EFFICIENT QUALITY OF SERVICE PROVISIONING FOR MOVING NETWORKS

Kamel, George

Awarding institution:
King's College London

EFFICIENT QUALITY OF SERVICE PROVISIONING FOR MOVING NETWORKS

George Kamel

Centre for Telecommunications Research
King's College London
London WC2R 2LS

May 2010

A thesis submitted to the University of London for the degree of

Doctor of Philosophy

A particular shot or way of moving the ball
can be a player's personal signature, but efficiency
of performance is what wins the game for the team.
Pat Riley

Abstract

Providing Internet connectivity on public transport has the potential to open up a new dimension of consumer entertainment and productivity. However, the potentially large population density and high handover rates of mobile users aboard a public transport vehicle necessitate robust and scalable quality-of-service (QoS) provisioning mechanisms designed for such environments that can improve both pre-session and in-session signalling efficiency.

A number of Quality of Service (QoS) aggregation policies are proposed that reduce the frequency with which QoS requests are made to a network, and hence increase overall pre-session signalling efficiency. However, since these policies are based on a static, request-rate-dependent parameter, operational inefficiency can occur under highly variant rates of request. Therefore, a cost-driven policy is proposed that is shown to increase signalling efficiency compared with other policies, while, at the same time, not putting users at a disadvantage with long and unpredictable waiting times to establish a session.

When the access network becomes congested, signalling efficiency is drastically reduced under the cost-driven policy. Therefore, two separate "overlay" policies are proposed to work in place of the cost-driven aggregation policy during periods of congestion: a dynamic policy and a static policy. The static policy is shown to significantly out-perform the case in which no overlay policy is used, significantly increasing cost-efficiency whilst reducing user waiting times.

Finally, attention is given to the issue of in-session QoS provisioning. Micro-mobility protocols play an important role in providing seamless handover support to terminals. However, such protocols typically suffer from bottleneck congestion, which can lead to a degradation of signalling efficiency and reduced QoS when they are used by moving networks. Therefore, a novel mechanism is proposed that alleviates these problems, and an implementation of the protocol within the Next Steps in Signaling framework is detailed.

Acknowledgements

Throughout my research, I have had the valued support and guidance of many, to whom I would like to express my sincere gratitude and appreciation. First, I wish to thank Prof Hamid Aghvami for his invaluable advice and support throughout this research. My sincere thanks also to Dr Andrej Mihailovic, who has been a cornerstone of this research through all the time and energy that he has devoted. In particular, I wish to thank him for our many fruitful discussions which have been the inspiration for a lot of the ideas contained in this thesis, and for his thorough reviews and feedback of my reports and publications.

I was particularly fortunate to have been part of the Virtual Centre of Excellence in Mobile & Personal Communications (Mobile VCE) Core 4 project, through which I have had the opportunity and privilege of working with many people from both industry and academia, and of presenting my work before them. Specifically, I would like to thank Dr Paul Pangalos for his leadership and support throughout the project. Sincere appreciation also to Dr Walter Tuttlebee, Dr Haitham Cruickshank and Mr Stephen Hope for their supportive guidance and advice. I would also like to acknowledge the Engineering and Physical Sciences Research Council and Mobile VCE for providing funding for this work.

Over the course of my research, I have had the pleasure of working with many colleagues and friends at King's College London. I particularly wish to thank Dev Pragad Audsin who has been a wonderful friend and travel companion, and whose moral support has made these past few years all the more enjoyable. My thanks also to Toktam Mahmoodi, Sampath Ranasinghe, Haffiz Shuaib, Thikriat Al-Mosawi, Uma Shanker, Alireza Attar, and many others who have all made the Centre for Telecommunications Research a pleasant and productive place to work.

I owe a debt of gratitude to my family and friends who have supported and encouraged me throughout these past years. Most of all, I thank God for guiding me to this point, for He is *able to do immeasurably more than all we ask or imagine, according to His power that is at work within us.*[NIV] (Ephesians 3:20).

Table of Contents

List of Figures

List of Symbols

α	Aggregation-Utility Parameter of the C-Policy
α_{RQ}	Aggregation Utility of RQ Queue
α_{TD}	Aggregation Utility of RQ Queue
β	Lower Congestion Threshold
β^*	Optimal Lower Congestion Threshold
Γ^*	Set of Optimal Lower Congestion Thresholds for all MMPP states
Γ_P	Set of Optimal Lower Congestion Thresholds for a P-State MMPP
λ	QoS Request Arrival Rate
λ_A	Expected QoS Arrival Rate at State A
λ_B	Expected QoS Arrival Rate at State B
λ_P	Maximum QoS Request Arrival Rate of P-State MMPP Model
$\mathcal{R}$	Amount of Resources Associated with Indexed Message
$\Pr(S)$	Probability of a State in Set, S
$\Pr(AB_p)$	Transition Probability from State A to B_p
ω	Waiting Time of Indexed QoS Request
ω_x	Waiting Time of Any Request, x
ϕ	General Policy Parameter
ϕ^*	Optimal General Policy Parameter
τ	Maximum Allowable Waiting Time of a QoS Request
θ	General Cycle Duration
$\tilde{\alpha}_{RQ}$	Aggregation Utility of RQ Queue under the D-Policy
$\tilde{C}_c$	Cost of Aggregation Cycle under the D-Policy
A	Ambient State
B	Bursty State of 2-State MMPP

B_p	Bursty State with Expected Rate, λ_{B_p}
c	Aggregation Cycle Index
C_c	Cost of Aggregation Cycle, c
C_d	Cost of Tear-Aggregation Cycle, d
C_h	Base Cost of Holding a QoS Request
C_r	Cost of Users Reneging
C_s	Cost of Sending a QoS Request
C_t	Cost of Tearing Down Resources
C_u	Cost of Resource Under-Utilisation
d	Tear-Down Cycle Index
i	Aggregated Request Index
j	Reneged Request Index
K	K-Policy Parameter
k	Tear-Down Message Index
K^*	Optimal K-policy parameter
M_c	Number of Reneged QoS Requests During Aggregation Cycle, c
N_c	Number of Queued Requests
N_d	Number of Queued Tear-Down Messages
P	Number of Bursty States in Multi-State MMPP Model
R	R-Policy Parameter
R^*	Optimal R-policy parameter
r_A	Sojourn Time of State A
r_B	Sojourn Time of State B
T	T-Policy Parameter
T^*	Optimal T-policy parameter
T_c	Duration of Aggregation Cycle, c
T_d	Tear-Down Cycle Index
x	Aggregated Request Index of Any Cycle
$\lceil \cdot \rceil$	Ceiling Value
$E[\cdot]$	Statistical Expectation

List of Acronyms

3GPP	Third Generation Partnership Project
AF	Assured Forwarding
AN	Access Network
AR	Access Router
BA	Binding Acknowledgement
BB	Bandwidth Broker
BE	Best Effort
BU	Binding Update
CL	Controlled Load
CN	Correspondent Node
CoA	Care-of-Address
DAD	Duplicate Address Detection
DiffServ	Differentiated Services
DRO	Dynamic Route Optimisation
DSCP	DiffServ Code Point
EF	Expedited Forwarding
EN	Enhanced Node
FCFS	First-Come, First-Served
FMIPv6	Fast Handovers for Mobile IPv6
GIST	General Internet Signaling Transport
GS	Guaranteed Service
HA	Home Agent
HMIPv6	Hierarchical Mobile IPv6
HoA	Home Address
HSDPA	High-Speed Downlink Packet Access

HSPA+	High-Speed Packet Access Evolution
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IP	Internet Protocol
IPv6	IP Version 6
LCoA	Local CoA
LFN	Local Fixed Node
LMN	Local Mobile Node
LOS	Line-of-Sight
LTE	Long Term Evolution
MAP	Mobility Anchor Point
MEXT	Mobility EXTensions for IPv6
MIP	Mobile IP
MIPv6	Mobile IPv6
MMPP	Markov-modulated Poisson process
MN	Mobile Node
MNN	Mobile Network Node
MNNHA	Mobile Network Node Home Agent
MR	Mobile Router
MRHA	Mobile Router Home Agent
MRI	Message Routing Information
MRSVP	Mobile RSVP
NBS	NEMO Basic Support
NE	NSIS Entity
NEMO	Network Mobility
NEMOR	NEMO Reservation
NSIS	Next Steps in Signaling
NSLP	NSIS Signalling Layer Protocol
NTLP	NSIS Transport Layer Protocol
PAN	Personal Area Network

PHB	Per Hop Behaviour
PSBU	Prefix-Scope Binding Update
PTV	Public Transport Vehicle
QENEMO	QoS-Enabled Micro-Mobility for Network Mobility
QBU	QoS-Extended Binding Update
QNE	QoS-NSLP NSIS Entity
QNI	QoS-NSLP NSIS Initiator
QNR	QoS-NSLP NSIS Receiver
QoS	Quality of Service
QOSPF	QoS Open Shortest Path First
RA	Router Advertisement
RCoA	Regional CoA
RG	Routing Group
RSVP	ReSerVation Protocol
RQ	Request Message
SNMP	Simple Network Management Protocol
TD	Tear-Down Message
VMN	Visiting Mobile Node
VMR	Visiting Mobile Router
VoIP	Voice-over-IP
WiMAX	Worldwide Inter-operability for Microwave Access
WLAN	Wireless LAN

Chapter 1

Introduction

Providing reliable Internet access on public transport has the potential to open up a new dimension of commuter entertainment and productivity, as well as lucrative opportunities for network operators and public transport operators alike. From a survey conducted in 2004 [1], it was found that 78% of business passengers questioned in the United Kingdom would take advantage of Internet connectivity if it were available to them. However, five years on, only a handful of train operators have rolled out some form of broadband service offering, whereby users connect to a *mobile router* (MR) located on the vehicle which manages the connection to the fixed network on their behalf. However, such offerings are still very much in their infancy, with service quality paling in comparison to that of fixed broadband.

Admittedly, much of the difficulty of providing reliable Internet connectivity on Public Transport Vehicles (PTVs) is the mere scarcity of wireless resources. For example, 3GPP's release 7, High-Speed Packet Access Evolution (HSPA+), is able to offer peak data rates of 14.1 Mb/s [2] which drops significantly for terminals located towards the edge of a cell. Forthcoming 3GPP releases such as Long Term Evolution (LTE) and "LTE Advanced" promise yet further increases to data rates. However, increasing wireless throughput usually serves only to encourage the use of applications with higher quality content, resulting in an increase of the typical

resource requirements of applications, and for users that are in turn more discerning over service quality. A good example of this is the progression of video technology from traditional "standard-definition" to the current "high-definition" to the future "three-dimensional," each one requiring significantly greater resources than the previous.

The problem of limited network resources is further exacerbated by the high population density of mobile users within a PTV. Users that would usually be dispersed across multiple cells in the network are suddenly found clustered together in cells within a very small region, potentially leading to overloading of the wireless resources of those cells. However, such problems can be ameliorated to some degree by ensuring more efficient management and utilisation of the resources that *are* available. In this respect, the MR provides a convenient platform upon which such management tasks can be implemented and carried out, as it lies directly between the terminals and the wireless interface. In addition, since the MR is not bounded in design by the same size and power constraints faced by mobile terminals, the MR is better placed to combat physical phenomena such as Doppler-shifting and fast-fading that become particularly prominent when travelling at high velocities.

The principle of using an MR to improve the operational efficiency of a moving network is not new in itself, and has already been applied successfully to a number of protocols. For instance, the Network Mobility (NEMO) Basic Support protocol [3] standardised by the IETF allows for the IP mobility of a moving network to be managed in a way that is independent of the number of terminals on the vehicle, based on a single address prefix that is common to all terminals. This therefore prevents congestion from occurring in the network during handover as a result of mobility control signalling being sent by or for each individual terminal. Similarly for QoS provisioning, protocols such as NEMO Reservation (NEMOR) [4, 5] facilitate the re-establishment of QoS forwarding states within a network after a handover, again in a scalable manner that is independent of the number of terminals on the vehicle. However, there is a lack of interaction between QoS and mobility mechanisms which

can lead to a number of inefficiencies arising in the way in which QoS is provisioned. Current QoS mechanisms are agnostic of both the mobility of passengers as they board and alight from a vehicle, as well as of the mobility patterns of the vehicle as a whole. This results in a situation in which the provisioning of QoS, originally designed to ensure more efficient use of resources, can potentially become more expensive than the resources it manages to save [6]. Therefore, this thesis takes a more holistic approach to QoS provisioning by considering the dynamics of both passengers and the moving network as a whole, in order to efficiently deliver QoS-enabled services to potential on-board users.

1.1 Motivation

Support for QoS-provisioning mechanisms represents an important way of leveraging the capacity of a network and of ensuring that resources are fairly and proportionally rationed amongst its multiple users according to the requirements and service agreements of each. QoS skeptics commonly argue that the benefits of QoS tend to be outweighed by the complexity of deploying such mechanisms, and that a simpler way of providing QoS is to ensure that resources are sufficiently over-provisioned and distributed amongst potential users on a best-effort basis [7]. However, this argument has a number of flaws. First, over-provisioning can be feasibly done within only the wired part of the network, as the bandwidth of the wireless part is fundamentally limited. Second, even if it were possible to over-provision the resources of the wireless channel, best-effort does not allow for the regulation of bandwidth consumption by individual users, as transport-layer protocols usually tend to want to maximise end-to-end throughput [8]. Finally, complexity aside, studies such as that carried out by Abella *et al.* [9] have shown that to obtain the same QoS levels in the best-effort case as the QoS-provisioned case, resources would need to be over-provisioned by 10% to 40%, depending upon the mix of traffic being carried across the network. This translates to an increase in the cost of the system in the same order of magnitude.

In order to maintain the benefits that QoS provisioning can offer, the QoS provisioning mechanism must operate efficiently at all times. In this respect, moving networks offers an immediate advantage in that QoS can be managed on aggregate for all users aboard the PTV; the NEMOR protocol mentioned before (and which will be elaborated on in the next chapter) is a good example of this. However, in spite of the existence of such protocols, providing efficient QoS-support to users is still a challenging task. This is due in part to the highly dynamic nature of moving networks, and in part to the sheer number of users that may be present on a vehicle which may potentially be in the order of hundreds. Therefore, such challenges can be divided into two main categories: *pre-session* and *in-session*, as illustrated in Figure 1.1. The first of these can be attributed to the characteristics of passenger movement with respect to the vehicle, and the latter to the movement of the vehicle as a whole. This thesis makes contributions to tackle both of these difficulties, and are presented in this thesis in that particular order. However, the following sections discuss the challenges of in-session QoS support before those of pre-session, so as to ease understanding.

1.1.1 In-Session QoS Support

As a PTV moves, it will be required to maintain session continuity for the terminals that it is serving by performing handovers seamlessly between networks. However, due to the high velocities at which PTVs travel, the frame of time within which a handover must be executed can be very small, as the time spent in an overlapping region of network coverage is reduced in comparison to the case of individual terminals moving at walking pace. The handover task of an MR is compounded by the sheer amount of resources for which it must find and reserve capacity, particularly as resource availability is likely to vary between networks.

A number of QoS protocols have been proposed which attempt to reduce the time for which sessions are without QoS support during a handover. These are based on the principle of aggregation, whereby the MR communicates to the network the require-

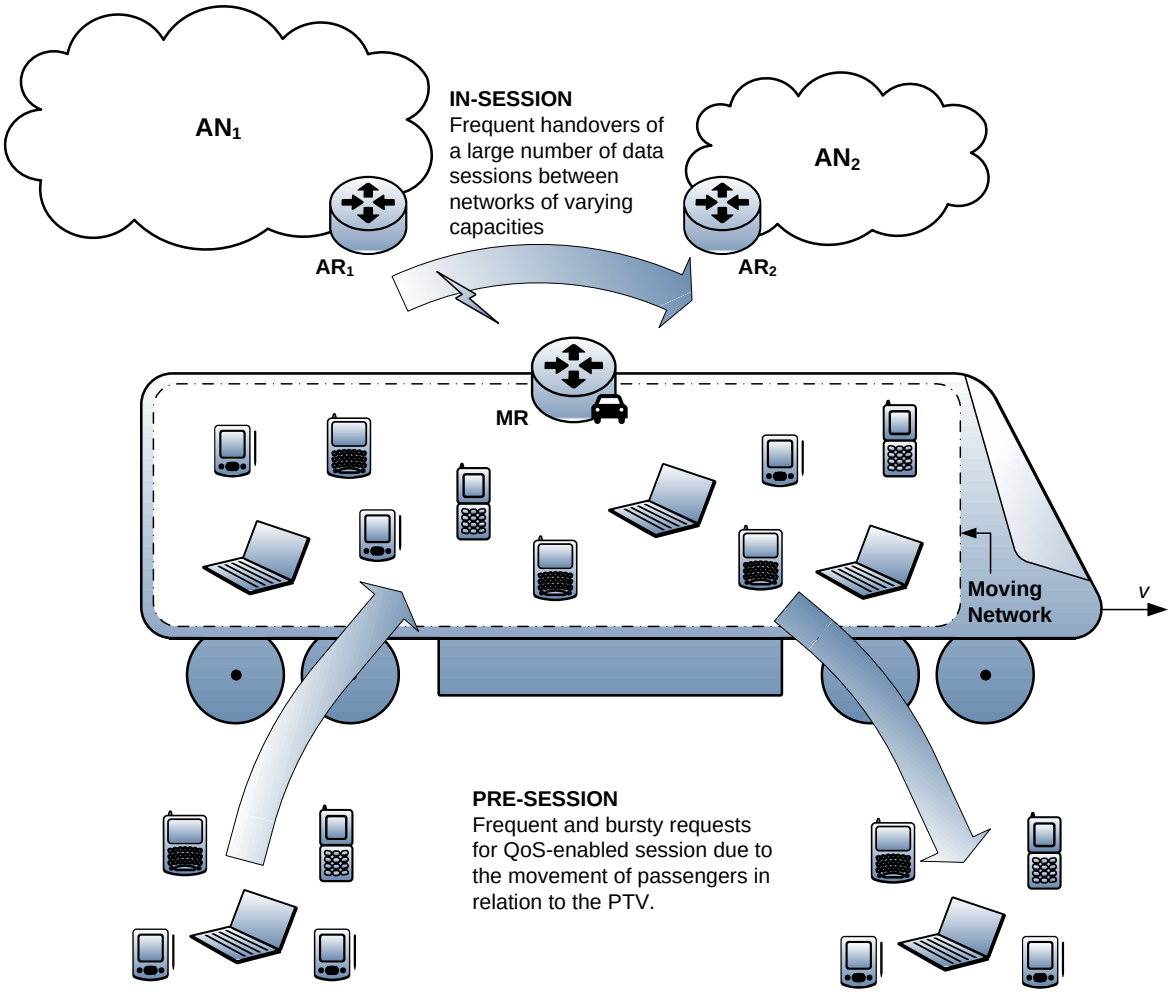


Figure 1.1: Dynamic nature of moving networks that makes the delivery of efficient QoS challenging.

ments of all its users as opposed to each individual user. For example, the NEMOR protocol [4, 5] will establish a single QoS state along a path within the network, within which queues are established to provide application-based packet prioritisation. However, with the use of micro-mobility protocols (see Section 2.3.1)—which are typically used in host mobility scenarios to reduce the average packet loss during handover by expediting the handover process—a conflict between seamless mobility and QoS-support arises. Specifically, micro-mobility protocols rely on the use of mobility agents within the Access Network (AN) to deliver seamless mobility support to users which, by virtue of its operation, places a bound on network capacity, despite the likely existence of less congested paths. This can be problematic for an MR needing to handover sessions with more requirements than any single mo-

bility agent within the network can support. An efficient mechanism is therefore required to ensure QoS-support during handover for moving network with support for micro-mobility.

