the MN sends a Binding Update to its HA and CNs the traffic is routed directly to the MN.

Table 4.6 presents the IPDV before and after the handover. IPDV confirms that in the MIPv4/MIPv6 handover packets suffer OWD variability before the handover due to wireless signal degradation.

Table 4.4: OWD before/after the handover for Mobile IPv6 and Mobile IPv4 (ms)

	MIPv4				**MIPv6**			
	Mean		**Std. Dev**		**Mean**		**Std. Dev**	
	Before	**After**	**Before**	**After**	**Before**	**After**	**Before**	**After**
CN to MN (VoIP)	5.0	6.0	0.8	1.5	9.7	7.4	9.6	1.6
CN to MN (Data)	24.1	12.8	14.7	1.3	109.2	14.3	108.3	3.8
MN to CN (VoIP)	4.5	2.0	5.0	0.2	5.9	4.6	3.3	2.8
MN to CN (Data)	16.2	4.0	22.0	0.2	19.0	6.8	15.6	0.5

Table 4.5: OWD before/after the handover for FMIPv6 (ms)

	Fast Handovers			
	Mean		**Std. Dev**	
	Before	**After**	**Before**	**After**
CN to MN (VoIP)	2.7	5.1	1.3	1.7
CN to MN (Data)	6.3	7.5	2.8	3.8

Table 4.6: IPDV before/after the handover for Mobile IPv6 and Mobile IPv4 (ms)

	MIPv4				**MIPv6**			
	Mean		**Std. Dev**		**Mean**		**Std. Dev**	
	Before	**After**	**Before**	**After**	**Before**	**After**	**Before**	**After**
CN to MN (VoIP)	9.5	6.8	5.2	3.7	76.2	33.9	104.7	27.7
CN to MN (Data)	160.0	8.4	108.6	8.5	332..2	12.1	297.3	11.9
MN to CN (VoIP)	11.7	1.0	22.9	0.2	43.2	8.1	54.1	8.2
MN to CN (Data)	63.6	1.4	86.4	0.8	131.6	9.1	198.4	11.2

4.4 Fast Handovers for Mobile IPv6 Implementation

The FMIPv6 implementation is written in C and runs on Linux Kernel 2.4.26, it enhances the Mobile IPv6 MIPL 1.1 implementation and complies with the draft–ietf–mipshop–fast–mipv6-03.txt. The basics parts of the draft are implemented, some optional and error recovery parts that does not affect the performance of the protocol are not developed. Our implementation also supports any wireless card (with Linux support) through the "Wireless Tools for Linux". Lastly it is worth to note that the implementation can be found at [222].

4.4.1 Architecture

The architecture of the FMIPv6 implementation is mainly divided into two modules:

- fh-base: This is a "dumb" module that runs into the kernel and interacts with the IPv6 module, the MIPL module and Netfilter. It receives commands from the user space.
- fh-daemon: This is a user-space daemon, interacts with the user, the wireless interface (through netlink) and actually implements the FMIPv6 protocol. It communicates with the "dumb" fh-base kernel module to perform the protocol operational procedures.

The motivation behind this architecture is to split the implementation into two parts, user-space and kernel-space. Most of the functionality is at the user-space, this way the implementation can be easily modified. Modifying the kernel-module is a complex and costly task.

4.4.2 Development environment

Developing support for a new protocol for the Linux Kernel is not an easy task, especially if it has to interact with other modules. In order to have a productive development environment we used User-Mode-Linux (UML) [121]. UML provides a virtual machine that emulates a Linux Box. We recreated our real testbed using UML on a single physical machine, all the virtual machines had the same configuration than the real ones, we used the same network topology, the same kernel and software versions. IEEE 802.11

is not supported by UML, however we emulated the handover using IEEE 802.1 and we simulated movement between two switches. The IEEE 802.11 part of the implementation was only tested in the real testbed. With this development environment we were able to intensively test our implementation in an easy and affordable way. Only after the implementation was mature enough, we moved it to the actual testbed to test it and to measure the FMIPv6 handover.

4.5 Related Work

In this section we detail related publications to this specific contributions. First *X.Perez-Costa* in [122] presented a mathematical model that account for the handover lateny. The authors in [123] studied the Mobile IPv6 handover by simulation obtaining similar conclusions than in this work. Regarding empirical evaluations the authors in [124] made an empirical analysis of the IEEE 802.11 handover. In their study the authors measured the handover of a set of 802.11 chipsets highlighting the main differences. Finally in [125] the researchers studied the WLAN/MIPv6 handover in a testbed and proposed a new algorithm to speed up the handover latency.

4.6 Summary and Conclusions

We have provided an empirical evaluation of the handover. By applying a methodology that uses a mix of passive and active measurement we have measured the handover latency and the impact on QoS metrics of the handover. More specifically we have evaluated public implementations of Mobile IPv4 and Mobile IPv6 and our public implementation of Fast Handover for Mobile IPv6. From the results we conclude:

- In terms of handover latency the fastest protocol is Fast Handover for Mobile IPv6, as expected. Mobile IPv4 performs better than Mobile IPv6 because it does not need to perform the Duplicate Address Detection and the Neighbor Unreachability Detection algorithms. These algorithms are used to ensure that its old access router is no longer reachable and that the new CoA is unique on that link. In fact a recent standard, Optimistic Duplicate Address Detection [126] reduces significantly the delay of the DAD algorithm by assuming that the address is not duplicated.

- For Mobile IPv4 and Mobile IPv6 the packet losses depend on the handover latency and the rate. Basically during the handover packets are lost. Regarding Fast Handovers for Mobile IPv6 has a zero packet loss. The protocol stores them (delays them) instead of loosing them. Our implementation behaves as expected.
- The provided QoS level for Mobile IPv4 and Mobile IPv6 is severely impacted before the handover. This is mainly because the wireless card changes from one access point to another when it detects that the signal quality becomes poor. However, Fast Handovers for Mobile IPv6 does not need to wait until the signal degrades, it forces the wireless card to switch from one access point to another, and hence, it does no suffer QoS fluctuations. Although with FMIPv6 the packets are delayed during the handover (as much as the IEEE 802.11 handover latency).
- According to [116] and having into considerations the results, MIPv4 and FMIPv6 are acceptable for VoIP applications, however MIPv6 has too large handover latency and losses too many packets.

Finally it is worth to note that considering the analytical model presented in the last chapter the results obtained for Mobile IPv6 are applicable to NEMO and HMIPv6. Both protocols also suffer from large handover latency and this can be unacceptable for real-time applications.

Part III

Near-Future: Deployment of Layer-3 Mobility Protocols

5

The Mobility Agent Architecture

5.1 Introduction

This chapter explores the solution space of two potential issues identified in the deployment phase of the transition to the Mobile Internet. On the one hand, in case of an hybrid deployment, some IPv6 nodes do not include Mobile IPv6 support and thus, if they act as correspondent nodes they do not provide route optimization support. Deploying such support may be very expensive since it requires updating the kernel of all the nodes (potential issue II.B). On the other hand, the Return Routability procedure, which provides Route Optimization, is incompatible with some Load Balancing techniques. Again this prevents mobile clients from communicating directly with this type of peers (potential issue II.D). This lack of route optimization has a significant impact in the performance of the Mobile Internet (see chapter 2 for more details).

We consider a solution for both problems at the same time since they are related. It is reasonable to assume that mobile nodes are basically clients that communicate with large servers (i.e. web sites). Therefore correspondent nodes are usually large servers that belong to Content Providers. Both issues affect mainly these type of hosts, either if they operate behind a load balancer device or not. That is we address both issues at the same time and we focus our solution to this particular scenario. The design of the solution should take into consideration the following requirements. On the one side the solution should be transparent to these servers and should avoid modifying them. On the other side the deployment and maintenance cost of the solution should be low. Basically it must be a simple plug & play solution useful for Content Providers.

Considering these design principles we propose the Mobility Agents as a solution for both issues. A Mobility Agent is a new mobile entity located at the correspondent network that acts as a transparent proxy for the different servers (CNs) and load balancing devices. It hides mobility related issues to these CNs and allows Mobile IPv6 clients to communicate directly. This entity processes Return Routability's messages on behalf the nodes. Additionally it is compatible with legacy mobile nodes. This way Route Optimization support can be deployed flawlessly, it does not require server support and has zero-configuration.

This chapter is divided as follows, first we provide some basic background about the Return Routability procedure and Load Balancing techniques. Next we present the Mobility Agents' architecture and its basic procedural operations. Then we detail the incompatibility between the Return Routability procedure and Mobile IPv6 clients and we show how the Mobility Agents can be extended to solve this issue. Finally we provide a qualitative evaluation of the proposed solution.

5.2 The Return Routability Procedure (Background)

This section describe the Return Routability (RR) procedure, the reader familiar with Mobile IPv6 can skip this section safely. This procedure is defined in [14] and explained in [15]. This procedure authorizes the direct communication between the MN and the CNs. The RR's main objective is to enable the CN to obtain some reasonable assurance about the identity of the MN (WHO) and its location (WHERE). Only with this assurance the CN is able to accept Binding Updates from the MN allowing direct communications (Route Optimization).

Basically, the RR is done by testing if packets addressed to the HoA (WHO) and CoA (WHERE) are routed to the MN. Only if the MN receives both packets it is able to communicate directly. Figure 5.1 shows the schema of the RR procedure.

Once the MN has finished the *Agent Discovery* and *Registration* phases (messages 1 and 2, see Section 2.2.7 for further details) the RR procedure starts. The MN sends two messages at the same time, the Home Test Init and the Care-of Test Init to obtain two keygen tokens from the CN. The first message is sent through the HA while the second one is sent directly. The CN replies with the Home Test and the Care-of Test messages. The first one is sent to the MN's Home Address while the second one is sent

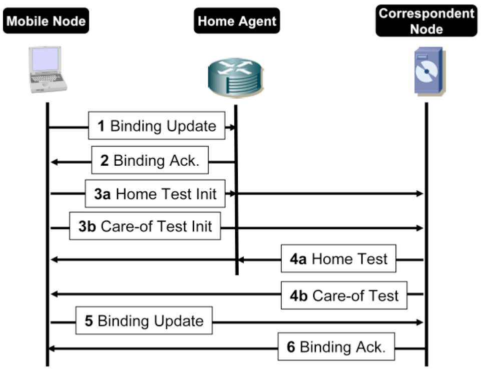


Figure 5.1: Return Routability Schema

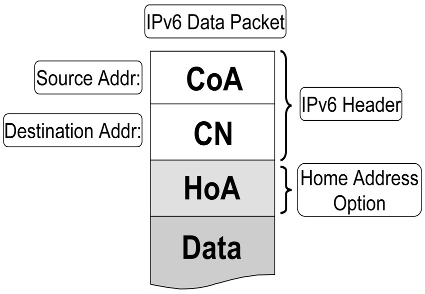


Figure 5.2: Extension Headers for data packets (MN to CN)

directly. Both keygen tokens are combined by the MN into a binding management key used to authenticate the Binding Update. At this point the MN and the CN are able to communicate directly.

In an optimized-route data packets exchanged between the MN and the CN use special IPv6 Mobility Extension Headers. Figure 5.2 shows which extensions headers are used for packets sent by the MN to the CN.

If a MN has performed the RR procedure with its CNs the packets are sent using the Home Address Option Extension Header defined in the MIPv6 RFC [14] (figure 5.2). Packets sent by the MN to the CN have the CoA as source address. Destination address for these data packets is the CN's address. The MN's Home Address is included into the Home Address Option. In this way, when the CN receives the packet it will know the identity of the MN (WHO) by inspecting the Home Address Option Extension Header and the location of the MN (WHERE) by inspecting the packet's source address. If

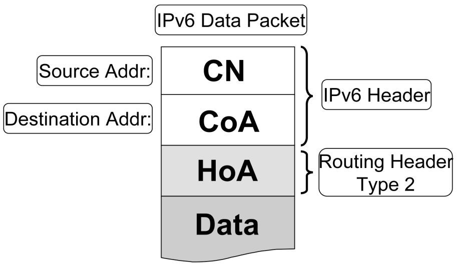


Figure 5.3: Extension Headers for data packets (CN to MN)

the CN has a binding between the MN's CoA and the MN's Home Address the packet will be accepted and sent for upper layer processing.

Packets sent by the CN to the MN use the Routing Header Type 2 Extension Header. Figure 5.3 shows these data packets.

Packets sent by the CN towards the MN must be addressed to a given location (the MN's CoA) but to a given identity (the MN's Home Address). The RFC defines that packets include the MN's CoA as destination address while the Home Address is included into the Routing Header. When the packet reaches the MN it will forward it to the address defined in this extension header, in this case its own identity. Then the packet is processed by upper layers as it was sent to the MN's Home Address.

5.3 Load Balancing Techniques (Background)

This section presents an overview of the existing load balancing techniques, again the reader familiar with these devices can skip this section safely. Although these techniques can be applied to any service we focus on web services load balancing techniques.

Web server administrators face the challenge of increase web server capacity as the Internet grows up. The first option is to add more hardware resources or improve the web server itself. While these strategies relieve short-term pressure it is neither a cost-effective nor long-term solution. A more appealing solution is to deploy a distributed web server with multiples nodes. Some system component is needed to distribute client requests among the servers. The multiple web servers are loosely coupled and under the client's point of view act as a single server. Depending if this virtualization is extended to the IP level or not there are two different techniques. In the following subsections these techniques are detailed.

Distributed Web Systems

A distributed web systems consists of a set of web servers whose IP addresses are visible to client applications and thus, the virtualization is not extended to the IP level. A client request is routed to a single web server that belongs to distributed web system by using two different approaches. The first one uses the DNS servers while the second one uses web servers to route incoming client requests.

1) DNS-Based Techniques: The DNS-based technique uses DNS servers to route incoming client requests to a target web server. This technique was initially presented in [127] and it is intended to geographically distributed web systems. DNS-routing is performed during the client lookup procedure. DNS Servers reply to DNS requests not with a single IP address but with a list of IP addresses (the servers' IP addresses). The list's order follows a certain policy. Usually the first returned IP address belongs to the nearest available server or to the less busy server. Basic DNS clients simply use the first entry and discard the rest.

This technique has the main drawbacks from the DNS hierarchy itself and TTL (Time to live) values [12; 128]. Firstly DNS servers usually cache DNS replies since DNS information changes very little. This means that even if a server becomes unavailable some DNS servers may continue redirecting traffic to it. Secondly this technique may not distribute traffic uniformly just because OS's do not usually make requests to the authoritative name servers but to their pre-configured name servers. Those name servers then forward the requests to the authoritative DNS servers and cache the reply. Finally, new information on the DNS hierarchy takes a while to propagate. This issue does not allow a site to quickly increase its capacity.

2) Web-Based Techniques: The second approach uses web servers to route client's requests. In this approach a single web server receives all the incoming clients' requests and redirects them to other web servers through the HTTP redirection [129] message or the URL rewriting mechanism [130]. The main drawback for these approaches is that they increase delay, as every redirection requires the client to initiate a new TCP connection. Even more, the web server that redirects incoming clients' requests may be overloaded adding extra delay.

Cluster-Based Web Systems

Cluster-based web systems extend the virtualization to the IP level. In this technique a set of web servers that are interconnected through a high-speed network and in a single location can be viewed as a single computer. The cluster system is accessible under a single IP address, known as virtual IP address. This virtual IP address is configured at a front-end node that will handle all the incoming clients' requests. The front-end node, known as load balancer, intercepts the servers' communications to the Internet making the whole system transparent both to the clients and to the servers. The load balancer device is able to identify all the servers through a private IP address or a layer-2 address. This load balancer will distribute the inbound packets to a target server according to a certain policy. Mainly there are two types of load balancers. The first type uses layer 4 information to make the routing decision while the second type uses the whole protocol stack to make the decision.

1) Layer 4 Load Balancers: Layer 4 Load Balancers assign packets that belong to the same TCP connection to the same server persistently. Thus clients are identified by a source IP address and port. There are different mechanisms to redirect the packets to the selected server.

The first mechanism, Packet Rewriting [131], is based on the IP Network Address Translation [25] (NAT) and it is implemented by many commercial products. The Packet Rewriting load balancer consists of a virtual server which has a virtual IP address. Clients will always send their requests to the virtual IP address. In turn, the load balancer will rewrite the destination IP address of the client's packet to the IP address of a server according to a given policy. Next, the load balancer will forward the packet. Then the server will process the packet. Server's responses will flow through the load balancer that will rewrite the packet's source IP address to its virtual IP address. In this way, clients will receive packets as they were sent from the virtual IP address. As it has been said before, when a given client has been redirected to a given web server further client's requests must be redirected to the same server.

The second mechanism is actually a set of mechanisms known as One-way architectures. In One-way architectures inbound packets pass through the load balancer device while outbound packets flow directly from the servers in order to avoid that the load balancer becomes the bottleneck of the whole system. There are different proposals of one-way architectures such as Packet Tunneling [132] and Packet Layer-2 forwarding [133].

2) Layer 7 Load Balancers: Layer 7 Load Balancers distribute client's requests according to information from the application level (HTTP). This way the load balancers device, acting as a TCP proxy, establishes a separate TCP connection with the client and with the target server in order to receive the whole HTTP request. In this case the load balancer can distribute different HTTP requests from the same client to different servers because HTTP is a stateless protocol [129]. This technique is called TCP Gateway (which actually is a simple proxy).

The TCP Splicing [134] technique is an enhancement of the TCP Gateway technique where IP packets are forwarded from one endpoint to the other one without having to cross the TCP layer. Once the client-to-server binding has been established, the load balancer handles the subsequent packets by changing the IP and TCP headers so that the process is transparent for the client and for the server.

Layer 7 Load Balancers also work in One-Way architectures where outbound packets flow directly from the server to the clients. Approaches such as TCP Handoff [135] and TCP Connection Hop [136] (a proprietary mechanism) are good examples. With these approaches, the load balancer hand offs the TCP connection endpoint to the selected server. This mechanism is transparent to the client as data sent by the servers appear to be coming from the load balancer.

5.4 Mobility Agents

This section presents the Mobility Agents (MA), a new entity located at the correspondent network that performs the operations of Mobile IPv6 on behalf the CNs. When a MN initiates a Route Optimized connection with a CN it runs the Return Routabilty procedure. The messages of this procedure is intercepted by the MA that process them on behalf the CN. The MA replies to these messages, it computes the required cryptographic operations to authenticate the Binding Update and finally, it stores the binding between the Care-of Address (CoA) of the MN and the Home Address of the MN. With MAs the CNs are unaware of mobility issues. In fact, they do not need Mobile IPv6 support at all. In further sections we show how the MAs can be extended to support load balancing.

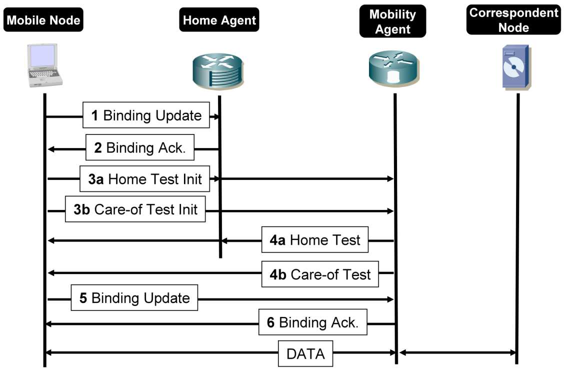


Figure 5.4: Mobility Agents interaction with Return Routability

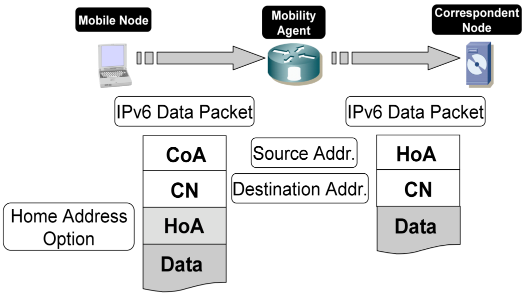


Figure 5.5: Home Address Option processing

Signalling Interaction

Figure 5.4 shows how the MAs perform the Return Routability procedure on behalf the CNs.

The MA acts as a transparent proxy for the MNs, receiving and processing all the signaling messages. When the Binding Update of the MN has been authorized it stores it and it replies with a Binding Acknowledgement. MAs can work with other Return Routabilty procedures such as [64; 137] in the same way.

Data Exchange Interaction

When the MN sends packets to the CN it includes the Home Address Option. Figure 5.5 shows how it is processed by the MA.

When a data packet including a Home Address Option is received by the MA it first checks if it has a binding between the source address of the packet (CoA) and the

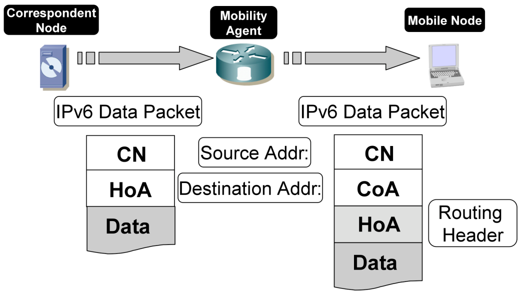


Figure 5.6: Routing Header processing

Home Address. If it has a binding it removes the extension header and it replaces the source address of the packet (CoA) with the Home Address of the MN included into the Home Address Option. This way the CN receives a packet from MN as if it were at home, and it processes it as usual.

When the CN sends packets to the MN in Mobile IPv6 it includes the Routing Header, however with MAs, CNs do not need Mobile IPv6 support and thus, they send the packets as stated by the IPv6 RFC [26]. As shown in figure 5.5, the packet is received by the MA that checks if it has a binding for the destination address of the packet (Home Address). If it does not have a binding it forwards it as defined in the IPv6 standard. However if a binding exists it replaces the destination address of the packet with the CoA of the MN and it adds the Routing Header Type 2 Extension Header. This extension header includes the Home Address of the MN. The MA also sets the Next Header field according to the new extension header (figure 5.6).

The MAs act as the tunnel endpoints. The tunnel is established between the MN and the MA. It is important to remark that this tunnel is conceptual and does not use the traditional IP over IP technique.

Location

The MA has to receive all the packets exchanged between the MN and the CN and thus, it must be placed on the path between both nodes. Due to movements of the MN the path between both nodes may change and thus, the MA must be placed at the border router of the correspondent network. We assume that the correspondent network is single-homed. In the case that the Correspondent Network is very large or multihomed, the MAs can be placed in each access router to avoid scalability issues.

Interaction with IPSec

The MAs should take into account what happens when the Route Optimization of Mobile IPv6 is used with IPSec in transport mode (IPSec in tunnel mode is fully compatible with MAs). IPSec in transport mode uses one parameter that affects the operations of our MAs. This parameter is the Integrity Check Value (ICV) which belongs to the Authentication Header [139] protocol. The ICV field is used to provide integrity to the communications and covers the entire packet. Mobile IPv6 specifies that packets sent by the MN that use IPSec in transport mode must be assembled in the following way:

1. The data packet is created by higher layer protocols and applications (e.g., by TCP) as if the MN were at home and Mobile IPv6 were not being used.
2. The Home Address Option Extension Header is inserted. The source address of the packet is set to the CoA of the MN and the Home Address Option is set to the Home Address of the MN.
3. The Authentication Header authentication data (ICV) is calculated as is the following was true. The source address of the packet contains the Home Address and the Home Address Option contains the CoA.

Please note that the Mobile IPv6 RFC defines that the ICV calculation must be done exchanging the source address field with the Home Address Option field.

The packets sent by the CN to the MN when Route Optimization and IPSec are used are assembled in the following way:

1. The data packet is created by higher layer protocols and applications (e.g. by TCP) as if the MN were at home and Mobile IPv6 were not being used.
2. The Routing Header is inserted. The destination address of the packet is set to the CoA and the Routing Header to the Home Address
3. The AH authentication data (ICV) is calculated as if the following was true. The destination address of the packet contains the Home Address and the Routing Header contains the CoA.

Once again, the RFC defines that the ICV calculation must be done exchanging the destination address field with the Routing Header field.

The MAs cannot deal with IPSec protected packets because they modify the packet. If MAs were used with an IPSec connection the ICV verification would fail at the destination.

Different approaches to the Route Optimization of NEMO [64] solve this issue by creating a tunnel between the Mobile Router and the Correspondent Router. This way, inner packets are not modified. However, packets sent in an optimized route of Mobile IPv6 are actually tunneled. Instead of inserting a whole new IP header, Mobile IPv6 uses the Home Address Option and the Routing Header. The traditional tunneling technique [140] uses four IP address: the source and destination addresses and the two tunnel endpoint IP addresses. The tunnel of Mobile IPv6 requires just three addresses: the CoA, the Home Address and the address of the CN. The tunnel of Mobile IPv6 is configured between the address of the CN and the CoA while the inner packet has the address of the CN and the Home Address. This tunneling [141] technique was specially designed for Mobile IPv6 because it has less overhead (25%).

In fact, in standard Mobile IPv6, packets are processed at the destination as if they were tunneled. In this case, the extension headers (Home Address Option or Routing Header) represent the outer IP header of the traditional tunnels. This header is removed and the IP source or destination address is replaced by the address contained in the extension header. Finally, the modified IP header represents the inner IP header. The resulting packet of this process reaches its destination as if the MN was at home and it was sent from or to the Home Address.

We claim that, if we accept that Mobile IPv6 tunnels packets, data packets exchanged between the MN and the CN with Route Optimization should be processed as tunneled packets. Under this assumption, a Mobile IPv6 data packet should be assembled without including the Home Address Option or the Routing Header into the ICV calculation. Please note that the ICV calculation does not include the outer IP header of a tunnel [138; 139]. In this case, the outer IP header is included after the ICV calculation and the header will be removed before it arrives to its destination. Thus, we propose that packets sent by the MN to the CN are assembled as follows (figure 5.7):

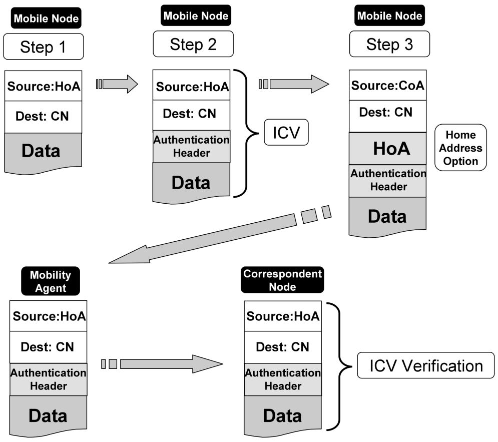


Figure 5.7: Proposed packet assembly at the MN

1. The data packet is created by higher layer protocols and applications (e.g., by TCP) as if the MN were at home and Mobile IPv6 were not being used. The source address of the packet contains the Home Address while the destination address of the packet contains the address of the CN.

2. The AH authentication data (ICV) is calculated.

3. Next the packet is tunneled by inserting the Home Address Option Extension Header. The source address of the packet is set to the CoA and the Home Address Option is set to the Home Address

With MAs, the CN does not need mobility support. In this case, the CN will assemble the packet following the rules defined in the IPv6's standard [26]:

1. The data packet is created by higher layer protocols and applications (e.g., by TCP) as if the MN were at home and Mobile IPv6 were not being used. The destination address of the packet contains the Home Address while the source address of the packet contains the address of the CN.

2. The AH authentication data (ICV) is calculated.

The CN sends the packets addressed to the Home Address. This packet is intercepted by the MAs. As explained before, the MA checks if it has a binding for this particular Home Address. If it does not have a binding it forwards it as defined in the IPv6 RFC [26]. Otherwise it tunnels the packet by inserting the Routing Header. The MA also changes the destination address of the packet to the CoA. Then, the Routing Header of the packet includes the Home Address. Finally, when the packet reaches the MN it removes the Routing Header and replaces the destination address with its own Home Address. Finally it verifies the ICV by calculating it on the same way than the CN did.

With this simple modification our MAs are also compatible with IPSec connections. It is important to remark that we propose to modify how Mobile IPv6 data packets are assembled, not the IPSec or the IPv6 standard. This modification only affects to the Mobile IPv6 implementations.

Security Analysis

Modifying how data packets of Mobile IPv6 are assembled when they interact with IPSec in transport mode can introduce new security threats. In this subsection we analyze these threats.

It is very important to remark that although the Mobility Extension Headers are not protected by the ICV the contained Home Address is indeed protected. The ICV was computed as if the MN was at home. Then, the MA or the MN replaced the Home Address with the CoA. This means that actually the CoA is not protected by the ICV.

We consider two separated cases. First we analyze the security considerations when packets are sent from the CN to the MN. Next when packets are sent from the MN to the CN. Packets sent by the CN to the MN are as follows:

- Source address of the packet: Address of the CN
- Destination address of the packet: CoA
- Routing Header : Home Address

However, the ICV has been computed with the following IP header and without including the mobility extension header:

- Source address of the packet: Address of the CN
- Destination address of the packet: Home Address

This means that an attacker could change the destination address of the packet (CoA) by using a Man-in-the-Middle attack. This would route the packet to a different destination. This attack is also possible even if the ICV computation includes the mobility extension header. Packets sent by the MN to the CN are as follows:

- Source address of the packet: CoA
- Destination address of the packet: Address of the CN
- Home Address Option : Home Address

However, the ICV has been computed with the following IP header and without including the mobility extension header:

- Source address of the packet: Home Address
- Destination address of the packet: Address of the CN

This means that an attacker could change source address of the packet (CoA) using a Man-in-the-Middle attack. The packet would reach the MA. This entity would check if a binding exists between the source address and the Home Address of the MN. As the source address has been modified by the attacker the binding does not exists and thus the packet is discarded. If the binding exists because the attacker is trying to impersonate another MN then the CN would also drop the packet. As it has been explained before, the Home Address is included into the ICV computation and the attacker is not able to forge and encrypt an ICV which includes the Home Address of the victim. In fact, the CoA is protected by the binding stored at the MA which is, at the same time, protected by the Return Routability procedure. The Home Address is always protected by the ICV value.

An attacker could also modify the packet, sending a malformed mobility extension header. This would drop the packet at the destination. This is also possible even if the ICV computation includes the mobility extension header.

As it has been shown our modification of how data packets are assembled in Mobile IPv6 when Route Optimization and IPSec in transport mode are used does not introduce new security threats.

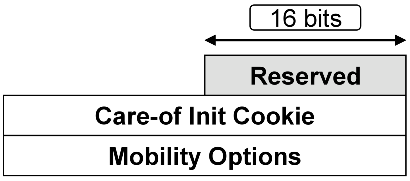


Figure 5.8: Care-of Init Message

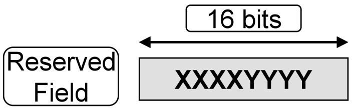


Figure 5.9: Modified Reserved Field

Legacy Support

Modifying how Mobile IPv6 data packets are assembled when IPSec in transport mode and Route Optimization are used can introduce incompatibility issues with legacy Mobile IPv6 nodes. In this subsection these issues are considered. It is important to remark that our MAs are compatible with legacy Mobile IPv6 nodes when IPSec in tunnel mode is used. Only if IPSec in transport mode is used incompatibility issues arise.

To provide compatibility with legacy Mobile IPv6 nodes we use the reserved field of the Care-of-Test Init message (figure 5.8). This message is the first one sent in the Return Routability procedure. By using this message we are minimizing the amount of useless signaling sent by unmodified Mobile IPv6 nodes.

Modified Mobile IPv6 nodes must compute the Reserved field using the following rules (figure 5.9). The first byte indicates whether the Mobile IPv6 node has MA IPSec support or not. When an unmodified Mobile IPv6 node sends the Care-of-Test Init message with the Reserved field set to zero the MA recognizes it as an unmodified node and sends an ICMP parameter problem code 1. The Mobile IPv6 node then reverts to bidirectional communication with the Home Agent to reach the CN. The unmodified MN is be able to communicate with the CN but without Route Optimization. This particular case is the same than when the CN does not have mobility support.

5.5 Mobility Agents Extension for Load Balancing

In this section first we detail the incompatibility between load balancing techniques and route optimized connections, then we extend the Mobility Agents to solve this issue.

5.5.1 Problem Statement

Distributed Web Systems

Distributed Web Systems are compatible with the return routability procedure because they do not extend the virtualization to the IP layer. The communications are always established between a MN and a single server without any further packet processing.

Layer-4 Load Balancers

MIPv6's return routability is not compatible with any Layer 4 Load Balancing technique because it requires that some state is stored at the CN. The CNs must store a list of bindings between the MN's Home Addresses and the MN's CoA in a structure called Binding Cache [14].

Layer 4 Load Balancers are required to establish client-to-server bindings. In this way each client has an assigned target server. Packets sent by the client are forwarded always to the same server. Upon a client connection establishment, the load balancer identifies the client according to its source port and IP address and creates the appropriate binding. Subsequent packets are forwarded to the selected server according to this binding.

With MIPv6, data packets flow with the CoA (the temporal IP address) as source address. This address changes according to the MN's movements. Thus, load balancers cannot identify clients by inspecting the packet's source address.

Load balancers should identify MNs by their Home Address. This address does not change even if the MN changes its point of attachment. Each MIPv6's data and signaling packet includes the Home Address except for the Care-of Test Init message (figure 5.10).