In-session QoS is addressed in Chapter 5.

1.1.2 Pre-Session QoS Support

As PTVs typically make several stops throughout a journey to let passengers board, surges in session requests are likely to occur. Each surge will increase the signalling burden placed on the network, as each session request would trigger the MR to extend the aggregate resource reservation state of the moving network within a very short space of time. Similarly, as passengers alight from a PTV in batches, a surge in session terminations (as far as the MR is concerned) will arise. This will lead to the MR having to make individual reductions to the aggregate QoS states of the fixed network, contributing to the signalling burden and the resultant operational inefficiency of the network.

For users that request pre-session QoS support, i.e. the QoS support being requested is not for a session being handed over from another network, then it is possible to let those requests be buffered at the MR. This will allow it to aggregate several requests that arrive within a window of time, such that the frequency with which the MR signals to the AN is reduced. However, one of the main drawbacks of this approach is that whilst users are waiting to be connected, they are not transferring data. This can result in reduced network operator revenue, as well as reduced overall user satisfaction. Therefore, there is a clear need for QoS aggregation policies that can efficiently control the aggregation of requests at the router, whilst ensuring that the disadvantage to the user in terms of waiting time is minimised.

Pre-session QoS is tackled in both Chapters 3 and 4.

1.2 Contributions

This thesis has made the following contributions:

- To address some of the issues of in-session QoS support, a patent-pending mechanism called QoS-Enabled Micro-Mobility for Network Mobility (QENEMO) was proposed that allows for the use of a micro-mobility protocol without the operational inefficiencies that arise from the large number of sessions for which QoS states must be established.

- An implementation of the proposed QENEMO mechanism was designed within the Next Steps in Signaling (NSIS) framework, setting out a suitable node architecture that reuses, where possible, the features of an existing NSIS QoS protocol. Designs of the signalling messages in terms of both format and content are also detailed.

- In terms of pre-session QoS support, the cost-efficiency of existing QoS aggregation policies was studied by means of mathematical analysis and computer simulation, and it was shown that the control parameter on which each of the existing policies is based is dependent upon the rate at which users request sessions, making them unsuitable for the bursty environment of a moving network.

- To counter the problems of existing QoS aggregation policies, a novel cost-driven policy (C-policy) is proposed that operates in such a way that aims to achieve cost-optimality (and hence operational efficiency), regardless of the burstiness of QoS requests at the MR. It was shown that in addition to reducing operating costs and increasing operational efficiency, the average time that a user must wait before obtaining QoS-connectivity was also reduced over existing policies.

- Finally, the assumption that network capacity is always greater than user demand was relaxed in the following part of the work on QoS aggregation, which

led to the need for a mechanism to manage the aggregation and processing of QoS requests when the network is congested. Therefore, two "overlay" QoS aggregation policies were proposed, which attempt to increase the QoS signalling efficiency during periods of high network congestion. The proposed S-policy in particular provided benefits in both cost and user waiting time over the case in which no specific policy is used.

1.3 Publications

Journals

1. G. Kamel, A. Mihailovic, and A. Hamid Aghvami, "A Cost-Optimal QoS Aggregation Policy for Network Mobility: Analysis and Performance Comparisons," IEEE Transactions on Vehicular Technology, vol. 58, no. 7, September 2009, pp. 3547–3557.

2. G. Kamel, A. Mihailovic, and A. Hamid Aghvami, "Case Analysis of a Cost-Optimal QoS Aggregation Policy for Network Mobility," IEEE Communications Letters, vol. 12, no. 2, February 2008, pp. 130–132.

Patents

3. G. Kamel, P. Pangalos, A. Mihailovic, and A. Hamid Aghvami, "Improvements in or Relating to Network Mobility," UK Patent Application No. GB08 22815.7, December 2008.

Conferences

4. G. Kamel, A. Mihailovic, P. Pangalos, and A. Hamid Aghvami, "A Seamless, QoS-Enabled Mobility Management Mechanism for Moving Networks," *in* Proc. International Conference on Telecommunications, ICT '09, May 2009, pp. 228–231.

5. A. Dev Pragad, G. Kamel, P. Pangalos, and A. Hamid Aghvami, "A Combined Mobility and QoS Framework for Delivering Ubiquitous Services," IEEE Inter-

national Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC'08, Cannes, France, 15–18 September 2008.

6. G. Kamel, A. Mihailovic, P. Pangalos, and A. Hamid Aghvami, "Cost-Optimal QoS Aggregation for Network Mobility," *in* Proc. IEEE Global Telecommunications Conference, GLOBECOM'07, November 2007, pp. 5006–5010.

7. G. Kamel, K. Altali Alhames, P. Pangalos, A. Mihailovic, and A. Hamid Aghvami, "Synergistic QoS and Mobility Management for Moving Networks: Problems and Solutions," Wireless World Research Forum (WWRF) 16th meeting, Shanghai, China, 26–28 April 2006.

1.4 Thesis Structure

The remainder of this thesis is structured as follows:

The following chapter provides a review of existing work surrounding moving networks relevant to the contributions of this thesis. The scope of the background extends from the architecture of moving networks, to mobility management protocols and QoS provisioning techniques proposed for both host and network mobility. The chapter is concluded with a discussion and summary of the mechanisms that will found the contributions of this thesis, as well as the cross-issues between mobility and QoS that these contributions will address.

The main contributions of this thesis are embodied within Chapters 3 to 5. Chapter 3 presents a mathematical analysis of the cost-efficiency of QoS aggregation policies that have been proposed in previous works, and shows how such policies are dependent on the rate of QoS-enabled sessions being requested by users on the PTV, proving its unsuitability for situations in which QoS requests are bursty. Therefore, the chapter then proposes a cost-driven QoS aggregation policy (C-policy), proving that this policy is cost-optimal for all arrival rates of QoS requests. A performance

comparison is then given of all discussed policies, by means of both mathematical analysis and simulation via MATLAB.

Chapter 4 extends the work of Chapter 3 by tackling the problem of inefficient QoS provisioning when the network is close to saturation. Two "overlay" policies (the S-policy and D-policy) are proposed, which work in place of the C-policy during periods of high congestion to increase overall QoS-provisioning efficiency. Both of these policies were simulated using MATLAB, and the performance of each is compared to the case in which no specific overlay QoS aggregation policy is used.

Chapter 5 then focuses on the problems of inefficiency surrounding in-session QoS support for moving networks. A mechanism is proposed, called QENEMO, which facilitates handover of a large number of sessions in such a way that eliminates excessive signalling to the fixed network, and as a result minimises the time for which sessions are without QoS support during handover. Signalling procedures are given for both *inter*-AN and *intra*-AN scenarios, and other supporting procedures required for the operation of the proposed mechanism are also touched upon. The latter part of the chapter then sets out the implementation design of the QENEMO mechanism within the NSIS framework, detailing the node architecture, and proposed message and object formats.

Finally, the thesis is concluded in Chapter 6, summarising the salient contributions and results of this work, and providing potential avenues for future research.

Chapter 2

Background and Related Work

2.1 Introduction

Since the dawn of mobile communications, new communication protocols that manage some aspect of the connection of the mobile terminal to the fixed network have continually been appearing. However, such protocols were designed to run on individual terminals, and manage the connection of only that user. Using these protocols to manage groups of terminals moving in unison can lead to a reduction in the overall protocol efficiency, resulting in a reduction of the resources available for data plane traffic. Therefore, in order to achieve efficient protocol operation, tailored solutions are required that take into account the characteristics of the environment under which moving networks usually operate.

In this chapter, the background and state-of-the-art research with respect to the work presented in the remainder of this thesis is presented. The first part of this chapter surveys the various architectural approaches that have been taken for moving networks. This is followed by a description of both mobility and QoS protocols; first those proposed for host mobility scenarios, and then those proposed for network mobility scenarios which tend to build upon the principles of the former. After separately covering both mobility and QoS, a discussion and summary is given

of the background presented in this chapter, as well as the cross-issues that exist *between* mobility and QoS that can lead to a degradation of the efficiency of QoS provisioning.

2.2 Architectural Overview of Moving Networks

One of the earliest proposed communication architectures designed to increase the data connection reliability of a user moving in a vehicle was proposed in the early 1990s by Hager *et al.* in his paper entitled, *MINT – A mobile Internet router* [10]. This paper outlined the workings of a device with *sufficient computational power to perform all necessary communication protocol operations*. One of the primary motivations for such a device was to eliminate the additional burden placed on the power consumption of Mobile Nodes (MNs) in having to carry out extra communication procedures such as mobility management as it roams between access routers and networks. However, as battery life continues to rise with improvements to battery technology, and with the recent *universal* drive towards reduced-energy radio and network communications, the focus of subsequent generations of moving network architectures have shifted towards improving the experienced QoS of moving network users in a way that can scale to vehicles with large populations of terminals.

In 2001, Ernst published his thesis entitled *Network Mobility support in IPv6* [11], in which he proposes an architecture based on an MR that can manage the mobility of a large number of terminals in an efficient manner. In a nutshell, the solution that Ernst proposed allows each terminal served by an MR to possess a globally routable IP address but with a common prefix that is unique to each moving network. This allows the MR to manage mobility based on a single IP address prefix, rather than on each individual terminal IP address, thus making significant improvements to scalability. The work carried out by Ernst led to the formation of the NEMO working group[1] within the IETF [13] in 2002, which has proposed a number of

[1]Since November 2007, the NEMO working group has merged with other related working groups to form the Mobility EXTensions for IPv6 (MEXT) working group [12].

architectural extensions for moving networks to support features such as multiple connection management [14] and route optimisation [15, 16]. The working group has also standardised a number of protocols based on the MR-architecture, most notably, the NEMO Basic Support (NBS) protocol [3] which extends Ernst's early work on mobility management, and which we describe in Section 2.3.2.

In the following subsections, we describe and qualitatively evaluate some of the main architectures for moving networks that have been proposed in the literature.

2.2.1 IETF NEMO Architecture

Ernst and Lach [17] have formalised the architecture considered by the NEMO working group, which is summarised in Figure 2.1. A moving network may be made up of three main types of devices that maintain their network connection to the fixed network through the MR: Local Fixed Nodes (LFNs), Visiting Mobile Nodes (VMNs) and Local Mobile Nodes (LMNs). LFNs and LMNs are nodes that belong to the same administrative domain as the MR, but whereas an LMN can maintain its session continuity with changes to its point of access within the vehicle, an LFN cannot. VMNs on the other hand are foreign to the domain of the MR, and are able to maintain session continuity with changes to the point of access, be it between MRs, or between the MR and the fixed network, and vice versa.

The MR itself consists of an ingress interface (that between the MR and vehicular devices), and one or more egress interfaces (those between the MR and the fixed network(s)). The egress interfaces of an MR may be of heterogeneous technologies, allowing the Mobile Network Nodes (MNNs) to access networks for which they do not necessarily possess an interface, through the use of only a single short-range technology such as Bluetooth or Wireless LAN (WLAN). Since the position of the MR remains static relative to the MNNs, the MNNs will naturally rely upon the MR to manage their mobility and maintain their connection to the fixed network.

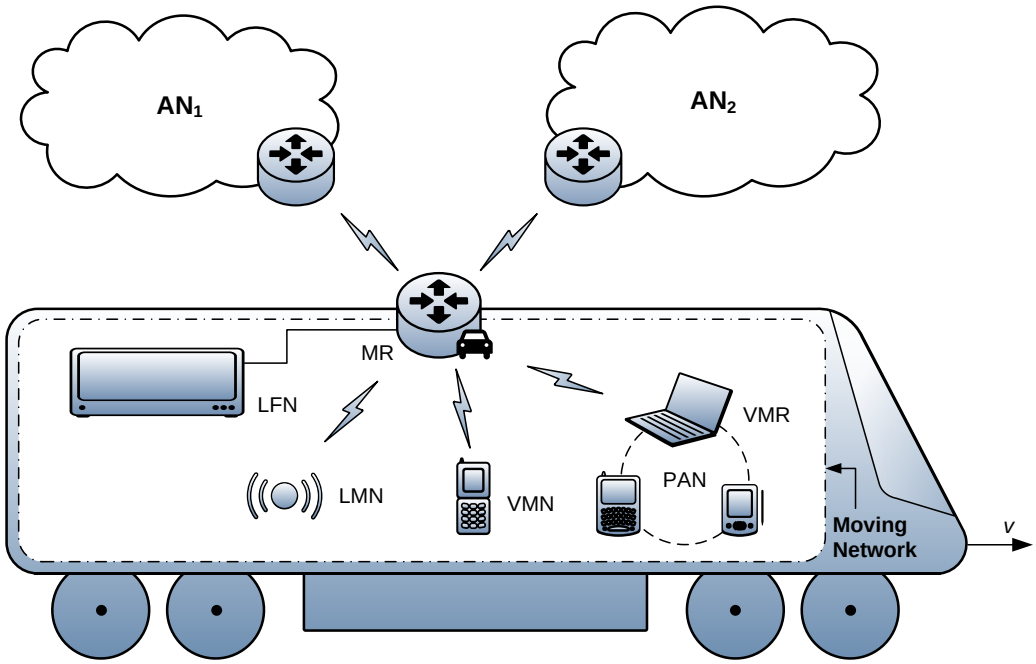


Figure 2.1: Moving network architecture set out by the IETF NEMO working group.

One of the other main advantages of the IETF NEMO architecture worth mentioning here is its hierarchical property, which can extend even below the MR, granting passengers the possibility to create their own Personal Area Network (PAN) of devices. For example, if a passenger has three devices, one of these can be designated the Visiting Mobile Router (VMR) of the other devices, thus allowing session continuity to be maintained for devices that do not support mobility, even if, for instance, the user leaves the vehicle and connects directly to the fixed network.

2.2.2 IST Ambient Networks Architecture

The IST Ambient Networks project [18] of the EU's sixth framework programme has sought to research and develop advanced control-plane mechanisms to provide autonomous cooperation and interworking of heterogeneous networks in such a way that makes the network appear homogeneous to potential users of the system. One of the underlying principles that the project has developed is that of *network composition* [19], which enables networks of arbitrary size, ranging from single nodes and PANs to entire network operators, to create associations with each other on-

the-fly. This allows for improvements to be made to the overall efficiency of network operation, particularly on the control plane.

One of the main concepts being developed in the project with respect to moving networks is that of a Routing Group (RG) [20]. When individual nodes are moving together as a group, e.g. on a PTV, it is advantageous to be able to dynamically detect this property, such that optimisations can be applied to mobility management and routing. This is useful in situations in which an MR is not present on the vehicle. With an RG, a node is delegated to act as an MR to other nodes within the PTV, assimilating the structure of a normal moving network, and its associated benefits.

2.2.3 Other Architectural Contributions

A number of other contributions have been made to the architecture of moving networks that advance those standardised by the IETF. Of most relevance to this thesis, Bonnin and Ben Rayana [21] have proposed a three-component architecture within the MR, consisting of monitoring, decision and enforcement modules. The monitoring module collects data such as vehicle speed and network coverage that can help the MR carry out its duties more efficiently. The decision module takes input from the monitoring module, as well as the different actors of the system and the flow requirements of the users, so as to be able to make decisions about the optimal connectivity characteristics of the moving network. Finally, the enforcement module translates the output of the decision module into commands and actions that are recognised by its target network. A similar approach of a monitoring- and decision-based architecture has also been taken by the IBBT Tr@ins project [22].

2.3 IP Mobility Management Support

Mobility management protocols are essential for allowing a mobile device to maintain session continuity even as mid-session changes to its point-of-attachment to the fixed network are made. However, providing mobility support to high populations

of terminals moving together as a group is inherently more challenging than for individual terminals moving independently, due primarily to the terminals' high velocity and thus significantly higher rate of handovers. This section surveys several of the key mobility management protocols that have been proposed in the literature, looking first at the protocols designed for host mobility, and then at those proposed for moving networks which build in many ways upon the principles and mechanisms of the former.

2.3.1 Host-Based Mobility Management

One of the oldest protocols for managing host mobility is the Mobile IP protocol [23] standardised by the IETF for IPv4 networks. Its successor, Mobile IPv6 (MIPv6), specified in the IETF document *RFC 3775* [24], brought support for IPv6 networks, and also made a number of enhancements to the former protocol (see, for example, [25]). The essential aim of MIPv6 is to allow a Correspondent Node (CN)—a node with which an MN is communicating—to send packets to an MN in such a way that is agnostic of its current location, i.e. of its current IP address. This is achieved by means of a redirection entity called a Home Agent (HA), located at the MN's home network.

Whenever the MN connects to a network besides its home network, it registers the IP address it is allocated by its current Access Router (AR) with its HA by means of a Binding Update (BU) message. The HA in turn creates an entry in its binding cache, associating the MN's Home Address (HoA) with its foreign "care-of" address (CoA), in essence creating a logical IPv6 tunnel. Thereafter, as illustrated in Figure 2.2, any packets destined to an MN's HoA are naturally intercepted by its HA due to the prefix of the HoA, and encapsulated at the network layer with another IPv6 header whose destination address is set to the MN's CoA. The MN will in turn take the encapsulated packets it receives, decapsulate them, and process the original IP packet in the normal manner.

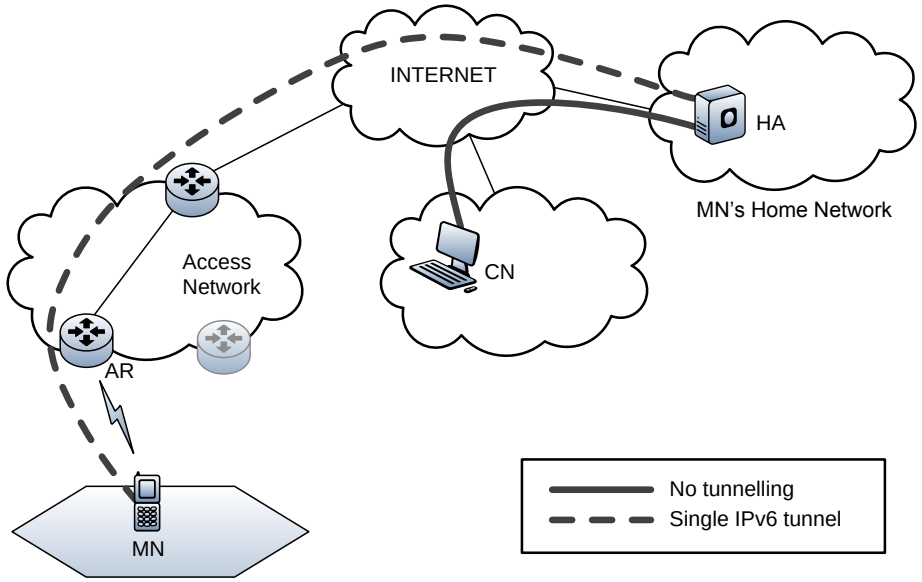


Figure 2.2: Mobile IPv6 architecture and example operation.

RFC 3775 also specifies a *route optimisation* mechanism whereby the MN may register its CoA directly with the CN, which may reduce network inefficiency and packet transmission latency occurring due to the sub-optimal routing paths introduced by the base MIPv6 protocol. Once the MN has registered its new CoA with the HA, it sends an authenticated binding cache update message directly to the CN to inform it of its CoA. The CN can thereafter tunnel packets directly to the MN.

A vast number of other mobility protocols have been proposed and studied in the literature that attempt to reduce to some degree the *handover latency* of MIPv6, that is, the time in which packets cannot be delivered to the MN during a handover operation. Amongst these protocols is the Fast Handovers for Mobile IPv6 (FMIPv6) protocol [26], which allows an MN to inform its previous AR of its new CoA, allowing it to tunnel packets to the MN until a new CoA is registered with its HA. However, the problem with FMIPv6 is that these temporary data paths can become long during a handover, which could lead to increased packet latency.

There exists an entirely different subset of mobility protocols known as *micro-mobility protocols*, which aim at reducing the handover latency of an MN when moving within an AN. Such protocols typically utilise mobility agents located in

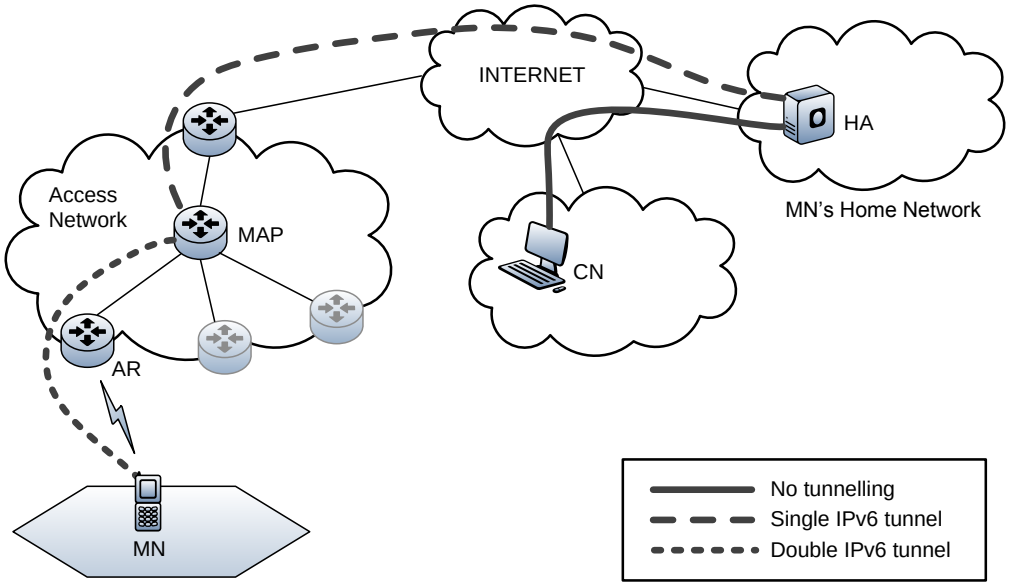


Figure 2.3: Architecture and example scenario of the Hierarchical Mobile IPv6 protocol.

the AN to track *local* location changes of an MN, reducing the frequency with which the MN need contact its HA. Hierarchical Mobile IPv6 (HMIPv6) [27] for example, which is perhaps one of the most well-known protocols due to its re-use of core principles from MIPv6, operates based on maintaining a hierarchy of nested IPv6 tunnels between the MN and CN through a Mobility Anchor Point (MAP) located in the AN.

When an MN enters a new AN, it it is able to discover the IPv6 address(es) of any MAP(s) from the information contained in the Router Advertisement (RA) it receives or solicits from the AR. Therefore, once it has configured a *local* CoA (LCoA) with the AR, it will also form a *regional* CoA (RCoA) and send a BU message to the MAP with the source address of the message set as the LCoA, and the *home address* option set to the RCoA. Once the MAP has checked that the address is not currently in use by another terminal, it will send a Binding Acknowledgement (BA) message back to the MN. The MN will in turn send a BU message to its HA to create or update the binding of its HoA with the new RCoA. Future changes to the MN's AR will therefore only require the MN to send a BU

to the MAP, having to contact its HA only when it is required to configure a new RCoA.

When all bindings are established, the operation of HMIPv6 is virtually the same as that of MIPv6, except for an extra level of IPv6 encapsulation that is necessary between the MAP(s) and MN, as shown in Figure 2.3.

2.3.2 Network-Based Mobility Management

While seamless mobility is attainable for mobile hosts through the use of MIPv6 and its variants, use of such protocols within group mobility scenarios could give rise to a number of problems and drawbacks for both users and network operators. First, the onus would be placed on each terminal to manage its own mobility. This would place an immense strain on network resources, as terminals located on a single vehicle would need to signal to perform handovers (i.e. establish new CoAs, inform HAs, etc.) at virtually the same points in time. In addition, if nodes were to connect to the MR of the vehicle (with the MR acting as solely as a form of repeater), they would no longer be able to directly receive link-layer triggers to assist in handover decisions without some form of intervention by the MR itself [28].

The NEMO working group have therefore proposed and standardised the NBS protocol [3], which addresses mobility management under the specific operating conditions experienced by moving networks. This protocol essentially extends the core mechanism of MIPv6 by introducing a home agent with which the MR itself maintains a bi-directional tunnel, as illustrated in Figure 2.4. Packets destined for an MNN are directed first to the MNN's home agent (MNNHA), which encapsulates and sends them to the MR's home agent (MRHA). The MRHA, in turn, redirects the encapsulated packets to the currently recorded location of the MR through another IP encapsulation. The MR in turn decapsulates these packets and delivers them to their final destination. Hence, by this mechanism, MNNs aboard the vehicle need register only a single, constant CoA prefixed from the MR's home network upon

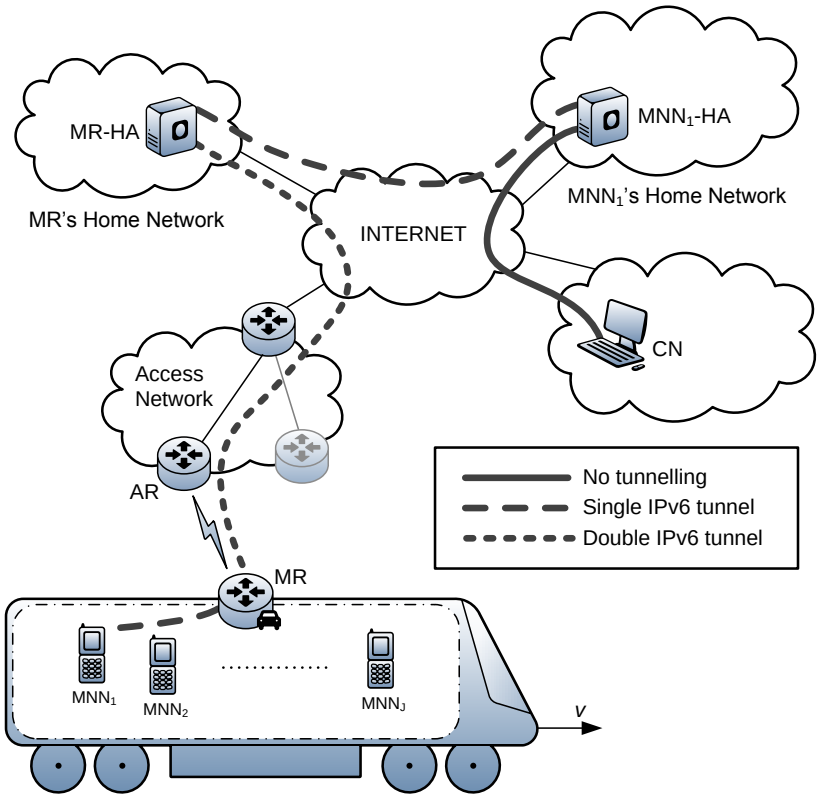


Figure 2.4: Architecture and example operation of the NEMO Basic Support protocol.

establishment of a connection to the MR. Thus, future IP handovers would require only a single *prefix-scope* binding update (PSBU) to be sent by the MR to update the location of the entire subnet of the moving network, and thereby eliminate signalling redundancy.

The handover latency of the NBS protocol is determined by the time required for a BU message to traverse the path between the MR and MRHA. If the MR is located far from its MRHA, handover latency will be large. Micro-mobility protocols would naturally be able to resolve this by limiting the depth with which handover signalling need propagate into the access network. Since the NBS protocol and HMIPv6 are both based on MIPv6, the operational feasibility of incorporating HMIPv6 within moving networks would pose no major problem, and has already been proposed by Hu *et al.* [29] with the Micro-NEMO protocol. However, one of the major problems of micro-mobility is that the mobility agents are known to be sources of bottleneck congestion (see Chapter 5), by virtue of the operation of micro-mobility protocols.

Therefore, the high aggregate resource requirements of moving networks will compound this problem further, resulting in a number of unwanted side-effects. These are elaborated further in Chapter 5, in which the problem is addressed.

2.4 QoS Architectures and Protocols

Networks have traditionally applied a best-effort approach to the transportation of packets across networks, whereby all packets are considered to have the same priority, and are thus forwarded on a First-Come, First-Served (FCFS) basis. However, as resource requirements of applications continue to grow in magnitude and diversity, it becomes increasingly necessary to use *QoS mechanisms* that can limit the use of network resources according to individual user requirements. This is to allow for maximum network utility to be achieved in the long-term whilst maintaining the satisfaction of users.

The term *QoS* can mean different things to different people. For network operators, the "QoS" of a flow[1] is defined in terms of a set of performance criteria, which can be given in terms of metrics such as minimum throughput, maximum delay and maximum jitter. For users, "QoS" represents the perceived quality of a given service, which is of course a lot more subjective to define than the former, particularly as user perception is often relative to previous experiences. For example, Armitage [30] points out that if a user were to frequently receive a level of throughput from network A that is significantly higher than the user's requirement, and then connect to a network B that meets (but does not exceed) the user's minimum throughput requirement, the user will likely have a negative opinion of the service quality of network B, even though it had met the user's requirement.

At any rate, the subject of this thesis is focussed more on QoS from the perspective of the network operator. Accordingly, the following section sets out the general

[1] In this context, the term "flow" is intended to mean the set of packets for which a certain QoS treatment is applied.

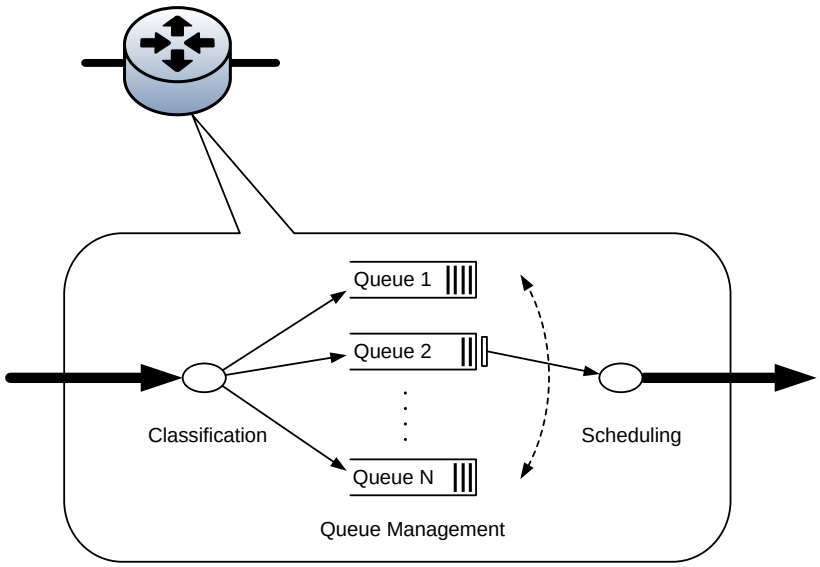


Figure 2.5: Data-plane architecture of routers with support for QoS.

components of QoS provisioning, while Section 2.4.2 defines the two broad QoS architectures defined by the IETF, Integrated Services (IntServ) and Differentiated Services (DiffServ) and their commonly-associated protocols and mechanisms, as well as the QoS approach of the IETF NSIS working group [13].

2.4.1 Components of QoS Provisioning

Traditional best-effort is based on routers forwarding the packets it receives onto the appropriate outgoing interface on an FCFS basis. However, since this approach does not differentiate between packet priorities, it is highly susceptible to the effects of transient congestion, which in turn makes it difficult for the network to provide QoS guarantees to individual flows. Therefore, in order for a network to be able to provide users with QoS support, packets at each router must be sorted into queues and scheduled for transmission in a manner that attempts to satisfy the service requirements of all users. Figure 2.5 shows the general data-plane architecture of a router with QoS support, which consists of three main functionalities: packet classification, queue management and packet scheduling. Further functionalities may be required on the control-plane, such as admission control, and configuration

of router queues. These mechanisms are elaborated on in the next section, in the context of specific QoS architectures.

2.4.2 QoS Approaches

A plethora of QoS forwarding protocols for fixed networks have been proposed in the literature, all of which are based on one or a combination of the two well-established IETF architectures, Integrated Services (IntServ) [7] and Differentiated Services (DiffServ) [31].

2.4.2.1 Integrated Services and RSVP

IntServ [7] is a QoS architecture that applies QoS treatment on a per-flow level, such that every flow that is agreed a particular QoS level along a certain path through the network has its own packet queue installed at each intermediate router. Accordingly, every router that supports IntServ QoS has both a traffic forwarding part, which classifies and schedules packets of each flow according to its registered state, and a background process part to control aspects such as the admission of flows and the configuration of queues and states.

The IntServ model supports two main types of QoS-enabled traffic forwarding: Guaranteed Service (GS) [32] and Controlled Load (CL) [33]. GS provides flows with firm bounds on end-to-end packet latency which are calculated by the routers based on the expected traffic profile (rate and burst size) of the receiver's application. If an IntServ flow violates these specifications, the excess traffic is treated as Best Effort (BE). Due to its ability to guarantee end-to-end latency, GS is well-suited for supporting real-time applications such as Voice-over-IP (VoIP).

Contrary to GS, CL does not provide flows with a guarantee on the maximum packet latency, but instead aims to deliver packets in a manner that emulates BE when the network is unloaded (uncongested). This allows for bandwidth to be shared amongst CL flows without the high degree of mutual interference that can occur under normal

BE. CL is thus well-suited for supporting applications such as file transfer, which can tolerate a certain degree of delay.

In order to install flow-specific states at routers along a flow's path, a signalling protocol is required. One of the first of such protocols proposed for IP is the ReSerVation Protocol (RSVP) [34, 35], which is defined and standardised by the IETF. This protocol requires *receivers* to initiate the reservation as opposed to senders, as this allows for more scalable operation for large multicast receiver groups. Prior to reservation of resources, the sender transmits a PATH message towards the receiver which contains two main message objects: a SENDER_TSPEC and an ADSPEC. The SENDER_TSPEC carries information pertaining to traffic characteristics, and the ADSPEC is used to convey information to the receiver as to what characteristics can be supported. Based on these messages, the receiver will construct an RESV message containing a FLOWSPEC object describing the desired QoS service to be applied to the traffic of the sender. The RESV message is used to *reserve* states in each RSVP-aware router towards the sender.

One of the main disadvantages of the IntServ model is its poor scalability that results from the need to perform hop-by-hop admission control, and maintain reservation states for every flow at each individual router. Another disadvantage, which is associated more with protocol operation during handovers, stems from use of the session identifier triplet, [DestAddress, DestPort, ProtocolID], that is used by RSVP to associate a flow to its reservation state. When an MN receives a new CoA as a result of a change of AR, the DestAddress is no longer valid, and resource reservations must be re-established end-to-end. As a result of such disadvantages, RSVP is particularly unsuited to moving networks, as a large number of reservation states would need to be re-created upon each IP handover. Subsequent contributions have extended RSVP with better support for seamless mobility of hosts. These have tended to be based on one or a combination of the following techniques:

- **Proactive Reservation Approach**

  The proactive reservation approach reserves resources in one or more neighbouring cells before a handover takes place, such that the MN can immediately be provided with the required service quality after a handover. One of the earliest forms of this protocol is Mobile RSVP (MRSVP) proposed by Talukdar *et al.* [36]. Various enhancements to this protocol that reduce resource wastage due to duplicate reservations have since been proposed; these are summarised in [37].

- **Address Transparent Approach**

  The address transparent approach reserves resources based on a mobility-independent flow identifier rather than on the changeable CoA of the MN. This ensures that even after a handover is performed, resource reservations are still valid, and only resources on the changed part of the path need be reserved. This eliminates the need for duplicate resource reservation on the entire end-to-end reservation path.

To address the mobility and scalability shortcomings of RSVP within moving networks, Malik *et al.* have proposed On-Board RSVP [38] as an extension to the original RSVP, that allows for a significant reduction in the number of reservation states in the network. It operates on the principle of aggregation, whereby PATH messages that arrive at the MRHA within a certain window of time are compressed by the MRHA into an OBPATH message. This message is identical in structure to a PATH message, save the addition of the ACC_Sender object, which carries information about each of the receivers' IP addresses and the flow specifications of their respective senders; the SENDER_TSPEC is modified by the MRHA to convey the aggregate traffic specifications of individual flows. The OBPATH is handled in the same way as in the normal RSVP, and when it arrives at the MR, it will be decompressed, and individual PATH messages reconstructed from the ACC_Sender object and delivered to each of the MNNs to which the PATH messages were originally destined. The same technique of compression and decompression of an OBRESV

message at the MR and MRHA, respectively, is also performed on the reverse path towards the sender. The authors of On-Board RSVP also specify techniques for proactive resource reservations based on the MRSVP protocol.

2.4.2.2 Differentiated Services

DiffServ [31] arose from the need for a simpler and more scalable way of providing QoS support to flows than IntServ. Rather than maintain queues for individual flows, DiffServ limits the number of queues based on a set number of *classes*. Transmitted packets are thus marked at edge routers with a DiffServ Code Point (DSCP) in the packet's IP header, which intermediate routers use to determine the packet's classification (and hence priority). The control of packet treatment (known as *per-hop behaviours* or PHBs) at each DiffServ-enabled router is achieved through the characteristics of the deployed queueing, queue management and scheduling schemes. However, the number of PHBs is open to custom implementation, and may vary from one region of a network to another. To maintain the consistency of QoS treatment across *DiffServ regions* (or "DS domains"), routers at the edges of a DS domain must be able to map the DSCPs of packets of its neighbouring domain.