MNs send the Care-of Test Init message when they start the return routability procedure due to a connection establishment or a handover. The message is used by the MN to request to the CN a care-of keygen token. This token, combined with the

```
                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                     Care-of Init Cookie                       +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                        Mobility Options                       .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5.10: Care-of Test Init Message [14]

home keygen token (requested trough the Home Test Init message) provides the binding key used to authenticate the Binding Update.

According to the information contained in this message the load balancer is unable to identify the MN (client). This message includes the new Care-of Address that used by the MN and a care-of init cookie which is a newly generated random number. The reserved field has not been yet standardized and the MIPv6 RFC [14] does not define any Mobility Options for such message.

This information is not enough to relate it neither to the client nor to the stored state at the server. In other words, the load balancer cannot relate the Care-of Test Init message with any pre-established client-to-server binding. This means that the load balancer is unable to process this message and thus, the return routability procedure fails forcing the MN to communicate through the HA (sub-optimal path). An obvious option would be to forward the Care-of Test Init message to all the servers. In this case each server would reply to the MN with its own care-of keygen tokens leading to an authentication failure.

Layer-7 Load Balancers

The TCP Gateway technique is compatible with the MIPv6's RR because the load balancer creates separate TCP connections with the clients and with the servers. In this case the MN would perform the return routability's procedure with the load balancer. The procedure's state would be stored at the load balancer. However the TCP Splicing technique is not compatible. As it has been explained in section 5.3 this technique also creates separate TCP connections but, in order to improve the performance of the TCP Gateway technique, it forwards IP packets from one endpoint to another without crossing the TCP layer. Packets are forwarded directly from one connection to another

Table 5.1: Mobile IPv6's Return Routability Compatibility

Load Balancing Mechanism	**Compatible**
Distributed Web Systems	
DNS-Based Routing	Yes
URL Rewriting	Yes
HTTP Redirection	Yes
Cluster-Based Web Systems	
Layer-4 Load Balancers	
Packet Rewriting	No
One-Way Architectures (Packet Single Rewriting and Packet Tunneling)	No
Layer-7 Load Balancers	
TCP Gateway	Yes
TCP Splicing	No
TCP Handoff	No

changing its IP and TCP headers appropriately. This means that the required state is stored at the server instead of at the load balancer. Once again, when a Care-of Test Init message arrives, the load balancer will be unable to identify the client and relate it to the appropriate client-to-server binding.

TCP Handoff and TCP Connection Hop are neither compatible with the return routability procedure. These techniques also create client-to-server bindings and they forward the TCP connection state to the selected server. Even if these protocols were updated to forward also the required state the load balancers would fail to identify the client when a Care-of Test Init message arrived.

Summary

Table 5.1 presents a summary of the compatibility issues between load balancing techniques and the return routability procedure. As we can see while Distributed Web Systems are compatible with mobile clients, generally, Cluster Based Web Systems are not.

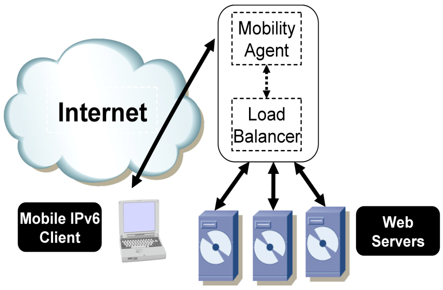


Figure 5.11: Proposed Load Balancer Architecture

5.5.2 Load Balancer Architecture

Figure 5.11 presents an extension to the MA architecture that supports load balancing. This includes two modules, a Mobility Agent and a regular load balancer device (a Packet Rewriting or a TCP Splicing device).

Clients address their requests to the server's network. The architecture receives those packets that are initially processed by the Mobility Agent. If the packet belongs to a non-MIPv6 client (i.e it does not use Mobility Extension Headers) the Mobility Agent forwards it to the load balancer device. In this case the load balancer processes the packet as usual. It identifies the client by inspecting the packet's source address and it forwards the packet to the selected server according to the client-to-server binding.

If the client is a MN that wants to establish a Route Optimized connection with the servers it starts the return routability procedure. As it has been detailed before, the Mobility Agent processes the mobility signaling on behalf the servers. This way the required MIPv6's return routability's state is stored at the Mobility Agent (not at the servers) and it is able to reply to Care-of Test Init messages with its own care-of keygen tokens.

Once the RR procedure has finished the MN will start to send data packets using the Extension Headers. For each data packet the Mobility Agent will replace the MN's CoA for the MN's Home Address and it will process the Extension Headers hiding mobility issues to the load balancer. In this way the load balancer can process the packet as usual, as if the MN was a fixed node or at home. It can identify the MN by inspecting the packet's source address, in this case the Home Address.

The operations of this architecture are similar to those presented in the previous

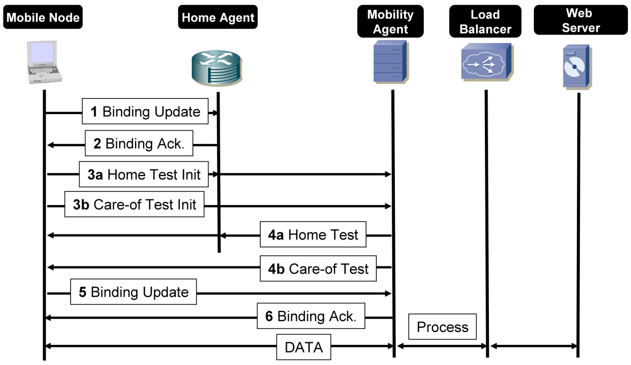


Figure 5.12: Mobility Agents interaction with RR (extended with a Load Balancing device)

section. As the figure 5.12 shows the Mobility Agent only forwards data packets to the load balancer.

5.6 Discussion

MAs are actually a Mobile IPv6 transparent proxy that runs the Return Routability procedure in a centralized fashion. This section presents the major benefits and drawbacks of this proposal.

- The MA's main advantage and goal is to avoid processing all the Return Routability signaling messages and the related cryptographic computations at the CNs. The solution is basically intended for large content providers where each CN may have thousands of simultaneous mobile clients. In addition it provides compatibility with some load balancing techniques.

- The MAs reduce the deployment cost. Mobile IPv6 requires specific support of the CNs to run the Return Routability procedure. This means that if a large content provider wants to support Mobile IPv6 clients it needs to modify the kernel of each of its servers. However with MAs a large content provider has just to upgrade its border router (or access router) with a MA implementation.

- The hardware of large servers may not be intended to process the signaling messages of the Return Routability procedure and its cryptographic operations. MAs can be implemented on a router that may have specific hardware to run these operations.

- A very interesting functionality of MAs is that they allow pre-established shared secrets with the MNs. The Return Routability procedure can be easily implemented using preconfigured shared keys. Standard Mobile IPv6 does not have this functionality because it is not feasible to distribute shared keys among all MNs and CNs on the Internet. However with MAs a large content provider can distribute shared secrets among its clients and implement its own Return Routability procedure. Another typical example scenario where this mechanism is applicable is within a corporation or between specific users. A Return Routability procedure based on symmetric keys [142] is much faster and securer than the traditional one. This Return Routability procedure requires just 2 direct messages (132 bytes) for each connection establishment or handover. This means that the Return Routability procedure based in shared keys reduces the handover latency to one round trip time and the signaling overhead to just 2 messages.

- When extended with a load balancer the Mobility Agents provide compatibility for the Packet Rewriting and the TCP Slicing load balancing techniques. While Distributed Web Systems are yet compatible, the Mobility Agents provide compatibility for the many existing Cluster-Based Web Systems. Table 5.2 and 5.3 present a classification of several products that provide Layer 4 and Layer 7 Load Balancers. Tables are based on [128; 133] and have been updated to reflect recent changes. As the table shows many commercial products use the Packet Rewriting technique and they can benefit from the proposed solution. Many other products use the Packet Forwarding technique, unfortunately this is a One-Way approach where the Mobility Agent cannot provide compatibility. Table 5.3 shows that TCP Splicing technique is also used by many major Layer 7 Load Balancers vendors.

5.7 Related Work

Several papers have presented solutions that run the Return Routability procedure on behalf the CNs. In [143] the authors present an agent-based route optimization

Table 5.2: Layer-4 Load Balancers

Packet Rewriting	**Packet Tunneling**	**Packet Layer-2 Forwarding**
Cisco's LocalDirector [148] LinuxVirtualServer [149] F5's BIG/IP [150] Foundry Networks ServerIron [151] IBM WebSphere Edge Server Network Dispatcher [152] Coyote Point Equalizer [153] Allot NetEnforcer [154]	LinuxVirtualServer [149]	IBM WebSphere Edge Server Network Dispatcher [152] Cisco's LocalDirector [148] LinuxVirtualServer [149] F5's BIG/IP [150] Foundry Networks ServerIron [151] IBM WebSphere Edge Server Network Dispatcher [152] Nortel Networks Application Switch [155] Radware's AppDirector [156]

for Mobile IPv4. In their proposal, a special entity located at the Correspondent Network border router achieves Route Optimization on behalf the CNs. Data packets are tunneled between that special entity and the MNs.

In [144] authors propose a bi-directional route optimization for Mobile IPv4. With the authors' solution, a special entity called Correspondent Agent is placed at the correspondent network border router. This entity also achieves Route Optimization on behalf the CNs. Another special entity (Foreign Agent) is placed at the visited network of the MN. The Correspondent Agent establishes a bi-directional tunnel with the Foreign Agent to send and receive data packets.

The NEMO protocol also introduces an entity called Correspondent Router that achieves Route Optimization on behalf the CNs. This entity was first proposed in [145]. Since NEMO has not yet a standard Return Routability procedure most of the research efforts have focused on this. Different papers such as [146; 147] propose different algorithms for the Return Routability procedure of NEMO. All the proposed solutions establish a tunnel between the Mobile Router and the Correspondent Router.

The MAs are intended for Mobile IPv6 instead of Mobile IPv4. Moreover, with this

Table 5.3: Layer-7 Load Balancers

TCP Gateway	**TCP Splicing**	**TCP Handoff**	**TCP Connection Hop**
IBM WebSphere Edge Server Network Dispatcher [152]	F5's BIG/IP [150] Foundry Networks Server-Iron [151] Nortel Networks Web OS [155] Radware's AppDirector [156] Lucent Web Switch [157] Cisco CSS [148] Zeus ZXTM-LB [158] IBM WebSphere Edge Server Network Dispatcher [154]	TCPSP [159]	Resonate's Central Dispatch [136]

solution packets are not tunneled but sent using the mobility extension headers. These headers provide less overhead (25%) than the traditional tunneling technique.

5.8 Summary and Contributions

This chapter describes the Mobility Agent architecture, a new entity located at the Correspondent Network that addresses potential issues II.B and II.D, that is high deployment cost of Mobile IPv6 and incompatibility with some load balancing techniques. We have focused the proposed architecture to Content Providers Networks since usually mobile nodes are clients that communicate with large servers. The following list summarizes the main contributions and conclusions of this solution:

- Incompatibility between load balancing techniques and Mobile IPv6's Return Routability arises due to the fact that the procedure is statefull while load balancing devices are stateless.

- A transparent proxy such as the Mobility Agents is a cost-effective solution for deploying Mobile IPv6 support at the content providers network solving both potential issues.

6

The flexible Home Agent Architecture

6.1 Introduction

This chapter explores novel Home Agent architectures that increase the overall reliability of Mobile IPv6-based networks. As the Potential Issue II.C shows, Home Agents represent a single-point-of-failure and are a potential bottleneck in such networks. In order to solve this issue we start our analysis by reviewing the state-of-the-art. Many research papers have been published addressing this problem. The solutions presented in [160; 161; 162; 163; 164] increase HA reliability by deploying several redundant HAs at the Home Link. In these solutions, all the HAs share the registration state and they define efficient mechanisms for HA recovery. These solutions reduce the service disruption time in front of Mobile IPv6. In addition, the MN's traffic is balanced among the different HAs. The main difference among them is that some [160; 161; 162] are MN-driven solutions while others [163; 164] are transparent to the MN.

Unfortunately, these proposals are focused on providing HA reliability and load balancing on just a single Home Link but they do not take into account the global requirements of an Autonomous System (AS). An AS that hosts MNs may have dozens of sub-networks. Deploying reliable HAs requires several redundant HAs on each link. The Mobile IPv6 protocol belongs to the IPv6 standard and, theoretically, any IPv6 node has mobility capabilities. Thus, these approaches are too expensive to deploy and to manage. A different proposal, which does not require deploying redundant

HAs on each Home Link, is the Virtual Mobility Control Domain protocol (VMCD) [165; 166]. The VMCD protocol allows multiple HAs to be placed at different domains. A MN may use multiple HAs simultaneously. The basic idea behind this proposal is that each HA advertises, through eBGP (exterior Border Gateway Protocol [167]), the same home network prefix from multiple routing domains. Each MN then picks the best HA according to its topological position. The main drawback for this proposal is that the impact on the exterior BGP routing system scalability is unpredictable.

As a summary, existing architecture effectively increase HA reliability, but they are very costly to deploy or they increase the load at the BGP subsystem. In addition, none of them reduce the load at the HA to avoid that it becomes a bottleneck. Taking this into consideration we base our solution on a different approach aiming to increase reliability and reducing the load. The HA can be seen as an entity that performs several differentiated operations. We have analyzed each operation and we have assigned each of them to an entity of the network. Our basic idea to distribute the operations is that a registration from a MN into a HA can be viewed as an internal route from the network's point of view. That is, when a MN registers a new location into its HA it is actually installing a new route (Home Address → Care-of Address). We believe that this route can be announced throughout the network and thus, it is not necessary to deploy a HA on each link. As we detail in further sections our solution only requires deploying one HA for the whole network. This HA should be reliable and our architecture allows deploying more than one HA to distribute the load. In addition, our solution reduces considerably the number of MN's data packets transmitted into the network and it is compatible with legacy MNs.

6.2 Home Agent Architecture

6.2.1 Design Rationale

This subsection analyzes the different operations of a Home Agent (HA) and how they can be distributed from a network's point of view. We define Exit Routers (ER) as the routers that connect the Home Network with the rest of the Internet. These ERs may or may not be the AS's border routers and an AS may have several Home Networks. Home Agents are responsible for maintaining bindings between the MN's identity and its location. The HAs forward the MN's signaling and the MN's data packets as well. MNs send data packets through their HA when communicating with their Home Network or with CNs. Since MNs can communicate directly with its CNs it is expected that communications through the HA are mainly used for short-term connections.

The Mobile IPv6 RFC states that packets sent through the HA may be secured through IPSec [138]. It should be taken into account that the MN can use IPSec with its peers regardless of the IPSec connection with their HA. We believe that it is not useful to secure MN to CN communications because the packets are only secured on half of the path (MN → HA) while the rest of the path (HA → CN) is not secured. Regarding the MN's communications with the Home Network, protecting the path is useful. In this case the HA is acting like a Virtual Private Network (VPN) gateway.

Under these assumptions and following the basic idea that a registration from a MN into a HA can be viewed as an internal route we can distribute the HA's operations throughout the network. In our architecture, a single HA is required for the whole network; we call it a flexible Home Agent (fHA). This fHA processes (using IPSec) the MN's signaling messages and maintains registration information. It also distributes this information throughout the network as internal routes. Hence, the network directly processes the MN's communications with the CNs while the fHA processes the MN's communications with the Home Network (using IPSec) in the same way as a VPN gateway.

6.2.2 Overview

Figure 6.1 presents an overview of the architecture. The proposal has only one HA (we call it a fHA) that serves all of the MNs of the network. This fHA is identified by an unicast address and the MNs addresses its registration messages to it. Upon reception of a registration message, the fHA validates it and sends a routing message announcing the new route towards the MN. This information is then sent to each ER. In addition, the fHA advertises the route to the Home Link's Access Router (AR). At this point, the network knows the location of the MN.

When communicating with a CN through the HA, MNs do not address packets to the fHA but to an anycast [26] address owned by the ERs. In this way, a given ER receives the MN's data packets and de-capsulate, lookup and forward packets to the CN. Similarly, CNs send packets to the MN's Home Address. Upon reception,

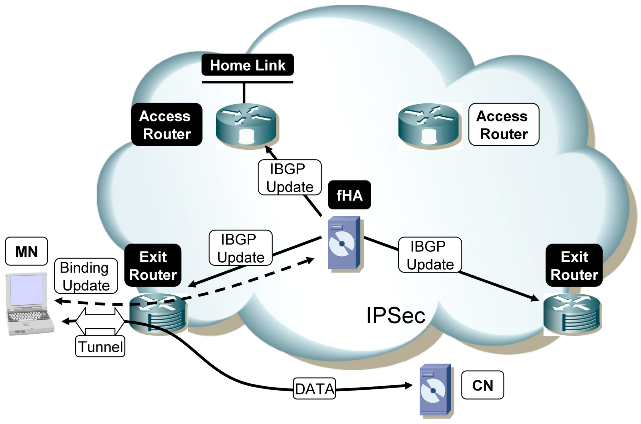


Figure 6.1: Overview of our proposal

the ER lookups the packet's destination address (the MN's Home Address). Since the fHA has previously installed a route at the ERs they know that the MN it is not at home. Therefore, the ERs encapsulate and forward the packet to the MN's location. The architecture manages efficiently MN to CN communications because some packets "bounce" at the ERs. This way the network's internal traffic is reduced considerably.

Regarding the communications from the MN to the Home Network, the MNs addresses its IPSec protected packets to the fHA that, in turn, de-capsulate and forward them to the MN's peer. The MN's peers address its data packets to the MN's Home Address. Since the fHA has announced to the Home Link's AR a route for the MN, the AR knows that the MN is away and it encapsulate the packet towards the fHA.

The Home Link's AR also multicasts Neighbor Advertisement messages on behalf the MN. This enables the AR to intercept communications from the Home Link to the MN and forwards them through the tunnel with the fHA. In the following subsections the detailed operations of the architecture are presented.

6.2.3 Dynamic fHA Architecture

This subsection specifies how the fHA announces their presence. In standard Mobile IPv6 HAs announce their presence through Router Advertisement messages. In this way, the MN's can automatically select a HA. Our architecture implements this functionality in exactly the same way that Mobility Anchor Points (MAP) announce their presence in the Hierarchical Mobile IPv6 protocol. The mechanism is also compatible with legacy MNs.

Each fHA sends Router Advertisement messages announcing its presence to the routers operating in the network. These messages include a preference value. In turn, the routers propagate the fHA's announcements to ARs that forwards them to the Home Link. Each router decrements the preference value. This way MNs can automatically discover their fHA's address and select the best one according to the preference value.

This mechanism has many benefits. On the one hand, it enables ARs to automatically discover the fHA thus avoiding manual configuration. On the other hand, it allows us to deploy more than one fHA on the network and distribute the load among them. The fHA's Router Advertisement messages include the prefix(es) of the Home Network that it is serving and the anycast address owned by the ERs. Including the Home Network's prefix enables the MNs to know if its peers are on the Home Network or not. Depending if the peer is on the Home Network or not MNs address the data packets to the fHA or to the ERs.

Finally, in order to provide compatibility with legacy Mobile IPv6 nodes, MNs may send its traffic to the fHA.

6.2.4 Signalling Processing

Each MN selects a given fHA through the above-mentioned mechanism. All the fHAes have pre-configured keys with the MNs as the Mobile IPv6 RFC states. Please note that ARs and ERs do not share any keys with the MNs. The fHAes receive registration messages from the MNs as stated by the Mobile IPv6 RFC.

Upon reception of a successful registration message, the fHA has to announce this information (route) to the ERs, to the Home Link's AR and to the rest of the fHAes. To distribute this type of information we use a routing protocol. Instead of designing a new routing protocol we use an already existing and deployed one. The routing protocol that best fulfills our requirements is the interior Border Gateway Protocol (IBGP) [167]. In the solution the fHAes, the ERs and Home Link's ARs create an IBGP domain. It is very important to remark that this IBGP domain may be an already existing IBGP domain or a separate one. The routes announced through this IBGP domain always have the longest prefix (/128) and *never* affect regular BGP routes. It should be noted that the routes announced by the fHAes are *never* distributed outside the network.

Finally, the entities participating in the IBGP domain have pre-configured keys to provide confidentiality, integrity and authentication to the communications.

For each successful received registration message, the fHAes send an IBGP UPDATE message to the ERs and to the AR responsible of the MN's Home Link. The fHAes are able to determine the appropriate AR by inspecting the MN's Home Address.

We introduce new options in the IBGP UPDATE message. The UPDATE message sent to ERs includes the following information: ⟨ *Home Address, Care-of Address, Lifetime* ⟩. Upon reception of this message, the ERs setup a tunnel endpoint with the MN. The tunnel source address is the anycast address while the destination address is the Care-of Address. In addition, each ER adds the following route to its routing table: ⟨ *HomeAddress/128* → *Tunnel* ⟩. The tunnel and the route are automatically deleted after Lifetime seconds.

The UPDATE message sent to the AR includes the following information: ⟨ *Home Address, Lifetime* ⟩. Upon reception of this message, the AR knows that the MN is away from home (note that the AR does not know the location of the MN). Next, the AR setups a tunnel endpoint towards the fHA that announced the route and adds the following route to its routing table: ⟨ *HomeAddress/128* → *Tunnel* ⟩. The AR also starts sending multicast Neighbor Advertisement messages on behalf of the MN at the Home Link. If a node of the Home Network (or Home Link) sends a packet to the MN, the AR intercepts it and encapsulates it towards the fHA. Once again, the tunnel and the route are automatically deleted after Lifetime seconds.

Once the MN returns home it sends a registration message to the fHA. Upon reception, the fHA sends an IBGP WITHDRAWAL message to the ERs and to the corresponding AR to immediately remove all the routes and tunnels related to the MN's Home Address.

6.2.5 Data Packets Processing

This subsection presents how packets are routed from/to the MNs.

MNs communicating with CNs encapsulate their data packets to the anycast address owned by the ERs (figure 6.2). The packets are received by the "nearest" ER that de-capsulates and forward them towards the packet's destination address (the CN's address). If the exit point of the CN's address is another ER then the packet traverses the network as a transit packet. It is important to remark that our solution does not

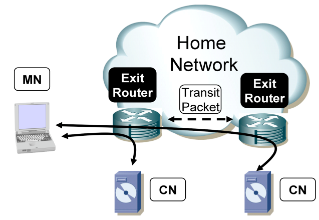


Figure 6.2: MNs to CNs communications

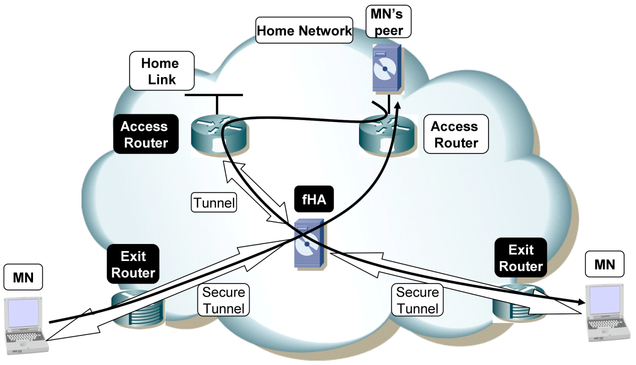


Figure 6.3: MNs to Home Network communications

require anycast routing. Packets addressed to the anycast address are routed normally (like unicast) and delivered to a given ER. We use anycast addresses because it is the standard procedure to assign the same address to different network interfaces.

MNs communicating with nodes located into their Home Network (figure 6.3) encapsulate their packets towards the fHA. However, packets sent by MN's peers are addressed to the MN's Home Address. The MN's AR intercepts those packets. Since the AR knows that the MN is away from home, it encapsulates the packet towards the fHA. Since the Mobile IPv6 RFC states that the packets are tunneled through the HA encapsulating the packet from the AR to the fHA does not affect the path's MTU. As has already been mentioned, the MN's communications with the Home Networks are protected with IPSec.

It is very important to remark that multihoming is only possible when BGP is used [168]. This means that a Home Network that is not running BGP will have just one exit router. Our proposal is flexible and works in both cases.

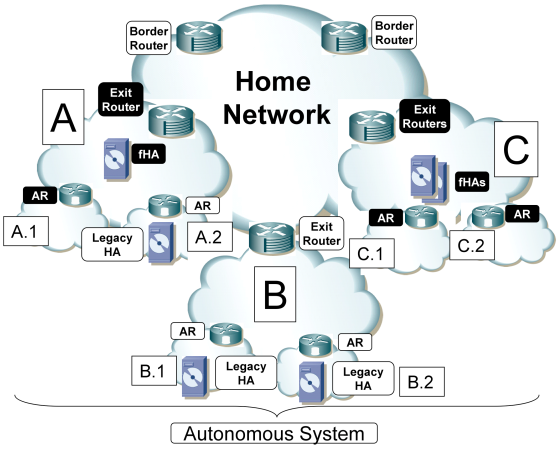


Figure 6.4: fHA location example

6.2.6 fHA Location

This subsection discusses the possible locations of the fHAes. Each fHA can be placed anywhere in the network, as a separate server, co-located with an ER/border router or even with a BGP Route Reflector.

One of the major benefits of this proposal is its flexibility. On the one hand, the architecture can serve all the MNs of a network with one or more fHAes. If more than one fHA is deployed MNs select the nearest one based on the preference value. This way the load is distributed among them. Each fHA thus only process signaling messages and communications from/to the Home Network (like a VPN gateway). MNs to CNs communications are then processed by ERs. On the other hand, the architecture is transparent to MNs running with legacy Home Agents and both technologies may co-exist on the same network.

Figure 6.4 shows an example of the flexibility. This AS has three networks and each one can independently select which approach it deploys. For instance, the A network can deploy both technologies. The fHA could serve MNs belonging to the A.1 sub-network while MNs belonging to the A.2 sub-network could be served by a legacy HA. The B network can deploy only legacy Home Agents on each sub-network. Finally, the C network can deploy two fHAes and all the MNs from C.1 and C.2 could be served by them.

Only routers labeled in black must belong to the IBGP domain with the fHAes of their network. There will be a separate IBGP domain for each Home Network. MNs served by an fHA send its data packets to an anycast address owned by the ERs. Since the prefix of the anycast address belongs to the Home Network's prefix, the AS's border routers knows how to forward the packets and do not need to be aware of our protocol.

6.3 Performance Evaluation

6.3.1 Analytical Evaluation

This section presents an analytical evaluation of the proposed scheme and a perfomance comparison with a network running Mobile IPv6 enhanced with existing solutions [160; 161; 162; 163; 164]. We do not consider solutions based on eBGP [165; 166] because their impact on the exterior BGP routing system scalability is unpredictable.

Signaling Overhead

Let N be the number of ER of a network that is running our proposal, let M be the number of deployed fHAes and let H be the total number of received registration messages per second (including foreign and home network registrations). Then our proposal requires sending $H \times (N + M)$ IBGP messages per second.

Transit Traffic Reduction

As has been commented previously, in this proposal some data packets bounce at the network's ER without being transmitted through the network. However in existing solutions [160; 161; 162; 163; 164] each packet sent through the HA has to be transmitted twice. One from the ER to the HA and another one in the opposite direction. In this subsection we compare this amount of transit traffic. We only consider the traffic exchanged between MNs and CNs that is routed through the HA.

Let I be the Kbps of traffic exchanged between all the MNs and its CNs through the HA. Then, existing solutions [160; 161; 162; 163; 164] have $2I$Kbps of transit traffic. If we assume that each ER of the network has the same probability of being the exit point of a given packet then, our proposal has $(1 - 1/N) \times I$ Kbps of transit traffic (figure 6.5).

In addition, transit traffic in existing solutions may follow a longer path than in our proposal. While in existing solutions [160; 161; 162; 163; 164] transit traffic must be

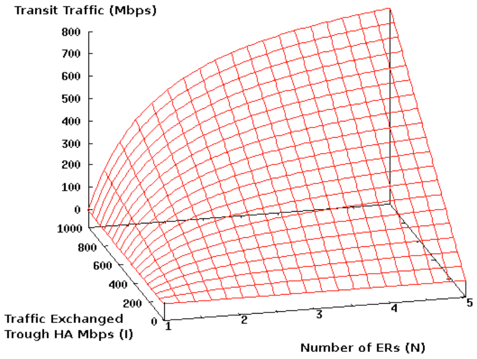


Figure 6.5: Transit Traffic in our proposal

transmitted to the HA in our proposal some transit traffic bounces at the ERs and the rest is transmitted from one ER to another. Home Links are usually deployed far away from the ERs while ERs may be close to one another (in terms of number of hops)

Stored State

This subsection analyzes the size of the routing tables and the number of tunnels configured at the ERs and ARs of a network running the proposal. Each ER has 1 tunnel and 1 route for each MN of its Home Network that is away from home. Each route and tunnel requires the Home Address, the Care-of Address and a lifetime, in total 34 bytes. Likewise, each AR will have just 1 tunnel with each fHA and 1 route for each of its nodes away from home.

6.3.2 Simulation Evaluation

In order to validate our proposal we have run a simulation. The simulation is intended to provide realistic values for the equations presented in section 6.3.1 and to compare our proposal with existing solutions [160; 161; 162; 163; 164].

In order to provide realistic values, we have configured a highly mobile environment by using a Random Trip mobility model [169]. Specifically, we have used the Random Waypoint on Generalized Domain model with a set of 8 domains. Each domain represents a layer-2 network where a MN can move without changing its point of attachment (i.e. default router). Only when the MN changes from one domain to another it must register its new location. Please, refer to [169] for further information.

Table 6.1: Simulation Results (Values in Mbps)

Traffic	**Existing Solutions** [160; 161; 162; 163; 164]	**fHA Architecture**
Traffic sent by MNs through the HA/fHA	1412.9 (465.04 to the HN, 947,86 to CNs)	
Traffic processed by HAs/fHAes	1412.9	465.04
Traffic processed by ERs	N/A	947.86
Transit Traffic	1895.72	473.93

The first domain is considered to be the Home Network while the rest of the domains are foreign networks. The Home Network has 1000 MNs, 2 ERs and 5 sub-networks. When running this proposal the Home Network has 2 fHAes while when running existing solutions [160; 161; 162; 163; 164] the Home Network has a set of reliable HAs on each sub-network (5 sets in total). In addition, each MN sends 64Kbps (VoIP) of unidirectional traffic towards its Home Network and 128 Kbps (Data) towards a CN. It should be taken into account that when a MN is at home traffic is sent directly and thus we do not consider it. Similarly, we do not take into account route optimized traffic. Finally, we have simulated this environment during 10000 seconds (roughly 2.7 hours).

The mobility model produces a mean of 4.68 foreign network registration messages per second (messages/s) and 0.80 Home Network registration messages/s. This means that our proposal requires sending 18.72 IBGP UPDATES messages/s and 3.2 IBGP WITHDRAWAL messages/s. Summarizing, our proposal introduces 21.92 signaling messages/s where each ER must process 5.48 messages/s.

Regarding the transit traffic table 6.1 presents the results. The fHAes have to process 465.04 Mbps. The simulated network has two fHAes and each one processes 232.52 Mbps of data traffic. In [160; 161; 162; 163; 164] the HAs process 1412.9 Mbps of traffic, the simulated network has 5 sets of HAs, this means that each set of HA processes 282.58 Mbps. In our work, the data traffic addressed towards CNs is directly processed by ERs (947.86 Mbps). Regarding the transit traffic, the proposal reduces it by 75% compared to existing solutions. It should be taken into account that existing solutions [160; 161; 162; 163; 164] must send each data packet twice, one from the ER to the HA and another one in the opposite direction.

Finally, during the simulation a maximum of 900 nodes were away from home at the same time (average 717, minimum 685). This means that the maximum stored state on each ER is 29.9KB.

6.4 Summary and Conclusions

In this chapter we have presented the flexible HA architecture, a solution that increases HA reliability and reduces its load (potential issue II.C). The main idea behind this proposal is to view a registration from a MN into a HA as a route that can be announced into the network. This way routers are aware of the location of the MN and can forward packets. As a summary the main conclusions of this chapter are:

- The fHA architecture increases reliability of Mobile IPv6-based networks. A failure in a fHA can be easily addressed since it supports efficient failure recovery mechanisms [160; 161; 162; 163; 164]. A failure on a ER does not disconnect the MN. In this case, the network announces the failure of the ER through the exterior routing protocol and the packets will be re-routed.

- The fHA architecture reduces the load of the HA. MN's data packets are processed by ERs or by a set of fHAes. Moreover it reduces the transit traffic thought the network (75% according to our simulation)

- The downside is that it requires adding some extra load at the ER. Nevertheless our simulation of a highly mobile environment shows that each ER would require processing only an average of 5.48 signaling messages per second.

7

The fP2P-HN Architecture

7.1 Introduction

This chapter addresses the lack of route optimization in Mobile IPv4 and NEMO clients (Potential Issue II.A). This means that communications between the MN (or MR) and its peers are routed through the HA. Unfortunately, packets routed through the HA follow a sub-optimal path. This reduces considerably the communications' performance, increases the delay and the infrastructure load. In addition, since a single HA may be serving several MNs and forwarding several connections, the HA itself may become the bottleneck of the whole system and represents a single point of failure in Mobile IP-based networks [64].

Again we start by analyzing the existing solutions to this particular issue. Basically the approach taken by researchers is to deploy multiple HAs at different Autonomous Systems (ASes) [166; 170; 171; 172]. The main idea behind this approach is that a MN may pick the best HA according to its topological position thus, reducing the delay of the paths towards its peers. The main challenge then is signaling the location of the different HAs throughout the Internet. Some of authors use the exterior Border Gateway Protocol (eBGP) protocol [166; 170; 171] while others [172] use Anycast routing. The main issue of these proposals is the scalability. On the one hand, using the exterior BGP protocol means increasing the load in the already oversized global routing table [173]. On the other hand, anycast's defiance of hierarchical aggregation makes the service hard to scale [174]. In addition, these solutions force the MNs to send the data packets through the HAs, increasing the load on these devices that may become the bottleneck of the whole system [64]. Later we provide a detailed related work on this particular topic.

A solution that provides route optimization for these mobile clients must be scalable and reduce the load at the HA. We propose using an overlay Peer-to-Peer (P2P) network to signal the location of the different HAs. When a MN detects that its current HA is too far it queries its *Original HA* (the one serving the MN's Home Network) that belongs to the fP2P-HN network for a closer HA. Then, the fP2P-HN network uses BGP information to locate a HA that reduces the delay of the paths between the MN and its peers, for instance by choosing a HA located in the same AS than the MN. Since security is one of the main concerns in mobility, we also present an architecture that provides trustworthiness to the HAs belonging to the P2P network and that allows that the MNs are authenticated by the HAs (and viceversa). In order to solve the second issue (i.e reduce the load at the HA) we benefit from the fHA architecture, detailed in the previous chapter.

The fP2P-HN architecture is simple, scalable and secure. Moreover it does not require deploying any new entities on the Internet. At the Inter-domain level, we signal the location of the HA using a P2P network instead of using eBGP or anycast. At the Intra-domain level we signal the location of the MN using IBGP, this way the Border Routers are aware of the location of the MN and the load of the HA is significantly reduced.

7.2 Flexible P2P Home Agent Network

In this section we detail the fP2P-HN architecture.

Overview

The main goals of the fP2P-HN architecture are to reduce the delay of the communications of the MNs and the load at the fHAes. Figure 7.1 shows an overview of the architecture.

When a Mobile IP or NEMO client changes its point of attachment to the Internet it establishes a new tunnel with its HA to communicate. Depending on the MN's topological position, this new path may have a large delay. We propose to deploy several HAs throughout the Internet in order to reduce this delay. When the MN detects that

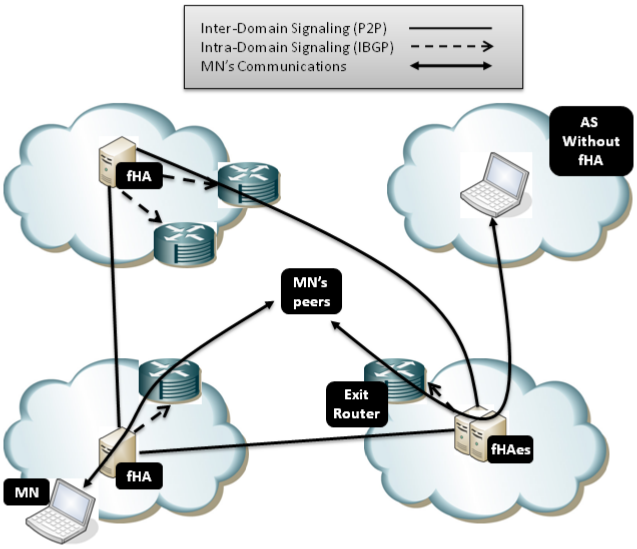


Figure 7.1: Overview of the fP2P-HN architecture

the new path towards its currently assigned HA has an unacceptable performance (e.g. RTT ≥ a given threshold) it queries its *Original* HA (the HA serving the MN's Home Network at the MN's administrative domain) for a closer one (i.e. an HA located in the MN's current AS). Our architecture is flexible and allows using any metric to trigger this procedure. We use the RTT because it is a simple metric able to capture the performance of a path. It is worth noting here that any other metric can be used.

Our proposal requires deploying several HAs throughout the Internet and has three differentiated phases. The HAs organize themselves in a P2P network which stores the information regarding their addresses and their topological position (HA's AS number). This P2P network is formed during the *P2P Setup phase*. The MNs are always bind to a HA belonging to this P2P network. Thus, when the MN detects that the RTT to its current HA is unacceptable it triggers the *fHA Discovery phase* and queries the P2P network for a closer one. Once the MN has the IP address of this closer HA it sends a registration message (Binding Update) and obtains a new HoA (*fHA Registration phase*). The MN keeps using this HoA while the RTT is below a given threshold.

All the HAs deployed in the fP2P-HN architecture are in fact flexible HAs. This means that they belong to the IBGP domain of its AS. When their assigned MNs are attached directly to their AS they act as a regular HA. However, when the MNs are outside their AS, they announce the location of the MNs (Care-of Address) through IBGP to the AS' Border Routers (BR). This is the last phase of the proposal known as *Data Packet Forwarding*.

P2P Setup Phase

This subsection details how the P2P network is created. The P2P network is used to store the location of the fHAes (AS number) and their IP addresses. This information is used by MNs to locate a closer fHA to its topological position.

fHAes organize themselves forming a structured P2P overlay (also known as DHT-based P2P overlay). The fP2P-HN is fully flexible and can be deployed using any of the proposed structured P2P schemes [175]. In the remainder of the paper we will consider Chord [176] as the P2P scheme, thus, the overlay's structure is a ring. In the fP2P-HN, the search key is the *AS-key* that is computed as *hash(AS number)*. When a new fHA joins the fP2P-HN it chooses an identifier (Peer-ID). In our case this is the *hash(fHA's IP Address)*. The fHA's position in the ring is determined by its *Peer-ID*: the fHA is placed between the two overlay nodes with the immediately higher and lower *Peer-ID* to its own id. Each overlay node has direct references to its two neighbors and also to other overlay nodes (crossing the ring) thus making the routing within the fP2P-HN faster. These nodes are named *fingers*. Each overlay node uses these fingers to create its fP2P-HN routing table. Finally, each fHA must register its AS number within the fP2P-HN. The fHA obtains the AS-key by computing the *hash(AS number)*. Then, it looks for the overlay node with the immediately higher *Peer-ID* to the *AS-key*, named *successor*, and sends to this node the *AS-key*, its IP address and its AS number. Moreover, the fHA sends some security information. The *successor* stores an entry with all this information.

fHA Discovery Phase (Inter-Domain)

This subsection details (figure 7.2) how a MN can use the fP2P-HN to discover a closer fHA. An MN connected to fHA_1 eventually detects (after a handover) that the RTT to fHA_1 is above a given threshold. Then, it triggers the procedure to discover a closer HA. The MN sends to its *Original fHA* a special BU soliciting the IP address of a closer fHA. At this point, the *Orginal fHA* discovers (using BGP) the AS number associated to the MN's CoA. Afterwards, it obtains the *AS-key* by computing the *hash(AS number)*.

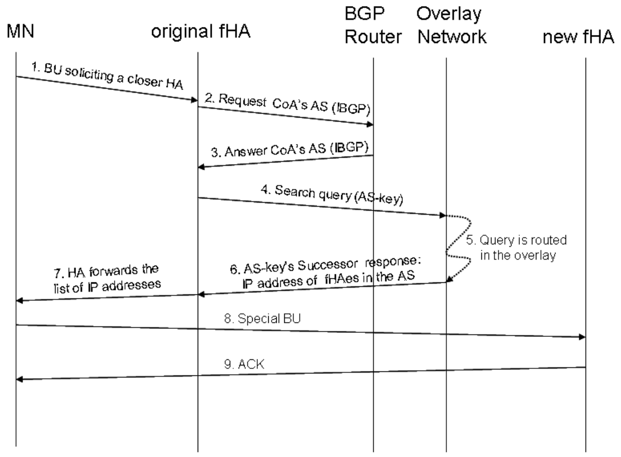

MN
original fHA
BGP Router
Overlay Network
new fHA
1. BU soliciting a closer HA
2. Request CoA's AS (IBGP)
3. Answer CoA's AS (IBGP)
4. Search query (AS-key)
5. Query is routed in the overlay
6. AS-key's Successor response: IP address of fHAes in the AS
7. HA forwards the list of IP addresses
8. Special BU
9. ACK

Figure 7.2: fHA Discovery Phase in the fP2P-HN architecture

The search method within the fP2P-HN is as follows. The *Original fHA* sends a query with the AS-key. The search query is routed in the overlay towards the AS-key's *Successor*. This fHA (e.g. fHA_2) is responsible of storing the information regarding the AS-key. Thus, it stores the IP addresses of all the fHAes located in the AS where the MN is currently attached to. Then, fHA_2 sends these IP addresses to the *Original fHA* which in turn forwards them to the MN. Finally, the MN selects one of them and sends a special BU message to the new fHA in order to obtain a new HoA.

Although the fHAes are expected to be very stable entities, the fP2P-HN includes the mechanisms to make the solution dynamic and adaptive. For this purpose, every fHA periodically checks if its neighbors and fingers are still reachable and running. If necessary, the fHA reconfigures its fP2P-HN routing table and establishes new neighbors or fingers.

Moreover, to make the solution more robust, reliable and load-balanced we use redundancy. Each AS-key is stored for several *successors* instead of just one. Then, in case of failure of a *successor* the others are still available and can reply to the queries. In addition, each MN has the list of the fHAes obtained during the last fHA discovery phase. Thus, if its current fHA fails, the MN can re-connect to one placed on the same AS.

fHA Registration Phase (Intra-Domain)

At the Intra-Domain level, each MN selects a given fHA through the above-mentioned mechanism. The fHA acts just as a regular HA when the MN is directly attached to its network. When the MN is attached to a different AS then the routers forward the MN's data packets.

Data Packet Forwarding Phase (Intra-Domain)

Finally MN's data packets are forwarded as detailed in the previous chapter (6), basically packets are processed by the routers if the MN is away from the fHA or by itself if it is attached to the same AS.

Final Remarks

In this subsection we discuss the final considerations of the fP2P-HN. First, changing the MN's HoA may break the existing connections. In order to solve this issue we propose that these connections are forwarded through the previous fHA while new connections are forwarded through the new fHA. A MN changes its HoA only when it is outside of its currently assigned fHA's AS and the RTT is above a given threshold. ASes usually provide connectivity to very large geographical areas, thus, this will occur rarely. In addition, 98% of the connections last less than 15 minutes [181], this means that very few connections may be affected. Regarding the inbound connections, the MN may still use its original HoA (the one from its Home Network). Thus MNs are always reachable through its regular Home Address. It is worth to note that MNs are clients (not servers) and with the current deployment of firewalls and NATs inbound connections are almost non-existent.

Finally the architecture requires minor modifications into the MNs and HAs. Obviously, the HAs must include an implementation of the fHA and the P2P algorithms. Regarding the MNs, they must include a triggering mechanism to discover a closer HA. As noted previously, this mechanism can use any metric, in our paper we have considered the RTT. In addition, the MNs must support multiples HoAs, this is already under standardization by the MEXT WG (see chapter 2 and [58]). The signaling between the MNs and the fP2P-HN can be accommodated into the Mobile IP signaling by exploiting the *Extensions* field present in the Binding Update messages (see [13] for details). Finally, the rest of the entities participating in the solution (CNs and routers) do not need to be modified. Since Mobile IP has not been deployed yet, we believe that the deployment cost of Mobile IP enhanced with the fP2P-HN does not increase.

7.3 Security Considerations

In Mobile IPv4 and NEMO, the mobile clients and the Home Agents are under the same administrative domain. That is why they are equipped with pre-configured keys. These keys provide, among others, two essential security properties to the mobile communications, trustworthiness and confidentiality. This means that the MNs and the HA can trust each other since they are authenticated. Additionally, ciphering techniques can protect the communications.

However, the MNs of the fP2P-HN may connect to different fHAes that, may or may not be under the same administrative domain. This section addresses the security at the fP2P-HN. Our goal is to achieve the same level of security than in Mobile IP and NEMO, that is: trustworthiness and confidentiality. In addition we also provide mechanisms to achieve a third security property, non-repudiation, only when it is required.

It must be considered that security solutions are highly dependent on the application scenario. In this section we analyze security in two potential fP2P-HN scenarios: (i) the fP2P-HN is deployed by an unique organization and (ii) the fP2P-HN is formed by fHAes belonging to different organizations, typically Internet Service Providers (ISPs). In both scenarios, we address the security of the two types of communications present in the proposed solution: fHA-fHA and fHA-MN communications.

7.3.1 Scenario I: fP2P-HN deployed by an unique organization

In the first scenario, all the fHAes are deployed by the same organization. Several approaches can be used in order to provide fHA-fHA trustworthiness. For instance, all the fHAes own a X.509 certificate [177] provided by the organization that authorizes them to use the fP2P-HN services. This certificate provides trustworthiness, because any fHA can require to another fHA its certificate in order to validate this second one as a legitimate entity. After being trusted, the fHAes involved in a communication can negotiate a shared key to provide confidentiality. This can be done by negotiating a session key based on Public/Private keys pair generated by each fHA (Public key could be also included along with the certificate provided by the organization). Finally, non-repudiation is achieved if each fHA is required to sign every data packet with its private key.

For fHA-MN communication, MNs are granted with a credential from the organization in charge of the fP2P-HN. This credential allows to identify uniquely a MN in the system and it could be provided in different ways: hardware device, SIM card, a user/password pair, a certificate, etc. Thus, in order to achieve trustworthiness, the MN obtains the fHA's certificate and the fHA request the credential to the MN. Again, confidentiality is obtained by negotiating a session key between the MN and the fHA. Finally, if non-repudiation is required, it is achieved if fHAes sign the data messages using their private keys and MNs include their credentials within the messages.

7.3.2 Scenario II: fP2P-HN deployed by several organizations

This second scenario requires more complex security mechanism because many different organizations are involved in the fP2P-HN deployment. Again, the most important requirements for the proposed solution are trustworthiness and confidentiality, but also non-repudiation is analyzed.

We propose using a trusted third party (TTP) in order to achieve these goals. This TTP is trusted by all the organizations participating within the fP2P-HN and thus, by all the fHAes belonging to these organizations.

In this scenario, the organizations that offer mobility services are typically the ISPs. In addition, an ISP is (usually) an AS within the Internet architecture. Thus, we assume that all the fHAes belonging to an AS are managed by a single ISP.

In this architecture, each ISP participating in the fP2P-HN is granted with an X.509 certificate obtained from the TTP. This certificate contains, among other elements: the AS Number, the AS public key (AS_pu_key) and the valid period. It must be taken into account that each ISP has an AS private key (AS_pr_key) paired with the AS_pu_key. Then, all the fHAes deployed in a given AS use that certificate within the fP2P-HN. Only fHAes belonging to an ISP participating in the fP2P-HN are provided with such certificate. Therefore, based on this approach, we are able to provide the required security properties in the fHA-fHA communications.

Trustworthiness is achieved because only fHAes owning such certificate (provided by the TTP) are trusted by the rest of fHAes within the fP2P-HN. Therefore, at any time a given fHA, fHA_1, could request to another fHA, fHA_2, its certificate to check whether fHA_2 is an authorized entity or not.

After both fHAes have trusted each other, they negotiate a shared key in order to provide confidentiality to the fHA-fHA communication. Several approaches could be applied at this point. For instance, the fHA_1 can provide a *nonce*$_1$[1] encrypted with the AS_pu_key2 to the fHA_2, and so does the fHA_2. Therefore, both peers create a shared key using the nonces as input parameters to a given function. For instance, *Shared Key = f(nonce*$_1$*,nonce*$_2$*) = nonce*$_1$ *XOR nonce*$_2$.

In order to secure the fHA-MN communications, we propose a similar approach to that used in GSM [178; 179; 180] that validates users owning a SIM card using a credential. In GSM, when an user is attached to a foreign operator (roaming), it has to present it credential to the new operator. Then, the new operator contacts the home operator and uses the received credentials to validate the user.

Following this approach, in the fP2P-HN the home AS (an ISP with the cerficate provided by the TTP) provides credentials to its MN clients. This credential could be: a certificate, an unique ID like in GSM networks, etc. Therefore, once a MN selects a new fHA from a different ISP, it presents its credential and its home AS number to the new fHA. In turn, the new fHA validates the MN by sending to one of the fHA in the MN's home AS the credential. Then, based on the received credential, the fHA in the home AS checks if the credential's owner is an authorized user and returns the validation result to the new fHA. If the validation is successful the new fHA can trust the MN.

Finally, each MN has a permanent trusted connection with its *Original fHA*. Thus, the MN also trusts the new fHA because it has been authenticated by its *Original fHA*. This means that the new fHA is trusted by the *Original fHA* and also by the MN. Therefore trustworthiness is achieved in both directions. After that, a shared key could be negotiatied between the fHA and the MN in order to provide confidentiality to the communications. Non-repudiation is achieved (if required) by applying the same mechanism introduced in the previous scenario.

[1] A nonce is a long random number

7.4 Performance Evaluation

The fP2P-HN architecture introduces two major improvements on Mobile IPv4 and NEMO which are: the reduction in the delay of the communications and the reduction in the load at the HAs. However, these improvements increase the signaling load in both, Intra (IBGP) and Inter-domain (P2P) levels. In order to evaluate the advantages (*reduction in the communication's delay* and *reduction in the load at the fHAes*) and the costs (*Inter-Domain Signaling* and *Intra-Domain Signaling*) we have implemented the fP2P-HN in a simulator.

7.4.1 Simulation Setup

In order to simulate the proposed solution we have used Internet-like topologies generated with the last version (3.0) of *Inet* [182]. An earlier version of this random topology generator was presented in [185]. We have chosen *Inet* as the topology generator because it has been designed based on the analysis of public NLANR (National Laboratory for Applied Network Research) data-traces [183]. These traces, well known by the passive measurements research community, have been collected from a variety of links at different networks. This means that *Inet* does not produce synthetic topologies, but realistic topologies based on real data-traces. In addition, *Inet* fulfills the requirements since it is intended to model AS-level connectivity instead of router-level connectivity. Regarding the mobility model, we have used the Random Waypoint Mobility simulator [169]. This simulator implements the well-known Random Trip Model [184] that was proposed as a generic mobility model. We refer the reader to [169] and [182] for further details.

Armed with a topology generator and a mobility model we have developed an ad-hoc simulator. Unless noted otherwise, we have simulated an average number of 100 mobile clients per fHA. The MNs are distributed randomly (uniformly) among the fHAes, this means that the fHAes do not necessarily serve the same amount of MNs. Each MN is assigned to a given Home Network (uniformly), the location if this Home Network is assigned randomly. For each handover, the MN has a 10% of probability of remaining into the same AS and, after a handover it remains attached to the same access router during a random amount of time distributed as (Gaussian) $N(5, 1)$ seconds. When the MN remains at the same AS, it means that it is changing its access router (CoA).

Obviously, these values produce highly mobile nodes compared to the movements in real environments, however we aim to evaluate our solution on a stressful scenario. Regarding the delays of the links, we consider that each link has a constant delay uniformly distributed as $U[10,25]$ms. Finally each MN sends 1 unit of bandwidth per second towards its Home Agent (for Mobile IP) and 1 unit towards its flexible Home Agent (for fP2P-HN). Since we aim to compare the load of both proposals a CBR data stream suffices. The MN's threshold to trigger the fHA discovery procedure is set to 75ms.

We run each simulation during 1000 seconds (simulation time) running fP2P-HN and Mobile IP/NEMO. We consider the following deployment scenarios $\{0.01, 0.1, 0.3, 0.6, 0.75, 0.9\}$. These numbers represent the probability of deploying one fHA for each AS. In the case of Mobile IP/NEMO, we consider the same amount of HAs and the same amount of MNs. Finally, we repeat the simulation of each deployment scenario 50 times with a different topology of 3500 ASes. The different topologies are generated using *Inet* (different seeds). In total, we have run 300 simulations. With this setup we simulate a wide range of scenarios, and we obtain the needed statistical information to assure the accuracy of the results. This accuracy is represented by the 90% Confidence Intervals included in every table and figure[1]. In order to run this huge amount of simulations we have used a cluster of 70 machines (Intel Xeon, 16Gb RAM) that uses Sun's N1 Grid Engine [186].

The graphics included in this section represent the Cumulative Distribution Function[2] (CDF) of the different evaluated aspects and also provides the Confidence Intervals of the calculated CDF. In order to obtain the CDF, first we compute the discrete probability density function (pdf) of the data. That is, we calculate the data distribution histogram. The histogram resolution (i.e. the width of the histogram intervals) was selected enough small to avoid information losses. Once we had the histogram, the CDF is the result of computing the histogram's cumulative sum. This process was repeated for each one of the 50 simulation samples. Thus, once we had the 50 CDFs we estimated the Confidence Interval for each one of the CDF points (that is, for each one of the histogram intervals). Since the histogram resolution is very high,

[1] In some figures the Confident Intervals are so narrow that appear as a point in the figure or are smaller than the symbol representing the point.

[2] In case of figure 7.4 the Complementary CDF is represented instead of the CDF.

the Confidence Intervals are not represented for every point since the figure would not be understandable.

7.4.2 Simulation Results

Reduction of the Communication's Delay

Firstly, we focus on the analysis of the communication's delay since this is the main issue of Mobile IP and NEMO. Figure 7.3 shows the delay of the communications in the path between the MN and its current HA, both for Mobile IPv4 and for the fP2P-HN. The figure presents the CDF of the average delay suffered by each MN. The results show that, for a very low deployment (1%), the fP2P-HN slightly outperforms Mobile IP/NEMO. However, increasing the deployment up to 10% the reduction of the delay achieved by the proposed solution is around 30%. This confirms, that even in the case of low deployments, our solution clearly outperforms Mobile IP or NEMO. Moreover, if we analyze the cases of higher deployments, fP2P-HN reduces the communication delay up to 6 times compared to Mobile IP or NEMO.

Table 7.1 summarizes the results on figure 7.3. It shows the mean MN-HA communication delay for both fP2P-HN and Mobile IPv4/NEMO.

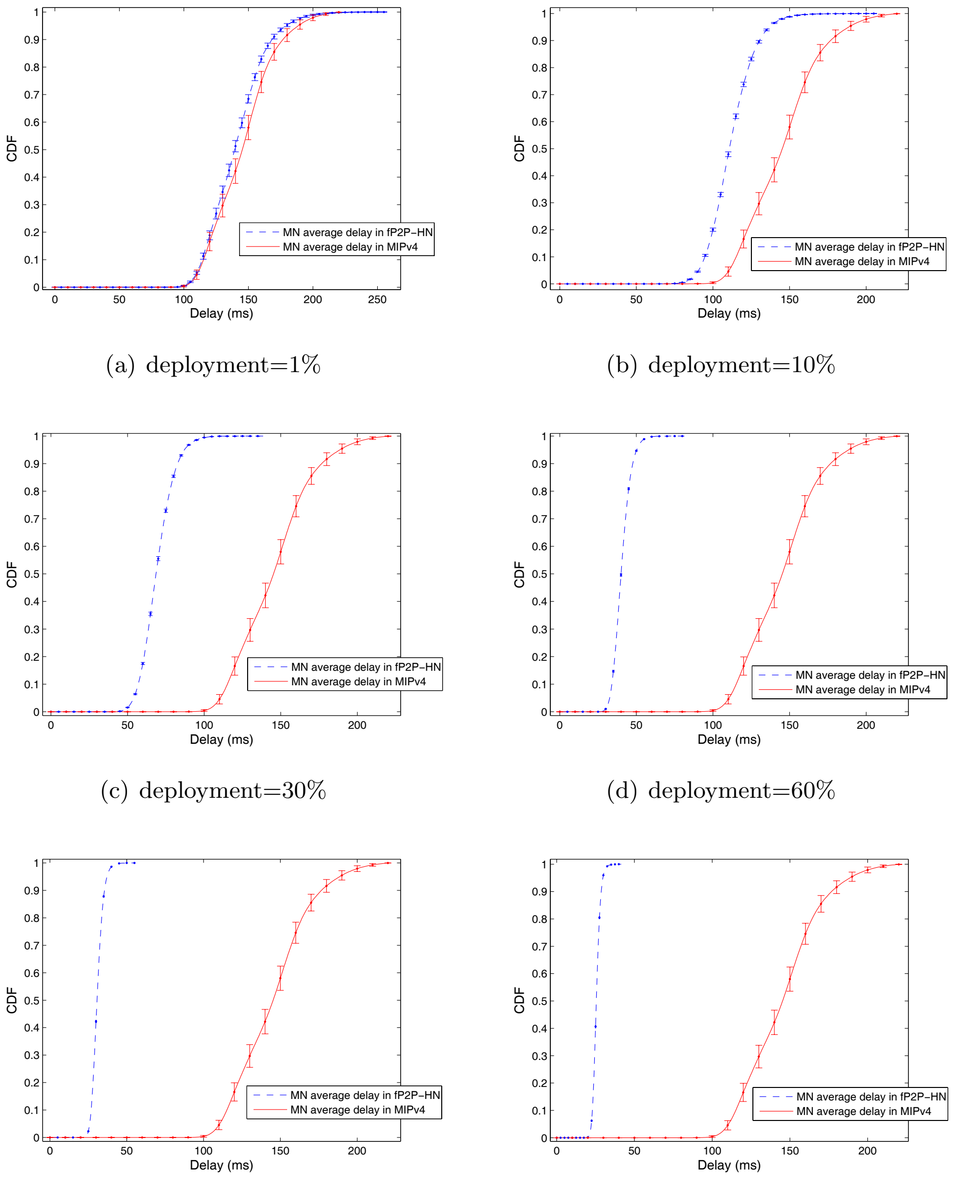
Table 7.1: Mean MN-HA communication's delay

Deployment	fP2P-HN (ms)	Mobile IP (ms)	Reduction of the delay (%)
0.01	140.86 ± 0.95	145.83 ± 0.29	3.41%
0.10	112.12 ± 0.31	145.83 ± 0.29	23.12%
0.3	69.63 ± 0.16	145.83 ± 0.29	52.25%
0.6	40.77 ± 0.07	145.83 ± 0.29	72.04%
0.75	31.22 ± 0.04	145.83 ± 0.29	78.59%
0.9	25.93 ± 0.03	145.83 ± 0.29	83.25%

Thus, we can conclude that in terms of delay, fP2P-HN introduces a major improvement compared to the Mobile IPv4 or NEMO solutions.

Reduction of the Load at the fHAes

In addition to the Route Optimization problem, the fP2P-HN addresses the reduction of the data traffic load at the HA as well. For this purpose we have introduced the concept



(a) deployment=1%

(b) deployment=10%

(c) deployment=30%

(d) deployment=60%

(e) deployment=75%

(f) deployment=90%

Figure 7.3: Average Communications Delay in the MN-HA path

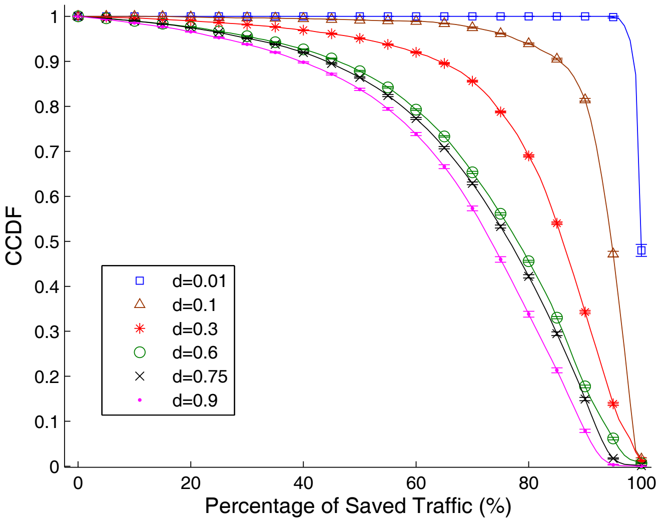


Figure 7.4: Percentage of fP2P-HN's Saved Data Traffic regarding MIPv4

of fHA. Figure 7.4 depicts the Complementary CDF (CCDF) of the percentage of saved traffic at the fHA compared to the regular Mobile IP's HA. The obtained results show that fP2P-HN introduces a major reduction of the load at the HA. The percentage of load reduction decreases along with the deployment. In the case of 1% of deployment we find out that around half of the fHAes are free of data traffic load. This means that they delegate the forwarding task to the Exit Routers. Even considering large deployments (d=0.9), 80% of the fHAes experiment a load reduction larger than 50%.

Table 7.2 shows the mean values. It must be noted that even in the worst case (d = 0.9) the mean load reduction with the fP2P-HN is the 54.56%.

The reader may wonder why the percentage of saved traffic decreases as the deployment increases. This is because the fHAes delegates the forwarding of traffic from/to the MN when this is not directly attached to the fHA's AS. Whereas, if the MN is attached to its fHA's AS, then the fHA is the responsible of forwarding the traffic from/to the MN. Hence if we consider a large deployment of fHAes, it is more likely that the MNs are attached to its current fHA's AS making the fHA suffers from higher load. On the other hand, in case of low deployments, the probability that the MN finds a fHA into its current AS is lower. Then, the MN maintains the connection to the fHA located in a different AS which delegates the forwarding task to the Border Routers.

Thus the fHA's load is lower with low deployments.

In a nutshell, as higher is the deployment as higher the probability that a MN uses a fHA placed at its current AS, thus more data traffic is forwarded by the fHAes.

Table 7.2: Mean load reduction at the fHA compared to Mobile IP

Deployment	Load Reduction (%)
0.01	99.31 ± 0.02
0.10	92.72 ± 0.03
0.3	78.94 ± 0.06
0.6	64.81 ± 0.04
0.75	59.35 ± 0.02
0.9	54.56 ± 0.72

Inter-Domain Signaling

As it has been explained above, existing solutions addressing the problem of Route Optimization for Mobile IPv4 and NEMO are not scalable. However the fP2P-HN uses P2P (an scalable technology) in order to signal the location of the HAs. In this section we evaluate the number of Inter-domain (P2P) signaling messages required to run the fP2P-HN. Figure 7.5 shows the inter-domain (P2P) signaling generated by the fP2P-HN to signal the location of the different fHAes. This figure depicts the CDF of the number of inter-domain signaling messages per second (sent + received) that a fHA has to support in the fP2P-HN. We can observe that the signaling overload introduced by the fP2P-HN remains between 50 and 100 messages/s for all the analyzed deployments. Therefore, the fP2P-HN requires a low number of Inter-domain signaling messages. Moreover it must be considered that these messages are usually short messages, thus the bandwidth consumption is negligible. For instance if we consider the worst case of the figure (50 sent + 50 received messages per second) and we assume that each message has 50 bytes (a Mobile IPv4's Binding Update message has 44 bytes, see [13]); then the amount of signaling traffic that a fHA has to support in the fP2P-HN is 20 kbps (both uplink and downlink).

Table 7.3 presents the mean number of total messages/s supported by the fHA. Again it is worth to analyze the signaling overload as function of the deployment. The reader can observe that the overload increases as the deployment goes from 1% to 10%,

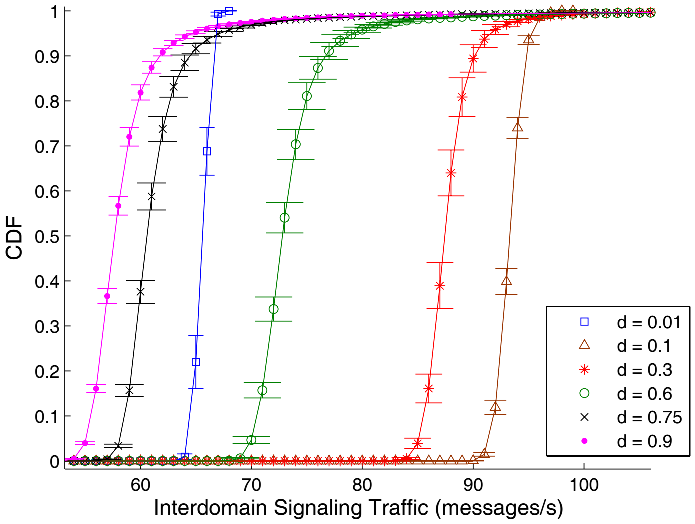


Figure 7.5: fP2P-HN Inter-Domain signaling traffic

and from this point it decreases along with the deployment increment. There are two parameters affecting the inter-domain signaling: the number of fHAes forming the fP2P-HN and the number of *special BUs* soliciting a new fHA (fHA discovery procedure). The number of fHAes affects since the fHA discovery procedure takes place at the overlay level and the query is routed by several fHAes within the fP2P-HN. The number of fHAes routing each query is bounded by $O(log_2(N))$ [175] (where N is the number of fHAes forming the fP2P-HN). Thus as larger is deployment (larger N) as more fHAes are involved routing each query. On the other hand the number of *special BUs* gets reduced as the deployment increases. With large deployments is expected that MNs will be always connected to very close fHAes and that the fHA discovery proccess will be rarely unsuccessful. Therefore, both parameters compensates each other. Thus when the deployment increases from 1% to 10%, the increment on the number of fHAes weighs more than the increment of the number of *special BUs* and the signaling load grows. For larger deployments the situation is reversed resulting in a signaling load reduction.

In order to further study this behavior let's consider table 7.4. This table details the probability of triggering the fHA discovery procedure for each deployment scenario (the values have been collected from the simulations). As the table shows, when the

Table 7.3: Mean Number of interdomain signaling messages/s per fHA

Deployment	Number of fHAes	Mean Number of (sent + received) messages/s
0.01	35	66.77 ± 0.14
0.10	350	94.46 ± 0.16
0.3	1050	89.23 ± 0.44
0.6	2100	75.21 ± 0.60
0.75	2625	63.32 ± 0.50
0.9	3100	67.63 ± 8.99

deployment is low, the MNs initiate the fHA discovery procedure more often. This is because MNs detect that the RTT is above a given threshold, ask for a closer fHA, but, since deployment is low, do not find one. Hence the probability of triggering the fHA discovery procedure decreases as the deployment increases.

Table 7.4: Probability of triggering the fHA discovery procedure

Deployment	Probability
0.01	0.73
0.10	0.64
0.3	0.47
0.6	0.35
0.75	0.29
0.9	0.27

Finally, we can conclude that the fP2P-HN is scalable. Considering a highly mobile simulation scenario and 100 MNs per fHA, the amount of signaling messages in the worst case is 20kbps. On the other hand, table 7.3 shows that the amount of signaling messages is irrespective of the number of deployed fHAes. In fact independently of the deployment the overload values are within the same order of magnitude (hundreds). Hence, the inter-domain cost of the proposed solution is $O(1)$.

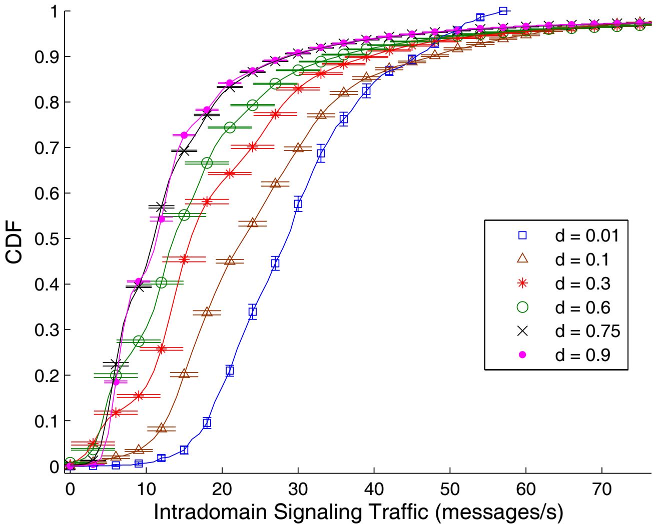


Figure 7.6: fP2P-HN Intra-domain signaling traffic

Intra-Domain Signaling

Finally we analyze the Intra-Domain signaling. This signaling includes the IBGP (UPDATE and WITHDRAWN) messages sent to the *Exit Routers* and the BGP queries sent to discover the MN's AS (see steps 2 and 3 in figure 7.2). This overload must be supported within each AS. Figure 7.6 shows the CDF of the amount of signaling per AS (per second), considering the different deployment scenarios. As the figure shows the number of signaling messages is bounded between 0 and 70 (sent + received) messages/s. Again, considering a message size of 50 bytes, the download/upload rate is inferior to 15 kbps. Additionally it has to be taken into account that this number is the total amount of signaling traffic supported inside each AS. Since the fP2P-HN allows deploying multiple fHAes within an AS (see section 6.2.6) each fHAes should only process a part of this signaling overload.

Regarding the mean values, Table 7.5 shows the results. The Intra-Domain signaling decreases as the deployment decreases. This is an expected result, since when MNs are directly attached to its fHAes no IBGP signaling is produced.

Table 7.5: Mean Number of intradomain signaling messages/s per AS

Deployment	Average Number of (sent + received) Messages/s
0.01	49.60 ± 0.03
0.10	45.96 ± 0.05
0.3	39.00 ± 0.09
0.6	32.57 ± 0.11
0.75	30.21 ± 0.12
0.9	29.24 ± 1.00

Summary of the obtained results

This section has evaluated the advantages and costs introduced by the fP2P-HN in front of the standard Mobile IPv4/NEMO protocols. The conclusion is that the fP2P-HN solves the main drawbacks of Mobile IPv4/NEMO (communication's delay and HA overload) with a low cost, some dozens of kbps in terms of extra signaling traffic. The obtained improvement depends on the deployment of the fP2P-HN. Figure 7.7 summarize in a single graphic the improvements (load reduction and communication delay reduction) introduced by the fP2P-HN as function of the deployment. This figure allows us to determine the required deployment in order to achieve a given performance. For instance if we aim to reduce both the communication's delay and the load at the HA over 60% then we should have a fHA deployment between 45% and 65%. Finally large deployments improve the communication's delays while low deployments improve the reduction of the load at the fHAes.

7.5 Related Work

Incorporating route optimization to Mobile IPv4 and NEMO clients is a key issue when considering the deployment of a truly Mobile Internet. That's why this topic has attracted the attention of the research community and many solutions have been proposed.

As we have mentioned in the previous chapter, first the research community focused on solving this problem specifically for Mobile IPv4 [143; 144] and NEMO clients

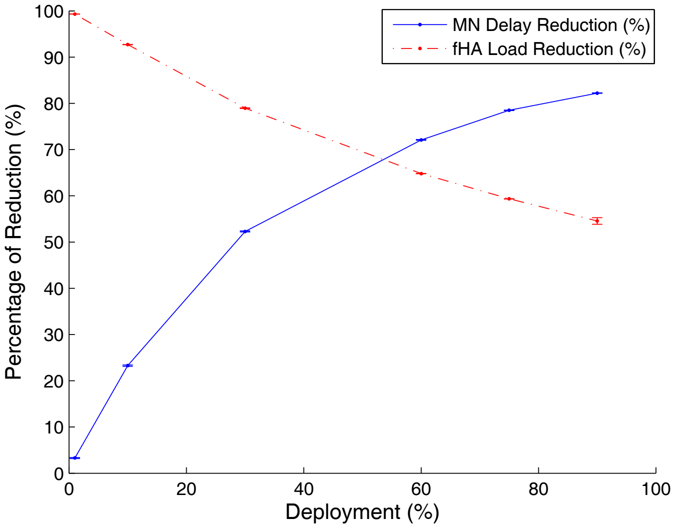


Figure 7.7: Reduction of the communication's delay and fHA's load

[146; 147]. The main idea behind these proposals is to deploy a new entity at the correspondent network that helps the MN to communicate directly with the CN. Usually this new entity authenticates the location (CoA) and the identity (HoA) of the MN. In addition this device acts as a tunnel endpoint, this way the MN can send the packets tunneled directly to the correspondent network. The main drawback of all these proposals is that they require deploying a new entity on each correspondent network. In the current Internet status this would imply deploying a new entity on each network or at least, on each AS (currently there are roughly 22.000 ASes on the Internet). That's why we believe that the deployment cost of these solutions.

As we mentioned in section 7.1, *R. Wakikawa* presented recently a different approach [166] used by other researchers [170; 171; 172]. Since these proposals are not scalable [173; 174] we propose using a P2P network that it is fully scalable and we benefit from the fHA that reduces the load at the HAs significantly.

7.6 Summary and Conclusions

In this chapter we have addressed the lack of route optimization in Mobile IPv4 and NEMO clients. We provide a scalable solution that also reduces the load at the Home

Agents: the fP2P-HN architecture. This solution uses a P2P network to signal the location of the HAes and benefits from the fHAes. The main conclusions that can be extracted from the results are:

- The fP2P-HN effectively reduces the delay of the mobile nodes compared to Mobile IPv4/NEMO. Even with low deployments (0.1) the reduction is 23%. As the deployment grows so does the reduction that can be up to 83% (0.9).
- Our architecture reduces the traffic processed by each flexible Home Agent compared to that of Mobile IPv4/NEMO. As expected, the reduction of the traffic decreases as the deployment increases. In the worst case the reduction of the traffic processed by a flexible Home Agent is 54% (0.9). This reduction grows further to 99% (0.01).
- Our architecture is highly scalable since the amount of Inter-Domain signaling is within the same order of magnitude (hundreds) and irrespective of the number of flexible Home Agents deployed, thus, the cost is $O(1)$. Additionally the amount of Inter-Domain signaling traffic per flexible Home Agent is around 20kbps.
- The extra Intra-Domain signaling of the fP2P-HN is very low, around 15kbps per Autonomous System. Since the architecture allows that multiple flexible Home Agents are deployed within an Autonomous System this overload may be shared between several entities.

Part IV

Future: New Architectures

8

Fundamentals of Bandwidth Estimation in Wireless Networks

8.1 Introduction

The subject of available bandwidth estimation has received much attention in the last decade. As a result of extensive research efforts, multiple techniques and tools have been developed to estimate different bandwidth related metrics from network measurements [93; 94; 95; 96; 97; 98; 99; 100; 101]. Most of these techniques are based on periodic probing processes. However recent studies [102; 103] have shown that these tools may fail when applied to wireless networks. As it has been shown in chapter 2, this is a key metric when considering the performance of multihomed mobile architectures.

The main reason of the failure of such techniques in wireless links is because they are based on the concept of a single bit-carrier multiplexing several users in FIFO order. Additionally, they assume that the transmission link presents a constant service rate along the measurement phase. Further, these techniques commonly assume that the impact of low-layer overheads can be neglected and measurements taken with a given packet size can be easily extended to other packet sizes.

However these assumptions do not hold any longer in the presence of wireless links. First, multiple-user access schemes compromise the FIFO assumption [102]. Second, the service rate may change along the measurement process [208]. Finally, previous studies [102; 103] have shown that bandwidth metrics of a wireless link cannot be easily normalized and that measurements have to take into account the packet size used.

The main objective of this chapter is to revisit some of the principles governing periodic probing processes in the presence of wireless links. Specifically we deal with the problem of estimating bandwidth metrics in a single-hop wireless path using the IEEE 802.11 standard as a reference.

We begin the study of the bandwidth estimation problem using a fluid approximation of the WLAN system. The fluid approximation assumes that packets have an infinitely small size and it is taken over cross-traffic, probe-traffic and the service delay that the WLAN system offers to the probing flow. This analysis reveals how periodic-based bandwidth measurement models [93; 94; 95; 96; 97; 98; 99; 100; 101] target the achievable throughput rather than the capacity or the available bandwidth in a WLAN system (as previously assumed in [102]). The main reason behind this result is the non-FIFO scheduling properties of the WLAN access protocol.

Next, we relax the fluid approximation and present a non-fluid analysis of bandwidth measurements over WLAN scenarios. This study reveals how the interaction between probing traffic and cross-traffic presents a transitory stage. Specifically, it shows that the first packets of a periodic probing sequence experience, in average, a lower delay than the rest to get service. As a consequence, these packets constitute erroneous samples of the stationary regime and introduce biases in bandwidth measurements.

In its last section the chapter describes some important implications of these results. On one side it shows how the packet-pair technique [218], extensively used in recent wireless routing literature [217], provides optimistic measurements of the achievable throughput instead of the capacity (as previously assumed). On the other side the chapter shows how removing the first packets of a probing sequence when measuring bandwidth is an effective means to improve the accuracy and/or reduce the intrusiveness of bandwidth measurement tools.

8.2 Validation Setup

The study presented in this chapter is based on theoretical analysis, simulation and experimentation. This section introduces the simulation and experimentation setups used to gather measurement data and validate theoretical findings.

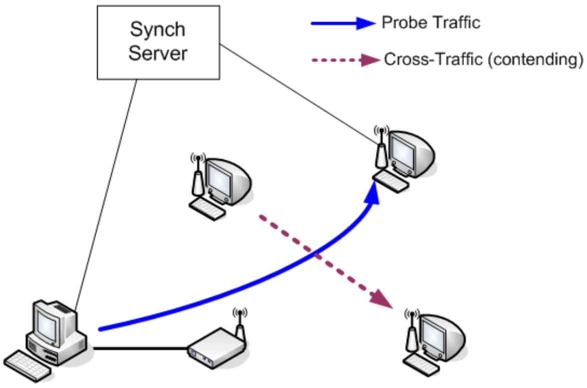


Figure 8.1: Experimental/simulation scenario

Experimentation have been carried out within the EXTREME framework (see [210]). This is a multi-purpose networking experimental platform developed at the Centre Tecnòlogic de Telecomunicacions de Catalunya (CTTC). The main advantage of this platform is its high automation capabilities that allow automatic execution, data collection and data processing of several repetitions of an experiment.

The WLAN devices used are Z-COM ZDC XI-626 cards which carry the popular Prism chipset. These wireless devices are controlled using computer nodes of the EXTREME cluster. In all cases these nodes are Pentium IV PCs with a 3GHz processor, 512MB of RAM memory and running Linux OS, with kernel 2.4.26. To control these devices, the EXTREME automation system makes use of the wireless extensions API.

In order to generate the traffic (probing and cross-traffic), we make use of the Multi-GENerator toolset [211]. However, in order to increase the accuracy of the time-stamping procedure, both at sender and receiver sides, network device drivers have been conveniently modified to timestamp packets just before they are laid down to the hardware (sending side) and just after getting them from the hardware (receiving side). This follows some of the ideas described in [212].

Figure 8.1 shows the basic setup used throughout the chapter for experimentation. The probing traffic is sent between two stations that are conveniently synchronized. This synchronization is achieved by sending frequent NTP updates through a parallel wired interface between the NTP server and the measurement nodes. Using this method we achieve accuracies of delay measurement in the order of ten microseconds.

Both types of cross-traffic considered in this study can be generated separately or at the same time depending on the case. The cross-traffic generated follows the Poisson distribution in order to provide variability to the scenario conditions.

Some of the experiments required a large amount of repetitions to achieve accurate convergence of results. Since this is difficult to achieve in a testbed we have also used a simulator. Specifically we have replicated the tesbed (figure 8.1) using NS2 (ver. 2.29 [213]). The main difference between the testbed and the simulator is that the latter includes scenarios with up to 5 contending nodes.

The simulator has been patched with the ”Speeding up Scheduler” in order to increase its performance. This new scheduler will be included into the next release of NS2. We use the NO Ad-Hoc Routing Agent (NOAH). This agent, a static routing agent, only supports direct communication between wireless nodes. NOAH does not send any routing related packets and thus, it does not interfere with probe or cross-traffic.

Regarding the configuration, all the experiments use the default MAC and PHY 802.11 layers included into the NS2 package. The radio propagation model is the TwoRayGround. This model is used when line-of-sight path exists and reflection of ground is considered, however it does not take into account fading effects such as the *Shadowing* or the *Ricean* models (also available in NS2). We left as future work the evaluation of W-Path's accuracy under other models, this could reveal interesting relations between the relation of the variability of the PHY and the MAC layer with that of the available bandwidth (see section 9.7 for further details). All the layer-2 queues are PriQueues, this means that they priorize signaling packets. The queues used are infinite, this way we avoid dealing with packet losses, which are irrelevant for our study. Finally all the wireless nodes are static and equally spaced from the Access Point. The physical transmission rate is set to 11Mbps and RTS/CLR are not used.

Finally, we have also developed a queuing simulator using Matlab. The motivation for this is that the probing process in a WLAN presents multiple components that are difficult to isolate from each other in an experimentation setting or even through simulations. The simulator convolves a series of packet arrivals with a series of service times in order to measure several metrics such as the queuing length distribution and the output dispersion (inter-arrival) of packets. The input parameters are gathered from experimentation measurements in order to keep the results as close to the real behavior as possible.

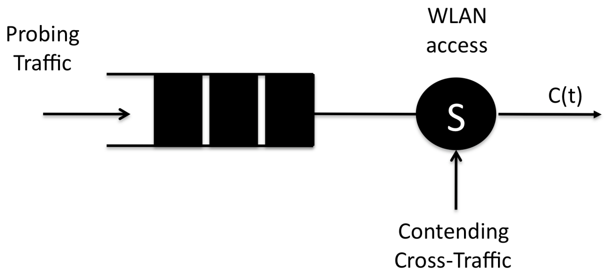


Figure 8.2: Model of the interaction between probing traffic and (contending) cross-traffic in a WLAN system

Unless noted otherwise the results presented in this work have been obtained from repeating experiments over 80 times while the simulations have been repeated 20.000 (NS2) to 70.000 (Matlab) times.

8.3 Bandwidth Measurements over WLAN

Measuring bandwidth related metrics in paths that include wireless transmissions has shown to be a difficult task [101; 102]. First this section introduces a model of the wireless link used in the chapter. Further it reviews bandwidth metrics in the presence of such links and presents results of bandwidth measurements when taking a fluid approximation of the model.

8.3.1 Model of a WLAN link for bandwidth measurement

Active bandwidth measurement techniques consist in sending a sequence of probing packets through a targeted path and inferring bandwidth metrics by observing the characteristics of the sequence at the reception side. Among others, the characteristics of the outgoing flow depend on the specific interaction between the probing traffic and the cross-traffic present in the path under measurement.

When considering a WLAN link, cross-traffic contends for the wireless medium with the probing stream. This is the case for example when two flows contend for channel access in the uplink towards an AP, or when two stations contend for channel access in an ad-hoc setting.

Figure 8.2 presents the model used all along the chapter. First probing packets enter a transmission queue. Once at the head of the transmission queue, probing packets contend for channel access with cross-traffic packets. As the figure shows and without loss of generality we model the contending cross-traffic as a process that affects the service time of a sequence of probing packets. In other words, probing packets enter a FIFO queue and, once at the head of the queue, suffer a random delay before being completely transmitted.

This model is non-parametric, hence it does not make any assumption about the distribution of the service delay or the cross-traffic.

8.3.2 Revisiting bandwidth metrics

As recent literature has shown, randomness in wireless transmissions has a direct impact on the bandwidth metrics traditionally considered. In some cases such metrics require more accurate definitions [102] and in some cases new metrics have been defined to account for the particularities of wireless environments [101].

Traditional metrics associated to bandwidth measurements are capacity and available bandwidth. On one side, the capacity, as defined in [101], is the maximum possible layer-3 (IP) transfer rate at a given network hop or path. However in a wireless environment, this is a time dependent random process $C(t)$. On the other side, the available bandwidth refers to the portion of the capacity that is not being used: $A(t)$. Capacity and available bandwidth depend on factors such as the variability of the wireless channel, cross-traffic intensity and the packet size used.

Recent literature on bandwidth measurement over wireless systems raised some debate around the measurement of available bandwidth in WLAN scenarios. In [102] the authors show how traditional techniques fail in measuring such metric in wireless settings. Following such debate we use the achievable throughput metric. As shown along the work, this metric better fits bandwidth measurement results in wireless environments. The following equation provides a specifi definition of the achievable throughput,

$$B = sup\{r_i : \frac{E[r_o]}{r_i} \geq 1\} \tag{8.1}$$

In this expression, r_i is the rate at which a traffic flow enters the path under measurement and r_o is the rate at which this traffic leaves the path. The achievable throughput is also a time varying metric $B(t)$ that depends on the specific characteristics of cross-traffic, channel access scheduling and channel propagation. Note that under these definitions the relation between the metrics is $A(t) \leq B(t) \leq C(t)$. This model assumes that, during the measurement interval, the capacity, the available bandwidth and the achievable throughput are stationary random processes with asymptotic averages $\overline{C}$,$\overline{A}$ and $\overline{B}$ respectively.

8.3.3 Bandwidth measurements under fluid assumptions

In order to get a basic understanding about bandwidth measurement in WLAN environments we place fluid assumptions on the model presented above and review the concept of the rate response curve [214]. The rate response curve relates the rate of a probing flow at the output of a network path (r_o) with the rate at which it entered the path (r_i). Fluid assumptions taken over a wireless link apply to the cross-traffic and to the service rate. Both processes lose their random and time dependent properties under this assumption and become constant over the measurement interval. As a consequence under fluid assumptions we have that during the entire measurement interval $A(t) = \overline{A}$,$B(t) = \overline{B}$ and $C(t) = \overline{C}$.

Recalling from [214], the fluid rate response curve of a FIFO queue with constant service rate (i.e. the probing and cross-traffic share a FIFO queue) can be expressed as,

$$r_o = min(r_i, C\frac{r_i}{r_i + C - A}) \tag{8.2}$$

Now let us consider a case when probing packets contend for (wireless) channel access with the cross-traffic instead of sharing a FIFO queue, such as figure 8.2 depicts. Figure 8.3 plots an experimentation result showing the evolution of the rate response curve when measured with one contending station. In order to obtain the fluid rate response curve we use very long packet probing trains (10000 packets). The figure shows also the evolution of the cross-traffic throughput for each probing rate. As it can be seen, when the cross-traffic starts experiencing a decrease in its throughput, that is, when the probing traffic arrives at the available bandwidth (∼2Mbps), the rate

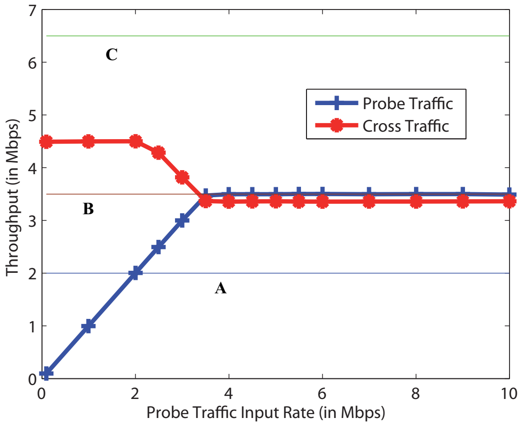


Figure 8.3: Experimental fluid rate response curve of probe traffic in a WLAN setting versus throughput of cross-traffic flow. C=6.5Mbps, A=2Mbps, B=3.4Mbps

response curve shows no flattens (as one would expect from eq. 8.2). Instead, the rate response curve deviates when reaching the fair share (∼3.5Mbps) that is gets from the wireless medium. This fair share corresponds, in fact, to the achievable throughput metric defined above.

This observation leads to reformulating eq. 8.2 for a wireless link:

$$r_o = min(r_i, B) \tag{8.3}$$

Equation 8.3 is the first conclusion of this work. The rate-response curve of a periodic probing process in a wireless link deviates at the achievable throughput. Therefore, existing tools based on this curve [93; 94; 95; 96; 97; 98; 99; 100; 101] measure this metric rather than the available bandwidth. In fact the study presented in [102] evaluated the accuracy of such tools in a wireless link assuming that they targeted the available bandwidth, hence the conclusion was that the accuracy was very poor. These results should be revisited taking into account this equation.

8.4 Non-fluid Analysis of the dispersion-based measurements in WLANs

Recent studies [214; 216] have taken a non-fluid approach to the bandwidth measurement problem. They reveal that dispersion based measurements of the available

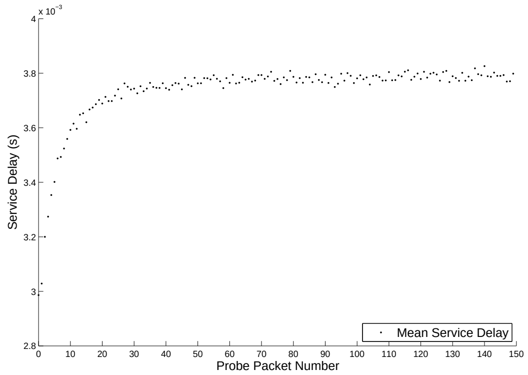


Figure 8.4: Mean Service Delay vs. Probe packet num

bandwidth present deviations from the fluid model that may lead to erroneous measurements. As they show, such deviations constitute biases that are difficult to remove when the number of packets used to infer bandwidth metrics (the train length) is not large enough.

This section takes a similar approach but applied to bandwidth measurements in WLAN environments. First it presents a detailed study of the service delay that probing packets experience when trying to access the shared WLAN medium. This study focuses on the transitory regime that this service delay experiences rather than on its stationary behavior.

After characterizing the service delay, the section presents an analytical approach to the bandwidth measurement problem. We first introduce the analytical framework and later show how both the randomness of the service delay and its transitory regime bias bandwidth measurements based on packet dispersion.

8.4.1 Analysis of service delay

This section analyses the characteristics of the service delay that probe packets experience. We take as service delay the time since the probing packet is ready to be transmitted (it is at the head of the FIFO queue) until it is completely transmitted. The service delay in WLANs has been repeatedly studied in the literature. Indeed, different researchers have analyzed its exact distribution using stochastic tools, such as

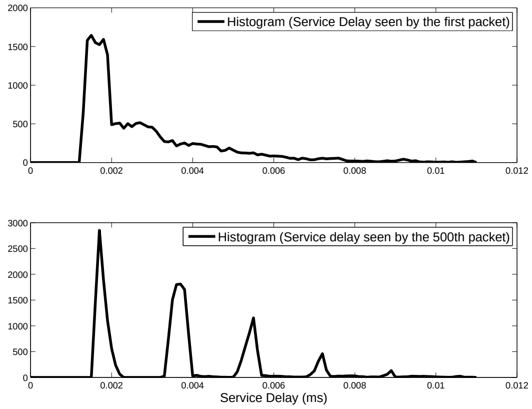


Figure 8.5: Histogram of the s.d seen by the first and 500th packet (simulation)

Markov Chains [206; 209]. Even more, others show how the exponential distribution provides a good fit [207]. All these studies have focused on the distribution of the service delay in stationary state. However, in general, dispersion based measurements are gathered using packet trains with a limited length (limited number of packets). As a consequence, for the purpose of this work, we are interested in analyzing how the service delay evolves in time as an increasing number of probing packets are sent through the WLAN.

In order to analyze such evolution we have conducted the following simulation. Using NS2 we send 1000 probe packets at a given rate (5Mbps) and with a static load of contending cross-traffic (4Mbps). We have repeated the experiment 25000 times and, for each probe packet (indexed from 1 to 1000), we compute the distribution of the service delay (considering all the repetitions).

Figure 8.4 plots the average service delay that each one of the first 150 packets observes. The figure shows how the average service delay perceived by the first packets is lower than for the rest of them. This suggests that, in fact, the distribution of the service delay changes as more probe traffic keeps on arriving to the WLAN link. In order to verify this hypothesis, figure 8.5 plots the histogram of the service delay seen by the first probe packet and by the 500th. As the plot shows, the distribution changes significantly. The main rationale behind this is that as new probing packets keep on arriving they keep on increasing the load of the network (transitory regime) until reaching a stationary state of interaction with the (contending) cross-traffic.

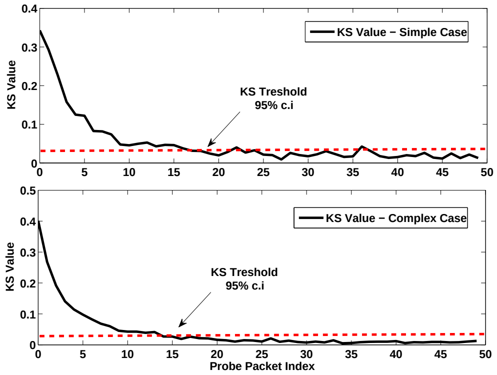


Figure 8.6: (Top) Analysis of the distribution simple case (Bottom) complex case

In order to better understand the transitory regime we use a metric able to capture the relation between the distribution of the service delay during the transitory regime and once in stationary state. For this purpose we use the Kolmogorov-Smirnov[1] goodness-of-fit test [215]. This test is a non-parametric test that analyzes whether two different sets come from the same distribution. Using this test we compare the distribution of each individual probing packet in the sequence with the service delay distribution of the last 500 probing packets. Figure 8.6 plots the results for the example in this section. The dashed line represents the threshold at 95% of c.i that rejects the null hypothesis (i.e both samples come from the same distribution). The figure shows two cases, the simple one with one contending station (poisson) at 5Mbps and probe traffic at 4Mbps. The second one, a complex case, with 4 contending stations using different packet sizes (40, 576, 1000 and 1500 bytes) and the following rates respectively (0.1, 0.5, 0.75 and 2Mbps). As both cases show, we need to send tens of packets until the service delay observed reaches the stationary state.