The IETF have published standards for two main types of PHBs: Expedited Forwarding (EF) [39] and Assured Forwarding (AF) [40]. Under EF, packets are always serviced at or above a particular rate, regardless of the amount of non-EF traffic waiting to be serviced. This makes EF well-suited to applications with low loss, low latency and/or low jitter requirements. To ensure that this service can be maintained, traffic shaping must be performed on entry to a DS domain, such that EF queues are ensured to be small or virtually empty on average.

AF [40] defines four classes of service, which is controlled through the assignment of a specific *service class* and *drop precedence* to each queue. The service class of a queue influences the priority with which packets in that queue are scheduled, and the drop precedence influences the probability with which packets are dropped from a queue according to the queue's size. Through appropriate control of these

parameters, network resources can be flexibly and dynamically shared amongst the different flows. However, unlike EF, AF cannot guarantee low delay, loss or jitter.

In contrast to IntServ, DiffServ provides a more simplified approach to QoS provisioning by reducing queueing stage complexity and state management required in individual routers. Complexities such as admission control are pushed to the edge of a network, allowing access routers and core routers to dedicate more of their resources to traffic forwarding. However, DiffServ cannot provide the same hard and fine-grained QoS guarantees to flows as IntServ. In addition, the use of packet dropping mechanisms can be seen to be wasteful of resources, as packets that have travelled some distance into a network and are subsequently dropped would likely need to be retransmitted, leading to temporal increases to network congestion.

2.4.2.3 Next Steps in Signalling

The NSIS working group [41] was formed within the IETF with the goal of designing a more generic approach to QoS signalling (and other network applications) that allows for greater operational flexibility, and improved interaction between other network functionalities. The working group have defined a general framework [42] based on a two-layer approach which separates the signalling application logic from the actual transport of signalling messages. The lower NSIS Transport Layer Protocol (NTLP) layer is responsible for the transport of signalling messages between NSIS-aware routers (known as *NSIS Entities* or NEs) of a network as shown in Figure 2.6, and relies on the General Internet Signaling Transport (GIST) pro-

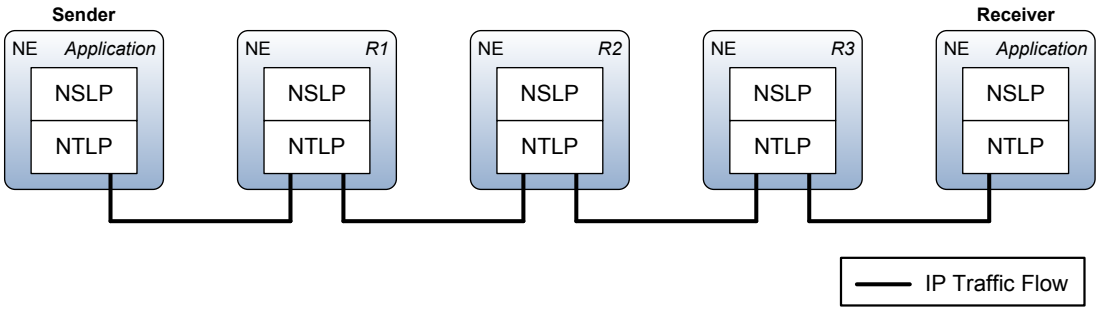


Figure 2.6: NSIS protocol stack and example traffic flow across NSIS entities.

tocol [43] to fulfil this. The upper NSIS Signalling Layer Protocol (NSLP) layer conveys information pertaining to a specific signalling application such as one related to QoS provisioning. For example, the QoS NSLP aims to provide a flexible protocol for the establishment and maintenance of QoS forwarding states in a network, independent of any particular QoS architecture. The main difference in scope of these two layers is that the NTLP operates only between adjacent NEs, where as NSLP operates on a larger scope such as end-to-end and end-to-edge.

The decoupling of the NTLP from the functionalities of the NSLP allows for significant flexibility in application design and operation. For example, whereas RSVP is designed for only end-to-end resource reservations where the "ends" are the originator and receivers of a flow, the QoS NSLP allows for signalling to be initiated or terminated at any NE. In addition, resource reservations can be either sender- or receiver-initiated, and uni- or bi-directional, with the possibility to define aggregate "tunnel" reservations and various other reservation models [44]. Such features have been exploited by Tlais *et al.* [4, 5] in the proposal of the NSIS-based NEMOR protocol, which is designed to provide flexible and scalable QoS support specifically for moving networks. In this protocol, resource reservation is, like On-Board RSVP, broken down into two distinct phases or "legs", as illustrated in Figure 2.7. The first phase involves setting up a single reservation state in each intermediate router between the MR and MRHA with sufficient resources for the entire moving network. The second phase is concerned with setting up reservation states between the MRHA and the different CNs according to the resource requirements of the individual sessions.

Upon successful construction of the QoS tunnels, flows arriving at the ingress of the MR are aggregated and marked with appropriate priorities before being transmitted through the egress interface. In each NE along the tunnel, the aggregated flows are scheduled according to their priority, as is performed in the DiffServ protocol.

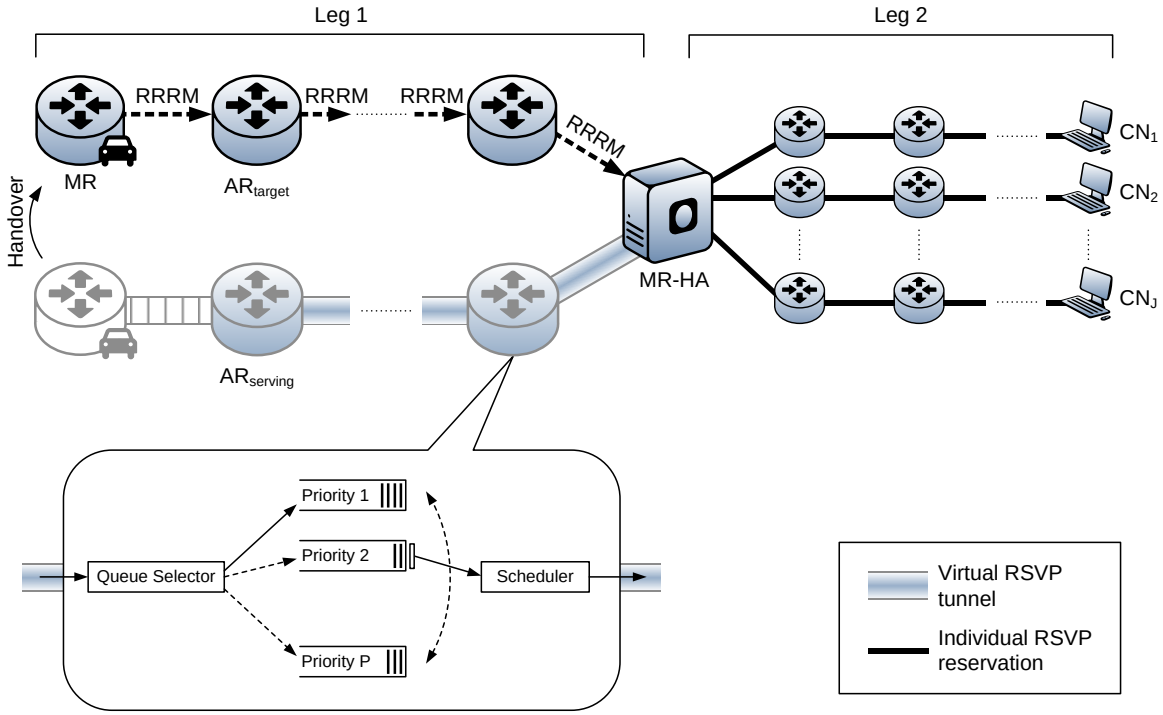


Figure 2.7: Scenario operation of the NEMOR protocol.

If a new or existing MNN initiates a new session, then only the path between the MRHA and the CN needs to be set-up. If there are insufficient resources available on the virtual QoS tunnel to support the new session, the MR can renegotiate to extend the resources allocated to it. When the MR performs a handover to a new AR, only the virtual tunnel (between the MR and MRHA) need be re-established.

One of the main advantages of NEMOR compared with On-Board RSVP is that less signaling needs to be exchanged for each individual user during handover, thus reducing the cost of protocol operation. In addition, with the novel use of DiffServ *within* a virtual QoS tunnel, the number of users contained within a DiffServ aggregation can be tightly controlled to allow for optimal performance and cost. On the other hand, NEMOR and On-Board RSVP are both dependent upon individual applications explicitly signalling their resource requirements which the MR can intercept and process. Therefore, for applications that use more transparent forms of QoS such as DiffServ, the MR would have no way of determining the application's

QoS requirements without additional application-layer functionality being added to the MR, such as is proposed by Rayana and Bonnin in [45].

2.5 Discussion and Summary

This chapter has presented a background study on the research efforts pertaining to moving networks including the underlying work relating to host mobility upon which much of the moving-network-specific research is based. One of the evident characteristics of moving network protocols such as the NBS and NEMOR protocols is their ability to scale well to large populations of terminals that may potentially be present on a moving network. For example, the NEMOR protocol works by maintaining a single virtual QoS tunnel for the entire moving network, wherein only a limited number of queues are maintained that aim at fairly distributing network resources amongst terminals. Similarly, the NBS protocol minimises the control signalling that needs to be transferred between the MR and MRHA upon handover by updating the location of a moving network based on an IP address prefix that is common to all MNNs. However, such protocols work efficiently only in their own right, with little or no cooperation between them. Therefore on the one hand, handovers are performed without knowledge of the extent to which an AN can fulfil the requirements of a moving network. On the other hand, QoS signalling is performed without awareness of the mobility of passengers with respect to the PTV. Both of these problems can lead to situations in which QoS provisioning efficiency can be severely compromised, requiring mechanisms that can minimise such inefficiencies.

As discussed in Chapter 1, QoS provisioning inefficiencies can be classified into *pre-session* and *in-session* support. To address the former issue, the scalability of the NEMOR protocol is leveraged through the use of QoS aggregation policies that operate between the MR and MRHA. QoS aggregation attempts to minimise the frequency with which the resources allocated to the virtual QoS tunnel are altered.

On the other hand, in-session QoS support is addressed through the proposal of a combined QoS and micro-mobility mechanism that is able to efficiently transfer the QoS requirements of the moving network upon handover to another network.

Chapter 3

QoS Aggregation Policies

3.1 Introduction

One of the differentiating characteristics of PTV-based moving networks from other types of networks is the high frequency and density with which sessions are created. This can be attributed to two main factors. First, there is likely to be a strong correlation between the rate at which sessions are requested and the number of passengers that board a PTV. Second, since passengers are usually idle during a journey, the presence of Internet connectivity on a train is likely to tempt passengers to use it to keep themselves occupied [46]. As an example, if the average session length is two minutes, and two hundred passengers are accessing Internet services with their handheld terminal or laptop computer, then this will result in approximately five resource reservations every three seconds, assuming, of course, that passengers make use of Internet connectivity throughout their journey.

QoS aggregation is a technique that can help reduce the number of resource reservations made from a moving network. It works by introducing a time lag between a request for resources by a user and the actual reservation process. This allows for multiple queued requests at an MR to be combined into a single request, thus reducing the frequency with which QoS requests are sent to an AN (assuming a

NEMOR-like protocol is used). Since the procedure involved in processing each resource reservation request message (message processing, admission control, updating of reservation states, etc.) is resource intensive, QoS aggregation can significantly reduce the average load on resources in the fixed network. However, given a business model based on metered data consumption, delaying the allocation of resources to session requests can reduce long-term operator revenue. Additionally, users must incur a delay before being able to commence their sessions, creating an inconvenience to the user, and potentially denting their overall satisfaction with the PTV Internet service, and possibly even with the PTV operator itself. Hence, the goal is to ensure that the time for which resource requests are held at an MR is adjusted such that both operator revenue loss and network operational inefficiency are minimised without significantly increasing the duration a passenger must wait before being connected.

Therefore, this chapter analyses a number of QoS aggregation policies proposed in previous work, and proposes a novel cost-driven aggregation policy that is shown to operate more efficiently than other policies. The next section provides an overview of the general mechanisms of existing policies, which is followed by a more formal definition of QoS aggregation in Section 3.3. Section 3.4 then applies this definition in deriving expressions for average cost rate and user waiting time of existing policies. After proving the dependency of these policies on the rate of QoS requests being made by users, a novel cost-driven policy [47–49] is proposed in Section 3.5 that is mathematically proven to maintain efficiency under varying QoS request arrival rates. Section 3.6 then sets out the framework used to evaluate the performance of both existing policies (herein referred to as *parameter-driven* policies), and the cost-driven policy. The results of the performance evaluation are presented in Section 3.7, and finally the chapter is concluded in Section 3.8.

3.2 Overview of QoS Aggregation Policies

Three QoS aggregation policies have previously been proposed by Malik *et al.* [50] that to attempt to reduce the rate at which control signalling is sent across the wireless link between the MR and the AN. These are the temporal operating policy (T-policy), the cardinal operating policy (K-policy) and the resource-threshold operating policy (R-policy).

The T-policy waits for a time T to elapse before launching or modifying an aggregated QoS reservation for users queueing to establish a session. The policy module in the MR will begin counting the cycle time only upon receipt of the first request into the queue. The main advantage of the T-policy lies in its ability to place an upper bound on the waiting time of users. On the other hand, there is little control over the amount of resources reserved and the number of requests per aggregated QoS message. This can cause QoS requests to exceed the available resources in the network, and can also make the minimisation of operator costs[1] challenging.

The K-policy operates by switching off the QoS aggregation server until K QoS messages have been received into the queue. It is therefore required to keep a running count of the messages waiting in the queue as they arrive; however, there is no requirement for the contents of the packets themselves to be analysed for the operation of the policy. Unlike the T-policy, the K-policy cannot provide any firm guarantee on the maximum user waiting time. On the other hand, the K-policy provides a benefit to the operator over the T-policy, in that the costs can be more readily controlled through appropriate selection of the value of K.

The R-policy is similar to the K-policy, in that it is triggered by a characteristic of the QoS requests; however, in this case, it is triggered not merely by the number of requests awaiting service, but by the total amount of resources R requested during a

[1]Cost here refers to a subjective measure of protocol efficiency and is strongly related to operator revenue; a full discussion of the physical meaning of cost is given in Section 3.3.2.

dormant period[1]. Although, again, this policy does not place any upper limit on the user waiting time, a number of operator advantages are gained, such as control over the amount of resources being requested by an aggregated QoS reservation. This is particularly useful when the MR must reserve resources across a low-capacity or congested network and can be used to avoid the situation of a request by the MR to extend the virtual QoS tunnel being rejected by the network due to insufficient availability of resources.

One of the main problems of the T-, K-, and R-policies is that the optimal policy parameter is dependent on the rate at which requests are made (the mathematical proof of this is given in Section 3.4, but for now, this fact is taken for granted). Therefore, if the rate at which requests are made varies with time, due, for example, to the bursty nature of passengers boarding and alighting a vehicle, the optimum policy parameter will also vary with time, in turn making optimisation of operational efficiency difficult.

3.3 General QoS Aggregation Policy Framework

QoS aggregation policies can be characterised by a framework consisting of a queue model describing the typical buffering and servicing behaviour of users' QoS requests at the MR, and a cost function that provides a way of measuring the efficiency of QoS provisioning under moving networks. Both of these aspects are described in turn in the following subsections.

3.3.1 Queue Model

Generally, a QoS aggregation policy can be viewed as the buffering and aggregation of QoS request messages at an MR, which can be modelled as a $G/G^{[N_c]}/1$ vacation queue (see, for example, [51]), as shown in Figure 3.1. As implied by the notation,

[1]The *dormant period*, which is sometimes known as the "vacation period," is that in which requests are being buffered only at the MR and not being served. Conversely, the *service period*, to which reference is made later in this chapter, is the period in which buffered requests are actually being processed for aggregation and transmission.

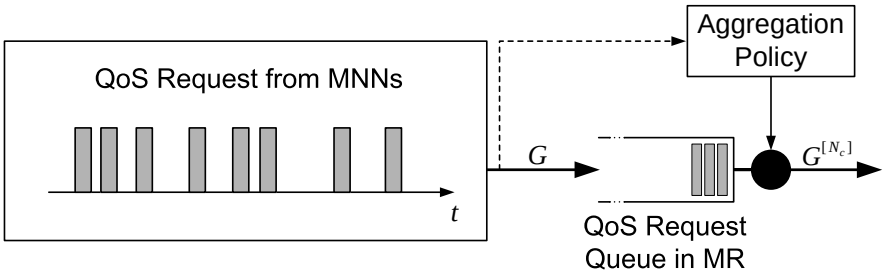


Figure 3.1: $G/G^{[N_c]}/1$ QoS request queue within an MR.

and contrary to the approach of Malik *et al.* in [52], we assume an infinite queue capacity at the MR. QoS requests from MNNs arrive at the MR according to some general distribution and are queued until the threshold of the deployed QoS aggregation policy is reached, which represents the end of the current dormant period. At this point, all queued requests are combined into a single request for resources, sent across the wireless link to the fixed network, and propagated further in accordance with the deployed QoS protocol. This represents the end of the service period (the combination of a dormant period and a service period form a *cycle*). Queued requests are thus served in batches, where N_c is a random variable describing the number of packets in each serviced batch. To this end, the exact queueing discipline can be said to be irrelevant, since all buffered requests are effectively served at the same time. Requests that arrive in the queue *during* a service period must wait until the end of the next dormant period before being served.

Figure 3.2 shows a generic *expected* arrival pattern of QoS requests at the MR with time under exponentially distributed interarrival times with parameter $1/\lambda$. At the end of an aggregation cycle c, the number of queued requests is N_c, and the cycle duration is $\theta(\phi) + \delta\theta(\phi)$ where potentially $0 \leq \delta\theta(\phi) < 1/\lambda$, depending on the deployed aggregation policy.