We have simulated multiple cases with different degrees of complexity. In all of them the Kolmogorov-Smirnov test reveals that the presence of the transitory in distribution of the service delay. Summarizing, this analysis reveals how the service delay that probe packets experience, when observed on a per-packet basis, presents a transitory in distribution. As a consequence, the first packets that enter a WLAN system do not

[1] Since we are using the KS test to compare two empirical discrete distributions we convert one of them to a continous one using linear interpolation.

capture the asymptotic behavior of a flow, and, as revealed in the next subsections, this biases dispersion measurements obtained with short probing trains.

8.4.2 Analytical framework

Here we present the basic analytical framework used to deal with this problem. This framework was originally formulated in [214] but it is extended here to focus on the particularities of WLAN transmissions.

The probing sequence: Arrivals, departures and input gap

The probing sequence consists of a series of n packets that enter the transmission queue at instants $\{a_i, i = 1, 2, \cdots, n\}$. Their departure instants, meaning the time at which they are completely transmitted, form the series $\{d_i, i = 1, 2, \cdots, n\}$. Finally, we are considering here periodic probing flows with a fixed inter-packet arrival time or input gap: $g_I = a_i - a_{i-1}$.

The service delay process

As shown above, the service delay that probing packets experience is a random process. This process is the result of the interaction between probing traffic, contending cross-traffic and backoff. To account for this let us define the sequence $\{\mu_i, i = 1, 2, \cdots, n\}$ to denote the random service delay that each one of the n probing packets of a probing sequence experiences when contending for medium access.

As shown above, the service delay presents a transitory period until reaching a certain stationary distribution. Thus, $\exists n_0 : \forall\{i > n_o\}, \mu_i$ is i.i.d. Further, we assume that the service delay distribution is upper and lower bounded. In other words, we assume that $\exists\{\mu^{max}, \mu^{min}\} : \forall i, Pr(\mu^{min} \leq \mu_i \leq \mu^{max}) = 1$.

Intrusion residual: amount of probe traffic in the FIFO queue

The intrusion residual $W_d(t)$ accounts for the sum of the service time of all probing packets in the FIFO queue and the remaining time to service the probing packet that may be in transmission. Next, we define the series $\{R_i, i = 1, 2, \cdots, n\}$ which captures the intrusion residual that every probing packet finds when it enters the transmission queue,

$$R_i(a_1) = W_d(a_i^-) = W_d(a_1 + (i-1)g_I^-) \tag{8.4}$$

Note[1] that R_i is a recursive process that under the assumptions in this work can be expressed as,

$$R_i = \begin{cases} 0 & i = 1 \\ max(0, \mu_{i-1} + R_{i-1} - g_I) & i > 1 \end{cases} \tag{8.5}$$

Finally, we define the series $\{Z_i, i = 1, 2, \cdots, n\}$ that encloses the queuing plus service delay that each one of the probing packets experiences. Under the assumptions taken,

$$Z_i = d_i - a_i = \mu_i + R_i \tag{8.6}$$

Dispersion based measurements:The output gap and its relation to the probing rate

Dispersion based measurements of bandwidth metrics consist on measuring the dispersion (or inter-departure time) of packets at the output of a path (receiving side). This measure is then used to infer the value of bandwidth related metrics. The output gap (or dispersion) of a train of probing packets is defined as follows,

$$g_O = \frac{d_n - d_i}{n-1} \tag{8.7}$$

Figure 8.7 illustrates the contribution of the processes defined above to the value of the output gap. From the arrival of the first probing packet at the transmission queue (a_1), probing packets keep on arriving at a constant interval of g_I. The cross-traffic, service delay and the intrusion residual of previous probing packets (Z_i) randomize the departure times of probing packets (d_i) and thus, their output dispersion (g_O).

Observing figure 8.7 we can obtain the output gap in relation to the different processes involved.

$$g_O = \frac{d_n - d_i}{n-1} = \frac{(n-1)g_I + Z_n - Z_1}{n-1} \tag{8.8}$$

[1] The minus superscript refers to the *a priori* state of the queue.

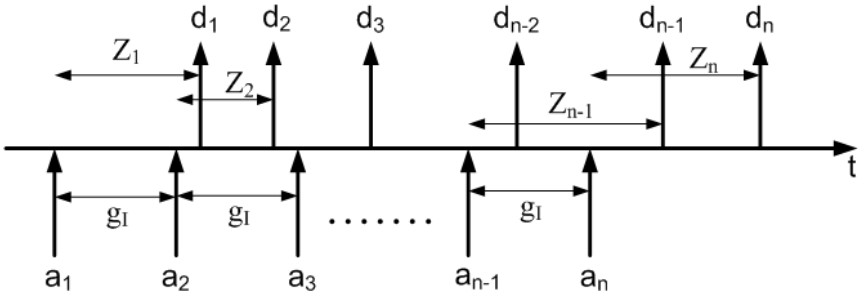


Figure 8.7: Inter-relation between probing arrival sequence (a_i), departure sequence (d_i) and cross-traffic related processes (Z_i).

Expanding this expression we get the following,

$$g_O = g_I + \frac{R_n}{n-1} + \frac{\mu_n - \mu_1}{n-1} \tag{8.9}$$

Problem formulation

We are interested in studying whether the rate response curve gathered in a realistic setting follows that of the fluid model. Thus, we are interested in analyzing whether equation 8.3 still applies here,

$$E[r_o] \overset{?}{=} \begin{cases} r_i & r_i \leq \overline{B} \\ \overline{B} & r_i \geq \overline{B} \end{cases} \tag{8.10}$$

However, dispersion based measurements take inferences of bandwidth metrics based on the output gap of trains of a limited length. We can reformulate this problem in terms of the output gap. If we use packets of length L to gather dispersion measurements, the problem can be stated as,

$$E[g_O] \overset{?}{=} \begin{cases} g_I & g_I \geq \frac{L}{\overline{B}} \\ \frac{L}{\overline{B}} & g_I \leq \frac{L}{\overline{B}} \end{cases} \tag{8.11}$$

Thus, taking expectation over eq. 8.9, the rest of this section deals with the evaluation of the behavior of the following expression,

$$E[g_O] = g_I + \frac{E[R_n]}{n-1} + \frac{E[\mu_n] - E[\mu_1]}{n-1} \tag{8.12}$$

8.4.3 The impact of the randomness of service delay on dispersion measurements

The aim of this section is to present the impact of a random service delay on bandwidth measurements, but isolated from the transitory regime detected above. As we show below the random service delay is a source of bias on its own, regardless of the presence of the transitory. We take the assumption, in this section, that the service delay does not present the transitory behavior described above. Thus, we consider that for all probing packets the service delay is i.i.d. In others word, we have that $\mu_i = \mu$ for any probing packet.

Expected output dispersion and achievable throughput

Assuming that the service delay does not present a transitory stage, expression 8.12 reduces to the following,

$$E[g_O] = g_I + \frac{E[R_n]}{n-1} \tag{8.13}$$

Under the assumption of no other (cross-)traffic in the FIFO queue the system can serve, in average, up to one probing packet every $E[\mu]$. As a consequence we can state that,

$$\overline{B} = \frac{L}{E[\mu]} \tag{8.14}$$

Expectation on output gap based on bounds of the intrusion residual

From expression 8.13, we learn that the expected output gap depends on the expected value for the residual that the last packet of the probing train (i.e. with index n) finds in the queue. Considering eq. 8.5 and that the service delay presents upper and lower bounds (i.e. μ^{max} and μ^{min}), one can define the following (loose) bounds for the probing residual:

$$\begin{cases} R_n = \sum_{i=1}^{n-1}(\mu_i - g_I) & g_I \leq \mu^{min} \\ max(0, \sum_{i=1}^{n-1}(\mu_i - g_I)) \leq R_n \leq \sum_{i=1}^{n-1}\mu_i & \mu^{min} \leq g_I \leq \mu^{max} \\ R_n = 0 & g_I \geq \mu^{max} \end{cases} \tag{8.15}$$

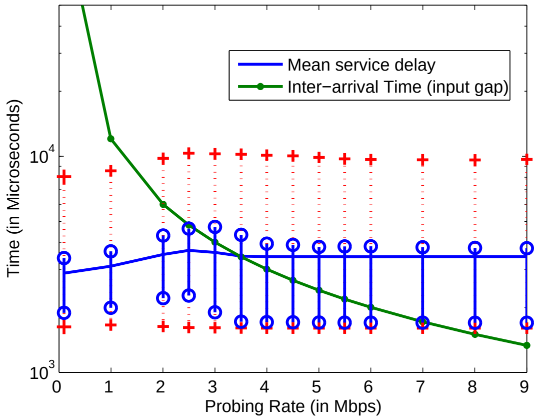


Figure 8.8: A comparison between the input inter-arrival of probing packets and the service delay distribution of the packets for a very large probing flow

Taking expectation over R_n, we can identify four differentiated regions,

$$\frac{E[R_n]}{n-1} = \begin{cases} E[\mu_s] - g_I & g_I \leq \mu_s^{min} \\ \frac{\beta_n}{n-1} & \mu_s^{min} \leq g_I \leq E[\mu_s] \\ \frac{\alpha_n}{n-1} & E[\mu_s] \leq g_I \leq \mu_s^{max} \\ 0 & g_I \geq \mu_s^{max} \end{cases} \tag{8.16}$$

The parameters α_n and β_n depend on the specific characteristics of the random cross-traffic but can be (loosely) bounded as follows,

$$\begin{cases} E[\mu_s] - g_I \leq \frac{\beta_n}{n-1} \leq E[\mu_s] \\ 0 \leq \frac{\alpha_n}{n-1} \leq E[\mu_s] \end{cases} \tag{8.17}$$

To illustrate all this, figure 8.8 depicts the relation between the inter-arrival time of probing packets (depending on the probing rate) and the service delay that these probing packets experience. The service delay is depicted in a boxplot fashion (i.e. with inter-quartile and max-min values). In the example of the figure the achievable throughput is 3.5Mbps.

To understand the α_n term, one can observe in the figure that when the probing rate reaches 2.5Mbps there is a non-negligible possibility that probing packets experience a service delay higher than the inter-arrival of packets. Thus, when probing at a rate between 2.5Mbps and 3.5Mbps, probing packets have a non-negligible probability of meeting in the queue. This is not expected in a fluid model as we are probing at a rate lower than the achievable rate. For short probing trains this leads to a non-negligible bias of the measure.

On the other side, the β_n term appears when probing at a rate higher than 3.5Mbps but lower than 7Mbps. In this region there exists the possibility that the service delay is lower than the inter-arrival time. Thus, when probing within this range there is a non-zero probability that consecutive probing packets do not meet in the transmission queue. Again, this is not expected in a fluid' system as we are probing at a higher rate than the achievable throughput. Also, for short packet trains this leads to a non-diminishable bias in the measure.

Finally, we are interested in the output dispersion. Thus, substituting eq. 8.16 into eq. 8.13 we get the following,

$$E[g_O] = \begin{cases} E[\mu_i] & g_I \leq \mu^{min} \\ g_I + \frac{\beta_n}{n-1} & \mu^{min} \leq g_I \leq E[\mu_i] \\ g_I + \frac{\alpha_n}{n-1} & E[\mu_i] \leq g_I \leq \mu^{max} \\ g_I & g_I \geq \mu^{max} \end{cases} \tag{8.18}$$

Two important observations about eq. 8.18 are that, first, α_n and β_n are deviation terms that depend on the length of the packet train (n) and that disappear when the input gap (g_I) falls outside the limits of the random service delay. Second, the lower bound of the output gap corresponds to the fluid response curve.

Numerical results on the rate response curve

The transitory stage of the service delay of a WLAN system cannot be removed in a testbed and is hard to remove from NS2 simulations. In order to do so we use the queuing simulator developed in Matlab and introduced in section 2. Figure 8.9 plots the expected rate response curve inferred using the dispersion of trains of different lengths (2, 10 and 20 packets) at the output of a system with exponential service delay. The mean service delay that probing packets experience corresponds to an achievable throughput of 3.5Mbps (probing packets are 1500 bytes long) and follow an exponential random distribution (see [207]). The figure shows also the fluid response curve. When inferring the achievable throughput with a limited number of packets the rate response curve deviates from the fluid one. The deviation is more intense as the number of packets used for the measure decreases. The figure plots also, as a reference, the

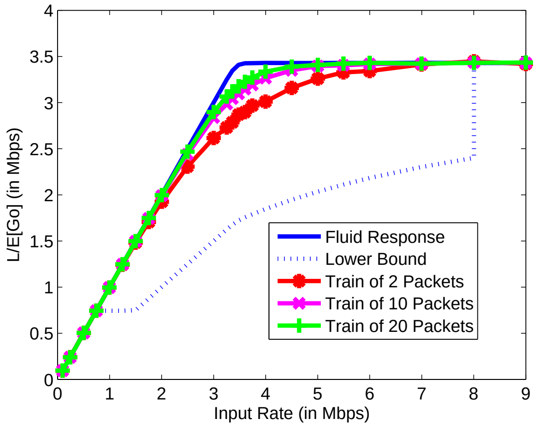


Figure 8.9: Rate response curve when probing a system with exponentially distributed service delay with probing sequences of different length.

(lower) bound on the maximum deviation of the rate response curve coming from the (upper) bounds on the expected output gap in expression 8.18.

The origin of the bias

The biases detected in eq. 8.18 and shown in figure 8.9 have their origin in the evolution of the expected queuing delay that probing packets suffer when traversing the FIFO queue. To illustrate this, figure 8.10 plots the difference in the mean delay experienced by each one of the first 100 packets of a long probing flow with respect to their immediate predecessor. In other words it plots the process $\{E[Z_i - Z_{i-1}, i = 2, \cdots, n\}$. This process is shown for different probing rates. It can be seen from the figure that it takes some probing packets until the process $E[Z_i - Z_{i-1}]$ becomes stable (constant). This is precisely what deviates the α_n and β_n terms in equation 8.17 from the fluid response curve.

This figure reveals that, when probing packets are served with random delay, it takes some packets until they start experiencing a stationary behavior. The first packets, then, constitute biased measures of the delay required to traverse the link.

Asymptotics of the deviation terms

As figure 8.9 suggests, the longer the packet train the lower the bias of dispersion measurements. Following a similar procedure as in [214] it can be shown that eq. 8.18

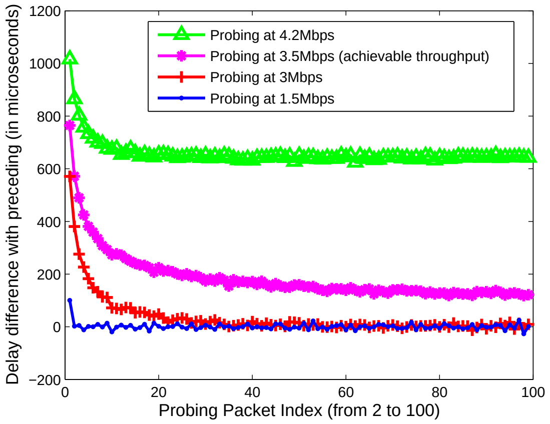


Figure 8.10: Expected per-packet delay difference between consecutive probing packets sent through a FIFO queue with exponential service delay

tends asymptotically to the fluid response curve as n increases. That is,

$$\lim_{n \to +\infty} \frac{\alpha_n}{n-1} = 0 \tag{8.19}$$

and,

$$\lim_{n \to +\infty} \frac{\beta_n}{n-1} = E[\mu_i] - g_I \tag{8.20}$$

Furthermore, it can be shown that there exists a strong relation between the variance of the service delay and the intensity of the deviation of the rate response curve. If we unbound the service delay then,

$$\lim_{Var[\mu] \to +\infty} \frac{\alpha_n}{n-1} = E[\mu] \tag{8.21}$$

and,

$$\lim_{Var[\mu] \to +\infty} \frac{\beta_n}{n-1} = E[\mu] \tag{8.22}$$

8.4.4 The impact of the transitory regime of service delay on dispersion measurements

This section reintroduces the transitory of the service delay and studies its impact on dispersion measurements.

Expected output dispersion and achievable throughput

Now the expression of the output gap cannot be reduced and our objective is studying this expression,

$$E[g_O] = g_I + \frac{E[R_n]}{n-1} + \frac{E[\mu_n] - E[\mu_1]}{n-1} \tag{8.23}$$

We can define again a relation between the achievable throughput and the service delay that probing packets receive.

$$\frac{L}{\overline{\overline{B}}} = \frac{1}{n}\sum_{i=1}^{n}(E[\mu_i]) \tag{8.24}$$

Note also that as the number of probing packets grows the expected service delay becomes constant and we can say that,

$$\frac{L}{\overline{\overline{B}}} \xrightarrow{n} E[\mu_n] \tag{8.25}$$

Expectation on output gap based on bounds of the intrusion residual

Following similar reasoning as in the previous section, the expected output dispersion of a train of n packets presents four differentiated regions such as,

$$E[g_O] = \begin{cases} \frac{1}{n-1}(\sum_{i=2}^{n}(E[\mu_i])) & g_I \leq \mu^{min} \\ g_I + \frac{\beta_n}{n-1} & \mu^{min} \leq g_I \leq \frac{1}{n}\sum_{i=1}^{n} E[\mu_i] \\ g_I + \frac{\alpha_n}{n-1} & \frac{1}{n}\sum_{i=1}^{n} E[\mu_i] \leq g_I \leq \mu^{max} \\ g_I + \frac{[\mu_n - \mu_1]}{n-1} & g_I \geq \mu^{max} \end{cases} \tag{8.26}$$

The parameters α_n and β_n in the expression above are bounded as follows,

$$\begin{cases} \frac{1}{n}\sum_{i=2}^{n}(E[\mu_i] - g_I) \leq \frac{\beta_n}{n-1} \leq \frac{1}{n-1}\sum_{i=2}^{n}(E[\mu_i]) \\ 0 \leq \frac{\alpha_n}{n-1} \leq \frac{1}{n-1}\sum_{i=2}^{n}(E[\mu_i]) \end{cases} \tag{8.27}$$

Expressions 8.26 and 8.27 reveal some interesting features about dispersion measurements for WLAN environments.

First, note that considering that the expected service delay in a WLAN environment is an increasing function with respect to the packet index (i), the following is true for

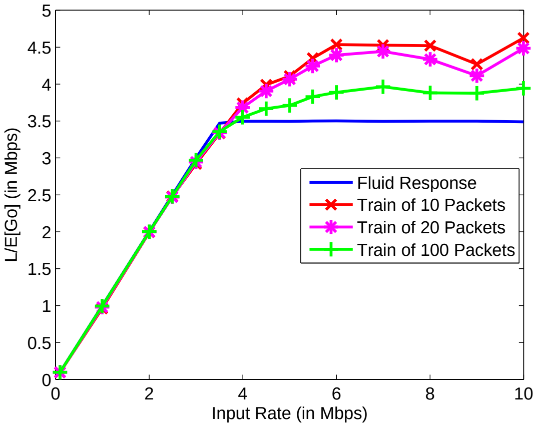


Figure 8.11: Experimental rate response curve when probing a WLAN system with probing sequences of different length

any value of n,

$$\frac{1}{n-1}\sum_{i=2}^{n}(E[\mu_i]) < E[\mu_n] \tag{8.28}$$

As a result, when probing packets arrive faster than the achievable throughput (i.e. $g_I \leq \frac{1}{n}\sum_1^n E[\mu_i]$), packets at the output experience a compression effect with respect to the fluid response that leads to inferring higher output rates than those that can be actually achieved in this region.

Second, as eq. 8.27 reveals, the upper bound of α_n is lower than without the presence of the transitory. The bias introduced by this term in dispersion measurements is, thus, lower than would be expected without the presence of the transitory.

Third, when the input rate is low enough (i.e. when $g_I \geq \mu^{max}$) the transitory causes an expansion of the expected output dispersion.

Figure 8.11 shows an experimental result showing these observations. The rate response curves plotted correspond to those of packet trains probing a WLAN link at different rates. The figure clearly shows how, when short packet trains are used, the rate response curve gathered leads to inferring higher rates than the achievable throughput. The figure also shows how the bias introduced by the α_n term and the expansion suffered at low probing rates are not important enough to distort the measurement process.

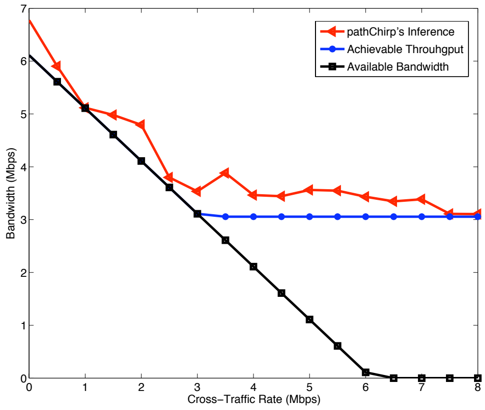


Figure 8.12: Estimation of pathChirp in a wireless link (1 contending node, exponential inter-departure time, 1500bytes as packet size, intensity varies).

8.5 Discussion

This section discusses the main consequences of the findings of this chapter. Since the results are based on modeling the WLAN as a FIFO queue with a random service delay, these findings could be extended to any system following such a model.

A consequence: bandwidth estimation in WLAN links

As shown in section 2 traditional methodologies and tools, such as [93; 94; 95; 96; 97; 98; 99; 100; 101], designed to estimate the available bandwidth target the achievable throughput in the presence of wireless links. To ilustrate this consider the experiment depicted in figure 8.12 (NS2). We have run a state-of-the-art available banwdith estimation tool (pathChirp [97]) in the presence of a wireless link. pathChirp bases its inference in the delay of probe traffic, such as Pathload [95] and others.

As the figure shows pathChirp points at the achievable throughput, this is clearer when both metrics are different (i.e cross traffic intensity is above 3Mbps).

Another consequence: packet pair measurements in WLAN links

Packet pairs that are sent back-to-back through a network path have been traditionally used to measure the capacity of the path [218]. However, as a consequence of the results

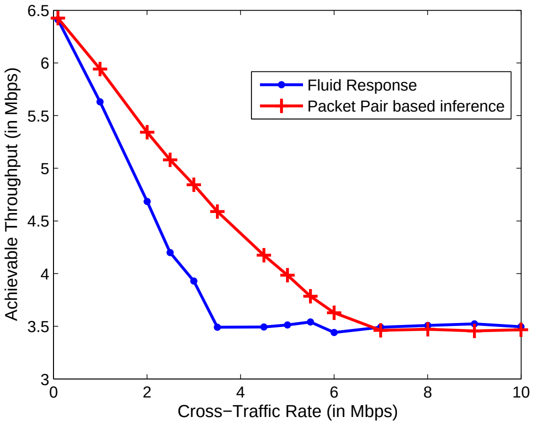


Figure 8.13: Experimental comparison between packet pair based bandwidth measurements and the actual fluid response in a WLAN link

presented in section 3 the packet pair (an infinite input rate) targets the achievable throughput when used in a WLAN link.

However, the results of section 2 reveal that probing at high rates leads to overestimating the achievable throughput due to the transitory regime that the service delay presents. This result has a particular relevance as packet-pair measurements have been extensively used to gather bandwidth metrics in wireless environments, especially in proposals related to routing in all-wireless multi-hop networks [217]. In order to illustrate this, figure 8.13 shows an experimental comparison of the achievable throughput measured using packet pairs together with the actual one (when using long packet trains). This is done for different rates of contending cross-traffic. As the figure shows back-to-back packet pairs overestimates the achievable throughput unless there is no contending traffic or it is already saturating the WLAN system (i.e. when cross-traffic throughput is higher than 7Mbps in the figure).

An application of results: improving the convergence and accuracy of traditional tools

The results presented in section 4 entail a second important observation that can be used to improve the accuracy of bandwidth measurement tools. As figure 8.4 and figure 8.10 reveal, the first packets that traverse a WLAN link constitute non-accurate samples of the stationary behavior. A direct consequence of this is that first samples

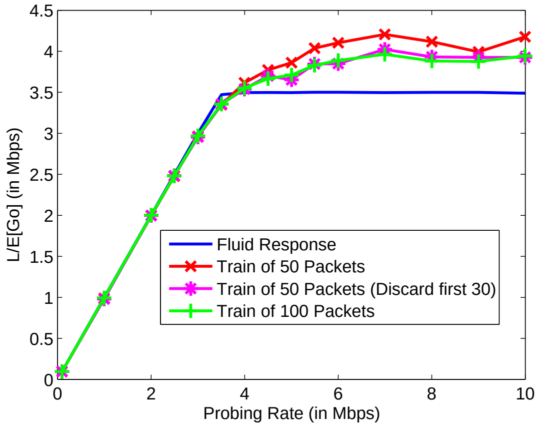


Figure 8.14: Experimental rate response curves showing the bias incurred by different probing strategies

(packets) of the probing train should be removed from bandwidth estimates as they may distort the measurement.

Traditionally the approach to remove measurement biases consists in enlarging the number of packets used to gather measurements. However, this comes at the cost of increasing the intrusiveness of the measurement process over the measured path. The results in section 4 reveal that removing the first packet samples from bandwidth measurements helps reducing the measurement bias and can help improve the measurement accuracy with a limited number of probing packets.

Figure 8.14 illustrates this observation. As the figure shows one can achieve the same measurement accuracy using trains of 50 packets (but removing the first 30 from the measure) as when using trains of 100 packets. This could be easily applied to existing tools [93; 94; 95; 96; 97; 99; 100; 101] improving its accuracy and/or reducing its convergence time.

8.6 Summary and Conclusions

In this chapter we have analyzed the traditional bandwidth estimation models and tools in the presence of wireless links. These tools base its inference in periodic probing processed. Our analysis has revealed that:

- The rate response curve, when applied over a WLAN system, points at the achievable throughput rather than the available bandwidth (as assumed previously [102]).

- Dispersion based measurements of bandwidth metrics are biased. There are two sources of bias in a WLAN system. On one side there is the randomness of the service delay that probing packets experience. On the other side, there is the transitory in distribution that the random service delay presents. In both cases the origin of the bias lies in the fact that it takes a while (some packets) for the probing traffic to completely interact with the contending cross-traffic. This implies that the first packets in a probing sequence do not follow the stationary state and are not valid estimates of the stationary behavior.

This has mainly two important consequences:

- Periodic probing processes can be inadequate to estimate the available bandwidth in wireless networks. As we show in the next chapter, poisson-based measurements can be effectively used to infer congestion in the presence of such links.

- Tools based on periodic probing can be significantly improved by removing the first packets of the probing sequence, hence removing the bias. This can lead to build lighter (less intrusive) tools while achieving the same (or even more) accuracy.

9

Available Bandwidth Estimation Tools

9.1 Introduction

In the previous chapter we have established the fundamentals of bandwidth estimation in wireless links. The main conclusion of this analysis is that existing tools, mostly based on periodic probing processes [93; 94; 95; 96; 97; 98; 99; 100], target the achievable throughput rather than the available bandwidth. In this chapter we focus our research on exploring Poisson probing process in order to design methodologies and tools for bandwidth estimation in a wide range of scenarios.

First we address bandwidth inference in wireless scenarios. Specifically we consider a wireless-cum-wired scenario, this is a multi-hop wired path where the last link is an IEEE 802.11 access network. Such paths are the usual operational scenarios of multihomed mobile architectures (see chapter 2 for further details).

Reviewing the existing literature shows that, at the best of our knowledge, the only existing tool able to infer congestion in wireless links is ProbeGap [102]. ProbeGap also bases its estimation in Poisson-probing processes that, as the PASTA [187] property states, sample the average queue sizes as an outside observer. It is worth noting here that in the previous chapter we have shown that the rate-response curve of periodic probing does not show any deviation at the available bandwidth, hence it may be difficult to estimate this metric with such probing process.

Although ProbeGap has been proved as very accurate it is intended for a single-hop wireless link scenario. In fact, it fails when applied to wireless-cum-wired scenarios. A reason for this is that the technique used to infer congestion it does not take into account the interference of the wired cross-traffic. A case when the most congested link is a wired one would not be identified correctly. Also, ProbeGap requires the a priori knowledge of the transmission rate of the bottleneck link (assumed in the wireless hop) to infer the available bandwidth. This rate might not be always available and varies over time in IEEE 802.11 links.

ProbeGap shows that Poisson-probing is a feasible approach for available bandwidth estimation in wired-cum-wireless scenarios. The first methodology presented in this chapter infers congestion on the detection of queuing of probe packets in the end-to-end path. To avoid the problems suffered by ProbeGap our methodology is able to remove the effects derived from the random scheduling of packets in a wireless link from those induced by the cross-traffic. In further sections we apply the methodology to design a tool (W-Path) and we provide an extensive performance evaluation.

Furthermore, poisson-based methodologies have not been explored in bandwidth inference in wired scenarios. As stated before, the traditional approach for bandwidth estimation tools for such scenarios is periodic probing processes. Most of the proposed tools designed to estimate the AB fall into two categories: the Probe Rate Model (PRM) [95] and the Probe Gap Model (PGM) [98]. The first model uses packet trains and it is based on the concept of self-induced congestion. Informally, if one sends a packet train at a rate lower than the AB along the path, then the arrival rate of the packet train at the receiver will match the rate at the sender. However if the sending rate is greater or equal than the AB then the packet train will congest the queues along the path and the receiving rate will be lower than the sending rate. Tools such as Delphy [93], TOPP [94], PathLoad [95], IGI/PTR [96], pathChirp [97], BART [99] and Forecaster [100] use this model. The second model (PGM) uses packet pairs and bases its estimation on the differences of input and output time gaps of the packet pairs [98].

Although the PRM has been shown as very accurate [95], it suffers from one basic problem: PRM-based tools must send probe traffic at a rate equal or greater than the AB. This will fill the queues along the path congesting it. This means that, for each estimation, a PRM-based tool congests the measured path during a certain period of time. In fact *A.Shriram* showed recently in [190] that tools such as PathLoad can significantly impact the response time of TCP connections.

That is why we explore a poisson-based approach in wired paths with the main goal of designing low-intrusive accurate tools. As shown later we apply this methodology to create two different tools: AKBest (Active Kalman-based Estimation) and PKBest (Passive Kalman-based Estimation). The first one is an active tool that it is able to infer congestion sending packet trains at a lower rate than the available bandwidth, hence reducing the impact on the path under measurement. The second one is a passive available bandwidth tool able to estimate the AB of a given path without introducing probe traffic.

The remainder of this chapter is as follows, first we detail the network model that is the basis of our proposed methodology. Next we present the methodology, for W-Path, AKBest and PKBest and finally a performance evaluation of all three tools. In order to validate the accuracy of these tools we use the simulator as described in the previous chapter (section 8.2).

9.2 Mathematical Model

This section presents the mathematical model used by the methodology. First, the utilization of a queue i in a single-hop scenario is:

$$u_i = 1 - \pi_i \tag{9.1}$$

Where π is the probability that the queue is void. Eq. 9.1 does not make any assumption about the nature of cross-traffic or the scheduling used in the link (i.e FIFO or non-FIFO).

If we transmit probe traffic at a rate r through this link, then the utilization can be expressed as:

$$u_i(r) = min(1, u_i + \frac{r}{C_i}) \tag{9.2}$$

Where C_i is the capacity of link i. For the multi-hop case *K. Harfoush* showed in [100] a first order approximation of eq. 9.2:

$$u(r) \approx min(1, ar + b) \tag{9.3}$$

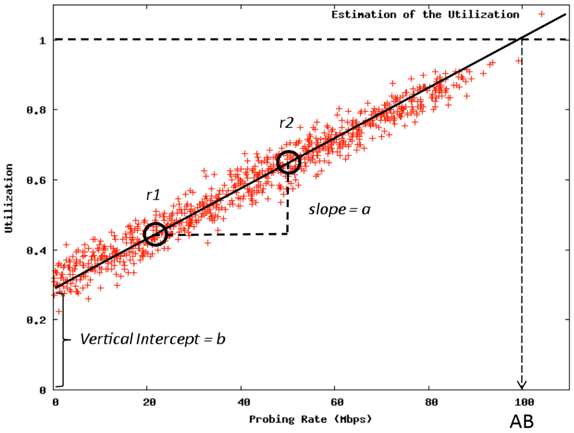


Figure 9.1: The mathematical model

Where a and b are constants. This equation states that there is a linear relation between the utilization of a path and the rate of the probe traffic sent. Figure 9.1 shows that, as the probe traffic rate increases so does the utilization (linearly). At a certain rate r_{ab}, the utilization will reach 1 (the path is fully loaded) then, the AB of the path is r_{ab}.

This means that with this model we can estimate the AB without sending the probe traffic at the AB rate. Once the linear equation (eq. 9.3) has been estimated the AB can be computed as:

$$AB = \frac{1 - \widetilde{b}}{\widetilde{a}} \tag{9.4}$$

9.3 Methodology

This section details methodology used to infer bandwidth in wired and wired-cum-wireless paths. The methodology is based on the model described in the previous chapter. As eq. 9.3 states, if the utilization of a path can be estimated, then inferring the available bandwidth is straightforward through eq. 9.4. First we show an estimator for the utilization in a wired path (for AKBest and PKBest), then we discuss the challenges that this metric presents in wireless links and how to overcome them (for W-Path).

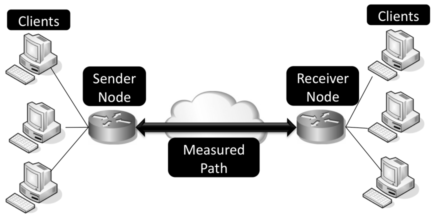


Figure 9.2: The passive AB estimation architecture

9.3.1 Estimating the Utilization in a Wired Path

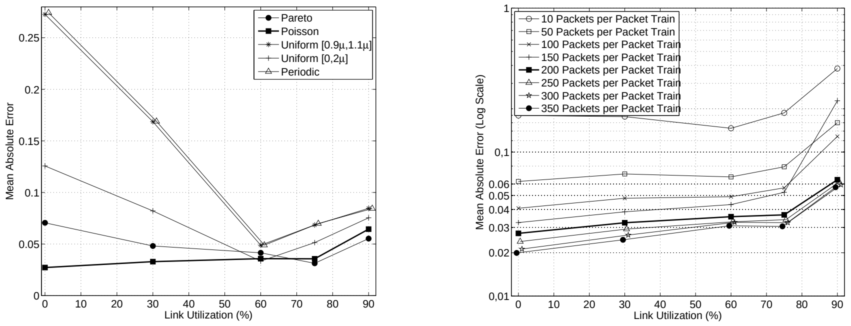
Eq. 9.1 defines the utilization as the probability that there is at least one packet in the queues of the path. In order to estimate the utilization of a path the authors in [100] suggest sending a packet train (a sequence of probe packets), end to end, and compute the fraction of packets that have experienced queuing delay along the path. Probe packets are time stamped at the sender and at the receiver. Then the minimum one-way delay of the set of packets is computed. This minimum delay corresponds to the delay suffered by a packet that has not encountered queuing delay. Therefore the fraction of packets with a greater delay than the minimum delay is the fraction of packets that suffered queuing delay. Let $D = \{d_1 \cdots d_N\}$ be the set of one-way delays suffered by the packets of the train. Then the utilization is estimated as:

$$\widetilde{u} = \frac{\|\{d_i > min\{D\}|d_i \in D\}\|}{\|D\|} \tag{9.5}$$

The utilization of the path is estimated through eq. 9.5. AKBest, which is an active tool, operates sending packet trains at different rates and estimating the utilization (for each rate). Through this process AKBest estimates the linear equation 9.3 and thus, the available bandwidth. On the contrary, PKBest is a passive tool intended to be applied between two separate nodes (see figure 9.2): the sender and the receiver nodes.

PKBest aims to estimate the available bandwidth on the path by analyzing specific parameters of the already existing traffic exchanged between both nodes. Throughout this chapter we refer to this traffic as *data-traffic.* The main challenge then is that it cannot relay on any given rate or pattern of the data-traffic.

In the next subsection we investigate which are the optimal parameters of the packet trains to accurately estimate the utilization. These packet trains are used by AKBest



(a) Mean absolute error for different distributions and link loads

(b) Mean absolute error for different packet trains lengths and link loads

Figure 9.3: Analysis of the Probe Traffic

and later, we investigate if such trains are present in the Internet *data traffic* and hence, can be used by PKBest.

Distributions

First we are going to investigate which is the optimal distribution of the inter-departure times of the packets within a packet train. According to the PASTA [187] property, if a packet train is sent with exponential inter-departure times, the packets arriving at the queuing system will sample the system queues, on average, as an outside observer would, at an arbitrary point in time. However *F. Baccelli* showed in [191] that poisson probes are not unique in their ability to sample without bias. That is why we evaluate other distributions in order to analyze which one samples the queues better for our particular estimator. All the experiments have been carried out using NS2.

We evaluate a range of different distributions by sending 104 packet trains (of 200 packets and 1500 bytes as packet size) at different rates through a single link fed with Poisson packet arrivals. The cross-traffic packet sizes are distributed as in the Internet (see [200] for details): 50% (40 bytes), 10% (576 bytes) and 40% (1500 bytes).

The experiment is performed with different packet trains distributions: Periodic, Poisson, Uniform ($[0.9\mu, 1.1\mu]$), Uniform ($[0\mu, 2\mu]$) and Pareto (α index = 1.16) and with different link loads (utilization=$\{0, 0.3, 0.6, 0.75, 0.9\}$). For each packet train we have computed the *absolute* error when estimating the utilization.

As Figure 9.3 shows the distribution that minimizes the error when estimating the utilization is the poisson distribution (as expected). The worst one is the Periodic distribution. In fact the exponentially distributed packet trains are not severely impacted by the load of the link and the mean error is always below 0.07.

The periodic and Uniform distributions ($[0.9\mu, 1.1\mu]$) create ”constant” packet trains where the inter-departure time of the packets is very similar. The Probe Rate Model roughly describes the behaviour of these packet trains. If the rate of the packet train is above the AB, then almost all the packets are queued and thus, the utilization is overestimated. If the rate is below the AB, then the packets do not congest the tight link queue and thus, the utilization is underestimated. Regarding the Pareto and Uniform ($[0\mu, 2\mu]$) distributions, Figure 9.3 shows that they are more accurate under high link loads. This is because these distributions have also some ”periodicity”. When the link load is at 90%, the cross-traffic rate congests the tight link queue and almost all the packets are queued. This means that the utilization, for this very special case, is accurately estimated.

From this experiments we can conclude that the optimal inter-departure time of the trains is poisson.

Length

In this subsection we evaluate the optimal packet train length. We have sent packet trains using different lengths {10,50,100,150,200,250,300,350} in a single-hop scenario. The link is loaded with cross-traffic {0,0.3,0.6,0.75,0.9} and the packet trains are sent at different rates, ranging from 0.1Mbps to the AB rate. In this case, the distribution is fixed to poisson and the packet size is fixed to 1500 bytes. The rest of the simulation parameters are the same than in the previous experiment.

Figure 9.3(b) (note that the X-Axis uses a log-scale for clarity) shows the results of the experiments. Packet trains with a length lower or equal to 150 packets suffer from a large error when estimating the utilization.

Regarding packet trains longer than 150 packets, the mean error is bounded to 0.06. In addition the accuracy is not significantly impacted by the utilization of the link under study. In fact the error is slightly reduced as the packet train length increases. However it is important to remark that long packet trains suffer from one basic problem, which is that the utilization may change during the transmission of the train and this may lead to incorrect estimates. Thus there is a tradeoff between the accuracy of the estimations and the duration of the measurement. Longer packet trains (250, 300, 350...) have slightly less error, however as the figure shows, this extra accuracy is not justified since the duration of the measurement increases dramatically (i.e. packet trains of 250 packets last 20% longer than packet trains of 200 packets). In addition we have to take into account PKBest's requirements: larger trains have less probability of being present in Internet traffic. That is why we believe that the optimal length for our trains is 200 packets.

Size

At this point we have concluded that the optimal packet inter-departure time distribution is Poisson and the optimal length is 200. The last parameter to evaluate is the packet size. AKBest can choose an optimal packet size, however PKBest cannot rely on any given packet size or distribution of the data-traffic. That is why in this subsection we analyze also the impact of random packet sizes in the packet trains. We have used the same parameters for this simulation than for the previous one, but the distribution is now fixed to Poisson, the length to 200 packets.

Firstly consider the experiment depicted by figure 9.4 that shows the mean error when estimating the utilization using different packet sizes (40B,100B,500B,1000B and 1500B). As the figure shows the optimal packet size is 1500B (this result agrees with the conclusions reached in [218]). As the figure shows when we use packet sizes of 40 and 100 bytes the mean error is larger (up to 0.08) than when we use packet sizes of 500, 1000 and 1500 bytes. In fact, these packet sizes have a very similar error, where the maximum mean difference is $5.5\,10^{-3}$.

Secondly consider figure 9.5 that shows the accuracy of the regular packet trains when using random packet sizes, randomly chosen (uniformly) between 40 and 1500 bytes. As the figure shows the accuracy is not severely impacted. By analyzing the results we notice that, for small rates (less than 5Mbps), and when the link is near congestion (90%), the estimation of the utilization is inaccurate. This can be seen in the long tail of the CDF. This is because for such small rates, with random packet sizes, the packet train has very few packets able to congest the link and thus, the utilization is underestimated.

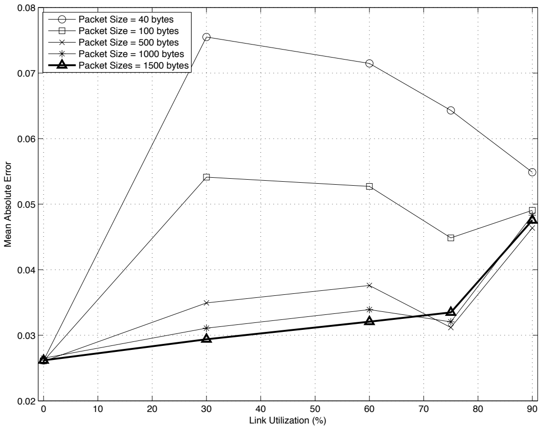


Figure 9.4: Mean absolute error using different packet sizes

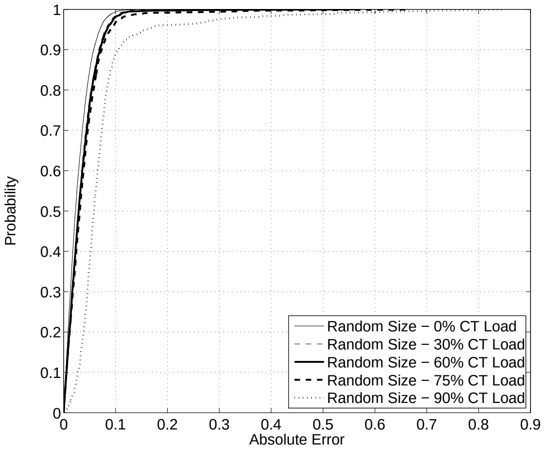


Figure 9.5: Absolute error when estimating the utilization with random packet sizes

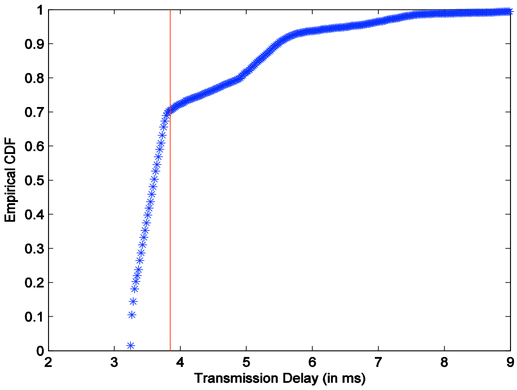


Figure 9.6: Empirical CDF of the transmission delay experienced by a Poisson flow of1 Mbps crossing an IEEE 802.11 wireless link (testbed)

Summary

At this point we have shown that in order to estimate the utilization in a path the packet trains of 200 packets, with exponential inter-departure time and with 1500 bytes as packet size, provide a good trade-off between accuracy and intrusiveness. We have also seen that random packet sizes does not have a significant impact in the accuracy of the estimations.

9.3.2 Estimating the Utilization in a Wired-Cum-Wireless Scenario

Estimating the utilization of a network path based on minimum delay observations cannot be used when the packets traverse an IEEE 802.11 wireless link (eq. 9.5). Two identical packets crossing such a link may suffer different transmission delays even when they are not queued before transmission. The access mechanism defined in the IEEE 802.11 protocol forces every packet being transmitted to wait for a variable random amount of time (i.e. backoff) in order to prevent transmission collisions. Such additional random delay turns the minimum delay as an erroneous estimator of an empty queue and thus, equation 9.5 would overestimate the probability that a packet is queued in the end-to-end path.

Figure 9.6 illustrates this by showing the empirical cumulative distribution function (CDF) of the transmission delay of a Poisson flow of 1Mbps when crossing an IEEE 802.11 wireless link (configured to transmit at 11Mbps phy rate). The figure shows how the minimum delay has a very low probability of occurring even though the wireless link is poorly utilized (approx. 20%).

However, as previously noted in [102], the CDF of the delay presents two differentiated regions. The authors showed how the turning point between the two regions (they named it knee of the CDF) is related to the available bandwidth.

A more detailed study of the CDF reveals that this knee is, in fact, related to the values that the backoff distribution of the 802.11 protocol gets. Indeed, when an 802.11 card starts transmission of a packet and enters backoff it raises a random value β between 0 and $CW_{MIN} - 1$ and delays the transmission a total of β time slots. The turning point in the figure coincides with the value $min\{D\} + (CW_{MIN} - 1) \times TimeSlot$ (e.g. $min\{D\} + 31 \times 20\mu s$ in 802.11b spec).

Based on these observations we propose to set this value as a threshold (red vertical line in the figure 9.6) to detect the utilization of the network path. Instead of using $min\{D\}$ as an indicator of non queued delays we use $min\{D\} + (CW_{MIN} - 1) \times TimeSlot$. Formally, we rewrite equation 9.5 as:

$$\widetilde{u} = \frac{\|\{d_i > (min\{D\} + (CW_{MIN} - 1) \times TimeSlot) | d_i \in D\}\|}{\|D\|} \tag{9.6}$$

Eq. 9.6 is the basic estimator of the utilization. This approach is fundamentally different from the one in ProbeGap [102]. The assumption in their approach is that the capacity of the wireless link is known but the threshold is unknown. They propose a methodology to find the threshold based on the detection of a "knee" in the CDF. This approach is hard to implement when the load of the wireless link is high, as the CDF does not show such clear properties as before (see figure 9.7). Also the approach in ProbeGap relies on probing the wireless link at a very low rate which may lead to low convergence rates of the estimation when the wireless link is medium or largely congested.

The threshold defined in the previous section can underestimate the utilization of a network path when it contains one or more high speed wired links (e.g. Gigabit Ethernet links). The following observation illustrates this fact. The time to transmit a packet of 1500 bytes in a Gigabit Ethernet link is in the order of tens of microseconds, which is in the same order of magnitude of the 802.11 slot time. As a consequence, probe packets being queued in wired links of the path may in the end suffer a delay lower than the threshold defined above, and thus, be categorized as non-queued. It is

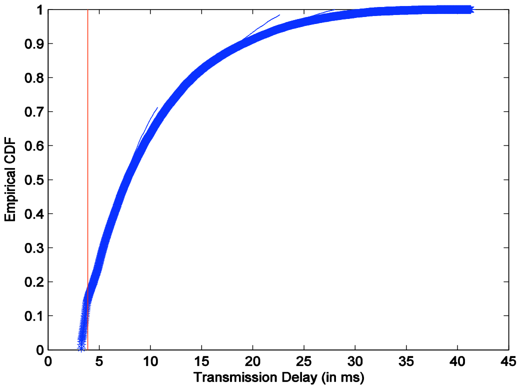


Figure 9.7: Empirical CDF of the transmission delay experienced by a Poisson flow of 5 Mbps crossing an IEEE 802.11 wireless link (testbed)

worth to note that this occurs only in high-speed links that have less probability of being the bottleneck link (i.e. a 2Mbps DSL link is more likely the bottleneck than a 1Gbps link).

In order to identify those packets that have been queued in the wired part of the network from those that have not, we use the following consideration. The backoff random value is taken, as defined in the standard, from a discrete uniform random distribution. This implies that backoff induced delays are discrete values at multiples of the time slot defined in the IEEE 802.11 spec (e.g. $20\mu s$ for 802.11b). The proposal here is to identify non-queued packets as those packets that have suffered a delay equal to the minimum delay ($min\{D\}$) plus a multiple of the slot time (up to the backoff threshold defined above). Formally, if we define the following filtering method:

$$g(x, x_{MIN}) = \begin{cases} 0 & \text{if } x \in \{x_{MIN} + k \times TimeSlot\} | k \in \{0, 1, \cdots, CW_{MIN}\} \\ 1 & \text{if otherwise} \end{cases}$$

The utilization of the end-to-end path can then be estimated as follows:

$$\widetilde{u} = \frac{\sum_{i=1}^{N} g(d_i, \min\{D\})}{N} \tag{9.7}$$

Eq. 9.7 is our Enhanced estimator of the utilization. It is worth noting here, however, that practical applicability of this enhanced solution may require a performance of the networking devices difficult to achieve in common off-the-shelf hardware and software solutions.

9.4 Design of W-Path

As we have seen from the previous section the AB can be estimated by finding the rate of the probe traffic at which the utilization of the network is 1 (turning point). In order to estimate the utilization of a path, given a probing traffic rate, W-Path sends Poisson sequences (packet trains) and measures the end-to-end delay of the probing packets. Poisson sequences are chosen, the packet train length is of 200 packets and the packet size is set at the start of the measurement. As previously shown, the capacity of an IEEE 802.11 link highly depends on the size of packets transmitted [105]. The W-Path accepts any probing size in order to account for this issue.

In order to estimate the parameters a and b of equation 9.3 W-Path implements a binary search algorithm that runs until it identifies the rate at which the utilization becomes 1 (the turning point in figure 9.1). The binary search has as input parameters an upper (MAX_{RATE}) and lower bounds (MIN_{RATE}), that is the range of the binary search. In our case, the lower bound is 0.1Mbps while the upper bound needs to be set by the user. Assuming a wired-cum-wireless scenario the upper bound can be safely set to 54Mbps. The time cost of the binary search algorithm is $O(\log_2(N))$ (i.e. $O(\log_2(MAX_{RATE}))$ in our case) until converging to the AB value.

Table 9.1: W-Path's algorithm