Within the period $\theta(\phi)$, the total expected waiting time of all queued requests can be expressed logically as

$$\sum_{i=1}^{E[N_c]} \omega_i(\phi, \lambda) = \frac{(E[N_c]-1)^2 + (E[N_c]-1)}{2} \frac{1}{\lambda} \tag{3.1}$$

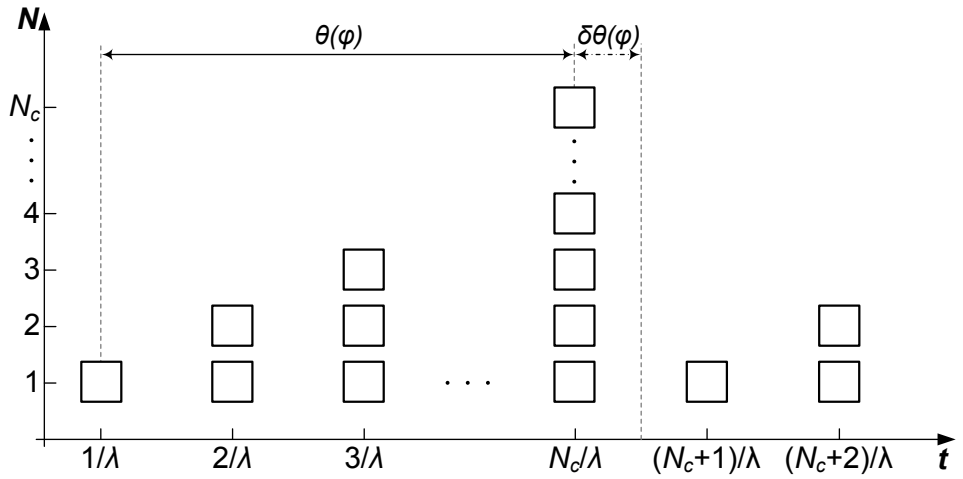


Figure 3.2: Generic expected arrival pattern of QoS requests at the MR.

where ω_i is a random variable describing the time each request has had to wait before being granted access to the network resources that it requires, and where from Little's theorem [53], $E[N_c]$ is given by

$$E[N_c] = \lambda\theta(\phi) + 1. \tag{3.2}$$

3.3.2 Cost Function

Based on the costs pertaining to M/G/1 queues with vacations identified by Heyman [54], two costs are of particular relevance to QoS aggregation: the holding cost and the sending cost. The holding cost, denoted C_h, can be considered as the revenue that the operator could have generated per unit resource had access to network resources been granted to the customer immediately upon request, and can therefore be regarded as a function of the flow specifications of the request (i.e. throughput, delay, etc). On the other hand, the signalling cost C_s is that resulting from the use of network resources such as processing overhead and bandwidth usage in the setup of an aggregated QoS request. Both of these costs are set by the operator and communicated to the MR upon first connection to the network using some form of network management protocol or otherwise, an aspect that remains outside the scope of this research. Based on these definitions of cost, we propose that the cost

rate of any cycle c be represented by

$$C_c(\phi, \lambda) = \frac{C_s + \sum_{i=1}^{N_c} (\omega_i(\phi, \lambda) \mathcal{R}_i C_h)}{T_c(\phi)}, \tag{3.3}$$

where N_c is the number of requests received at the end of aggregation cycle c, $T_c(\phi)$ is the duration of cycle c, $\mathcal{R}_i$ and $\omega_i(\phi, \lambda)$ are the resource requirements (e.g. throughput) and the waiting time as a function of the policy parameter value ϕ, respectively, of the ith QoS request to arrive during cycle c. The term $\sum_{i=1}^{N_c} (\omega_i(\phi, \lambda) \mathcal{R}_i C_h)$ represents the total cost of holding of all requests at the end of an aggregation cycle. Based on this equation, the *expected* cost rate of any aggregation cycle, c, can be expressed as

$$E[C_c(\phi, \lambda)] = \frac{C_s + \sum_{i=1}^{E[N_c]} (E[\omega_i(\phi, \lambda)] E[\mathcal{R}_i] C_h)}{E[T_c(\phi)]}, \tag{3.4}$$

For simplicity, C_s is assumed to be independent of the number of requests aggregated. This assumption is valid if a per-class mechanism is used between the MRHA and CNs, as then, the aggregate QoS messages would not need to contain additional specifications of individual flows. Thus, the problem can be formulated as the optimisation problem

$$\begin{aligned} &\text{minimise} \quad C_c(\phi, \lambda) \\ &\text{subject to} \quad \omega_x(\phi, \lambda) \leq \tau, \quad i \in \mathbb{Z} : 1 \leq i \leq N_c, \end{aligned} \tag{3.5}$$

where τ is the maximum allowable waiting time for any QoS request to be granted resources, which is defined by the moving network operator, and can be set to such a value according to the maximum acceptable user-reneging probability.

Due to the difficulty in absolutely quantifying C_s, the subsequent study examines the cost-effectiveness of QoS aggregation under a sample of values, which is represented as the ratio (tariff) between the sending cost, and the expected holding cost of any user, x, i.e. $C_s : E[\mathcal{R}_x] E[\omega_x] C_h$. In any case, we assume that $C_s > E[\mathcal{R}_x] E[\omega_x] C_h$,

which essentially implies that QoS aggregation is economically justified, as otherwise it would always be more cost efficient to admit a session than to cause it to wait for other requests to arrive.

3.4 Analysis of Parameter-Driven Policies

In this section, we derive the expected cost rate and user waiting time for the C-policy, and for each of the parameter-driven policies, under Poisson-distributed QoS request arrival rates. This is expanded in Section 3.6 for the case in which requests are bursty, allowing for an analytical performance comparison to be made between all policies under the operating conditions commonly encountered on most modes of PTVs.

3.4.1 T-Policy

The T-policy waits for a time, T, to elapse before launching or modifying an aggregated QoS reservation. Therefore, under this policy the expected cycle duration is

$$E[T_c(T)] = \theta(T) + \delta\theta(T) = T, \tag{3.6}$$

where $\delta\theta(T)$ is half of the expected request interarrival time of all queued requests at the end of an aggregation cycle, i.e.

$$\delta\theta(T) = \frac{1}{2\lambda}. \tag{3.7}$$

From Equations 3.6 and 3.7, $\theta(T)$ is given by

$$\theta(T) = \frac{2\lambda T - 1}{2\lambda}. \tag{3.8}$$

Therefore, from Equations 3.1 and 3.7, the summation of expected waiting times of all QoS requests arriving within an aggregation cycle can be expressed as

$$\sum_{i=1}^{E[N_c]} E[\omega_i(T,\lambda)] = \sum_{i=1}^{E[N_c]} E[\omega_i^{\theta}(T,\lambda)] + \sum_{i=1}^{E[N_c]} E[\omega_i^{\delta\theta}(T,\lambda)]$$
$$= \frac{(E[N_c]-1)^2 + (E[N_c]-1)}{2}\frac{1}{\lambda} + \frac{E[N_c]}{2\lambda}.$$

Using Little's theorem in Equation 3.2 to substitute for $E[N_c]$ we get

$$\sum_{i=1}^{E[N_c]} E[\omega_i(T,\lambda)] = \frac{\lambda}{2}\theta^2 + \theta + 1. \tag{3.9}$$

Substituting for θ, we obtain

$$\sum_{i=1}^{E[N_c]} E[\omega_i(T,\lambda)] = \frac{\lambda T^2}{2} + \frac{T}{2} + \frac{1}{8\lambda}, \tag{3.10}$$

from which we can obtain the expected waiting time of any request, x, as

$$E[\omega_x(T,\lambda)] = \frac{\lambda^2 T^2 + \lambda T + \frac{1}{4}}{2\lambda^2 T + \lambda}. \tag{3.11}$$

Substituting Equation 3.10 into Equation 3.4, we get the following expression for the expected cost rate of the T-policy

$$E[C_c(T,\lambda)] = \frac{C_s}{T} + \frac{(4\lambda^2 T^2 + 4\lambda T + 1)E[\mathcal{R}_x]C_h}{8\lambda T}. \tag{3.12}$$

3.4.2 K-Policy

The K-policy operates by switching off the QoS aggregation server until K QoS messages have been received into the queue. Since the QoS server is switched on as soon as the Kth request is received into the queue, $\delta\theta(K) = 0$, such that $T_c(K) = \theta(K)$. Hence, the summation of expected waiting times of all requests within an

aggregation cycle is given by

$$\sum_{i=1}^{K} E[\omega_i(K,\lambda)] = \frac{(K-1)^2 + (K-1)}{2}\frac{1}{\lambda}. \tag{3.13}$$

From this the expected user waiting time of any request is given by

$$E[\omega_x(K,\lambda)] = \frac{K-1}{2\lambda}. \tag{3.14}$$

From Little's theorem in Equation 3.2 the cycle time, $T_c(K)$, is given by

$$T_c(K) = \frac{K-1}{\lambda},$$

which we can substitute together with Equation 3.13 into Equation 3.4 to obtain an expression for the cost rate of the K-policy as

$$E[C_c(K,\lambda)] = \frac{\lambda C_s}{K-1} + \frac{KE[\mathcal{R}_x]C_h}{2}. \tag{3.15}$$

3.4.3 R-Policy

The formulation of the expected cost rate and user waiting time for the R-policy follows a similar derivation to that of the T- and K-policies. Upon arrival of the QoS request that brings the sum of all queued resources to R or greater, the aggregation server is immediately switched on, such that $\delta\theta(R) = 0$. Therefore, the expected cycle time $E[T_c(R)] = \theta(R)$, is given by

$$E[T_c(R)] = \frac{R - E[\mathcal{R}_x]}{\lambda E[\mathcal{R}_x]}. \tag{3.16}$$

If N_c requests are queued before the aggregation server is switched on, then $(N_c - 1)$ of these will have incurred a non-zero waiting time, such that

$$\sum_{i=1}^{E[N_c]} E[\omega_i(R,\lambda)] = \frac{(E[N_c]-1)^2 + (E[N_c]-1)}{2}\frac{1}{\lambda}. \tag{3.17}$$

From Equations 3.17 and 3.2, the expected user waiting time is given by

$$E[\omega(R,\lambda)] = \frac{R - E[\mathcal{R}_x]}{2\lambda E[\mathcal{R}_x]}. \tag{3.18}$$

Using Little's theorem, and substituting for $T_c(R)$ and $\sum_{i=1}^{E[N_c]} \omega_i(R,\lambda)$ into Equation 3.3 gives the expected total cost rate under the R-policy as

$$E[C_c(R,\lambda)] = \frac{\lambda E[\mathcal{R}_x] C_s}{R - E[\mathcal{R}_x]} + \frac{RC_h}{2}. \tag{3.19}$$

3.4.4 Cost-Optimal Policy Parameter Threshold

To find the cost-optimal policy parameter threshold ϕ^* for each of the policies, the first derivative of each of the respective expected cost-rate expressions in Equations 3.12, 3.15 and 3.19 must be obtained and analysed for any minima. Taking the T-policy as an example, the first derivative of $E[C_c(T,\lambda)]$ is given by

$$\frac{dE[C_c(T,\lambda)]}{dT} = \frac{\lambda E[\mathcal{R}_x] C_h}{2} - \frac{1}{T^2}\left(C_s + \frac{E[\mathcal{R}_x] C_h}{8\lambda}\right). \tag{3.20}$$

Setting Equation 3.20 to zero to obtain the optimal value T^*, we get

$$T^* = \sqrt{\frac{2}{E[\mathcal{R}_x] C_h \lambda}\left(\frac{E[\mathcal{R}_x] C_h + 8\lambda C_s}{8\lambda}\right)} \tag{3.21}$$

which is plotted against λ in Figure 3.3. Since T^* is dependent on λ, it is difficult to achieve cost optimality with just a single value of T. Therefore, one approach—assuming that all values of λ within a given range are equally likely to occur—would be to obtain an average value for T^*, i.e.:

$$\overline{T^*} = \frac{1}{(\lambda_{max} - \lambda_{min})} \int_{\lambda_{min}}^{\lambda_{max}} \sqrt{\frac{2}{E[\mathcal{R}_x] C_h \lambda}\left(\frac{E[\mathcal{R}_x] C_h + 8\lambda C_s}{8\lambda}\right)}\, d\lambda. \tag{3.22}$$

However, in this case, cost optimality would hold only for the mean value of λ.

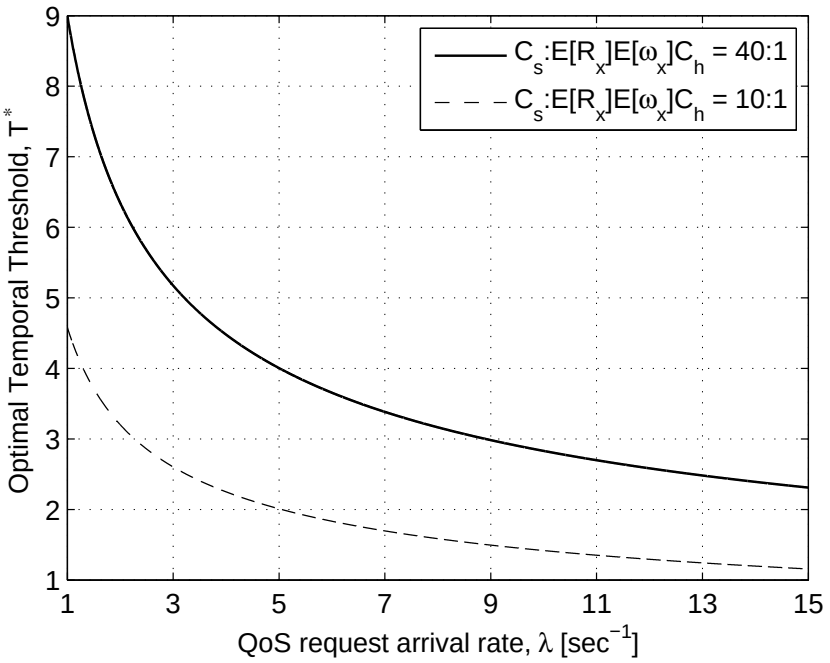


Figure 3.3: Cost-optimal temporal threshold T^* for signalling-to-holding-cost (per mean requested resource) ratios $C_s : E[\mathcal{R}_x]E[\omega_x]C_h$ of 10:1 and 40:1.

Similarly, by differentiating Equation 3.15 and setting to zero, we get for the K-policy

$$K^* = \sqrt{\frac{2\lambda C_s}{E[\mathcal{R}_x]C_h}} + 1. \tag{3.23}$$

Similarly, from Equation 3.19 for the R-policy, we get

$$R^* = \sqrt{\frac{2\lambda E[\mathcal{R}_x]C_s}{C_h}} + E[\mathcal{R}_x]. \tag{3.24}$$

For both of these expressions, we can obtain an average optimal value in a similar manner to that in Equation 3.22. Plots of these equations showing the degree of dependency of K^* and R^* on λ are given in Appendix A.

3.5 Proposal for a Cost-Driven Policy

In the previous section, it was clear that any single optimal policy parameter is optimal for only a fixed expectation of the arrival rate of QoS requests at the MR. Thus, with increasing variance of request rate, the cost efficiency of QoS aggregation

is reduced. We therefore propose a dynamic aggregation policy that is driven by the instantaneous cost of the current cycle, rather than by a particular characteristic of the received requests.

Defining an aggregation utility function as

$$\alpha = \frac{\sum_{i=1}^{N_c} (\omega_i(\alpha, \lambda)\mathcal{R}_i C_h)}{C_s} \tag{3.25}$$

the decision to launch a new QoS aggregation is determined by α reaching a certain value. For cost optimality, new QoS aggregations are to be launched when the holding cost of the current cycle is equal to the signalling cost, i.e., when $\alpha = 1$. Economically, this is equivalent to being allocatively efficient, whereby the price is equal to the marginal cost.

Referring to Figure 3.2, $\delta\theta(\alpha)$ is, like the T-policy, half of the expected request interarrival time of all queued requests at the end of an aggregation cycle, i.e.

$$\delta\theta(T) = \frac{E[N_c]}{2\lambda}, \tag{3.26}$$

such that

$$\theta(\alpha) = E[T_c(\alpha)] - \frac{E[N_c]}{2\lambda}, \tag{3.27}$$

Therefore, the sum of expected waiting times of all users can be expressed as

$$\begin{aligned}\sum_{i=1}^{E[N_c]} E[\omega_i(\alpha, \lambda)] &= \sum_{i=1}^{E[N_c]} E[\omega_i^{\theta}(\alpha, \lambda)] + \sum_{i=1}^{E[N_c]} E[\omega_i^{\delta\theta}(\alpha, \lambda)] \\ &= \frac{(E[N_c] - 1)^2 + (E[N_c] - 1)}{2} \frac{1}{\lambda} + \frac{E[N_c]}{2\lambda}.\end{aligned}$$

Using Little's theorem in Equation 3.2 to substitute for $E[N_c]$ we get

$$\sum_{i=1}^{E[N_c]} E[\omega_i(\alpha, \lambda)] = \frac{\lambda}{2}\theta^2 + \theta + 1. \tag{3.28}$$

Substituting for θ in Equation 3.27, we obtain

$$\sum_{i=1}^{E[N_c]} E[\omega_i(\alpha,\lambda)] = \frac{4\lambda^2 E[T_c]^2 + 4\lambda E[T_c] + 1}{8\lambda}. \tag{3.29}$$

Since the decision to aggregate is based on the value of $\sum_{i=1}^{N_c} (\omega_i(\alpha,\lambda)\mathcal{R}_i C_h)$ reaching a certain proportion, α, of the sending cost, C_s, we can state that

$$E[\mathcal{R}_x]C_h \left(\frac{4\lambda^2 E[T_c]^2 + E[T_c] + 1}{8\lambda} \right) = \alpha C_s,$$

from which we can solve for $E[T_c]$ using the quadratic equation to give

$$E[T_c] = \frac{\sqrt{8E[\mathcal{R}_x]C_h\lambda C_s\alpha} - k}{2E[\mathcal{R}_x]C_h\lambda}. \tag{3.30}$$

Therefore, substituting for $E[T_c]$ into Equation 3.29 and averaging over the number of QoS requests received at the end of an aggregation cycle gives the expected waiting time per user as

$$E[\omega_x(\alpha,\lambda)] = \frac{C_s\alpha}{\sqrt{2k\lambda C_s\alpha}}. \tag{3.31}$$

By virtue of the fact that the aggregation server is switched on when the total holding cost is equal to a proportion, α of the sending cost, then the expected cost rate can be represented as

$$E[C_c(\alpha,\lambda)] = \frac{Cs + \alpha Cs}{E[T_c]}.$$

Substituting for $E[T_c]$ gives the expected cost rate of the C-policy as

$$E[C_c(\alpha,\lambda)] = \frac{2k\lambda C_s(1+\alpha)}{\sqrt{8k\lambda C_s\alpha} - k}. \tag{3.32}$$

The complexity of the C-policy is comparable to that of the R-policy, as the MR is still needed to keep a record of the resources required by each request. However, the C-policy additionally needs to calculate α at regular intervals. The frequency of

calculation can be manually set by the moving network operator, but in general, as the frequency is reduced, so in turn will the cost optimality. For the purpose of this work, it is sufficient to calculate α at 0.1-second intervals—which provides a good balance between calculation accuracy and computational processing overhead—and let the requirement to launch a new QoS aggregation be such that $\alpha \geq 1$.

3.5.1 Cost-Optimal Aggregation Utility Value

To find the cost-optimal value of α for the C-policy, the first derivative of Equation 3.32 with respect to α is determined and analysed for any minima. After simplification, the first derivative of $E[C_c(\alpha, \lambda)]$ is given by

$$\frac{dE[C_c(\alpha,\lambda)]}{d\alpha} = \frac{8k^2\lambda^2C_s^2(\alpha-1) - 4k^2\lambda C_s\sqrt{2k\lambda C_s\alpha}}{(\sqrt{8k\lambda C_s\alpha}-k)^2\sqrt{8k\lambda C_s\alpha}} \tag{3.33}$$

where $k = E[\mathcal{R}_x]C_h$. Using the quadratic equation to solve for the cost-optimal value of α, denoted α^*, we get

$$\alpha^*(k) = \frac{4\lambda C_s + k + \sqrt{8k\lambda C_s + k^2}}{4\lambda C_s}. \tag{3.34}$$

Figure 3.4 shows the optimal aggregation utility α^* plotted against the arrival rate of QoS requests λ for various signalling-to-holding-cost ratios. It can be seen that $\alpha^* \simeq 1$ for $1 \leq \lambda \leq 15$, and it can be concluded that

$$\lim_{k\to\infty} \alpha^*(k) = 1.$$

3.6 Performance Evaluation Framework

To evaluate the cost efficiency of the QoS aggregation policies, we consider two different models for characterising the *number* of QoS requests from MNNs arriving at the MR according to the user population dynamics of the PTV. The first is a standard Poisson process with parameter λ (in requests per second), which is a

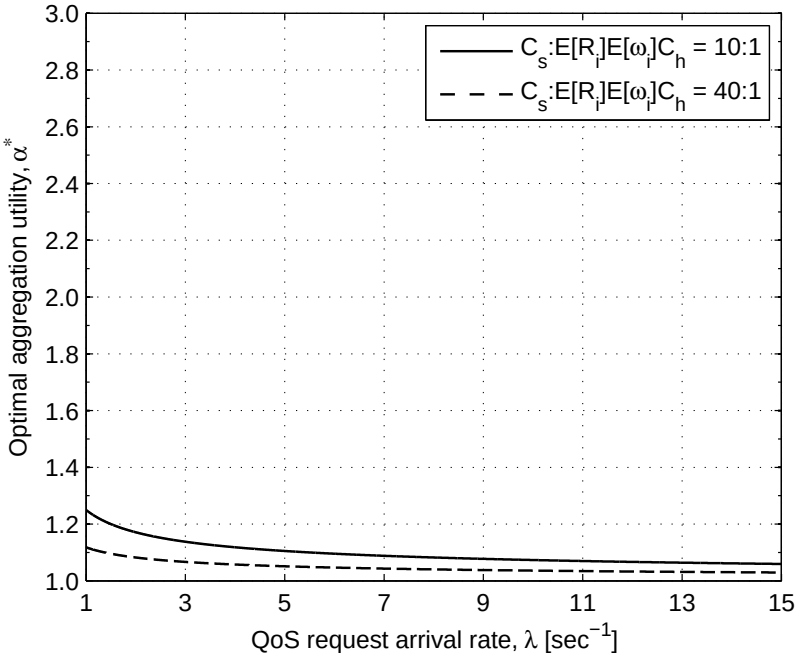


Figure 3.4: Optimal aggregation utility α^* for signalling-to-holding-cost (per mean requested resource) ratios $C_s : E[\mathcal{R}_x]E[\omega_x]C_h$ of 10:1 and 40:1.

common process used in the analysis of many queueing models due to its simplifying analytical and probabilistic properties [55]. This model is particularly effectual in the case when passengers remain on the vehicle throughout the entire journey. In this case, the queue model reduces to $M/G^{[N_c]}/1$.

The second model considered is a two-state Markov-modulated Poisson process (MMPP) [56], through which one is able to simply yet realistically capture the bursty nature of the QoS requests arising as a result of the batch arrivals of passengers on the vehicle at each station stop, which will naturally tend to correlate with the Internet activity of passengers. A state diagram of this model is shown in Figure 3.5. State A represents that of low-traffic or *ambient* requests, with a mean QoS request arrival rate, λ_A requests/s. State B represents the high-traffic or *bursty* state, with a mean QoS request arrival rate, λ_B requests/s.

The sojourn times of states A and B of the MMPP are exponentially distributed with mean $r_A = 50$ seconds and $r_B = 10$ seconds, respectively. These sojourn times were chosen to illustrate the ability of the C-policy to operate cost-efficiently under

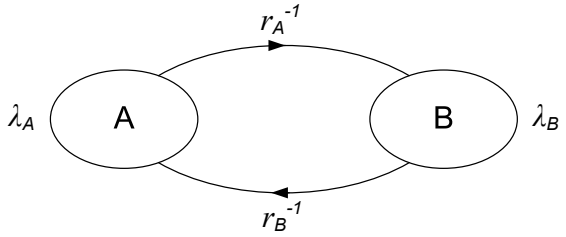


Figure 3.5: Two-state MMPP used to model bursty QoS requests.

varying rates of request. In reality these times would be in the order of minutes, but setting it to that order in this study would have no significant effect on the accuracy and validity of the results. Under this model, the queueing system can be described as $MMPP/G^{[N_c]}/1$.

The probability of each state is defined as

$$\Pr(S) = \frac{r_S}{r_A + r_B}, \quad S = \{A, B\}$$

and the mean arrival rate $\overline{\lambda}$ of the model as

$$\overline{\lambda} = \sum_S \Pr(S)\lambda_S.$$

Under the MMPP model shown in Figure 3.5, the expected cost rate is the sum of the cycle costs in each state proportioned by the respective state probability. Therefore, the expected cost rate of the C-policy under bursty requests is given by

$$E[C_c(\alpha, \lambda_A, \lambda_B)] = \sum_S \Pr(S)E[C_c(\alpha, \lambda_S)] \tag{3.35}$$

and the expected user waiting time is given by

$$E[\omega_x(\alpha, \lambda_A, \lambda_B)] = \sum_S \Pr(S)E[\omega_x(\alpha, \lambda_S)]. \tag{3.36}$$

In the case of the parameter-driven policies, the expected cost rates and user waiting times are obtained by replacing λ with $\overline{\lambda}$ in each of their respective equations in Section 3.4.

Under the Poisson model, λ was varied between 1 and 15 requests/s, and under the MMPP model, λ_B was varied between 1 and 15 requests/s while λ_A was fixed at 1 request/s. The distribution of requested throughput across QoS requests was assumed to be exponential with mean $E[\mathcal{R}_\S] = 64$ kB/s. Results were generated for two different cost tariff ratios $C_s : E[\mathcal{R}_\S]E[\omega_x]C_h$ of 10:1 and 40:1, which are the same as those used by Malik *et al.* [50]. For each permutation of parameters, 25,000 aggregation cycles were simulated, which was that maximum number possible that avoided out-of-memory errors from occurring during the the simulation. However, in order to ensure a degree of fairness between simulations of the various policies, the input QoS requests were pre-generated, such that for a given arrival rate λ, each policy was subject to identical arrival patterns. These pre-generated requests were validated separately by plotting the probability distribution and observing the shape of the curve.

The simulations of the QoS aggregation policies were carried out under MATLAB based on discrete events. Under the parameter-driven policies, these events consisted primarily of the arrival of a new QoS request at the MR and, under the T-policy, the time parameter being reached. Under the C-policy, the discrete events consisted of the arrival of new QoS requests and a regular 0.1-second calculation of the cost-utility, α.

3.7 Performance Evaluation

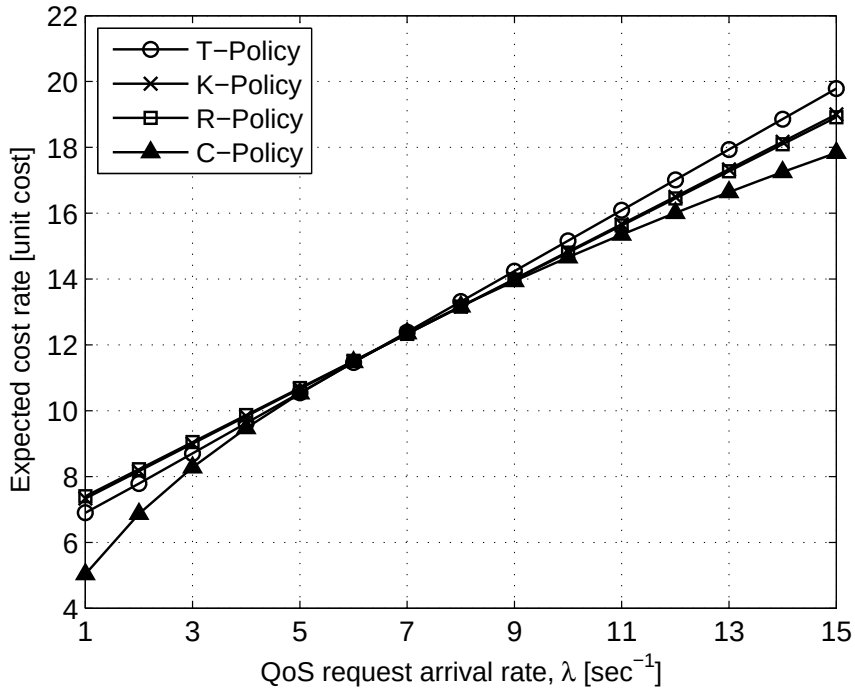
This section presents the results of a performance evaluation of the QoS aggregation policies set out in the prior sections. Focus is placed on the results of simulations carried out in MATLAB [57], but the results of the analysis are also presented to support the correctness of the simulations.

3.7.1 Operator Cost

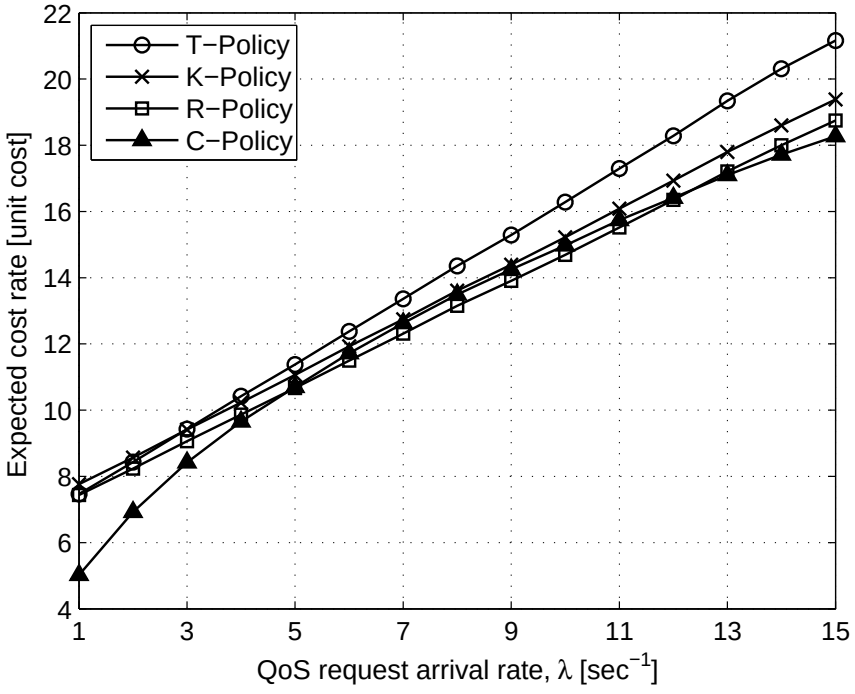
Figure 3.6(a) and (b) shows the variation of the expected cost per second versus the QoS request arrival rate for each of the QoS aggregation policies under Poisson (smooth) requests and from both the analytical and simulation models, respectively, for a signalling-to-holding-cost ratio $C_s : E[\mathcal{R}_x]E[\omega_x]C_h$ of 10:1. Each set of results uses the optimal policy parameter thresholds derived from our analytical models. It can be observed in Figure 3.6(a) that the cost rate of our proposed C-policy is either less than or equal to that of other parameter-driven policies for all request rates considered. This cost reduction is equal to 5.6% on average, in the range $1 \leq \lambda \leq 15$, which demonstrates the ability of the C-policy to reduce costs by only a single aggregation threshold of $\alpha = 1$ that is independent of the QoS request arrival rate λ.

The points at which the cost of the C-policy are equal to the parameter-driven policies—typically in the range, $5 < \lambda < 11$—are due specifically to the fact that the thresholds of the parameter-driven policies used in generating the results were cost optimal for the mean value of λ that could occur. With increasing deviation from this range of λ, the cost rate of each of the parameter-driven policies diverges from that of the cost-optimal C-policy with differing degrees according to the magnitude of the signalling-to-holding-cost ratio, $C_s : E[\mathcal{R}_x]E[\omega_x]C_h$.

The simulation-generated results are comparable to the analytical results, with the only notable difference being that the cost rate of the C-policy within the range $5 < \lambda < 11$ is marginally greater than that of the most efficient parameter-driven policy. This is due to the granular cost calculation interval of the C-policy (set to 0.1 seconds), which causes the actual aggregation utility α to exceed unity. Thus, in this case, the minimum cost saving is slightly lower at 3.3% relative to the R-policy but as much as 6.6% and 10.9% relative to the K- and T-policies, respectively.

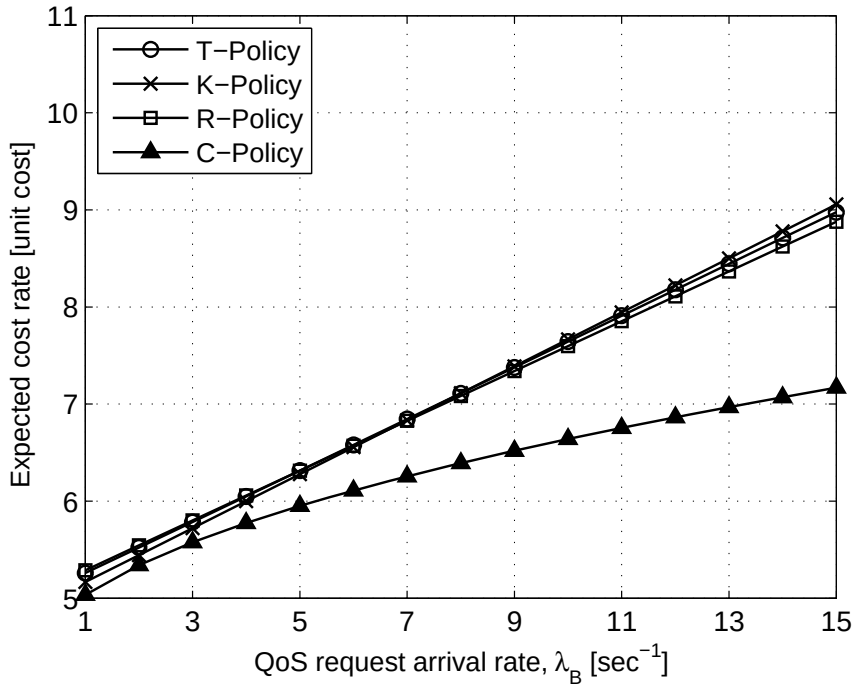


(a)

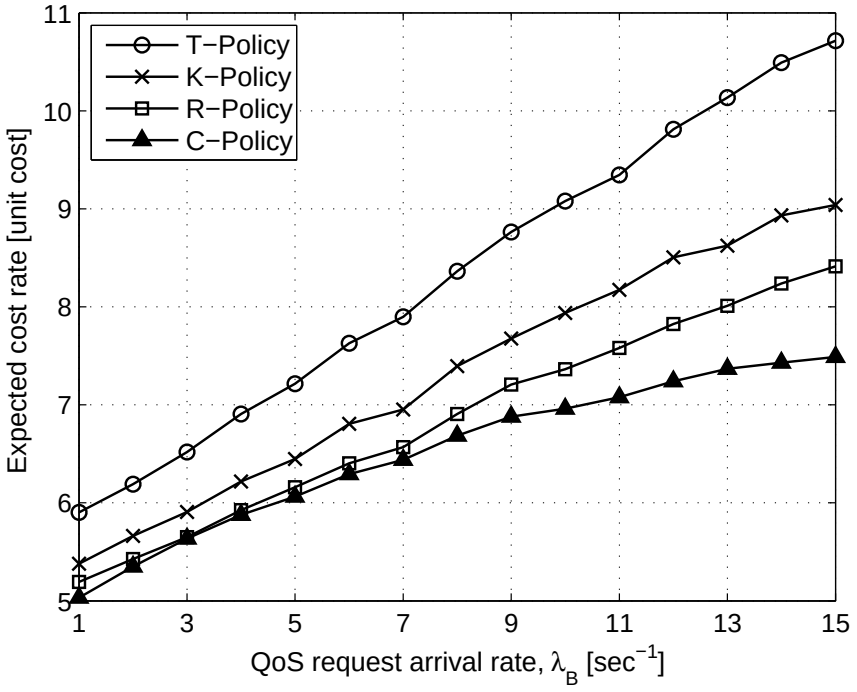


(b)

Figure 3.6: Relation between the expected cost rate of QoS aggregation policies and the QoS request arrival rate under smooth (Poisson) requests for C_s : $E[\mathcal{R}_x]E[\omega_x]C_h = 10 : 1$, $\alpha^* = 1$, $T^* = 1.85$ seconds, $K^* = 13$ requests, and $R^* = 842$ kB/s under (a) the analytical model and (b) the simulation model.