```
program W-Path
define $MAX_{RATE}$
define $ACC$
$CBW = 0.1Mbps$
$AB = 0Mbps$
$MIN_{RATE} = 0$
$MAX_{UTIL} = 1 - \frac{1}{PTLEN}$
while $((MIN_{RATE} - MAX_{RATE}) \geq ACC)$
    $UTIL = SendPT(CBW)$
    if $(UTIL \geq MAX_{UTIL})$
        $MAX_{RATE} = CBW$
        $AB = CBW$
    else
        $MIN_{RATE} = CBW$
    endif
    $CBW = MIN_{RATE} + \frac{MIN_{RATE} - MAX_{RATE}}{2}$
endwhile
```

Finally, we need to set the expected accuracy (ACC) and the threshold for the utilization (MAX_{UTIL}), in our evaluation we used 0.5Mbps and 0.95 respectively. Since we use finite Poisson-probing, packets may have a large inter-departure time. Even if the sending rate matches the AB, the probability that at least some packets are not queued is not negligible. It is worth to note that W-Path is an heuristic-based tool.

9.5 Design of AKBest

Section 9.3.1 has shown that poisson packet trains of 200 packets and 1500 bytes accurately estimates the utilization of a wired path (eq. 9.5). W-Path estimates the available bandwidth (parameters of eq. 9.4) by using a binary approach. Although this is a good approach for wired-cum-wireless paths we propose a different one for wired paths. In this case we apply Kalman Filters [192]. The Kalman Filters (KF) are an efficient recursive filter that estimates the state of a linear system from a series of noisy measurements. With KFs we are able to produce an estimation per measurement and filter noisy (erroneous) measurements. This means that KFs allows us to create a continuous monitoring tool able to track the AB and improve the overall accuracy.

The Kalman Filters

The Kalman Filters (KF) are able to estimate a system defined by a state vector x, affected by an input u through noisy measurements. In our case the network is our system and the noisy measurements are the estimations of the utilization. The system is also affected by a noise w and the measurements have a noise v. Then the system is governed by the linear stochastic difference equation:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{9.8}$$

With a measurement z that is:

$$z_k = Hx_k + v_k \tag{9.9}$$

Where the subscript k refers to the discrete time and A relates the state of the previous time step $(k-1)$ with the state of the new time step. Similarly B relates the control input to the state x while H relates the state with the measurement. Then the KF estimates the process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. The KF algorithm has two steps, in the first step (”time update“) the filter projects forward in time the state of the system and obtains an *a priori* estimate. In the second step (”measurement update”) the filter uses a new measurement to correct the *a priori* estimate to produce an improved *a posteriori* estimate. After each time and measurement update pair, the process is repeated with the previous *a posteriori* estimates used to project or predict the new *a priori* estimates. This recursive nature is one of the main advantages of the Kalman Filters. The KFs assume that the system is linear and that the system noise w and the measurement noise v are Gaussian and independent. We refer the reader to [192] for further details on Kalman Filtering.

In our case the state vector x that describes the system represents our linear model (the parameters of the sloping straight line from eq. 9.3):

$$x = \begin{pmatrix} a \\ b \end{pmatrix} \tag{9.10}$$

As it has been seen in the previous subsection our system is linear. We drop the input u (and consequently B) because in our particular case the network is affected by the intensity of our probe traffic *and* the cross-traffic. As we cannot estimate the intensity of the cross-traffic we do not use this particular parameter. In addition we drop A (i.e. $A = I$) because the state of the previous time step of the network will be the same as the state of the new time step.

Thus the following equation governs our system:

$$x_k = x_{k-1} + w_{k-1} \tag{9.11}$$

The measurements are governed by equation 9.9. We define H as:

$$H = \begin{bmatrix} r & 1 \end{bmatrix} \tag{9.12}$$

This way the measurements z (eq. 9.9) are seen by the KF as the actual utilization of the system under our probe-traffic load. The measurements z are in fact the estimations of the utilization. Finally the predictor equations defined by the KFs in our particular case are:

$$\widetilde{x}_k^- = \widetilde{x}_{k-1} \tag{9.13}$$

$$P_k^- = P_{k-1}A^T + Q \tag{9.14}$$

And the corrector equations are:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \tag{9.15}$$

$$\widetilde{x}_k = \widetilde{x}_k^- + K_k (z_k - H \widetilde{x}_k^-) \tag{9.16}$$

$$P_k = (I - K_k H) P_k^- \tag{9.17}$$

Where the "minus" superscript refers to the *a priori* estimates (before the measurement correction). P is the estimate of the error covariance matrix, its value will be updated by the KF each time step. K is the Kalman gain, a very important parameter of the KF. This gain is computed (in each time step) in eq. 9.15 and weights the new measurement with the *a priori* estimate in eq. 9.16. Finally Q and R represent the process and measurement noise covariance respectively. Q, the process noise covariance, is a 2x2 matrix that represents the variability of the system. This value must be set manually and it is a key parameter when considering the behavior of the KF. A high Q means that the KF will consider the prediction as less accurate while the measurements will be considered as very accurate. Therefore the KFs will set the Kalman gain accordingly and each new measurement will be weighted heavier. Low values for Q mean the opposite. We will come back to this in the results' section.

AKBest

Table 9.2 details the algorithm used by this tool to estimate the AB. The algorithm uses the following initial parameters: MAX_{RATE} which is an upper bound for the rate of the probe traffic and INTERVAL which is the measurement interval, x which represents the initial guess of the AB and Q, which is the covariance matrix of the process noise.

AKBest sends probe traffic at a random rate (uniformly distributed $[0, MAX_{RATE}]$ and estimates the utilization. Then it uses the KFs to estimate the parameters of the linear model (a and b) and finally it produces an estimation of the AB using eq. 9.3.

The algorithm can be easily implemented. It does not require time synchronization because it estimates the utilization by comparing the OWD of the packets (note that the actual value of the OWD it is not needed). It does not require the accurate computation of the dispersion of the packets. Therefore it may not suffer from the

Interrput Coalescence issue described in [193]. Since KFs are recursive, it does not require storing past values. The main drawback (and advantage) of the algorithm is that some parameters need to be set, evaluated and tuned.

Table 9.2: AKBest's algorithm