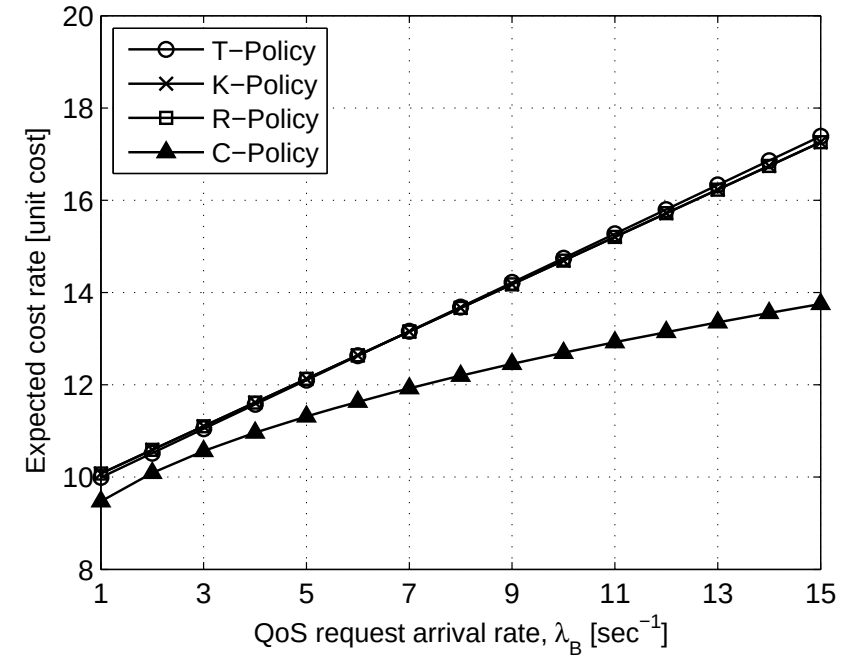


(a)

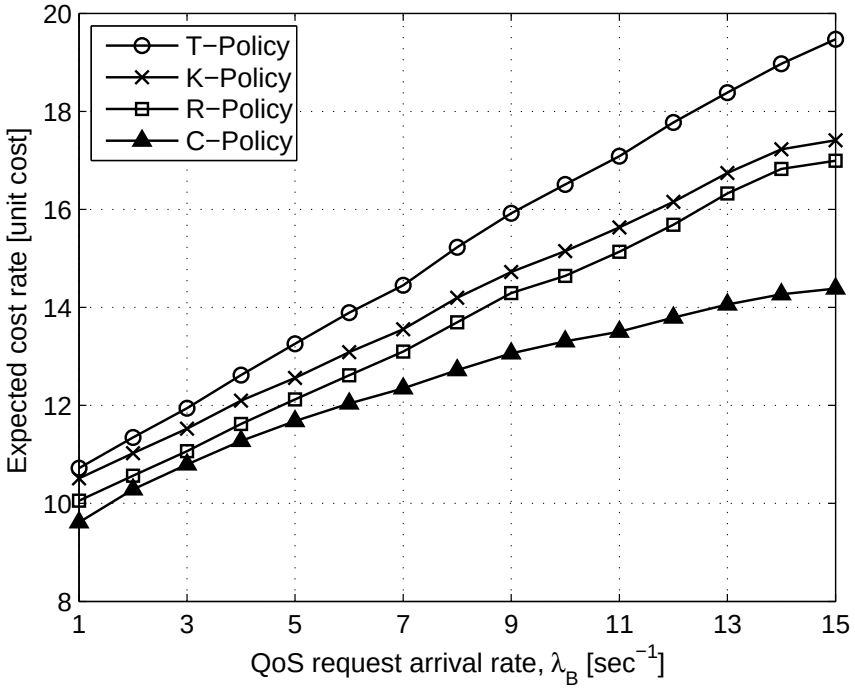


(b)

Figure 3.7: Relation between the expected cost rate of QoS aggregation policies and the QoS request arrival rate under bursty (MMPP) requests for (a) the analytical model and (b) the simulation model with $C_s : E[\mathcal{R}_x]E[\omega_x]C_h = 10 : 1$, $\alpha^* = 1$, $T^* = 3.21$ seconds, $K^* = 7$ requests, and $R^* = 480$ kB/s.



(a)



(b)

Figure 3.8: Relation between the expected cost rate of QoS aggregation policies and the QoS request arrival rate under bursty (MMPP) requests for (a) the analytical model and (b) the simulation model with $C_s : E[\mathcal{R}_x]E[\omega_x]C_h = 40 : 1$, $\alpha^* = 1$, $T^* = 6.35$ seconds, $K^* = 14$ requests, and $R^* = 896$ kB/s.

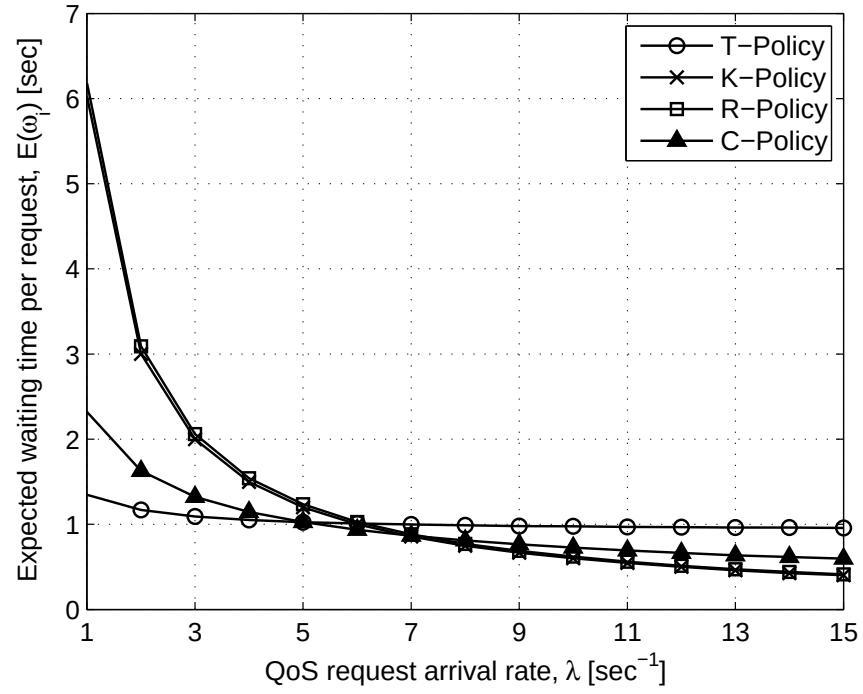
Figures 3.7 and 3.8 show the expected cost rate under bursty requests for cost ratios, $C_s : E[\mathcal{R}_x]E[\omega_x]C_h$, of 10:1 and 40:1, respectively, for both the analytical and

simulation models. From Figure 3.7(b) and (b), it can be seen that for all QoS request arrival rates in the range $1 < \lambda < 15$, the C-policy exhibits a lower cost rate than all parameter-driven policies. This is due to the cost optimality of α in both states of the MMPP model; the optimal threshold of the parameter-driven policies, on the other hand, is optimal for only the mean arrival rate of both states $\overline{\lambda}$. Quantitatively, Figure 3.7(b) shows the cost rate of the C-policy to be lower than that of the most cost-efficient parameter-driven policy (R-policy) by an average of 4.5% across the range of simulated arrival rates for a sending-to-holding-cost ratio of 10:1 and by 10.1% and 20.7% over the K- and R-policies, respectively. In the case of a 40:1 cost ratio, shown in Fig 3.8(b), the minimum average cost rate reduction is 7.9%, which is relative to the R-policy. A similar trend is confirmed by the analytical results in Figure 3.7(a) and 3.8(a).

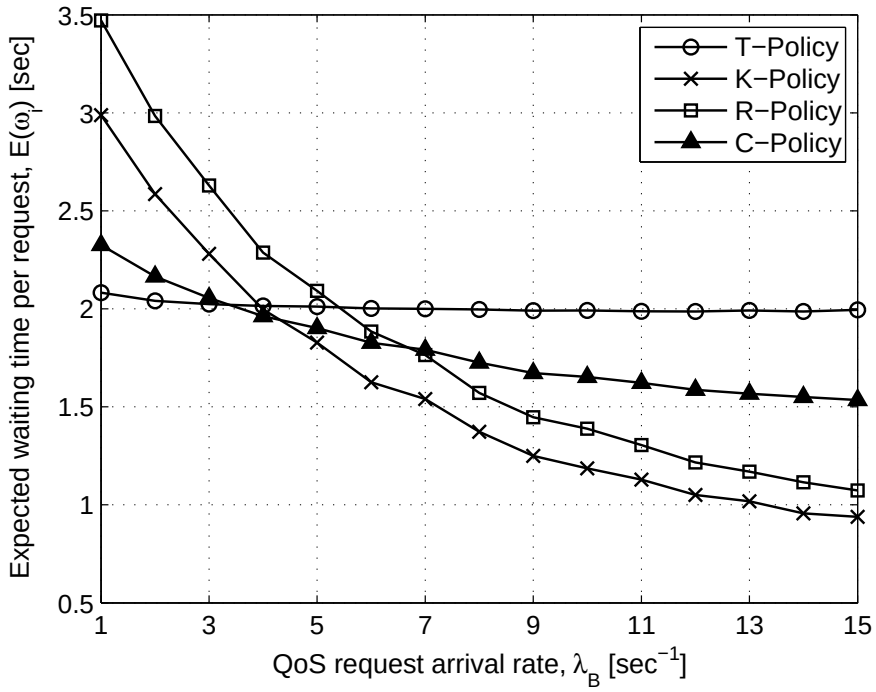
3.7.2 User Waiting Time

Figure 3.9(a) and (b) shows the simulated expected user waiting time to establish a QoS session versus the QoS arrival rate for a signalling-to-holding-cost ratio, $C_s : E[\mathcal{R}_x]E[\omega_x]C_h$ of 10:1 and under smooth and bursty traffic, respectively. Under smooth requests, the average expected waiting time of the C-policy across all arrival rates was marginally higher than that of the K- and R-policies by 0.4% and 3.2%, respectively, but lower than that of the T-policy by 7.5%. However, under bursty requests, the average expected waiting time of the C-policy is higher by an average of 7.2% and 23.7% over the R- and K-policies, respectively, yet lower than the T-policy by an average of 10.6%.

Whilst the *expected* waiting time of the C-policy is lower than that of other policies, it is not necessarily a good indication of the actual waiting times users may incur in any randomly sampled aggregation cycle. Figures 3.10 and 3.11 show the variance of the user waiting time obtained through simulation for each of the aggregation policies. In both the smooth and bursty cases, the T-policy exhibited the lowest user-waiting-time variance, due primarily to the determinism of the cycle duration.

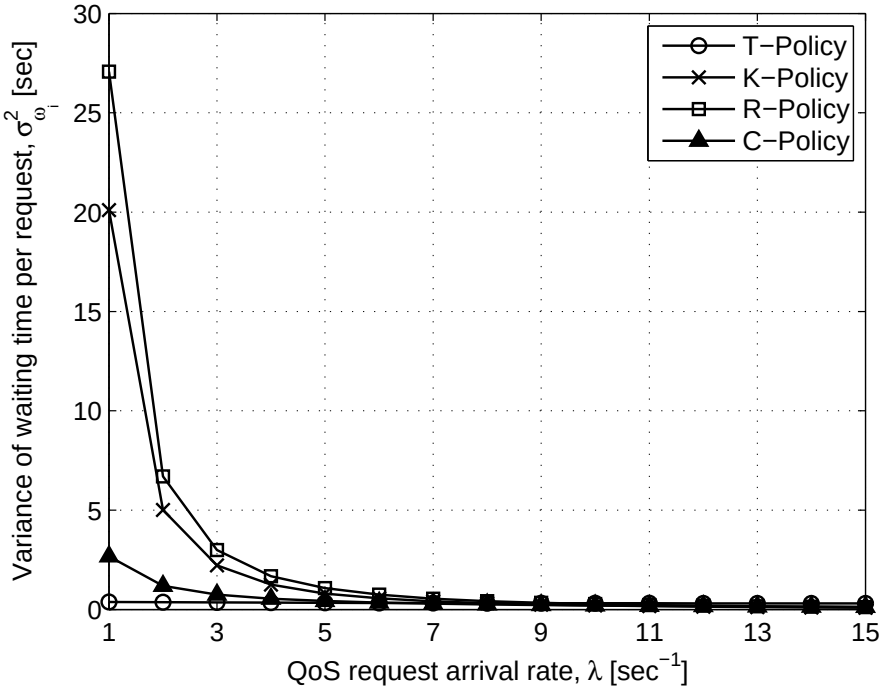


(a)

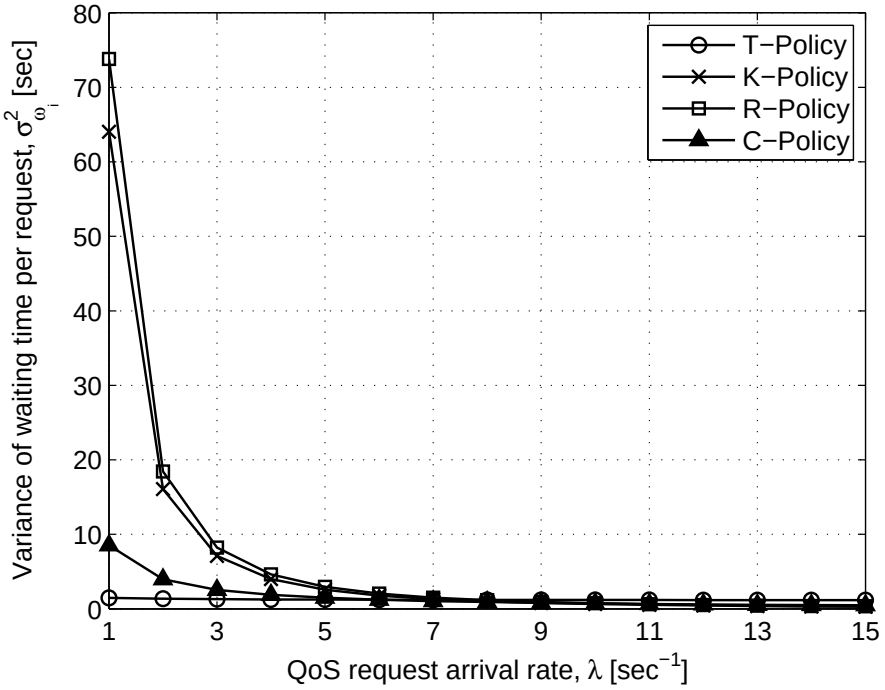


(b)

Figure 3.9: Relation between the *simulated* expected waiting time per request of QoS aggregation policies and the QoS request arrival rate for $C_s : E[\mathcal{R}_x]E[\omega_x]C_h = 10 : 1$, $\alpha^* = 1$ under (a) smooth (Poisson) requests with $T^* = 1.85$ seconds, $K^* = 13$ requests, and $R^* = 842$ kB/s and (b) bursty (MMPP) requests with $T^* = 3.21$ seconds, $K^* = 7$ requests, and $R^* = 480$ kB/s.

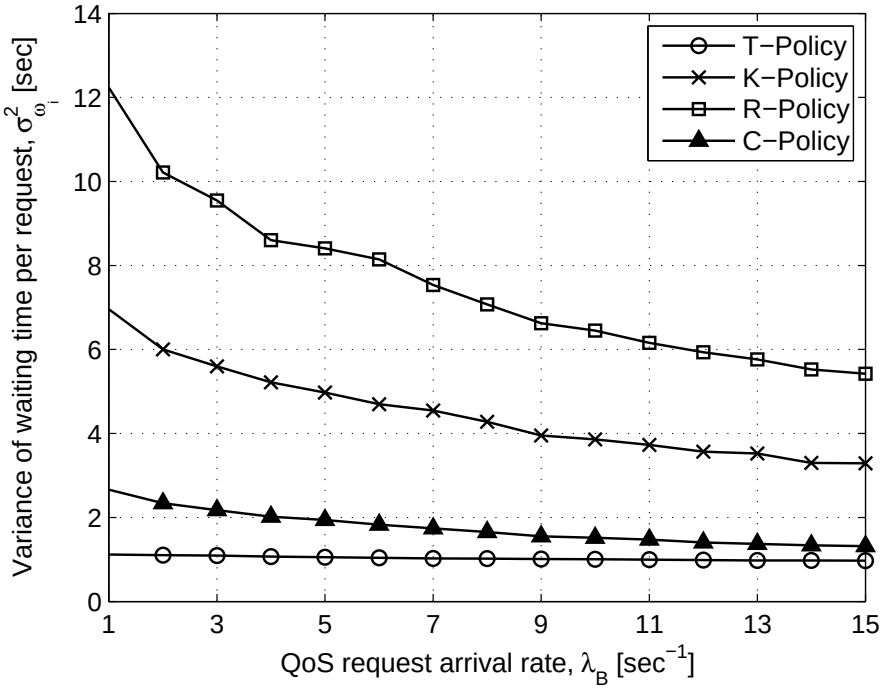


(a)

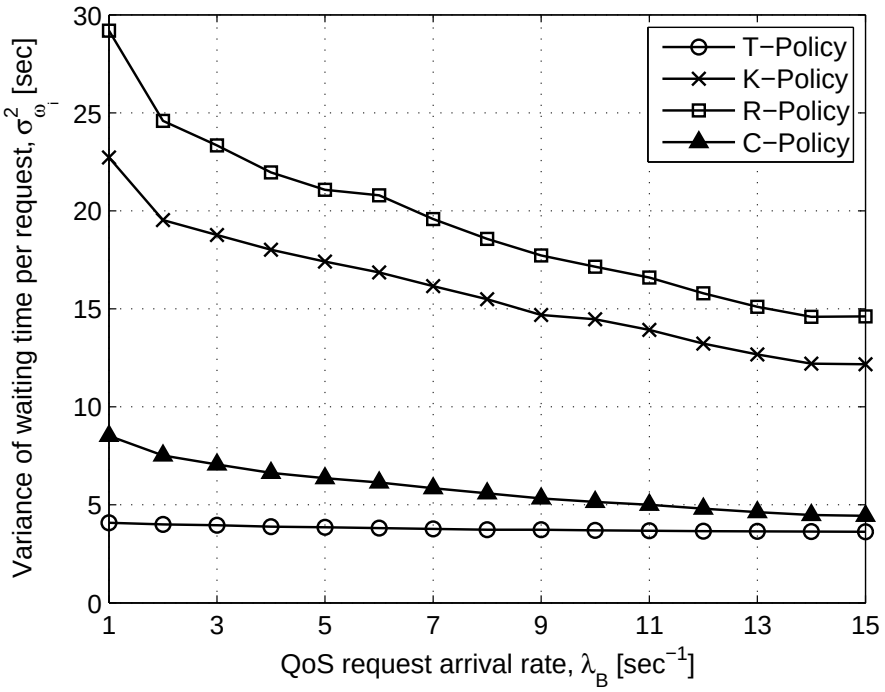


(b)

Figure 3.10: Relation between the *simulated* variance of waiting time per request of QoS aggregation policies and the QoS request arrival rate under smooth (Poisson) requests for (a) $C_s : E[\mathcal{R}_x]E[\omega_x]C_h = 10 : 1$, $\alpha^* = 1$, $T^* = 1.85$ seconds, $K^* = 13$ requests, $R^* = 842$ kB/s and (b) $C_s : E[\mathcal{R}_x]E[\omega_x]C_h = 40 : 1$, $\alpha^* = 1$, $T^* = 3.68$ seconds, $K^* = 25$ requests, and $R^* = 1620$ kB/s.



(a)



(b)

Figure 3.11: Relation between the *simulated* variance of waiting time per request of QoS aggregation policies and the QoS request arrival rate under bursty (MMPP) requests for (a) $C_s : E[\mathcal{R}_x]E[\omega_x]C_h = 10 : 1$, $\alpha^* = 1$, $T^* = 3.21$ seconds, $K^* = 7$ requests, and $R^* = 480$ kB/s and (b) $C_s : E[\mathcal{R}_x]E[\omega_x]C_h = 40 : 1$, $\alpha^* = 1$, $T^* = 6.35$ seconds, $K^* = 14$ requests, and $R^* = 896$ kB/s.

In contrast, the K- and R- policies gave the highest variance, particularly under the case of bursty requests, due to the potential prolongment of the cycle duration when the arrival rate of QoS requests becomes low. Under certain request rates, the variance can be seen to reach as much as 70 seconds under the smooth case and 30 seconds under the bursty case. The C-policy variance, on the other hand, follows closely to that of the T-policy, as cycle durations do not suffer from the theoretical indefiniteness of the K- and R-policies.