```
program AKBest
$MAX_{RATE}$ as CONSTANT
$INTERVAL$ as CONSTANT
$x = \begin{pmatrix} \widetilde{a} \\ \widetilde{b} \end{pmatrix}$
$Q = \begin{pmatrix} 10^{-5} & 10^{-7} \\ 10^{-7} & 10^{-2} \end{pmatrix}$
while(TRUE)
R = RandomRate($MAX_{RATE}$)
u = SendPacketTrain(r)
$\widetilde{x}_k^- = \widetilde{x}_{k-1}$
$P_k^- = P_{k-1}A^T + Q$
$H = \begin{bmatrix} u & 1 \end{bmatrix}$
$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$
$\widetilde{x}_k = \widetilde{x}_k^- + K_k(z_k - H\widetilde{x}_k^-)$
$P_k = (I - K_k H)P_k^-$
$AB = \frac{1 - x.\widetilde{a}}{x.\widetilde{b}}$
r = Variance(AB)
Wait(INTERVAL)
end
```

9.6 Design of PKBest

Figure 9.2 presents the main architecture of PKBest. The algorithm used to infer congestion is the same that for AKBest. However, as stated before, the main challenge then is that it cannot relay on any given rate or pattern of the data-traffic. That is, AKBest's algorithm requires 200 consecutives packets exponentially distributed, we refer to these packet trains as *valid*. In the next subsection we investigate if the *valid* packet trains, as defined in section 9.3.1, are present in Internet traffic. Taking this into consideration PKBest works similarly to AKBest. The receiver needs the delay of ρ consecutive packets. Then it processes these packets and extracts '*valid* packet trains. Then for each *valid* packet train, it estimates the utilization, the results are fed into the Kalman Filter, and it computes the AB. Since our methodology does not need the actual delay, but the *relatives* delays, it does not require clock synchronization.

It is important to remark that the delays of the packets can be obtained in several ways. First a generic passive measurements infrastructure can be used [194]. This type of architectures deploy two points of capture, one at the sender node and one at the receiving node. The points of capture, for each captured packet, send a hash and the timestamp of the packet to a central processing unit. This unit matches the hashes and extracts the delays of the packets. A different approach can be used using In-Line Measurements [195]. This method defines an IPv6 extension header that includes a timestamp and that can be used to compute several QoS metrics. This can be easily ported to IPv4 and a timestamp can be added in a special header after the transport header or in a new IP header using tunnelling. The receiver node always removes these headers. Since some IP packets already have the largest size (1500 bytes in Ethernet links) we cannot always add a new header. In these cases the timestamps can be included into the next packet that is smaller than the MTU (along with a special identifier).

PKBest considers this latter case for its operational procedures. The amount of timestamped packets by the sender node (ρ) must be enough to extract at least one *valid* packet train at the receiver node. In the next section we analyze the number of *valid* packet trains present in several Internet data-traces and we discuss possible values for ρ.

9.6.1 Applicability and Limitations of PKBest

In this section we analyze the number of *valid* packet trains present in Internet traffic. Additionally we evaluate the main limitations of PKBest.

Analysis of public data traces

In order to study if our tool will find such packet trains in real traces we have analyzed four different public NLANR (National Laboratory for Applied Network Research) data traces [196]. These traces, well-known in the passive measurements research community, have been collected from a variety of links at different research networks. The traces are public and were anonymized.

Specifically we have analyzed the CESCA-I, SanDiego-I, NCAR-I and Auckland-VIII data traces. It is important to note that among all the public data traces, we have analyzed all the traces that contained contiguous packets. Sampled traces are not useful for our study.

We have processed the traces in the following way. We consider a packet train as 200 consecutive packets within each trace. For each packet train, we determine if it can

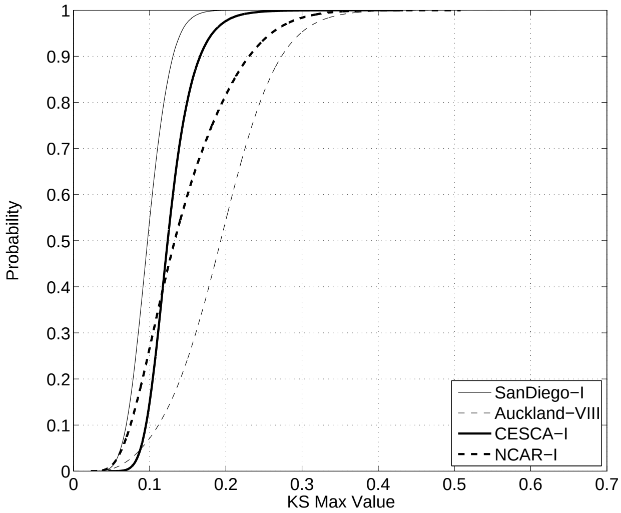


Figure 9.8: KS Test for the NLANR traces

be considered as exponentially distributed and we compute the number of *valid* packet trains per second.

Figure 9.8 shows a CDF of the Kolmogorov-Smirnov (KS) test [215] against a theoretical exponential distribution for each packet train. The critical value for this test, with a 95% confidence interval, is 0.096. This means that packet trains with a KS value below or equal 0.096 can be considered as exponentially distributed and thus, *valid* for our methodology. As the figure shows all the traces contain exponentially distributed packet trains: Auckland-VIII (8.7%), CESCA-I (17.6%), NCAR-I (26.5%) and SanDiego-I (53.2%). This variation of these results may arise from the different types of applications that each link is supporting.

Figure 9.8 also allows us to discuss the value of ρ. For the SanDiego-I data trace if we timestamp 400 packets we will likely have, at least, one *valid* packet train. For the NCAR-I and CESCA-I we should timestamp 1000 packets. Finally for the Auckland-VIII trace we should timestamp around 20000 packets.

Regarding the ratio of *valid* packet trains per second, Table 9.3 shows the results. As the table shows, in the worst case, the Auckland-VIII data trace, our tool would find a *valid* packet train each 3.376s. This means that PKBest can produce an estimation (roughly) each 3 seconds. We believe that this resolution is enough for many

Table 9.3: Ratio of *valid* packet trains per second

Data Trace	Min	Mean	Max	Std.Dev
Auckland-VIII	0.001	0.070	3.376	0.031
CESCA-I	0.084	0.231	0.633	0.051
NCAR-I	0.004	0.095	0.874	0.072
SanDiego-I	0.13	0.037	1.523	0.036

applications. For instance PathLoad [95] (considered as one of the most accurate tools [190]) produces an estimation each 10-100 seconds, depending on the scenario [190]. Taking into consideration that the ratio of *valid* trains per second is high, we use 4 *valid* trains to produce an estimation.

Evaluation of the Limitations

In the previous subsection we have analyzed different data traces and evaluated the number of *valid* packet trains present. The remaining parameter to evaluate is the rate of these packet trains. This is a key parameter when considering the accuracy of our methodology. Let's consider figure 9.9 as an example. In this case our tool is processing several packet trains to estimate the utilization. As eq. 9.3 states (and the figure shows) there is a linear relation between the rate of these packet trains and the estimated utilization. Our tool feeds the Kalman Filter with the estimations of the utilization and the rates. In turn the Kalman Filter estimates a and b (eq. 9.3). Finally using eq. 9.4, we estimate the AB. The main concern, in this case, is that the rates of these packet trains are very low (0-10Mbps in the figure) compared to the actual AB (80Mbps). This means that the error when estimating a and b will be "projected" to the AB point, producing a larger error. We can conclude that, the closer we operate to the actual AB, the more accurate the estimations will be.

In this subsection we evaluate the relation between the distance to the AB of the rates of the packet trains and the achieved accuracy. First we need to consider all the possible path scenarios. In section 9.2 we assumed that any path loaded with any cross-traffic can be modelled using eq. 9.3. This means that any scenario can be represented by two parameters: a are b. The valid range of values for b is $[0, 1]$ since a negative utilization is not possible. Consequently given an AB of α, $a = \frac{(1-b)}{\alpha}$. Thus all the

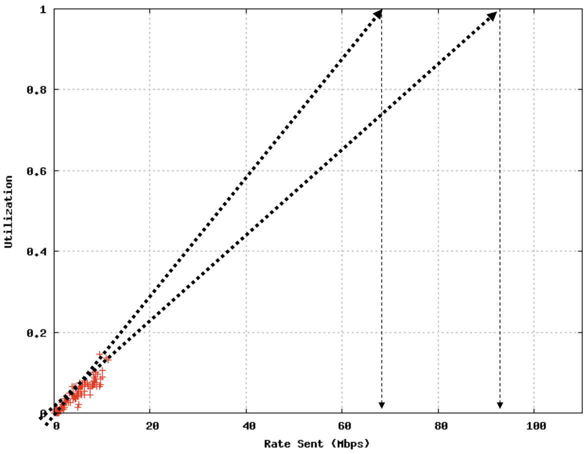


Figure 9.9: Limitation of our tool

possible paths are given by the following equation:

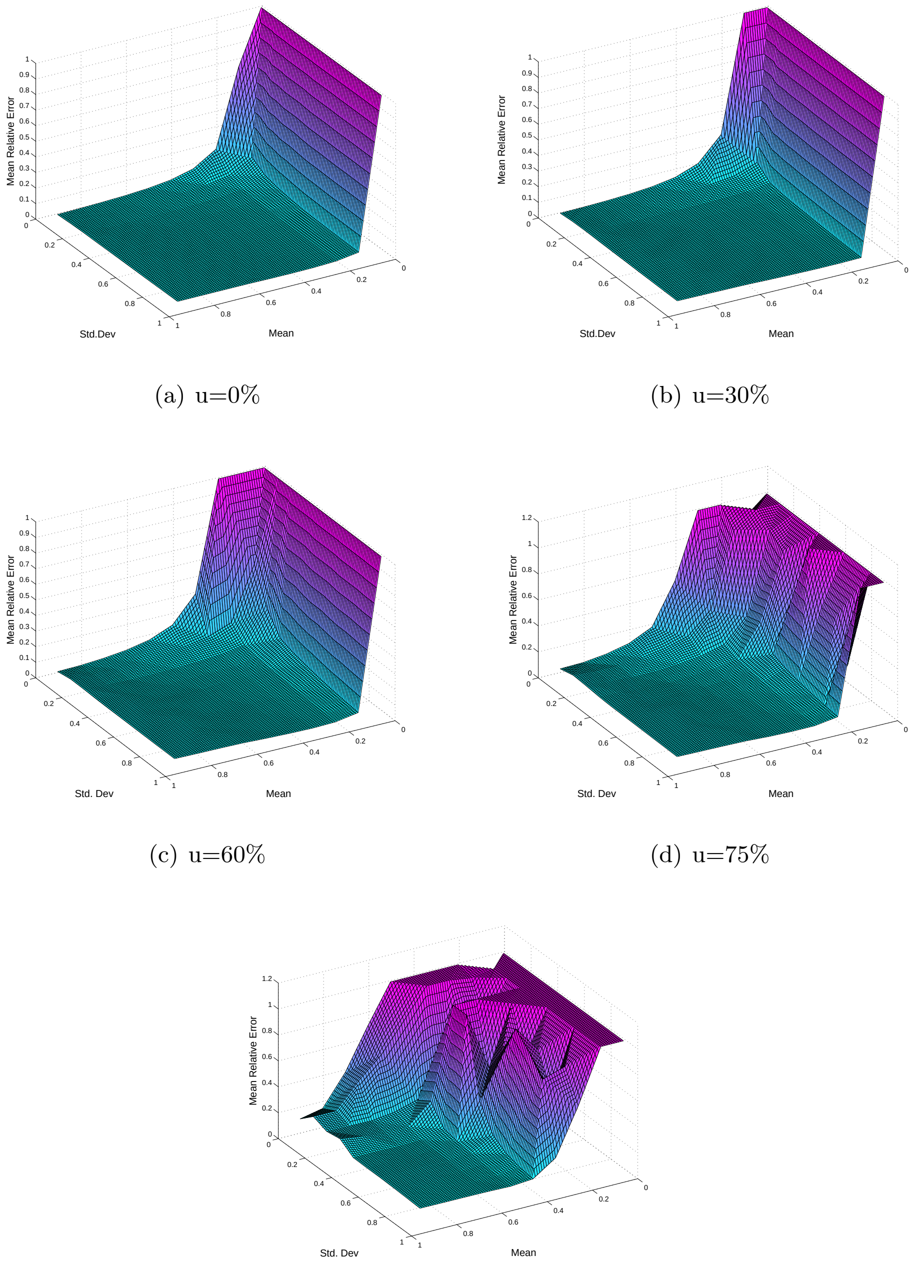
$$g(\alpha) = \{(a,b)|b = [0,1] \bigwedge a = \frac{(1-b)}{\alpha}\} \tag{9.18}$$

Second we need to model the rates of the received packet trains. This is the same as modeling the bandwidth of aggregated traffic. Bandwidth is assumed as Gaussian in [197] based on the measurements of [198] (and the references therein). Specifically the measurements show that the vertical aggregation of at least 25 users, with an aggregate average traffic rate of 25Mbps, is a good fit with the Gaussian model in time scales that are longer than 128msec.

Finally we need to model the error of the estimations of the utilization. Considering all the experiments carried out in section 9.3.1, the error can be modeled as a Gaussian distribution. Specifically it can be modeled as $N(0, 0.03)$.

Having modeled the path, the rates of the packet trains and the error of the estimations of the utilization, we perform the following experiment using Matlab. We consider the paths where $\alpha = 100$ and $b = \{0, 0.3, 0.6, 0.75, 0.9\}$. For each path, we fed the Kalman Filters with packet trains at rates distributed as $N(\mu, \sigma)$ We consider the following ranges: $\mu = [0.1\alpha, \alpha]$ and $\sigma = [0.1\mu, \mu]$. It is important to note that, for each rate, we compute the estimation of the utilization affected by an error modeled as: $N(0, 0.03)$.

Figure 9.10 presents the results of the experiments. The x-axis represent the mean of the rates while the y-axis its standard deviation. Note that the mean is related to



(a) u=0%

(b) u=30%

(c) u=60%

(d) u=75%

(e) u=90%

Figure 9.10: Evaluation of the limitations of our methodology

the AB (real mean=$\mu \times \alpha$), similarly the standard deviation is related to the mean (real std.dev=$\sigma \times \mu$) Finally the z-axis presents the accuracy of the final estimations of the AB. Specifically we present the mean relative error ($\epsilon = \min(1, \frac{abs(\widetilde{ab}-\alpha)}{\alpha})$) of the accuracy. We plot a figure for each value of b.

For low values of b the figure shows that the accuracy is not affected by σ unless it is 0. This is because even a small amount of variability in the rates of the packet trains is enough for the Kalman Filters. Regarding the mean, as expected, it impacts the accuracy of the estimation in certain cases. When the path is below 60% and the mean rate above 20% of the AB, then the achieved accuracy is bounded at 0.01. However as the utilization increases (75% and 90%) the error, for small mean values increases. In this case the accuracy is bounded at 0.01 when the mean rate is above 50% of the AB. In fact in theses cases (75% and 90%) σ also affects the achieved accuracy. This is also an expected results since a larger σ means that some trains are sent at a closer rate to the AB.

This analysis helps us identifying the main limitations of PKBest. When the path is not congested (below 60%) almost any packet train rates are valid to achieve good accuracy. When the path is near congestion (above 75%), then the packet train rates must be around 50% of the AB. It is worth to note that, when the path is near congestion the AB decreases. For instance when a 100Mbps link is near congestion, let's say 75%, the AB is 25Mbps. This means that the mean rate of our packet trains should be around 12.5Mbps. Thus, as the path utilization increases, the AB decreases and the required rates of our packet trains also decrease.

Table 9.4: Rates of the *valid* packet trains (in Mbps)

Data Trace	**Min**	**Mean**	**Max**	**Std.Dev**
Auckland-VIII	0.01	7.67	330.77	10.38
CESCA-I	84.38	230.61	633.63	50.89
NCAR-I	4.21	95.87	870.90	75.32
SanDiego-I	13.34	37.39	152.15	108.17

Finally table 9.4 shows the mean rates of the *valid* packet trains present in each data trace. Since these traces have been collected during large periods of time (days) they show a very large variability regarding the rate.

9.7 Variability of the process under measurement

This chapter presents three different tools aimed to estimate the available bandwidth. The common methodology presented in this chapter assumes this process as constant during the measurement time-scale and the tools presented here target the average over such period of time. The measurement time-scale of AKBest and PKBest is in the order of ms (i.e. one packet train) while for W-Path is in the order of seconds (i.e. N packet trains since it uses a binary search algorithm). Our experimental results (see the next section) show that W-Paths uses, on average, 5 packet trains to produce an estimation. It is worth to mention that the size of the probe packets and the length of the packet trains used in the common methodology can be set to any value.

However and as pointed by *C.Dovrolis* in [197] the available bandwidth is a time-varying quantity. It is important to remark that all previous work [93; 94; 95; 96; 97; 98; 99; 100; 101] (except [197]) assumes this metric as constant over the measurement time-scale, even when applied to wireless scenarios [102; 103; 208]. In this section we discuss the validity of this assumption.

In wired scenarios there is mainly one source of variability of the process under measurement: the variability of the cross-traffic. In [197] the authors study how this impacts the variance of the available bandwidth using Internet data traces. Their results show that three factors play an important role in the variation range of the available bandwidth: traffic load, number of flows and the rate of the flows. This variation is mostly present in sub-second scales. Furthermore the authors present a tool intended to estimate a given percentile of the available bandwidth distribution rather than its average.

Regarding wireless scenarios there are two additional sources of variability. The first one is related with the dynamic rate adaptation present in IEEE 802.11 links [104]. This mechanism allows a radio link to switch between different rates (e.g. 1,2,5.5 and 11Mbps for 802.11b) by switching modulation schemes depending on channel quality, which can vary dynamically due to environmental changes. In [208] the authors evaluate empirically the variability of the transmission rate of a 802.11g link in an stressful situation. Their results show that, in a dynamic scenario, the adaptation mechanism of such links change the transmission rate in the order of *tens* of seconds (20-40s). This time-scale is close to the measurement period of W-Path (seconds) and could impact its accuracy. In this case a good solution would be to use smaller probe packets in order to further reduce the convergence time.

Secondly specific conditions of the radio channel (slow/fast fading) can also affect the capacity of the link and hence, the available bandwidth. The variability of such conditions is related with the coherence time [229]. This is the time over which a propagating wave may be considered coherent, that is, the time interval within which its phase is, on average, predictable. Typically researchers assume values in the order of ms for this parameter (i.e. see [230; 231]). Hence this source of variability is in a much smaller time-scale than the measurement duration.

Additionally the adaptation mechanism changes the transmission rate depending on the characteristics of the radio channel. As we have seen this occurs in the order of tens of second. Thus it is reasonable to assume that the physical parameters of the wireless link can be considered as constant (from a layer-3 point of view) at this time scale. Otherwise, if the characteristics of the radio link change abruptly and can impact the performance of layer-3 communications, the adaptation mechanism is triggered.

9.8 Performance Evaluation

This section presents a performance evaluation of W-Path, AKBest and PKBest. In all the cases the tools have been evaluated by simulation and its accuracy is compared with that of state-of-the-art tools.

9.8.1 Evaluation of the Accuracy of W-Path

This section presents a complete evaluation of the performance of W-Path. Specifically, we evaluate through simulation (see section 8.2 for further details about the settings) the accuracy and the convergence time of this tool. Additionally we compare the accuracy of W-Path to that of ProbeGap. We use the Basic algorithm because it is simpler to implement, later we discuss its performance with that of the Enhanced algorithm.

Figure 9.11 details our simulated scenarios. We have setup two different wired paths (I and II, upper and lower parts of the figure). Each wired link is loaded with three cross-traffic flows with Pareto inter-departure times (α=1.19). Each flow uses a different packet size (40, 576 and 1500 bytes). The amount of packets of each size is distributed as in the Internet [200] (40%, 10% and 50%). It has been shown that N

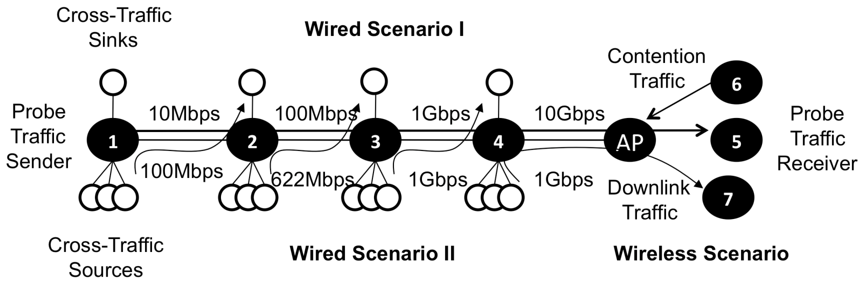


Figure 9.11: The simulated scenarios.

Pareto flows mimic the burstiness behavior of the Internet traffic [199]. With this setup we evaluate our tool under realistic wired paths.

The last hop of the path (AP) is the Access Point of a single-rate IEEE 802.11b link. In this link, node 6 generates uplink cross-traffic, this produces contention. Node 7 receives the downlink traffic generated at node 4. Finally node 5 receives the probe traffic (sent from node 1). The cross-traffic at the wireless link has exponential inter-departure times.

Table 9.5: CT in the wireless link (in Mbps)

	Simple Cross-Traffic		**Complex Cross-Traffic**	
Case	**CT Intensity**	**Measured AB**	**CT Intensity**	**Measured AB**
1	0	6.11	0	6.11
2	1	5.11	0.4	4.2
3	1	5.10	0.4	4.21
4	2	4.05	0.8	2.25
5	4	2.05	1.30	1.71
6	5	1.11	1.40	1.63

We have three sets of experiments. In the first set (Simple Wireless Cross-Traffic), we run W-Path in the wired scenario I without wired cross-traffic. The wireless link is loaded with fixed packet-size cross-traffic (1500 bytes). In this set, we also run ProbeGap, but only in the wireless link to compare the accuracy of both tools. In the second set, (Complex Wireless Cross-Traffic) the cross-traffic has different packet sizes as in the Internet [200]. The second set is intended to test our tool in a realistic

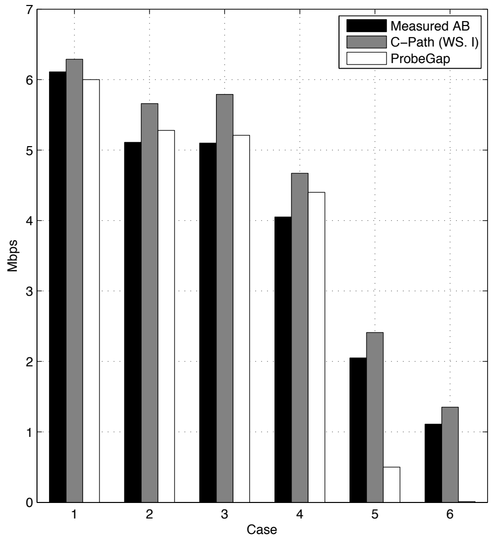


Figure 9.12: Results for the Simple Wireless CT

scenario. Finally, in the third set, the cross-traffic at the wireless link has different packet sizes but the bottleneck is at a wired link.

Simple Wireless Cross-Traffic

Figure 9.12 presents the mean AB estimation, for each case, of the first set of experiments. As the figure shows, W-Path's estimations agree with the measured AB in all the cases. Regarding ProbeGap's estimations, they also agree with the AB under low loads. When the cross-traffic intensity is high (cases 5 and 6) ProbeGap significantly underestimates the AB. This is because, in these cases, ProbeGap's heuristic is unable to find the knee of the delay's CDF. The CDF is too smooth, similar to the one shown in figure 9.7. It is worth noting here that this limitation was also pointed out in ProbeGap's evaluation [102].

Complex Wireless Cross-Traffic

In this set of experiments the wireless link is loaded with cross-traffic with different packet sizes. We test W-Path in both wired scenarios; the first one is at 30% of its capacity while the second one is at 50%. Our results (figure 9.13) show that W-Path is

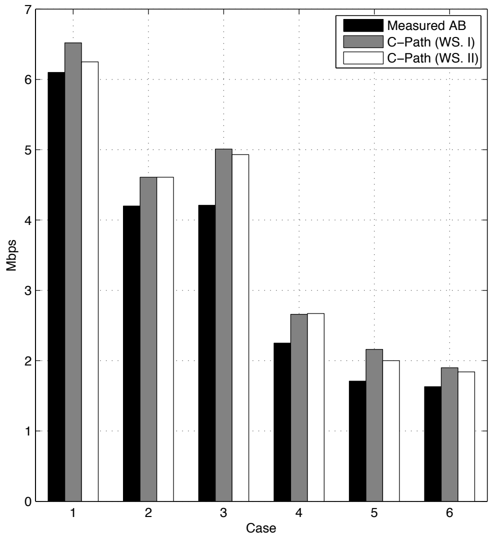


Figure 9.13: Results for the Complex Wireless CT

very accurate in all the cases. It gets no impact from the wired or wireless cross-traffic, even under heavy loads.

We have not run ProbeGap in this set of experiments. On the one hand it has not been designed for wireless-cum-wired scenarios (the previous experiments do not include wired cross-traffic). On the other hand, it is unable to produce estimations with variable packet size cross traffic since the CDF of the delay is too smooth in all the cases.

Basic vs. Enhanced Estimator

In this subsection we consider our Enhanced estimator and discuss its performance compared with the Basic one. We have repeated all the experiments mentioned before (Simple and Complex with both wired scenarios) using the Enhanced estimator. Both estimators have shown similar accuracy; the mean relative error when using our Basic estimator is 0.042 while for the Enhanced one is 0.036. As expected, when the bottleneck link is the wireless link, both estimators perform similarly.

Regarding the variability of our estimations, considering all the experiments, in the worst case, our tool using our Basic estimator has 0.18Mbps of standard deviation.

When using our Enhanced estimator has 0.14Mbps of standard deviation. We have produced 40 estimations per case. This shows that W-Path produces stable estimations.

The last parameter to evaluate is the convergence time of our tool. The mean convergence time of our tool, considering both estimators, is 22.09s and the maximum 34.02s. The standard deviation is 2.13s. W-Path sends (on average) 5 packet trains to produce an estimation. For instance, ProbeGap has a convergence time of approximately 50s.

Bottleneck at the wired path

Finally, in this subsection we evaluate the accuracy of W-Path when the bottleneck is located at the wired path. We test both wired scenarios (I and II) with complex wireless cross-traffic (case 3). Table II presents the results using both estimators. Specifically, it shows the mean AB estimated.

In the first two rows, the first link of the wired scenario was at 60% of its capacity. The last row presents a worst-case scenario where the first link of the wired scenario is at 98% of its capacity. As expected, the Enhanced version produces more accurate estimations than the Basic one since it is able to identify queuing at the wired hops. Nevertheless, the Basic's estimations have a reasonable low error.

Table 9.6: Estimations of the AB (in Mbps)

Case	$\widetilde{AB}$	**Basic Mean** $\widetilde{AB}$	**Enhanced Mean** $\widetilde{AB}$
Wired S.I (simple)	4	4.76	4.16
Wired S.I (complex)	4	3.59	4.11
Wired S.II (complex)	2	2.63	2.15

9.8.2 Evaluation of the Accuracy of AKBest

This section presents the evaluation of the accuracy of AKBest. Similarly to the W-Path's evaluation we have implemented AKBest in a simulator (NS2) and we have used four different scenarios with different number of links and different capacities. Our scenarios aim to represent the actual configuration of a backbone path, thus we simulate very high-speed links (up to 10Gbps). Usually these high-speed links are in the middle of the path, just like in a real Internet backbone path.

For each scenario we have run two different sets of experiments. In the first set we load the different scenarios with static cross-traffic rate. That is, the AB does not change during the experiments. The different links are loaded with non hop-persistent cross-traffic, this means that it interacts with our probe traffic on one link. With this set of experiments we aim to evaluate the accuracy of the tool under different loads. In the second set we load the different scenarios with variable cross-traffic rate. In this case the cross-traffic is hop-persistent and we aim to evaluate if the tool is able to accurately track the AB. It is important to remark that this is our main objective. Regarding the cross traffic the distribution for the packet sizes are as measured in the Internet: 50% (40 bytes), 10% (576 bytes) and 40% (1500 bytes) and the cross-traffic model is Poisson.

Table 9.7 summarizes the configuration of our experiments. For the static AB experiments we have three different cases. In the first case (case A) the path is not loaded with cross-traffic. This case is used as a best-case scenario. In the second case (case B) the path is at a half-load, this is a typical scenario. The last case (case C) is a worst-case scenario where at least one link is highly congested (up to 95%). This third case is used to evaluate the performance of the tool in an extreme situation and analyze its limitations. For the variable AB experiments we use variable cross-traffic rate. Since the tool is intended to collect long-term statistics of the AB we have simulated large variations of the AB (and thus, of the cross-traffic rate).

Finally, we compare the results of our tool with that of pathChirp [97]. pathChirp is considered as one of the best [190] available bandwidth estimation tools and uses the PRM model, this means that it congests at least one link along the path in order to estimate the AB. It is important to remark that it has also been shown in [190] that among the PRM-based tools pathChirp is considered as one of the less intrusive.

Results

On the one hand, figure 9.15 presents the results of the simulations for all the static scenarios (cases A,B and C). All the results are presented using Interquartile Range Boxes where the box represents the middle 50% of the estimations and thus, the line in the middle of the box is the median. The upper/lower whiskers extend to the minimum/maximum data point within 1.5 box heights from the bottom/top of the box. Finally the cross represents the mean of the estimations. Each case was simulated

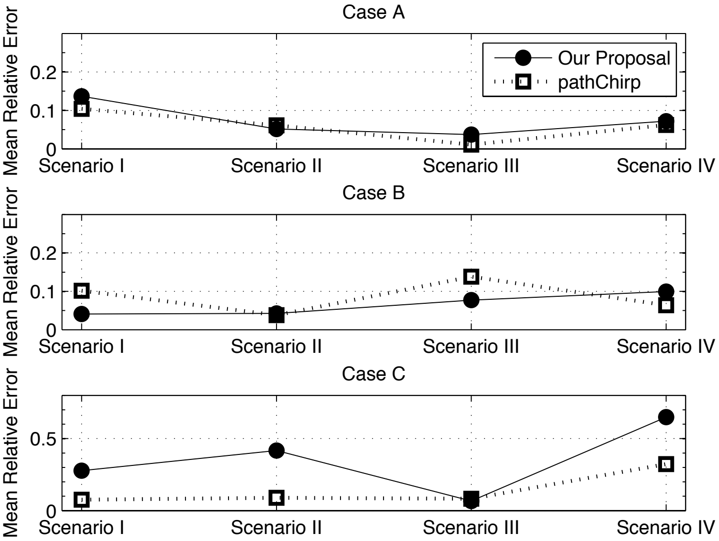


Figure 9.14: Mean error when estimating the AB for the static cases

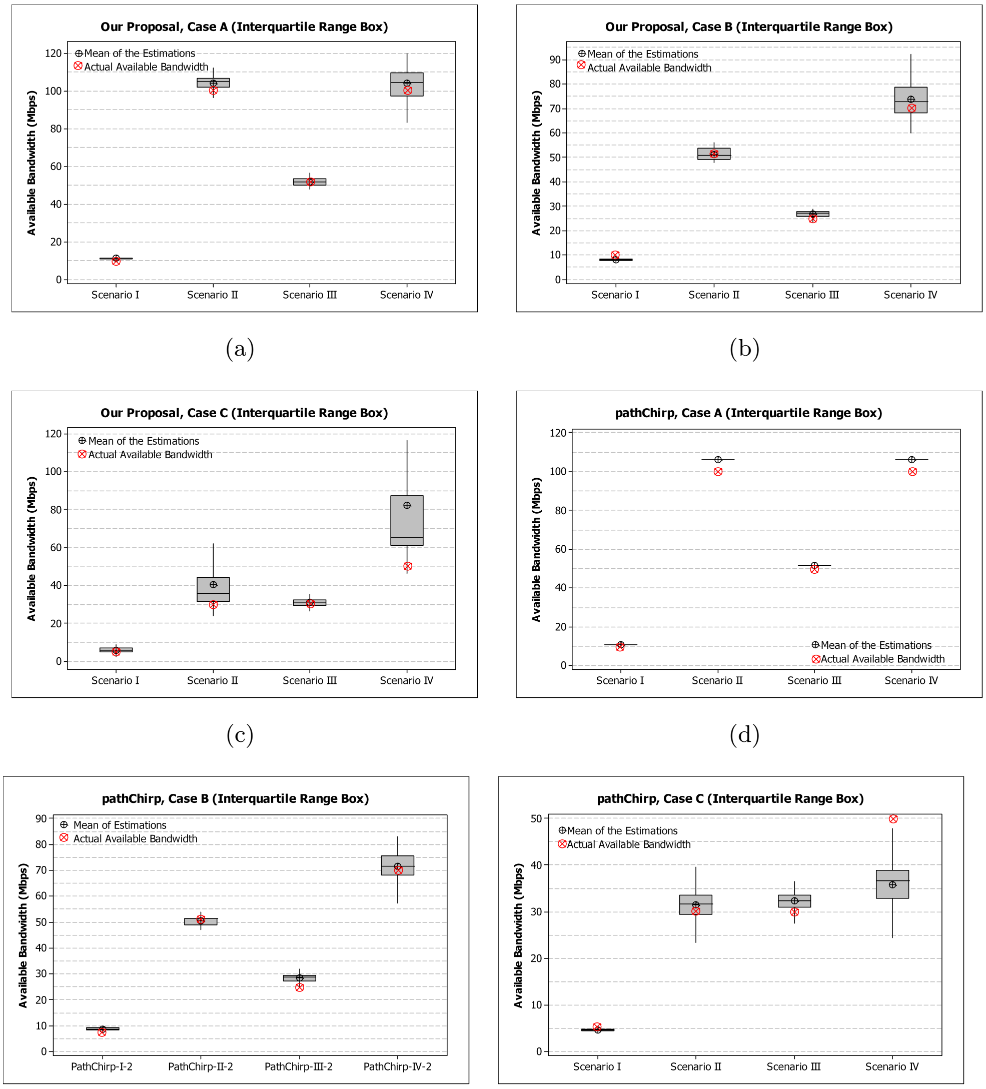
during 300 seconds for both tools. The (red) cross highlights the actual available bandwidth for each case and for each scenario. On the other hand, figure fig:IV-18 shows the mean relative error for all the different cases:

$$\epsilon = \frac{abs(\widetilde{AB} - AB)}{AB} \tag{9.19}$$

As both figures show both tools are very accurate for the case A (in all the different scenarios). In this particular case both tools show very good accuracy where the mean relative error is around 0.1. As it has been said before this is a simple case, from the figures we can also see that our tool has larger variability than pathChirp. This depends on the configuration of the covariance matrix of the process noise Q and can be tuned; we will come back to this later. However it is important to remark that this does not affect the accuracy of our tool.

Regarding the second case (B), where the network is at half load, both tools also show high accuracy with a similar mean error. We believe that this is the most common case in the internet and the mean relative error for the four scenarios is 0.065 for our tool and 0.085 for pathChirp.

Finally, for the third case (C) where the network is highly congested both tools show lower accuracy. Specifically the last scenario has a backbone link of 4976Mbps



(a)

(b)

(c)

(d)

(e)

(f)

Figure 9.15: Results of the simulations for the static cases

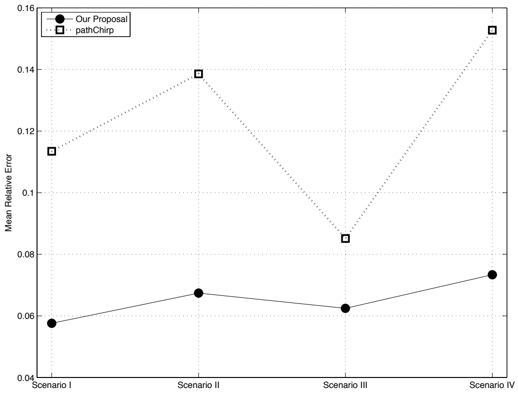


Figure 9.16: Mean error when estimating the AB for the variable cases

under a high utilization (0.95). This is considered as a worst-case scenario and it is not very common on the Internet. In this case the error of our tool is due to an incorrect estimation of the utilization. The problem is that when our tool is operating near the full utilization it is difficult for our Kalman Filters to estimate the slope of the straight line (in the mathematical model). This is because a highly congested link may drop packets, making difficult to accurately estimate the utilization. As the figure shows pathChirp's accuracy is also affected by the same reason. We believe that this is a tradeoff because we estimate the AB without sending packet trains at the same rate than the AB. This can be solved by designing a special estimator for this very special case (when the estimated utilization is very high). This special estimator should take into account losses. In any case if our tool detects that a link is highly congested it just needs to report it, this should be enough for upper layers (such as a routing algorithm) to make a decision, or to an ISPs to upgrade a link. The main benefit of our technique is that the impact on the performance of the network is lower.

Regarding the variable experiments, figure 9.16 shows the mean relative error for the different scenarios. In this experiment the AB has sudden variations and both tools need to rapidly adjust its estimations. As the figure shows our tool is more accurate than pathChirp, in this case the average error for pathChirp is 0.12 while for our tool is 0.06. It is important to remark that one of our objectives is to accurately track the changes of the AB and collect long-term statistics.

Summarizing our tool's accuracy is similar to that of pathChirp for the static experiments while it is better for the variable cross-traffic rate experiments. Our evaluation has also shown the main limitation of our tool in highly congested paths (the same applies to pathChirp). In addition we have seen that the covariance matrix of the process noise Q is a key parameter that can tune the behavior of our tool. This matrix represents the uncertainty of the process. Large values of Q will help our tool react quicker to large variations of the AB, however it will also increase the variability. Low values of Q mean the opposite. In fact, this parameter may be used to tune the behavior of our tool in different scenarios. If we want to monitor the AB at short time-scales it is better to use a large Q (it will react quicker) while if we want to monitor it at large time-scales it is better a low Q (it will be more stable). In this work we have used a large value of Q, thats why our tool has larger variability than pathChirp but it is more accurate for the variable experiments. We have used the same value of Q along all the experiments.

9.8.3 Evaluation of the Accuracy of PKBest

This section presents a complete evaluation of the accuracy of PKBest. In order to choose the evaluation scenarios we follow the methodology presented in [190] where the authors compare the performance of different AB estimation tools. Our evaluation is based on this methodology and on the evaluation of PathLoad [188]. Specifically we consider the cross-traffic load similarly to [188] while our evaluation scenarios are close to the ones depicted in [190].

In all the experiments each link is loaded with three different cross-traffic flows with Pareto inter-departure times ($\alpha = 1.19$). Each flow uses different packet sizes (40, 576 and 1500 bytes). The amount of packets of each size is distributed as in the Internet [200]: 50% (40 bytes), 10% (576 bytes) and 40% (1500 bytes). In addition we repeat the experiments using the same setup but with exponentially distributed cross-traffic for comparison. Each experiment is run for 600 seconds. In the last case of the evaluation, we use NLANR data traces as cross-traffic. With this latter case we aim to test our tool under a highly realistic scenario.

Regarding the data traces we test our tool using the NCAR-I and the CESCA-I traces. According to the parameters evaluated in section 9.3.1 the most suitable trace for our tool is the SanDiego-I while the less suitable is the Auckland-VIII. That is

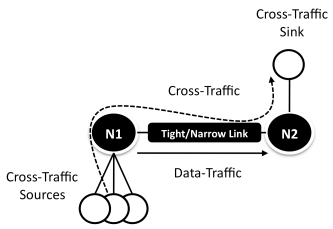


Figure 9.17: Single-Hop, Single-Bottleneck, same tight and narrow link scenario

why we choose these two traces to evaluate the accuracy of our tool. Since ρ does not affect the accuracy of our methodology but the amount of estimations per second, we timestamp all the packets (i.e. $\rho = \infty$). The value for the process covariance matrix Q (of the KFs) used in the evaluation is:

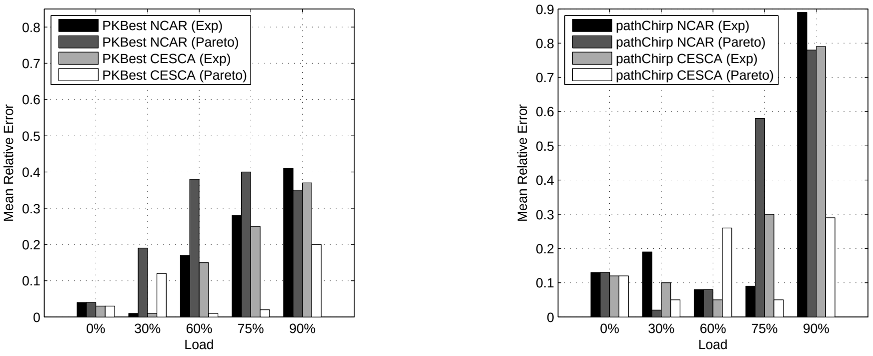
$$Q = \begin{bmatrix} 10^{-6} & 10^{-7} \\ 10^{-7} & 10^{-2} \end{bmatrix} \tag{9.20}$$

Finally for each experiment we compare the achieved accuracy of our tool with that of pathChirp. Again, we choose pathChirp because it is one of the less intrusive tools [190]. Since this is also one of the main advantages of our tool we believe that this is a fair comparison. pathChirp is evaluated under exactly the same scenarios and the same cross-traffic: the Pareto cross-traffic and the NLANR data trace. We have used the publically available implementation of pathChirp [201]. The authors of pathChirp claim that the parameters of pathChirp do not need to be set because it adapts them to the situation automatically. Therefore we use pathChirp's default parameters throughout the evaluation.

Single-Hop

In the first set of experiments we evaluate our tool in a single-hop scenario (Figure 9.17), this scenario represents paths on which our methodology is likely to encounter only a single congested link. In this set the cross-traffic has different loads at the bottleneck link {0,0.3,0.6,0.75,0.9} and the capacity of the bottleneck link is set accordingly {500Mbps,714Mbps,1250Mbps,2Gbps,5Gbps}.

Figure 9.18 shows the mean relative error. Unless noted otherwise we compute the error as ($\epsilon = \min(1, \frac{abs(\widetilde{ab}-ab)}{ab})$). As the figure shows the Pareto cross traffic has not



(a) PKBest's results

(b) pathChirp's results

Figure 9.18: Results for the single-Hop, single-Bottleneck, same tight and narrow link scenario

a noticeable impact on the accuracy of our tool (compared to the exponential cross-traffic). Our tool is based on a model that does not rely on a constant-fluid cross traffic and it is able to deal with the burstiness of the cross-traffic. Regarding the *data-traffic*, PKBest achieves a slightly higher accuracy when operating with the CESCA-I data trace. This is because, as has already been mentioned in section 9.6.1, the mean rates of the CESCA-I trace are higher.

The intensity of the cross-traffic at the tight link impacts the accuracy of PKBest. When the utilization of the tight link is high the slope of the linear eq. 9.4 is close to 0, and the error of the estimations is "projected". This has been analyzed and evaluated in section 9.6.1. Regarding pathChirp's accuracy it is severely impacted by the cross-traffic intensity. When the tight link is at 90% of its capacity the error of pathChirp's estimations are close to 1. In [190] pathChirp was evaluated under a similar scenario, specifically with a link load of 53%, and the mean relative error was between 0.1 and 0.25. In our case the closest scenario is when the link load is at 60% of its capacity, and the achieved accuracy of pathChirp in our case is between 0.08 and 0.27. Thus our results agree with that of [190].

Multi-Hop, Single Bottleneck, same tight and narrow link

In the second set of experiments we use a multi-hop scenario. Almost all the paths on the Internet are multi-hop. In this case the experiments are intended to evaluate the

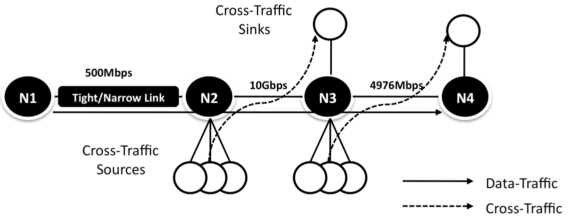
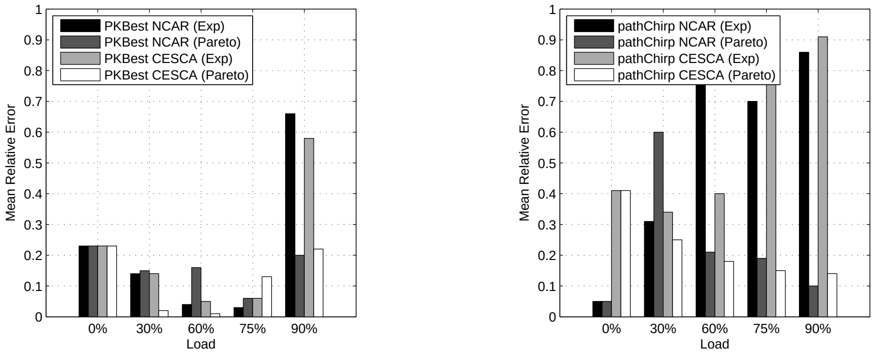


Figure 9.19: Multi-Hop, Single-Bottleneck, same tight and narrow link scenario



(a) PKBest's results

(b) pathChirp's results

Figure 9.20: Results for the multi-Hop, single-Bottleneck, same tight and narrow link scenario

accuracy of our tool when it is affected by non-tight link cross-traffic. In this case the tight link is located between N1-N2, the load at the links N2-N3 and N3-N4 varies: {0,0.3,0.6,0.75,0.9}. The AB is set to 500Mbps.

Figure 9.20 shows the results for this scenario. PKBest's estimates match the AB and the error is bounded to 0.27. Non-tight link cross traffic has not a noticeable impact in PKBest's estimates. This is because our model sees this cross traffic as a minor increase of b. However pathChirp is severely affected, especially when the non-tight link cross-traffic intensity is high. It is worth noting that these cases can be considered as extreme scenarios.

Multi-Hop, Single Bottleneck, different tight and narrow link

Many AB estimation tools have different accuracy depending on the location of the tight and narrow link. In this experiment we evaluate our tool in a scenario with

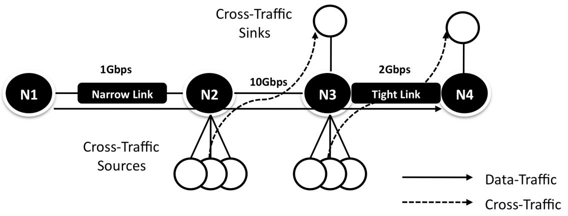
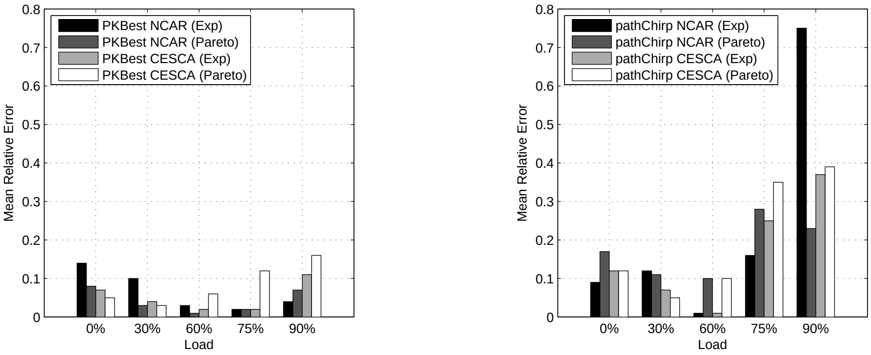


Figure 9.21: Multi-Hop, Single-Bottleneck, different tight and narrow link scenario



(a) PKBest's results

(b) pathChirp's results

Figure 9.22: Results for the multi-Hop, single-Bottleneck, different tight and narrow link scenario

different tight and narrow link (figure 9.21). This case may be very common on the Internet since an ISP access link that is shared among a large user population may have a lower AB. In this case the narrow link is between N1-N2 and the tight link is at N3-N4. The load at the link N2-N4 varies {0,0.3,0.6,0.75,0.9} while the utilization of the tight link is always 0.625 and thus, the AB=750Mbps.

Again figure 9.21 shows the results of these experiments. We can see that PKBest's estimates agree with the AB. This scenario does not affect the accuracy of our tool because the load at the tight link is reasonable low (62.5%) and, as we have seen earlier, it is not affected by non-tight link cross-traffic. Regarding pathChirp's it shows a larger error. A similar scenario has also been evaluated in [190] and the error of pathChirp increased by a factor of 2-3 compared to the scenario of figure 9.19. Again our results agree with the evaluation of pathChirp presented in [190].

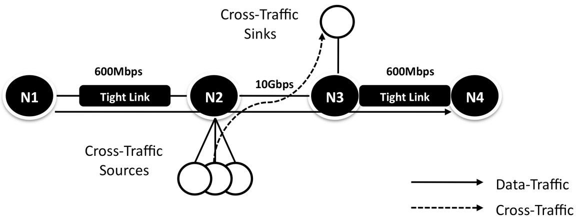


Figure 9.23: Multi-Hop, Multiple-Bottleneck, two potential tight links

Multi-Hop, Multiple Bottleneck, two potential tight links

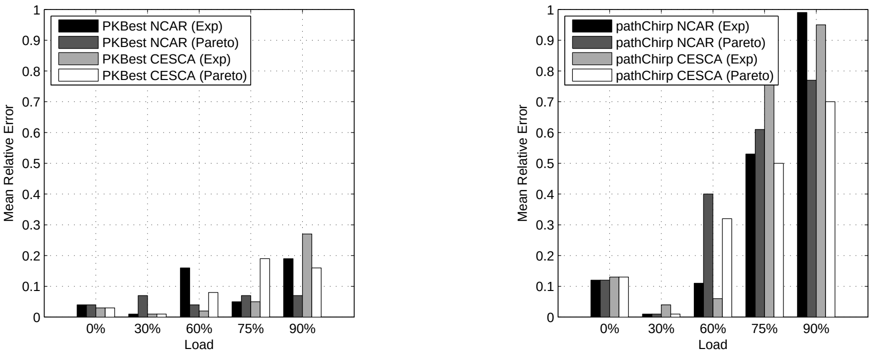
Finally most ABET assume the existence of only a single congested link on the path. For instance Spruce [98] (a PGM-based tool), needs the *a priori* knowledge of the capacity of the tight link. Therefore it may not perform well in this scenario. Additionally it is conjectured that PRM-based tools might underestimate the AB in the presence of multiple bottleneck links [190]. In order to study this scenario we simulate the topology of figure 9.23. With this setup we evaluate our tool in an scenario with one narrow link and two potential tight-links. On average the latter is the "tigher link" but the *data-traffic* experiences queuing at both links. In this case the utilization of the link N2-N3 is {0,0.3,0.6,0.75,0.9} and the utilization of the latter tight link is 0.083.

Figure 9.24 shows the results for these experiments. In this case PKBest shows a very good accuracy and the error is bounded to 0.17. In this scenario both 'tight links' have a low load and, as noted before, the non-tight link cross-traffic does not affect PKBest's accuracy. pathChirp's estimates also match the AB when the non-tight link is not congested.

NLANR data traces as cross-traffic

Finally in this set of experiments we setup a highly realistic environment (Figure 9.25). We evaluate the accuracy of PKBest in a path with 5 hops where each hop is loaded with a different NLANR trace as cross-traffic. This set of experiments is intended to evaluate the performance of our tool with realistic cross-traffic.

In this case each link is loaded with an NLANR trace as cross-traffic. The last link is the tight link and it is loaded either with Pareto or a NLANR data trace. Since some of the NLANR data traces have a low rate, we multiply the inter-departure times of the packets by a scale factor to increase its rate. This way the utilization in the last



(a) PKBest's results

(b) pathChirp's results

Figure 9.24: Results for the multi-Hop, multiple-Bottleneck, two potential tight links

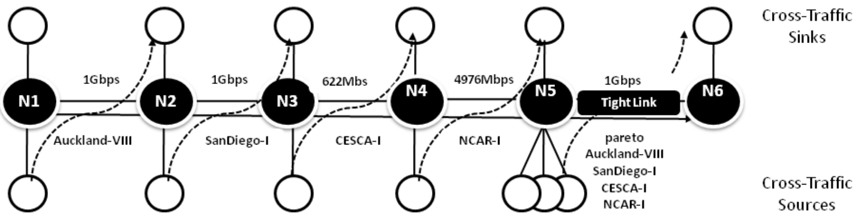


Figure 9.25: NLANR data traces as cross-traffic

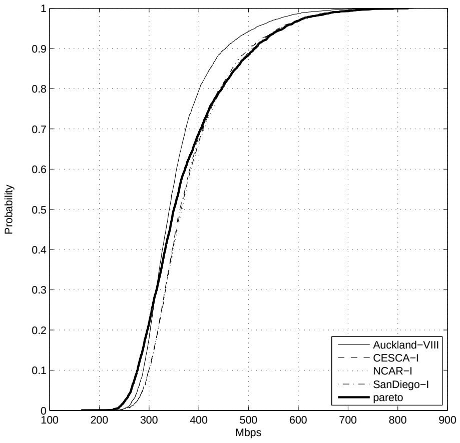


Figure 9.26: NLANR traces as cross-traffic (Results)

link is always around 50%. Note that the NLANR traces have a variable rate, and that the AB varies over the time. We run this experiment for 600 seconds and the AB=400Mbps. We use the CESCA-I data trace.

Table 9.8 shows the results of this set of experiments. As the table shows the mean estimations of the AB match with the real value (400Mbps), in fact the mean relative error is below 0.10 in all the cases. The table also shows that the achieved accuracy is similar when operating with realistic cross-traffic or with Pareto cross-traffic. This is an expected result since it has been shown that N Pareto flows mimics the burstiness behavior of the Internet traffic [199].

The figure 9.26 shows a CDF of the estimations, for all the cases. The figure shows that the behavior of our tool is very similar when operating through the different types of cross-traffic. The variability of the estimations is due to several factors. First the AB changes during the experiment (as noted previously). Second the configuration of the process covariance matrix Q and finally the error when estimating the utilization. As it has been stated before a high Q means that the KF will consider the prediction as less accurate while the measurements will be considered as very accurate. This means that a high Q helps the KFs to better track changes of the AB, but produces less stable estimations. A low Q means the opposite. Throughout the chapter we have used a high value for Q (eq. 9.20).

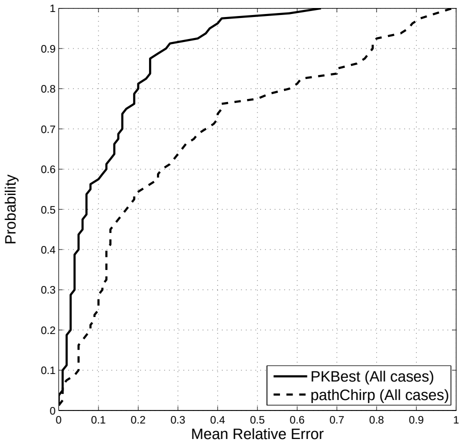


Figure 9.27: Summary of Results

Summary of the Results

This subsection presents a summary of the results. The figure 9.27 shows a CDF of the mean relative error of both tools (PKBest and pathChirp) considering all the cases. As the figure shows the mean relative error for PKBest is 0.12 while for pathChirp is 0.30. For both tools the maximum error (0.66 for PKBest and 0.99 for pathChirp) occurs when the tight link is near congestion.

9.9 Related Work

Most of the tools designed to infer congestion in wired paths are based on the Probe Rate Model [95] or the Probe Gap Model [98]. The research efforts have been focused on active-based tools for wired scenarios. Basically these tools show a trade-off between accuracy and intrusiveness [190]. AKBest is an active tool that exploits poisson probing and Kalman Filtering to achieve reasonable accuracy (comparable to that of pathChirp) but reducing considerably the intrusiveness. Kalman Filters were also used by S.Ekelin et al. in [189] with BART, a PRM-based AB estimation tool. BART uses a linear model that relates the interpacket strain (the time gap between two consecutives packets) with the probe traffic rate. Then the AB is computed as the point where the line intercepts the horizontal axis. In fact BART's model is similar to ours, the main difference is that their linear model is defined above the AB while ours is defined below the AB.

Thats why BART has to congest the tight link in order to produce an estimation. The only AB estimation tool that does not send probe traffic matching the AB is Forecaster [100]. AKBest is based on this mathematical model but produces estimations using KFs instead of projecting a line. In addition AKBest is able to track the dynamic variations in the AB.

Regarding passive tools for wired scenarios there is very little research. *M. Zangrilli* presented in [203] a one-sided, passive, PRM-based tool that uses TCP packets to estimate the AB. This tool uses the timestamps of data and ACKs packets to calculate round-trip times and then applies the PRM model. Second *C. Man* presented in [202] ImTCP, a new version of TCP that uses the arrival intervals of ACK packets as packet pairs to produce estimations using the Probe Gap Model. Both tools have the same issue, TCP cannot guarantee any given rate or pattern. This means that they are only able to produce estimations if the AB is similar to the actual TCP throughput. *S. Katti* presented in [204] MultiQ, a passive *capacity* measurement tool suitable for large-scale studies of Internet path characteristics. MultiQ is the first passive tool able to discover the capacity of multiple congested links along a path from a single flow trace and it is based on a modified version of the Probe Gap Model. Although this tool does not estimate the AB, both tools (MultiQ and PKBest) are truly passive and use NLANR data traces to evaluate their accuracy.

Finally and as it has been stated above ProbeGap [102] is the only tool able to operate in such scenarios. Unfortunately, as we have shown in section 9.7.1, ProbeGap fails when applied to wired-cum-wireless scenarios or with realistic settings (i.e different packet sizes and/or many contending stations).

9.10 Summary and Conclusions

In this chapter we have addressed the lack of available bandwidth estimation tools for wireless networks. This metric can be applied to many scenarios such as P2P networks, overlay routing, path monitoring and, as detailed in chapter 2, to efficient multihomed mobile router architectures and strip mechanisms. As we have seen in chapter 8, periodic probing processes target the achievable throughput rather than the achievable bandwidth. That is why we have explored poisson probing processes to design a methodology able to estimate the available bandwidth. Furthermore, we have applied this methodology to wired scenarios. Specifically this chapter has presented three different tools: W-Path, AKBest and PKBest

- *W-Path*: This is an heuristic-based tool able to estimate the available bandwidth in wired-cum-wireless scenarios (a multi-hop wired path with a single-hop IEEE 802.11 link). The tool is based on a poisson-based methodology that infers congestion by estimating the point at which the path is fully utilized. We have evaluated our tool through simulation and our results show that W-Path is accurate, even if the wireless link is near congestion. Additionally it is able to estimate the available bandwidth if the bottleneck link is located at the wired path.

- *AKBest*: This tool is based on the same methodology than W-Path but uses Kalman Filtering to infer congestion. This mathematical mechanism allows us to design a tool able to produce estimations by sending packet trains at a lower rate than the available bandwidth, hence reducing the impact on the measured path. We have evaluated this tool by simulation and compared its accuracy to that of pathChirp (a state-of-the-art tool). As our results show AKBest has better accuracy than pathChirp, especially where the available bandwidth varies over time.

- *PKBest*: This is a passive tool based on the same methodology than AKBest. The main difference is that while AKBest actively sends packet trains, PKBest is a passive tool intended to be applied between a sender and a receiver node. PKBest produces estimations by inspecting specific properties of the traffic exchanged. Specifically the sender node timestamps sequences of 200 packets with poisson distributed inter-departure time while the receiver one exploits delay information to infer congestion. We have analyzed several public data traces (collected at different access routers) and shown that considering subsets of 200 packets, in average, 20% of the traffic processed fulfils these requirements. Further we have evaluated the accuracy of PKBest in a wide range of scenarios and shown that it has better accuracy than pathChirp.

Table 9.7: Scenarios used to evaluate AKBest. For each scenario we show the capacity of the links and the load of each link. The rightmost column shows the actual

						Available Bandwidth
		100Mbps	622Mbps	10Mbps		
Scenario I	**Static**	0.0	0.0	0.0		10Mbps
		0.3	0.4	0.2		8Mbps
		0.95	0.2	0.01		5Mbps
	Variable	AB varies: 6, 8, 6, 10 Mbps				
		1000Mbps	10000Mbps	100Mbps		
Scenario II	**Static**	0.0	0.0	0.0		100Mbps
		0.2	0.1	0.5		50Mbps
		0.9	0.1	0.7		30Mbps
	Variable	AB varies: 80, 60, 40, 100 Mbps				
		51Mbps	4976Mbps	100Mbps		
Scenario III	**Static**	0.0	0.0	0.0		51Mbps
		0.5	0.1	0.2		25Mbps
		0.1	0.2	0.7		30Mbps
	Variable	AB varies: 21, 51, 41, 31 Mbps				
		1000Mbps	4976Mbps	622Mbps	100Mbps	
Scenario IV	**Static**	0.0	0.0	0.0	0.0	100Mbps
		0.2	0.1	0.1	0.3	70Mbps
		0.95	0.1	0.1	0.1	50Mbps
	Variable	AB varies: 80, 100, 40, 60 Mbps				

Table 9.8: Statistics of the results (Values in Mbps)

Data Trace	**Min**	**Mean**	**Max**	**Std.Dev**
Auckland-VIII	182.38	358.31	798.97	73.79
CESCA-I	184.00	386.15	823.97	86.30
NCAR-I	185.08	387.24	813.03	87.21
SanDiego-I	185.23	384.93	839.91	85.13
Pareto	163.87	374.82	821.72	95.21

Part V

Conclusions

10

Conclusions

This thesis analyzes the transition to the Mobile Internet. This is the process where the Internet becomes mobility-enabled and a node attached to it can change its attachment point seamlessly. However, deploying mobility to the current Internet architecture is a complex task. The research community has proposed many solutions, at different layers, to provide such functionality. This thesis has explored such question and our analysis reveals that solutions at the higher layers of the TCP/IP stack provide seamless transitions but present significant issues in terms of security and overhead. For instance none of the proposed solutions at the transport or the session layer provide an effective and scalable authentication mechanism. On the other side, solutions aimed for the network layer, show a good performance and provide simple but efficient authentication mechanisms. However they fail to provide seamless transitions. Our study suggests that possibly the optimal solution for mobility is a cross-layer protocol that takes advantage of seamless transitions of higher layers and the performance and the security provided by the network layer.

Furthermore the thesis also presents a cost-effective analysis of this issue. By considering the deployment costs of mobility at the different layers, network-mobility is the most cost-effective solution. A reason for that is because the Internet currently uses IPv4 as the default network protocol and it has other urgent issues. For instance it has a depleted address space and lacks of extensibility. IPv6, a replacement for IPv4, solves these issues and incorporates built-in mobility. Therefore, since IPv4 must be updated, the most cost-effective solution to deploy mobility is through IPv6.

Taking this into consideration this thesis analyzes the transition to the Mobile Internet considering the deployment of network-layer mobility. The Mobile IP family of protocols, designed by the IETF, extends IP to provide mobility. Hence this thesis has analyzed the transition to a Mobile IP Internet at three different stages. First at present, by analyzing the Mobile IP technology. This a family of protocols that provides mobility to IPv4 (Mobile IPv4), IPv6 (Mobile IPv6) and that has many extensions that enhance the performance of Mobile IPv6 or improve its functionalities. Additionally NEMO provides mobility to networks. Our study shows that one of the key issues of Mobile IP is the performance of the handover. The thesis present an analytical model and experimentation to study several metrics related with the handover of the main protocols of Mobile IP. In detail, the main conclusions of this thesis are:

- In all the cases the handover depends linearly on the distance between the MN and the HA
- Mobile IPv4 has a low latency handover, in the order of tens of ms, that enables it to support real-time services.
- Mobile IPv6 has a slow handover, in the order of seconds. This disrupts significantly the communications and prevents this protocol to support delay-sensitive applications. The main reason for that is the large amount of time that takes to reconfigure IPv6.
- Fast Handovers for Mobile IPv6 is an extension of Mobile IPv6 aimed to reduce its handover latency. This thesis presents the first public implementation of this protocol (see [222]for further details) and a performance evaluation. Our results reveal that this protocol effectively reduces the latency of the handover to 0 (without considering the link-layer handover).

The second stage of the analysis is during the deployment phase of Mobile IP. This thesis has focused on identifying potential issues and providing solutions. The main conclusions of this stage of the analysis are:

- Home Agents represent single-point-of-failure in Mobile IP-based networks. The thesis proposes the flexible Home Agent architecture. This architecture distributes the operations of a Home Agent throughout the network increasing reliability and reducing the load at this particular entity.

- There is a lack of route optimization in Mobile IPv4 and NEMO clients. This increases the delay of the paths of the mobile clients and impacts the performance of these protocols. This thesis proposes the fP2P-HN architecture, a P2P-based network of distributed Home Agents that reduces significantly the paths of the mobile clients. Our evaluation shows that it is scalable ($O(1)$) and that can reduce the delays of the paths 50% (on average), compared to Mobile IPv4 and NEMO.

- Additionally Mobile IPv6 clients can also lack of this functionality during certain stages of the deployment. This can occur if Mobile IPv6 is not deployed along with IPv6. Furthermore we have spotted and important incompatibility between Mobile IPv6's route optimization and some load balancing techniques. In order to solve both issues this thesis proposes the Mobility Agents, a transparent proxy located at the server's networks that manages mobility related signalling on behalf the correspondent nodes.

Finally the third stage of our analysis is in the future, were the Internet is Mobile IPv6-enabled and new architectures can improve its functionalities. Specifically this thesis has analyzed the advantages and the complexity of terminals equipped with multiple interfaces. This can provide more aggregate bandwidth, increase the reliability and the area of coverage. However supporting multiple interfaces can be a difficult task. Our analysis reveals that a generic architecture able to deal with these issues can be greatly enhanced if the available bandwidth of the different paths provided by the multiple interfaces can be estimated. Research on this topic has focused mainly on periodic probing processes and wired networks, however very little research has been conducted considering wireless links (a typical scenario of mobility). Therefore this thesis has focused on analyzing the existing methodologies and tools in the presence of wireless links, taking the IEEE 802.11 standard as a reference. Our study reveals that:

- Periodic probing processes target the achievable bandwidth instead of the available bandwidth. This metric is related with the fair share of the network.

- Measurements using periodic probing processes present a bias. This affects the first packets of the process that are served faster than the remaining ones. This bias appears due to the random nature of wireless networks and impacts the measurements. This means that such probing processes can to lead to erroneous estimations of bandwidth-related metrics.

Taking this into consideration this thesis has explored poisson-probing process to estimate the available bandwidth in wireless and wired scenarios. In particular our research has lead to design the following tools and methodologies:

- W-Path: This tool aims to estimate the available bandwidth in wired/wireless scenarios. W-Path uses poisson probing to estimate the utilization of the path and a binary search algorithm to estimate the available bandwidth.
- AKBest and PKBest: Both tools are intended for wired scenarios and are able to produce estimations without impacting the path under measurement. Specifically AKBest is an active tool that sends probe packets at a rate lower than the available bandwidth while PKBest is a passive methodology able to infer congestion by inspecting existing traffic.

Furthermore, our study in poisson probing processes has revealed that:

- They are useful to estimate the available bandwidth in wireless links. Periodic processes target a different metric.
- These processes can be used to design less intrusive tools. This is mainly because such processes present a linear relation between its intensity (rate) and the load of the network (utilization). On the contrary, periodic probing processes present a binary relation.
- Periodic probing processes can be enhanced in order to produce faster tools. Removing the first probing packets removes also the bias. Therefore, shorter probing processes (that do not take into account the first probe packets) can achieve the same accuracy.

11

Future Work

The analysis and studies carried out within this thesis have opened up new research directions that can be followed to enhance both mobility and bandwidth estimation. The proposed mobility architectures, such as the fP2P-HN or the fHA, have been developed taking into consideration the current Internet architecture. However, new architectures are under active research, and the Internet may adapt one of them in a short period of time. One of the most promising new architectures is LISP[1] (Location Identifier Separation Protocol) [219]. LISP is a CISCO proposal that is being standardized at the IETF [220]. LISP's main goals are to decouple the WHO from the WHERE identifier from IP addresses in order to provide multihoming. A new overlay BGP network (the LISP-ALT network [221]) provides bindings between the WHO and the WHERE identifiers. As stated in [219], LISP cannot include mobility, basically because for each new location of a mobile node the LISP-ALT distributed database must be informed, and this is clearly not scalable. Therefore mobility should be accommodated into LISP using a different approach. If we analyze LISP and Mobile IP we realize that it has mainly the same issues than Mobile IP and the current Internet's architecture. Therefore we believe that the mobility architectures proposed in this thesis (Mobility Agents, fHA and fP2P-HN) can be ported to LISP. Even more, these architectures can take advantage of the LISP-ALT network to enhance its functionalities and performance.

Regarding bandwidth estimation, the study and experimentation carried out within

[1] The reader may wonder which differences have LISP and NewArchs described in chapter 1. Both approaches are fundamentally different, LISP is a new architecture that runs on top of Internet while NewArchs aim to replace entirely the current Internet architecture

this thesis have shown several complex issues that remain unsolved. Firstly the transitory detected in chapter 8 biases measurements, not only in wireless networks, but also in any network. A reason for that is because the cross-traffic can be modelled also as a random process, just as the service delay. Hence, periodic probing processes are always affected by such transitory. This transitory has been detected in our analysis, but future research on this can provide an analytical model. This model could help analyzing its duration or could show important properties of this transitory stage.

Secondly, as we have shown in chapter 8, any periodic probing processes can be enhanced if the first probing packets are removed from the measurements. This can enhance the accuracy and reduce the intrusiveness. Therefore we believe that this can be further researched and design novel bandwidth estimation tools.

And third, in chapter 9 we have seen that poisson probing process can be very useful for bandwidth estimation. We believe that this thesis has just provided the first steps of the analysis of such processes and that further research can provide even faster and lighter tools.

Finally, an issue that this thesis leaves open, is related to multihoming mobile architectures. As we have shown in chapter 2 managing efficiently mobility in such scenarios is a complex task. For instance further research can be carried out in interface selection, strip modules or dynamic flow assignment taking into account bandwidth estimation.

Appendix A

Appendix - Complete list of publications

- Rubén Cuevas, Albert Cabellos-Aparicio, Ángel Cuevas, Jordi Domingo-Pascual, Arturo Azcorra "fP2P-HN: A P2P-based Route Optimization Architecture for Mobile IP-based Community Networks" to appear in Computer Networks, Elsevier, special issue in Content Distribution Infrastructures for Community Networks (January 2009)

- René Serral-Gracià, Albert Cabellos-Aparicio, Jordi Domingo-Pascual, "Packet loss estimation using distributed adaptive sampling", in Proceedings of 6th IEEE Workshop on End-to-End Monitoring Techniques and Services (E2EMon), Salvador de Bahia, April 2008

- Albert Cabellos-Aparicio, Francisco J. Garcia, Jordi Domingo-Pascual "A non-congesting available bandwidth estimation and tracking algorithm", PDNo:20080257, 25-Jan-2008.

- René Serral-Gracià, Albert Cabellos-Aparicio, Jordi Domingo-Pascual, "Network performance assessment using adaptive traffic sampling", IFIP Networking 2008

- Albert Cabellos-Aparicio, Francisco J. Garcia, Jordi Domingo-Pascual, "A Novel Available Bandwidth Estimation and Tracking Algorithm" in Proceedings of 6th IEEE Workshop on End-to-End Monitoring Techniques and Services (E2EMon), Salvador de Bahia, April 2008

A. APPENDIX - COMPLETE LIST OF PUBLICATIONS

- Albert Cabellos-Aparicio, Jordi Domingo-Pascual, ”Mobility Agents: Avoiding the Signaling of Route Optimization on Large Servers” in Proceedings of IEEE PIM-RC 2007, Athens, Greece

- Albert Cabellos-Aparicio, Jordi Domingo-Pascual, ”A Flexible and Distributed Home Agent Architecture for Mobile IPv6-based Networks” in Proceedings of IFIP Networking 2007, Atlanta, USA.

- Albert Cabellos-Aparicio, Jordi Domingo-Pascual, ”Load Balancing in Mobile IPv6's Correspondent Networks with Mobility Agents” in Proceedings of IEEE ICC 2007, Glasgow, UK

- Albert Cabellos-Aparicio, Jordi Domingo-Pascual, ”Enhanced Fast Handovers Using a Multihomed Mobile IPv6 Node” in Proceedings of IEEE New2AN 2005, St. Petersburgh, Russia

- Albert Cabellos-Aparicio, Jose Núñez-Martínez, Hector Julian-Bertomeu, Loránd Jakab, René Serral-Graciá, Jordi Domingo-Pascual ”Evaluation of the Fast Handover Implementation for Mobile IPv6 in a Real Testbed” in Proceedings of IEEE IPOM 2005, Barcelona, Spain

- Albert Cabellos-Aparicio, Hector Julian-Bertomeu, Jose Núñez-Martínez, Loránd Jakab, René Serral-Graciá, Jordi Domingo-Pascual ”Measurement-Based Comparison of IPv4/IPv6 Mobility Protocols on a WLAN Scenario” in Proceedings of Networks UK HET-NET Ilkley, UK 2005

- René Serral-Gracià, Albert Cabellos-Aparicio, H. Julian-Bertomeu, J. Domingo-Pascual ”Active measurement tool for the EuQoS project” in Proceedings of IPS-MOME 2005, Warsaw, Poland.

- Albert Cabellos-Aparicio, René Serral-Gracià, Loránd Jakab, and Jordi Domingo-Pascual ”Measurement Based Analysis of the Handover in a WLAN MIPv6 Scenario” in Proceedings of Passive and Active Measurements 2005, Boston, USA.

- Carlos Veciana-Nogues, Albert Cabellos-Aparicio, Jordi Domingo-Pascual, Josep Sole-Pareta, Verifying IP Meters from Sampled Measurements IFIP TestComm 2002 Munich, Germany.

In Spanish

- Albert Cabellos-Aparicio, Jordi Domingo-Pascual Visión General del Protocolo IPv6. Novatica 2005
- Josep Mangues Bafalluy, Albert Cabellos-Aparicio, René Serral-Gracià, Jordi Domingo Pascual, Antonio Gómez Skarmenta, Tomás P. de Miguel, Marcelo Bagnulo, Alberto García Martinez, Movilidad IP: macromovilidad, micromovilidad, calidad de servicio y seguridad. Novatica 2004
- Albert Cabellos-Aparicio, Lluís Calafell Liebanas, Jose Núñez-Martínez, Jordi Domingo-Pascual "Desarrollo y Evaluacin del Protocolo Fast Handovers for Mobile IPv6 en un Entorno Virtual". Jornadas Tcnicas Rediris, Logroo Octubre 2005

Appendix B

Appendix - Research Projects

Part of the material in this thesis has been used in the following Research Projects:

- CONTENT, IST-2006-NoE-0384239, FP6 Network of Excelence `http://ist-content.org`
- Provisión de Servicios de Red Interdomino (CEPS), MEC, TSI 2005-07520-C03-02, MCYT
- E-NEXT, UE IST, Thematic Network FP6-2002-IST/1 506869, FP6 Network of Excelence `http://ist-e-next.net`
- Servicios avanzados con movilidad (SAM), McyT, TIC2002-04531-C04-02
- Laboratories over Next Generation Networks (LONG), UE IST-1999-20393 `http://long.ccaba.upc.edu`

References

[1] Saltzer, J.H., Reed, D.P., and Clark, D.D, ”End-to-end Arguments in System Design” ACM Trans. Comput. Syst., Vol 2, Number 4, 1984[1] 3

[2] Clark, D, ”The Design Philosophy of the DARPA Internet Protocols” ACM SIGCOMM Comput. Commun. Rev, Vol 18, Number 4, 1988 3

[3] Carpenter, B, ”Architectural Principles of the Internet” RFC 1958, 1996 3

[4] A. Tanenbaum, ”Computer Networks” Prentice Hall, 1989 3, 5

[5] Eddy, W, ”At what layer does mobility belong?” IEEE Commun. Mag, Number 42, Vol 10, 2004 4, 5, 20

[6] Mika Ratola, ”Which Layer for Mobility? - Comparing Mobile IPv6, HIP and SCTP” Helsinki University of Technology (Technical Report), (online) http://www.tml.tkk.fi/Studies/T-110.551/2004/papers/Ratola.pdf 4, 20

[7] Henderson Thomas R, ”Host mobility for IP networks: A comparison” IEEE Network, Vol. 17, Num 6, 2003 4

[8] Alex C. Snoeren, Hari Balakrishnan, and M. Frans Kaashoek, ”Reconsidering Host Mobility” In Proc. 8th Workshop on Hot Topics in Operating Systems, 2001 4

[9] L. Ong, ”An Introduction to the Stream Control Transmission Protocol (SCTP)” RFC 3286, 2002 5, 22, 25

[10] Maltz, D.A. Bhagwat, P, ”MSOCKS: an architecture for transport layer mobility” In Proc. IEEE INFOCOM, 1998 5, 24

[11] Daichi Funato, Kinuko Yasuda, and Hideyuki Tokuda, ”TCP-R: TCP mobility support for continuous operation” In Proc. IEEE International Conference on Network Protocols, 1997 5, 24

[12] J.Jung, E. Sit, H. Balakrishnan and R. Morris, ”DNS performance and effectiveness of caching” IEEE/ACM Trans. Netw., 2001 5, 91

[13] C. Perkins, ”IP Mobility Support” RFC 3344, 2002 5, 19, 28, 128, 137

[14] D. Johnson, C. Perkins and J. Arkko, ”Mobility Support in IPv6” RFC 3775, 2002 xv, 5, 7, 28, 66, 88, 89, 102, 103

[15] P. Nikander, J. Arkko, T. Aura, G. Montenegro, ”Mobile IP Version 6 Route Optimization Security Design Background” RFC 4225, 2005 6, 88

[16] Blumenthal M Clark, ”Rethinking the design of the Internet: The end-to-end arguments vs. the brave new world” ACM Transactions on Internet Technology, Vol 1, No. 1, 2001 6

[17] Reed, D, ”The End of the End-to-end Argument?” ACM Trans. Comput. Syst. Vol. 2, Num. 4, 2000 6

[18] Moors, T, ”A Critical Review of End-to-end Arguments in System Design” In Proc. IEEE ICC, 2002 6

[19] Robert Braden, David Clark, Scott Shenker, and John Wroclawski, ”Developing a Next-Generation Internet Architecture” New Arch: Future Generation Internet Architecture, 2000 6

[20] Clark, D., Wroclawski, J., Sollins, K., Braden, R, ”Tussle in Cyberspace: Defining Tomorrow’s Internet” IEEE/ACM Trans. Netw., Vol. 13, Num. 3, 2002 6

[21] USC Information Sciences Institute Computer Networks Division, MIT Laboratory for Computer Science, International Computer Science Institute (ICSI), ”NewArch Project: Future-Generation Internet Architecture” (online) http://www.isi.edu/newarch 6, 20

[22] Braden, R., Faber, T., Handley, M, ”From Protocol Stack to Protocol Heap - Role-Based Architecture” ACM SIGCOMM Comput. Commun. Rev., Vol 33, Num. 1, 2003 6, 20

[23] Clark, D., Braden, R., Falk, A., and Pingali, V, ”FARA: Reorganizing the Addressing Architecture” ACM SIGCOMM Comput. Commun. Rev., Vol 33, Num. 4, 2003 6, 20

[24] IPv4 Address Report (online) http://www.potaroo.net/tools/ipv4/index.html 6

[25] K. Egevang, ”The IP Network Address Translator (NAT)” RFC 1631, 1994 7, 92

[26] S. Deering, ”Internet Protocol Version 6 (IPv6) Specificaton” RFC 2460, 1998 7, 95, 98, 99, 113

[27] The Kame Project (online) www.kame.net 7, 11, 36

[28] CISCO IPv6 Solutions (online) http://www.cisco.com/en/US/technologies/tk648/tk872/tk373/technologies_white_paper09186a00802219bc.html 7, 11, 36

[29] Dynamics HUT Implementation (online) http://dynamics.sourceforge.net/ 7, 8, 70, 72

[30] Boeing Connexion Service (online) http://www.boeingconnexion.com 7, 8

[31] R. Koodli, ”Fast Handovers for Mobile IPv6 (FMIPv6)” RFC 4068, 2005 9, 37, 75

[32] Kaushik Das, ”IPv6 Deployment Status” (online) http://www.ipv6.com/articles/deployment/IPv6-Deployment-Status.htm 7

[1] The numbers at the end of each reference point at the page number where the document is cited.

[33] B. Larsson, T. Ismailov, Y. Seneviratne A, "SLM, a framework for session layer mobility management" In Proc. International Conference on Computer Communications and Networks (ICCCN), 1999 22

[34] Alex C. Snoeren, Hari Balakrishnan, and M. Frans Kaashoek, "Reconsidering IP Mobility" In Proc. of Workshop on Hot Topics in Operating Systems, 2001 5, 22

[35] Alex C. Snoeren and Hari Balakrishnan, "An End-to-end Approach to Host Mobility" In Proceedings of the conference on Mobile computing and networking (ACM MOBICOM), 2000 5, 22

[36] Alex C. Snoeren, David G. Andersen, and Hari Balakrishnan, "Fine-Grained Failover Using Connection Migration" In Proc. 3rd USENIX Symp. on Internet Technologies and Systems (USITS), 2001 5, 22

[37] The Migrate Internet Mobility Project, Massachusetts Institute of Technology (online) http://nms.csail.mit.edu/projects/migrate/ 5, 22

[38] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, "SIP: Session Initiation Protocol" RFC 3261, 2002 5, 22

[39] Elin Wedlund, Henning Schulzrinne, "Mobility Support using SIP" In Proc. of the 2nd ACM international workshop on Wireless mobile multimedia (WOWCOM), 1999 5, 22

[40] R. Stewart, "Stream Control Transport Protocol" RFC 2960, 2000 22, 25

[41] Yongguang Zhang and Son Dao, "A persistent connection model for mobile and distributed systems" In Proc. IEEE International Conference on Computer Communications and Networks, 1995 24

[42] Tadashi Okoshi, Masahiro Mochizuki, Yoshito Tobe, and Hideyuki Tokuda, "MobileSocket: Toward continuous operation for Java applications" In Proc. IEEE International Conference on Computer Communications and Networks (ICCCN), 1999 24

[43] Xun Qu, Jeffrey Xu Yu, and Richard P. Brent, "A mobile TCP socket" In Proc. of International Conference on Software Engineering (SE), 1997 24

[44] Victor C. Zandy and Barton P. Miller, "Reliable network connections" In Proc. of the international conference on Mobile computing and networking (MOBICOM), 2002 24

[45] Marcus Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, and L. Jones, "SOCKS protocol version 5" RFC 1928, 1996 24

[46] Christian Huitema, "Multi-homed TCP" IETF Working Draft-work in progress, 1995 24

[47] E. Kohler, "Datagram Congestion Control Protocol (DCCP)" RFC 4340, 2006 25

[48] Seok Joo Koh, Moon Jeong Chang, Meejeong Lee, "mSCTP for soft handover in transport layer" IEEE Communications Letter, Vol. 8, Num. 3, 2004 25

[49] Xylomenos, G. Polyzos, G.C. Mahonen, P. Saaranen, M, "TCP performance issues over wireless links" IEEE Communications Magazine, Vol. 39, Num. 4, 2001 26

[50] R. Moskowitz, "Host Identity Protocol (HIP)" RFC 5201, 2008 26

[51] R. Moskowitz, "Host Identity Protocol (HIP) Architecture" RFC 4423, 2006 26

[52] Fumio Teraoka, Yasuhiko Yokore, and Mario Tokoro, "A network architecture providing host migration transparency" ACM SIGCOMM Comput. Commun. Rev., Vol. 21, Num. 4, 1991 28

[53] Sumit Gupta and A. L. Narasimha Reddy, "A client oriented, IP level redirection mechanism" In Proc. IEEE INFOCOM, 1999 28

[54] A. T. Campbell, Gomez, J., Kim, S., Turanyi, Z., Wan, C-Y. and A, Valko, "Internet Micromobility" Journal of High Speed Networks, Special Issue on Multimedia in Wired and Wireless Environment, Vol. 11, Num. 3-4, 2002 28

[55] Isidro Castineyra, Noel Chiappa, and Martha Steenstrup, "The Nimrod routing architecture" RFC 1992, 1996 28

[56] Ram Ramanathan, "Nimrod mobility support" RFC 2103, 1997 28

[57] Mobility for IPv4 (mip4) IETF WG (online) http://www.ietf.org/html.charters/mip4-charter.html 33

[58] Mobility EXTensions for IPv6 (mext) IETF WG (online) http://www.ietf.org/html.charters/mext-charter.html 33, 128

[59] Mobility for IP: Performance, Signaling and Handoff Optimization (mipshop) IETF WG (online) http://www.ietf.org/html.charters/mipshop-charter.html 33

[60] Keiichi Shima, Koshiro Mitsuya, Ryuji Wakikawa, Tsuyoshi Momose, Keisuke Uehara, "SHISA: The Mobile IPv6/NEMO BS Stack Implementation Current Status" (online) http://2007.asiabsdcon.org/papers/P10-paper.pdf 36

[61] Mobile IPv6 for Linux (online) http://tldp.org/HOWTO/Mobile-IPv6-HOWTO/mipv6.html 36

[62] H. Soliman, "Hierarchical Mobile IPv6 Mobility Management (HMIPv6)" RFC 4140, 2005 37

[63] V. Devarapalli, "Network Mobility (NEMO) Basic Support Protocol" RFC 3963, 2005 12, 39

[64] C. Ng, P. Thubert, M. Watari, F. Zhao, "Network Mobility Route Optimization Problem Statement" Internet Draft, Work in Progress 2006 94, 97, 123, 124

[65] Jongkeun Na, Seongho Cho, Chongkwon Kim, Sungjin Lee, Hyunjeong Kang, Changhoi Koo, "Route Optimization Scheme based on Path Control Header" Internet Draft Work in Progress, 2004 39

[66] Ryuji Wakikawa, Susumu Koshiba, Keisuke Uehara, Jun Murai, "ORC: Optimized Route Cache Management Protocol for Network Mobility" In Proc. of the IEEE 10th International Conference on Telecommunication (ICT), 2003 39

[67] Bernardos, C., Bagnulo, M., and M. Calderon, "MIRON: MIPv6 Route Optimization for NEMO" In. Proc. Workshop on Applications andServices in Wireless Network, 2004 39

[68] Ryuji Wakikawa, Masafumi Watari, "Optimized Route Cache Protocol (ORC)" Internet Draft Work in Progress, 2004 39

[69] C. Ng, "Network Mobility Route Optimization Problem Statement" RFC 4888, 2007 43

[70] Jahanzeb Faizan et al, "Problem Statement : Home Agent reliability" IETF Draft Work in Progress, 2004 45

[71] G. Huston, "Architectural Approaches to Multi-homing for IPv6" RFC 4177, 2005 14, 48

[72] R. Ogier, V. Ruenburg, N. Shacham, "Distributed algorithms for computing shortest pairs of disjoint paths" IEEE Transactions on Information Theory, 1993 14, 48

[73] I. Cidon, "Analysis of multi-path routing" IEEE/ACM Transactions on Networking, 1999 14, 48

[74] J. Raju, "A new approach to on-demand multipath routing" In Proc. Eight International Conference on Computer Communications and Networks (ICCCN), 1999 14, 48

[75] Mobile Nodes and Multiple Interfaces in IPv6 IETF Monami6 Working Group (online) https://www.labo4g.rennes.enst-bretagne.fr/twiki/bin/view/Monami6/ 49

[76] Jun Yao, "Implementation of a Multihoming Agent for Mobile On-Board Communication" University of New South Wales, School of Computer Science and Engineering, Technical Report, 2005 50

[77] Pablo Rodriguez, Rajiv Chakravorty, Julian Chesterfield,Ian Pratt andSuman Banerjee, "MAR: A Commuter Router Infrastructure for the Mobile Internet" In Proc. of International Conference on Mobile systems (MobySYS) 2004 50

[78] N. Thompson, "Flow Scheduling for End host Multihoming" In Proc. IEEE INFOCOM, 2006 50

[79] Luiz Magalhanaes, "Transport Level Mechanisms for Bandwidth Aggregation on Mobile Hosts" In Proc. IEEE International Conference on Network Protocols, 2001 50

[80] R. Wakikawa, "Multiple Care-of Addresses Registration" IETF Work in Progress, 2008 50

[81] H. Soliman, "Flow Bindings in Mobile IPv6 and Nemo Basic Support" IETF Work in Progress, 2008 50

[82] Cisco Documentation (online) `http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/diren.htm` 53

[83] M. Mirhakkak, "Dynamic Bandwidth Management and Adaptive Applications for a Variable Bandwidth Wireless Environment" IEEE Journal on Selected areas in Comm. Vol. 19, Num. 10, 2001 53

[84] B. Riddle, "A Quality of Service Proposal" (online) `http://qos.internet2.edu/may98Workshop/html/apiprop.html` 53

[85] Andreas Kassler, Holger Christein, Peter Schulthess, "A Generic API for Quality of Service Networking based on Java" In Proc. IEEE International Conference on Communications (ICC), 1999 14, 53

[86] D. Phatak, "A Novel Mechanism for Data Streaming Across Multiple IP Links for Improving Throughput and Reliability in Mobile Enviroments" In Proc. IEEE INFOCOM, 2005 54

[87] Kyu-Han Kim, "Improving TCP Performance over Wireless Networks with Collaborative Multi-homed Mobile Hosts" In Proc. of International Conference on Mobile systems (MobySYS), 2005 54

[88] Rajiv Chakravorty, "Flow Aggregation for Enhanced TCP Over Wide-Area Wireless" In Proc. IEEE INFOCOM, 2003 54

[89] Hung-Yun Hsieh, "A Transport Layer Approach for Achieving Aggregate Bandwidths on Multi-homed Mobile Hosts" In Proc. of ACM International Conference on Mobile Computing and Networking (MOBICOM), 2002 54

[90] J.R. Iyengar, "Concurrent Multipath Transfer Using SCTP Multihoming Over Independent End-to-end Path" IEEE Transactions on Networking, Vol. 11, Num. 1, 2006 54

[91] Luiz Magalhaes, "Transport Level Mechanisms for Bandwidth Aggregation on Mobile Hosts" IEEE International Conference on Network Protocols (ICNP), 2001 54

[92] D.P. Pezaros, F.J. Garcia, R.D. Gardner, D. Hutchison, J.S Sventek, "In-line Service Measurements: An IPv6-based framework for traffic evaluation and network operations" In Proc. of IEEE/IFIP Network Operations and Management Symposium (NOMS), 2004 54

[93] V. J. Ribeiro, M. Coates, R. H. Riedi, S. Sarvotham, and R. G. Baraniuk, "Multifractal cross traffic estimation" In Proc. of ITC Specialist Seminar on IP Traffic Measurement, 2000 14, 54, 147, 148, 154, 168, 170, 173, 174, 196

[94] B. Melander, M. Bjorkman, and P. Gunningberg, "A New End-to-end Probing and Analysis Method for Estimating Bandwidth Bottlenecks" In Proc. of Global Telecommunications Conference (GLOBECOM), 2000 14, 54, 147, 148, 154, 168, 170, 173, 174, 196

[95] M. Jain and C. Dovrolis, "Pathload: A Measurement Tool for End-to-end available bandwidth" In Proceedings of Passive and Active Measurements (PAM) Workshop, 2002 14, 54, 147, 148, 154, 168, 170, 173, 174, 192, 196, 214

[96] N. Hu and P. Steenkiste, "Evaluation and Characterization of available bandwidth Techniques" IEEE Journal on Selected Areas in Communications, Vol. 21, Num. 6, 2003 14, 54, 147, 148, 154, 168, 170, 173, 174, 196

[97] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell, "pathChirp: Efficient available bandwidth Estimation for Network Paths" In Proceedings of Passive and Active Measurements (PAM), 2003 14, 54, 147, 148, 154, 168, 170, 173, 174, 196, 202

[98] J. strauss, "A Measurement Study of available bandwidth Estimation tools" In Proc. of the ACM SIGCOMM conference on Internet measurement (IMC), 2003 14, 54, 147, 148, 154, 168, 173, 174, 196, 211, 214

[99] Svante Ekelin, "Real-Time Measurement of End-to-end available bandwidth using Kalman Filtering" In Proc. of the IEEE/IFIP Network Operations and Management Symposium (NOMS) 2006 14, 54, 147, 148, 154, 168, 170, 173, 174, 196

[100] Mradula Neiginhal, "Measuring Bandwidth Signatures of Network Paths" In Proc. of International IFIP-TC6 Networking Conference Networking (Networking), 2007 14, 54, 147, 148, 154, 168, 170, 173, 174, 175, 177, 196, 215

[101] Yu Cheng, "New Exploration of Packet-Pair Probing of available bandwidth Estimation and Traffic Characterization" In Proc. of IEEE International Conference on Communications (ICC), 2007 14, 147, 148, 151, 152, 154, 168, 170, 196

[102] Karthik Lakshminarayanan, "Bandwidth Estimation in Broadband Access Networks" In Proc. of the ACM SIGCOMM conference on Internet measurement (IMC), 2004 14, 55, 147, 148, 151, 152, 154, 171, 173, 183, 196, 199, 215

[103] M. Li, M. Claypool, R. Kinicki, "Packet Dispersion in IEEE 802.11 Wireless Networks" In Proc. IEEE Conference on Local Computer Networks (LNC), 2006 14, 55, 147, 196

[104] IEEE 802.11 Standard (online) http://grouper.ieee.org/groups/802/11/ 14, 37, 61, 62, 196

[105] Andreas Johnsson,Bob Melander,and Mats Bjorkman, "Bandwidth measurement in wireless networks" In Proc. of Mediterranean AdHoc Networking Workshop, 2005 14, 55, 185

[106] Shin, S., Forte, A. G., Rawat, A. S., and Schulzrinne, H, "Reducing MAC layer handoff latency in IEEE 802.11 wireless LANs" In Proc. of the International workshop on Mobility management & wireless access protocols (MobiWac), 2004 78

[107] T. Narten, "Neighbor Discovery for IP Version 6 (IPv6)" RFC 2461, 1998 63, 73

[108] Thomson, S. and T. Narten, "IPv6 Address Autoconfiguration" RFC 2462, 1998 63, 73

[109] Internet2 Consortium: OWAMP NTP Configuration (online) http://e2epi.internet2.edu/owamp/details.html#NTP (2004) 71

[110] Mogul, J., D. Mills, J. Brittenson, J. Stone and U. Windl, "Pulse-per-second API for Unix-like operating systems" RFC 2783, 2000 71

[111] Helsinki University of Technology: MIPL Mobile IPv6 for Linux (online) http://www.mobile-ipv6.org/ (2004) 70, 72

[112] Wireless Toolkit for Linux (online) http://www.hpl.hp.com/Jean_Tourrilhes/Linux/Tools.html 73, 76

[113] René Serral, Roberto Borgione, "NetMeter a NETwork performance METER" (online) http://www.ccaba.upc.es/netmeter (2002) 72

[114] Gerald Combs, "Ethereal: The worlds most popular network protocol analyzer" (online) http://www.ethereal.com 72

[115] Loránd Jakab, Albert Cabellos-Aparicio, René -Graciá, Jordi Domingo-Pascual, "Software Tool for Time Duration Measurements of Handovers in IPv6 Wireless Networks" Technical Report, UPC, UPC-DAC-2004-25, 2004 72

[116] ITU-T Recommendation Y.1541, "Network Performance Objectives for IP-based Services" 2005 84

[117] IP Performance Metrics WG IETF (online) http://www.ietf.org/html.charters/ippm-charter.html 69

[118] G. Almes, "A One-Way-Delay Metric for IPPM" RFC 2679, 1999 69

[119] C. Demichelis, "IP Packet Delay Variation Metric IP Performance Metrics (IPPM)" RFC 3393, 2002 69

[120] G. Almes, "A One-way Packet Loss Metric for IPPM", RFC 2680, 1999 69

[121] Jeff Dike, UML – User-Mode-Linux (online) http://user-mode-linux.sourceforge.net 82

[122] Marco Liebesch, Xavier Perez Costa and Ralph Schmitz, "A MIPv6, FMIPv6 and HMIPv6 Handover latency study: Analytical Approach" In Proc. of IST Mobile & Wireless Telecommunications Summit, 2002 83

[123] Xavier Perez Costa and Hannes Hartenstein, "A simulation study on the performance of mobile IPv6 in a WLAN-based cellular network" IEEE Computer Networks Vol. 20, Num. 3, 2002 83

[124] A. Mishra, M. Shin and W. Arbaugh, "An Empirical Analysis of the IEEE 802.11 MAC Layer Handover Process" ACM SIGCOMM Computer Communications Review, Vol. 33, Num. 2, 2003 83

[125] N. Montavont and T. Noel, "Handover Management for Mobile Nodes in IPv6 Networks" IEEE Communications Magazine Vol. 40, Num. 8, 2002 83

[126] N. Moore, "Optimistic Duplicate Address Detection" RFC 4429, 2006 83

[127] T. Brisco, "DNS Support for Load Balancing" RFC 1794, 1995 91

[128] Tony Bourke, "Server Load Balancing" O'Reilly, ISBN 0-596-00050-2 2001 91, 107

[129] T. Berners-Lee, R. Fielding and H. Frystyk, "Hypertext Transfer Protocol HTTP/1.0" RFC 1945, 1996 91, 93

[130] Q. Li and B. Moon, "Distributed Cooperative Apache Web server" In. Proceedings of the ACM International conference on World Wide Web, 2001 91

[131] P.Srisuresh, D.Gan, "Load Sharing using IP Network Address Translation (LSNAT)" RFC 2391, 1998 92

[132] C. Perkins, "IP encapsulation within IP" RFC 2003, 1996 93

[133] V. Cardellini, E. Casalicchio, M. Colajanni, P.S. Yu, "The state of the art in locally distributed Web-server systems" ACM Computing Surveys, Vol. 34, No. 2, 2002 93, 107

[134] A. Cohen, "On the performance of TCP splicing for URL-aware redirection" In Proc. of the USENIX Symposium on Internet Technologies and Systems, 1999 93

[135] V.S. Pai, "Locality aware request distribution in cluster-based network servers" In Proc. of the International conference on Architectural support for programming languages and operating systems, 1998 93

[136] Resonate Inc. (online) http://www.resonate.com 93, 109

[137] W. Haddad, L. Madour, J. Arkko, F. Dupong, "Applying Cryptographically Generated Addresses to Optimize MIPv6 (CGA-OMIPv6)" Internet Draft Work in Progress, 2006 94

[138] S. Kent, K. Seo, "Security Architecture for the Internet Protocol" RFC 4301, 2005 22, 97, 113

[139] S. Kent, "IP Authentication Header" RFC 4302, 2005 96, 97

[140] D. Farinacci, T.Li, S. Hanks, D. Meyer, P. Traina, "Generic Routing Encapsulation (GRE)" RFC 2784, 2000 97

[141] C. Perkins, "Minimal Encapsulation within IP" RFC 2004, October 1996 97

[142] C. Perkins, "Securing Mobile IPv6 Route Optimization Using a Static Shared Key" RFC 4449, 2006 107

[143] R. Vadali, J. Li, Y. Wu and G. Cao, "Agent-Based Route Optimization for Mobile IP" In Proceedings of IEEE Vehicular Technology Conference (VTC), 2001 107, 141

[144] Chun-Hsin Wu , Ann-Tzung Cheng , Shao-Ting Lee , Jan-Ming Ho and D. T. Lee, "Bi-directional Route Optimization in Mobile IP over Wireless LAN", In Proceedings of IEEE Vehicular Technology Conference (VTC), 2002 108, 141

[145] C. Ng, P. Thubert, M. Watari, F. Zhao. "Network Mobility Route Optimization Problem Statement" Internet Draft, Work in Progress, February 2006 108

[146] M. Calderon, C. J. Bernardos, M. Bagnulo, I. Soto, "Securing Route Optimization in NEMO" In Proc. of the International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPTS), 2005 108, 142

[147] C. Ng, J. Hirano, "Extending Return Routability Procedure for Network Prefix (RRNP)", Internet Draft, Work in Progress, 2004 108, 142

[148] Cisco Systems (online) http://www.cisco.com 108, 109

[149] Linux Virtual Server Project (online) http://www.linuxvirtualserver.org 108

[150] F5 Networks BIG/IP (online) http://www.f5labs.com/products/bigip/ 108, 109

[151] Foundry Networks ServerIron (online) http://www.foundrynet.com/products/webswitches/serveriron/docs.html 108, 109

[152] IBM Network Dispatcher (online) http://www.ibm.com/software/webservers/edgeserver/ 108, 109

[153] Coyote Point Systems Equalizer (online) http://www.coyotepoint.com 108

[154] Allot Communications NetEnforcer (online) http://www.allot.com 108, 109

[155] Nortel Networks (online) http://www.nortelnetworks.com 108, 109

[156] Radware Inc. (online) http://www.radware.com 108, 109

[157] Lucent Web Switch: (online) http://www.bell-labs.com/project/webswitch 109

[158] Zeus ZXTM-LB (online) http://www.zeus.com 109

[159] TCPSP for the Linux Kernel: (online) http://www.linuxvirtualserver.org/software/tcpsp/index.html 109

[160] F.Heissenhuber, W. Fritsche and A. Riedl, "Home Agent Redundancy and Load Balancing in Mobile IPv6" In Proc. of the Conference on Broadband Communications (BC), 1999 111, 119, 120, 121, 122

[161] H.Deng et al, "Load Balance for Distributed Home Agents in Mobile IPv6" In Proc. of the Personal, Indoor and Mobile Radio Communications (PIMRC), 2003 111, 119, 120, 121, 122

[162] H.Deng et al, "Load Balance for Distributed Home Agents in Mobile IPv6" IETF Draft, Work in Progress, 2003 111, 119, 120, 121, 122

[163] J. Faizan et al, "Efficient Dynamic Load Balancing for Multiple Home Agents in Mobile IPv6 based Networks" In Proc. of the Pervasive Services (ICPS), 2005 111, 119, 120, 121, 122

[164] R. Wakikawa et al, "Home Agent Reliability Protocol" IETF Draft, Work in Progress, 2006. 111, 119, 120, 121, 122

[165] R. Wakikawa et al. "Inter Home Agents Protocol Specifications" IETF Draft, Work in Progress, 2006 112, 119

[166] R. Wakikawa et al, "Virtual mobility control domain for enhancements of mobility protocols" In Proc. of IEEE INFOCOM, 2005 112, 119, 123, 142

[167] Y. Rekhter et al, "A Border Gateway Protocol 4 (BGP-4)" RFC 1771, 1995 112, 115

REFERENCES

[168] J.Abley et al, "Goals for IPv6 Site-Multihoming Architectures" RFC 3582, 2003 117

[169] PalChaudhuri, S. et al, "Perfect Simulations for Random Trip Mobility Models" In Proc. of IEEE Simulation Symposium, 2005 120, 132

[170] Y.S.Yet et al, "Global Dynamic Home Agent Discovery on Mobile IPv6" In Proc. of Wireless Communications and Mobile Computing, 2006 123, 142

[171] Boeing Connexion Service (online) http://www.connexionbyboeing.com 123, 142

[172] Marcelo Bagnulo et al, "Scalable Support for Globally Moving Networks" In Proc. of Wireless Communication Systems (ISWCS), 2006 123, 142

[173] G.Huston, "Commentary on Inter-Domain Routing in the Internet" RFC 3221, 2001. 123, 142

[174] Katabi, Dina et al, "A Framework for Scalable Global IP-Anycast (GIA)" SIGCOMM Comput. Commun. Rev. Vol. 31, Num. 2, 2001 123, 142

[175] K. Lua et al, "A Survey and comparison of peer-to-peer overlay network schemes" IEEE Communications Surveys & Tutorials, 2005 126, 138

[176] I. Stoica et al, "Chord: A scalable peer-to-peer lookup service for internet applications" IEEE/ACM Transactions on Networking, Vol. 11, Num. 1, 2005 126

[177] R. Housley et al, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile" RFC 2459, 1999 129

[178] European Telecommunications Standards Institute, "GSM 03.20: Security Related Network Functions" 1999 131

[179] European Telecommunications Standards Institute, "GSM 02.09: Security Aspects" 1993 131

[180] European Telecommunications Standards Institute, "TS 133 102: Universal Mobile Telecommunications System (UMTS); 3G Security; Security Architecture" version 3.6.0, 2000 131

[181] N.Brownlee et al, "Understanding Internet traffic streams: dragonflies and tortoises" IEEE Communications Magazine, Vol. 40, Nun. 10, 2002 128

[182] Internet Topology Generator (online) http://topology.eecs.umich.edu/inet/ 132

[183] Passive Measurement and Analysis (PMA) (online) http://pma.nlanr.net 132

[184] Amit Jardosh, Elizabeth M. Belding-Royer, Kevin C. Almeroth, and Subhash Suri, "Towards realistic mobilit ymodels for mobile ad-hoc networks" In Proc. International conference on Mobile computing and networking (MOBICOM), 2003 132

[185] S. Jamin et al, "On the Placement of Internet Instrumentation" In Proc. of IEEE INFOCOM, 2000 132

[186] N1 Grid Engine (online) www.sun.com/software/gridware/ 133

[187] Wolff. R, "Poisson Arrivals See Time Average" in Operational Research, 1982 173, 178

[188] M. Jain, C. Dovrolis, "End-to-End Available Bandwidth: Measurement methodology, Dynamics, and Relation with TCP Throughput" SIGCOMM Comput. Commun. Rev., Vol. 32, Num. 4, 2002 206

[189] Ekelin, S., Nilsson M., Hartikainen E., Johnsson A., Mångs J.E., Melander B., and Björkman M., "Real-Time Measurement of End-to-End Available Bandwidth using Kalman Filtering" In Proc. Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS), 2006 214

[190] Alok Shriram and Jasleen Kaur, "Empirical Evaluation of Techniques for Measuring Available Bandwidth" In Proc. IEEE INFOCOM, 2007 175, 192, 202, 206, 207, 208, 210, 211, 214

[191] Francçois Baccelli, Sridhar Machiraju, Darryl Veitch and Jean C. Bolot, "The Role of the PASTA property in Network Measurement" SIGCOMM Comput. Commun. Rev., Vol. 36, Num. 4, 2006 178

[192] Rudolph Emil Kalman, "A New Approach to Linear Filtering and Prediction Problems" Journal of Basic Engineering, Vol. 82, Num. 1, 1960 186, 187

[193] R.Prasad and M.Jain, "Effects of Interrupt Coalescence on Network Measurements" In Proceedings of Passive and Active Measurements (PAM), 2004 189

[194] Rene Serral-Gracia, Albert Cabellos-Aparicio and Jordi Domingo-Pascual, "Network performance assessment using adaptive traffic sampling" In Proc. of International IFIP-TC6 Networking Conference Networking (Networking), 2008 189

[195] D. P. Pezaros, D. Hutchison, R. D. Gardner, F. J.Garcia, J. S. Sventek, "Inline Measurements: A Native Measurement Technique for IPv6 Networks" In Proc. of International Networking and Communication Conference (INCC), 2004 190

[196] Passive Measurement and Analysis (PMA) (online) http://pma.nlanr.net/Special/ 190

[197] M. Jain and C. Dovrolis, "End-to-End Estimation of the Available Bandwidth Variation Range" In Proc. ACM SIGMETRICS international conference on Measurement and modeling of computer systems, 2005 193, 196

[198] J. Kilpi and I. Norros, "Testing the Gaussian Approximation of Aggregate Traffic" In Proc. of the ACM SIGCOMM Workshop on Internet measurment, 2002 193

[199] Zhang Z.-L., Ribeiro V.J., Moon S., Diot, C., "Small-time scaling behaviors of Internet backbone traffic: an empirical study" In Proc. of IEEE INFOCOM, 2003 198, 213

[200] Cooperative Association for Internet Data Analysis "NASA Ames Internet Exchange Packet Length Distributions" 178, 197, 198, 206

[201] pathChrip Software (online) http://www.spin.rice.edu/Software/pathChirp/ 207

[202] Cao Le Thanh Man et al, "A Merged Inline Measurement Method for Capacity and Available Bandwidth" In Proceedings of Passive and Active Measurements (PAM), 2005 215

[203] M. Zangrilli et al, "Applying Principles of Active Available Bandwidth Algorithms to Passive TCP traces" In Proceedings of Passive and Active Measurements (PAM), 2005 215

[204] Katti S., Katabi D., Blake C., Kohler E., and Strauss, J., "MultiQ: Automated Detection of Multiple Bottleneck Capacities Along a Path" In Proc. of the ACM SIGCOMM conference on Internet measurement (IMC), 2004 215

[205] E. Rescorla, "Diffie-Hellman Key Agreement Method" RFC 2631, 1999 26

[206] Issariyakul T. et al., "Exact Distribution of Service Delay in IEEE 802.11 DCF MAC", In Proc. of Global Telecommunications Conference (GLOBECOM), 2005 156

[207] Shahrokh Valaee, "Bandwidth Estimation and Distributed Traffic Regulation in Wireless Local Area Networks" In Proc. of the Personal, Indoor and Mobile Radio Communications (PIMRC), 2008 156, 163

[208] de A. Rocha, A. A. Leao, R. M. M. de Souza e Silva, "An End-to-End Technique to Estimate the Transmission Rate of an IEEE 802.11 WLAN" In Proc. of IEEE International Conference on Communications (ICC), 2007 147, 196

[209] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function" IEEE Journal on Selected Areas in Communications, Vol. 18, Num. 3, 2000 156

[210] M. Portoles-Comeras et al. "EXTREME: Combining the ease of management of multi-user experimental facilities and the flexibility of proof of concept testbeds" In proc. of the IEEE TridentCom, 2006 149

[211] Multi-Generator (MGEN) (online) http://cs.itd.nrl.navy.mil/work/mgen/ 149

[212] Attila Pásztor and Darryl Veitch, "PC Based precision timing without GPS" SIGMETRICS Perform. Eval. Rev. Vol. 30, Num. 1, 2002 149

[213] The Network Simulator NS2 (online) http://www.isi.edu/nsnam/ns/ 150

[214] Liu X., Ravindran K., Liu B., and Loguinov, D., "A Queuing-Theoretic Foundation of Available Bandwidth Estimation: Single-Hop Analysis" IEEE/ACM Transactions on Networking, Vol. 15, Num. 6, 2007 153, 154, 158, 164

[215] NIST/SEMATECH e-Handbook of Statistical Methods (online) http://www.itl.nist.gov/div898/handbook/ (Sec. 1.3.5.6) 157, 191

[216] P.Haga, K. Diriczi et al, "Granular model of packet pair separation in Poissonian traffic" Elsevier Computer Networks, Vol. 51, Num. 3, 2006 154

[217] Richard Draves,Jitendra Padhye and Brian Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks" Proceedings of the ACM International conference on Mobile computing and networking (MOBICOM), 2004 148, 169

[218] Dovrolis C., Ramanathan P., and Moore, D. "Packet dispersion techniques and a capacity estimation methodology" IEEE/ACM Transaction on Networking, Vol. 12, Num. 6, 2004 148, 168, 180

[219] David Meyer, "The Locator Identifier Separation Protocol (LISP)" The Internet Protocol Journal, Volume 11, No. 1, 2008 225

[220] Farinacci, D. et al., "Locator/ID Separation Protocol (LISP)" Internet Draft, Work in Progress, 2008 225

[221] Fuller, V. et al., "LISP Alternative Topology (LISP-ALT)" Internet Draft Work in Progress, 2008 225

[222] Fast Handovers for Mobile IPv6 Implementation, (online) http://personals.ac.upc.edu/acabello/fmipv6/ 9, 41, 70, 82, 222

[223] K. Leung et al, "Network Mobility (NEMO) Extensions for Mobile IPv4" RFC 5177 2008 39

[224] International Telecommunications Union, (online) http://www.itu.int 4

[225] Alex C. Snoeren, "A Session-Based Approach to Internet Mobility" PhD Dissertation, Massachusetts Institute of Technology, 2002 21

[226] J. Postel et al, "Transmission Control Protocol" RFC 793, 1981 5

[227] J. Postel et al, "User Datagram Protocol" RFC 768, 1980 24

[228] R. Droms, "Dynamic Host Configuration Protocol" RFC 2131, 1997 34

[229] T.S.Rappaport, Wireless Communications Prentice Hall, 1996 197

[230] Xiao-Hui Lin et al, On Channel-Adaptive Routing in an IEEE 802.11b Based Ad Hoc Wireless Network, In Proc. of Global Telecommunications Conference (GLOBECOM), 2003 197

[231] Steger, C.; Radosavljevic, P.; Frantz, J.P., Performance of IEEE 802.11b wireless LAN in an emulated mobile channel, In Proc. of the Vehicular Technology Conference (VTC), 2003 197