From the results relating to waiting time in general, particularly from the variance plots, it is clear that one of the main drawbacks of the K- and R- policies, is that the maximum waiting time of a user to establish a session cannot be guaranteed. This is unlike the case of the T-policy, which has time at the core of its policy, and hence is easily regulated, as can be seen from the T-policy's relatively constant waiting time across all arrival rates in Figures 3.10 and 3.11. For example, if the condition of the optimisation problem in Equation 3.5 is that $\omega_x(\phi) \leq \tau$ and $\tau = 2$ seconds, then there would be no explicit way of achieving this with the threshold value of the nontemporal policies, whereas for the case of the T-policy, it would be just a matter of using the most cost-optimal threshold value below $T = 2$. In comparison, while the C-policy is unable to provide strict guarantees on user waiting times, it does provide a way of limiting user waiting times when the arrival rate is low, thus giving the advantage of reducing both user waiting times and operator costs.

3.8 Discussion and Summary

QoS aggregation policies, in general, significantly reduce the cost of QoS provisioning under moving networks by reducing the amount of control signalling traversing the wireless link between the MR and the AN. However, this cost saving comes at the expense of users having to wait for a variable, non-negligible time lag between session request and session initiation, which is not existent if QoS aggregation is not used.

Comparing the cost saving of the various QoS aggregation policies, it has been found that the proposed dynamic C-policy reduces the cost of QoS provisioning in moving networks beyond that of other previously proposed aggregation policies. Whereas this cost saving is small when QoS requests arrive in a steady flow, the most significant cost saving is achieved when requests are made in bursts (due to passengers boarding/alighting a PTV in batches and subsequently requesting data services). Under these conditions, the percentage cost reduction of the C-policy over other policies has been found to vary considerably over other parameter-driven policies, depending on the cost ratio and traffic-arrival characteristics. In general, the highest average cost saving was found to be 20.7% over the T-policy, compared with 4.5% over the R-policy, i.e. the most cost-efficient parameter-driven policy.

As previously mentioned, QoS aggregation policies cause users to incur a delay from the time a session is requested to the time resources are granted to that session. Comparing the waiting time incurred by the C-policy with that incurred from the parameter-driven policies, the *expected* user waiting time of the C-policy under bursty request characteristics has been found to be higher than the R-policy by an average of up to 7.2% when compared against the R-policy, but reduced over the cost-inefficient T-policy by up to 10.6% on average.

In more absolute terms, the expected waiting time incurred under the C-policy is typically under 5 seconds, whereas under other policies, the expected waiting time can reach up to 12 seconds. In a similar fashion, the *variance* of the waiting time of the C-policy has been found to be no more than 10 seconds, compared with up to 30 seconds for other policies, which is a value likely to stretch user tolerability and thus substantially increase the probability of users reneging on their session request.

Chapter 4

Overlay QoS Aggregation Policies for Congested Networks

4.1 Introduction

The QoS aggregation policies described in the previous section made the silent assumption that the availability of network resources is always greater than user demand. When this assumption is relaxed, a situation could potentially arise, when the network is at the point of saturation, in which QoS requests are no longer aggregated in a cost-efficient manner. The paradox of this situation (and the resulting engineering challenge) is that QoS-related signalling is increased at a time when resources are most scarce, potentially leading to a degradation in the service quality of ongoing sessions. This chapter therefore tackles this problem by considering QoS aggregation policies—so-called "overlay policies"—that work in place of the C-policy.

The following section looks in greater detail at the problem of signalling inefficiency when the network is congested. In light of this problem, Section 4.3 presents revisions to the queue model and cost framework presented in the previous chapter. Following this, Section 4.4 proposes two overlay policies based on the revised cost framework

that attempt to reduce operator cost due to QoS signalling when the network is congested. Section 4.5 then presents the simulation framework used to evaluate the proposed policies; in particular, a new multi-state MMPP model is proposed to better represent the QoS request dynamics of passengers along a journey of a PTV. The results of performance evaluations of the proposed policies carried out in MATLAB are then presented in Section 4.6. Last, the results are discussed and summarised in Section 4.7.

4.2 Problem Description

QoS aggregation and its associated policies have been shown to improve the operational efficiency of QoS provisioning in moving networks, particularly when the rate of session requests is bursty as a result of passenger dynamics. However, when the network reaches saturation, it becomes difficult to aggregate a large number of requests due to the limited resources of the AN. To illustrate the problem, consider the set-up shown in Figure 4.1, in which the AN is represented as a queue with finite capacity, and in which QoS request (RQ) and QoS tear-down (TD) messages are handled in separate queues at the MR each with their own independently-running QoS aggregation policy. When the AN has reached near-saturation, and the aggregation utility of the RQ queue, α_{RQ}, has reached unity, then only the number of QoS requests that can be accommodated are aggregated and sent, even if this aggregate message consists of only one QoS request. Once more capacity eventually becomes available due to an aggregate TD message being sent by the MR to free-up unused network resources, the RQ queue will again aggregate and send only the requests that can be accommodated, despite the potential cost-inefficiency of doing so.

Further difficulties arise from the fact that as congestion increases in the network, the cost of signalling likewise increases. Therefore, by the principle of cost-efficiency shown in the previous chapter, in order to operate cost-efficiently, the dormant period of the MR's aggregation queue must increase. This brings up an entirely

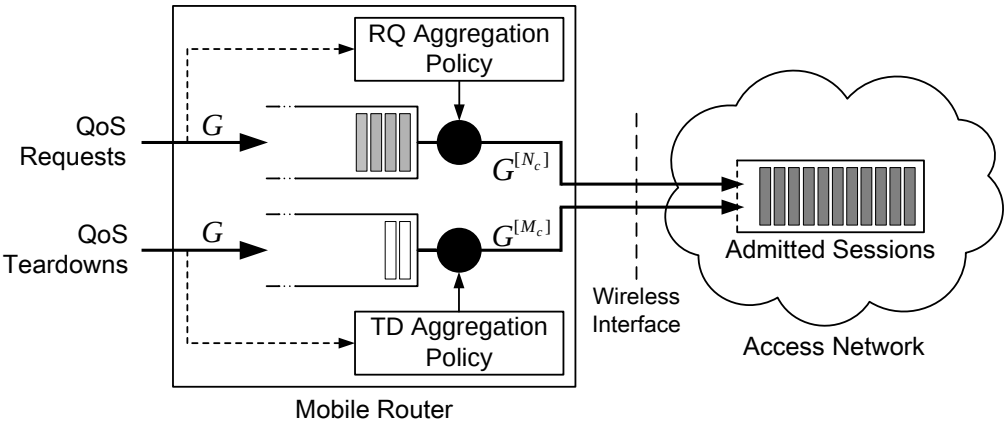


Figure 4.1: Queue configuration under constrained network resources.

different problem of the possibility of users reneging on their QoS requests due to prolonged waiting. With the system of costs previously introduced, this translates to a lost opportunity for generating revenue.

One possible way of addressing this problem is to introduce an element of hysteresis in the MR, such that when the network becomes saturated, the MR enters a "cooling-off" period during which it is prevented from sending any aggregate resource reservation request messages. This mechanism can reduce the cost of sending in the long-term, but can also lead to increased holding cost and increased probability of users reneging on their session request. Therefore, in order to achieve operational efficiency, the length of the cooling-off period must be controlled in such a way that ensures costs are minimised. This will require the MR to be able to monitor the degree of network congestion within the network to which the MR is attached, using some form of probing technique that can, for example, infer congestion levels from the round-trip time of sending a control packet through the network. This aspect, however, remains outside the scope of this thesis.

4.3 Revised Queue Model and Cost Function

When user demand exceeds the amount of available network resources, additional factors must be accounted for in the queue model and cost function that were not present (i.e. not relevant) in the previous chapter.

When a session is terminated, it is important to tear-down its associated resource reservation in the network by reducing the resources allocated to the virtual tunnel and the queues within it. We assume a simple approach to achieve this, whereby TD messages are queued at the MR in a separate queue from the RQ messages, as shown in Figure 4.1, and which are aggregated in the MR using the cost-driven approach proposed in the previous chapter. While it could be argued that this would prevent local exploitation of released resources for new requests, this approach is particularly necessary in situations where virtual reservation tunnels are to be established, as it is unlikely that an new request will have resource requirements that are similar in specification to an old reservation. Therefore, the aggregation processes of both the RQ and TD queue are triggered by their respective aggregation utility functions (denoted α_{RQ} and α_{TD}, respectively).

With respect to cost, three other costs must be taken into consideration besides the costs of holding and sending (C_h and C_s, respectively). These are the cost of under-utilisation of resources, denoted C_u, the cost of signalling to tear-down resources from the network, denoted C_t, and the cost of users reneging on their request, denoted C_r. C_u relates to the cost of keeping resources reserved in the network that are no longer needed, and which could have been used by other potential users of the network. On the other hand, signalling to tear-down resources would also incur a cost, C_t, but it can be argued that the cost of signalling to tear-down resources is not as high as that of requesting resources, as tearing-down does not require resource-expensive operations such as admission control. Finally, the reneging cost, C_r, relates to the cost due to users reneging from the RQ as a result of waiting too long for a QoS-enabled connection to be established by the MR.

Based on the new costs that have been introduced, the cost function of the RQ queue can be reformulated as

$$C_c = \frac{C_s + \sum_{i=1}^{N_c} (\omega_i \mathcal{R}_i C_h) + \sum_{j=1}^{M_c} (\omega_h \mathcal{R}_j C_r)}{T_c}, \tag{4.1}$$

where M_c is the number of requests that have reneged during cycle c, and ω_j and $\mathcal{R}_j$ are the waiting time before reneging and the resources that were requested by the jth user to have reneged.

Similarly for the TD queue, we can formulate its cyclic cost, C_d as

$$C_d = \frac{C_t + \sum_{k=1}^{N_d} (\omega_k \mathcal{R}_k C_u)}{T_d}, \tag{4.2}$$

where N_d is the number of tear-down messages that have been aggregated at the end of tear-down cycle d, ω_k and $\mathcal{R}_k$ represent the queueing time and amount of resources to be torn down of the kth queued user since the beginning of a cycle, and T_d represents the duration of cycle d.

4.4 Overlay QoS Aggregation Policies

When the network is unloaded (i.e. not congested), it is possible to run the aggregation policies of both the RQ queue and the TD queue using the cost-driven policies with each queue accounting for its respective cost-utilisation, α_{RQ} and α_{TD}. Extending the cost-utility given in Equation 3.25, α_{RQ} can be expressed as:

$$\alpha_{RQ} = \frac{\sum_{i=1}^{N_c} (\omega_i \mathcal{R}_i C_h) + \sum_{h=1}^{M_c} (\omega_h \mathcal{R}_h C_r)}{C_s}, \tag{4.3}$$

and α_{TD} as

$$\alpha_{TD} = \frac{\sum_{j=1}^{N_d} (\omega_j \mathcal{R}_j C_u)}{C_t}, \tag{4.4}$$

whereby cost optimality of each queue is attained when its respective cost-utility reaches unity. However, when the network is congested, Equation 4.3 will not yield cost-optimality since α_{RQ} is likely to exceed unity due to insufficient availability of network resources. Therefore, one approach is to let the MR enter a cooling-off period once the network reaches saturation, in which no further aggregation messages are sent. This would effectively let the enforcement of the C-policy be bypassed during this period. However, since preventing aggregation during the cooling-off period

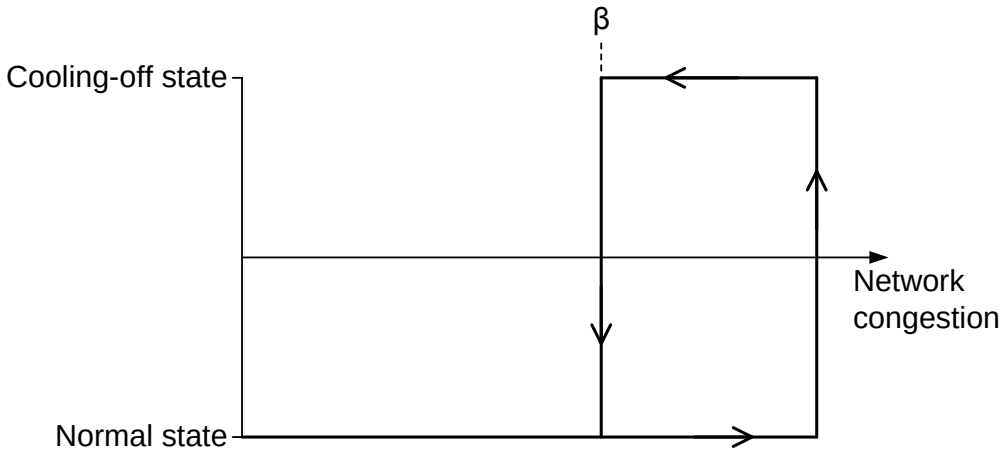


Figure 4.2: Hysteresis curve of the S-policy.

will lead to increased holding and reneging costs, a policy is required to ensure that overall costs are kept to a minimum in the long-term, through appropriate control of the duration of the cooling-off period.

The following subsections therefore propose two such policies—a static policy (S-policy) and a dynamic policy (D-policy)—as possible ways to control the duration of the cooling-off period.

4.4.1 S-Policy

The S-policy is a simple policy in which the lower congestion threshold is fixed to such a value that minimises cost across all rates of QoS requests. Therefore, when the network reaches saturation, the MR enters the cooling-off state, whereby no further requests are aggregated until the network congestion drops again below a lower congestion threshold β, and transitions back to the normal state, as shown in Figure 4.2. Once the MR enters back into the normal state, requests are aggregated again in the normal manner of the C-policy until once again the network reaches saturation and enters the cooling-off state.

Since no requests are aggregated during the cooling-off period, technically, no cost would be incurred by the RQ queue during this time. However, it also means that

the value of α_{RQ} will be significantly greater than unity for the first cycle after each transition from a cooling-off state to a normal state.

4.4.2 D-Policy

The D-policy works along the same principles as those of the C-Policy, based on a modified version of the cost-utility function, $\tilde{\alpha}_{RQ}$, that takes into account *only* the requests that can be admitted into the network as well as the requests that have reneged. It essentially provides a way of determining the "optimal" cooling-off duration according to the instantaneous request characteristics, however, the concept of a cooling-off period is perhaps not as clear cut as in the S-policy. Therefore, the aggregation utility function would, in such a case, have the form

$$\tilde{\alpha}_{RQ} = \frac{\sum_{i=1}^{\tilde{N}_c} (\omega_i(\tilde{\alpha}_{RQ}, \lambda)\mathcal{R}_i C_h) + \sum_{j=1}^{M_c} (\omega_j(\tilde{\alpha}_{RQ}, \lambda) C_r)}{C_s}, \tag{4.5}$$

where $\tilde{N}_c$ is the number of QoS flows out of those queueing for resources that can be admitted into the network; M_c is the number of requests that have reneged by the end of aggregation cycle, c; and C_r is the base cost of a user reneging. Accordingly, the cyclic cost under this scheme is given by

$$\begin{aligned}\tilde{C}_c(\tilde{\alpha}_{RQ}, \lambda) = \frac{1}{T_c(\tilde{\alpha}_{RQ})} C_s + \sum_{i=1}^{\tilde{N}_c} (\omega_i(\tilde{\alpha}_{RQ}, \lambda)\mathcal{R}_i C_h) \\ + \quad \frac{1}{T_c(\tilde{\alpha}_{RQ})} \sum_{j=1}^{M_c} (\omega_j(\tilde{\alpha}_{RQ}, \lambda) C_r).\end{aligned}$$

4.5 Performance Evaluation Framework

The cost-efficiency and expected user waiting time under the proposed policies were evaluated using event-driven simulations in MATLAB [57]. The session request dynamics of users was modelled as a P-state Markov-modulated Poisson process, which allows for a better approximation of the typical expected behaviour of passengers

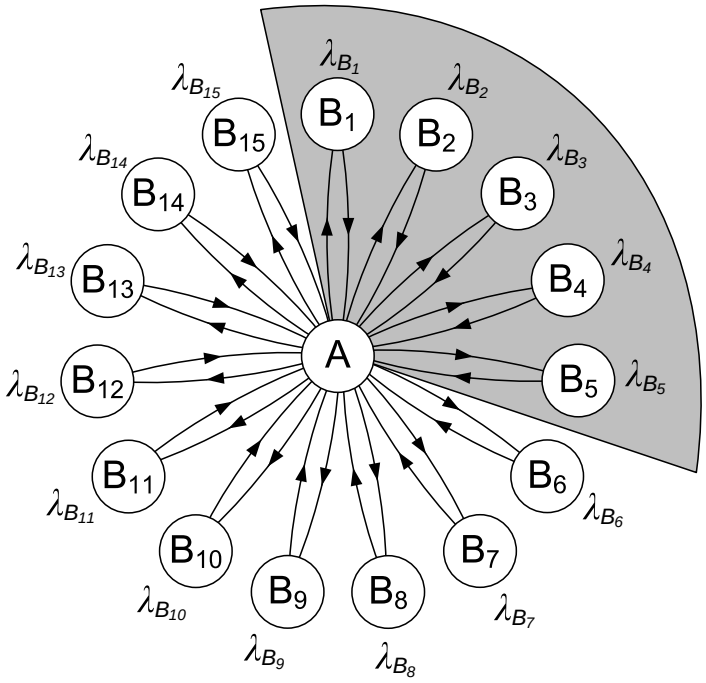


Figure 4.3: P-state MMPP used to model bursty QoS request behaviour.

boarding and alighting a vehicle along a route in which the busyness of stations is variable. It also prevents the unrealistic situation of the network being constantly congested at high rates of request. Figure 4.3 shows an example of the model that was used; QoS requests are made to vary between an *ambient* state, A, which represents the rate of requests per second, λ_A *between* station stops, and one to several *bursty* states, B_p, which represent different possible rates of requests per second, λ_{B_p} *at* each station stop. In our study, we look at the performance under various values of P, in which the probability of going from state A to state B_p is drawn from a discrete triangular distribution[1] with a lower limit of zero; mode, $\left\lceil \frac{P+1}{2} \right\rceil$; and upper limit of $P + 1$, where $P \in \mathbb{Z} : 1 \leq P \leq 15$. Therefore, the transition probability

[1]The MATLAB tool used to generate discrete random numbers from a triangular distribution is the TRIRND code implemented by Cavin [58].

from state A to state B_p is given by

$$\Pr(AB_p) = \begin{cases} \frac{2p}{(P+1)\left\lceil \frac{P+1}{2} \right\rceil} & \text{for } 1 \leq p \leq \left\lceil \frac{P+1}{2} \right\rceil \\ \frac{2(P+1-p)}{(P+1)\left(P+1-\left\lceil \frac{P+1}{2} \right\rceil\right)} & \text{for } \left\lceil \frac{P+1}{2} \right\rceil \leq p \leq P \\ 0 & \text{otherwise.} \end{cases}$$

Taking the shaded region of Figure 4.3 as an example of a 5-state MMPP, the maximum possible arrival rate of a station would be 5 $requests/s$, and the minimum, 1 $requests/s$; the transition probability from state A to state B_p for $p > 5$ would be zero.

In the following simulation study, the sojourn time of state A and state B_p was assumed to be exponentially distributed with mean 50 seconds and 10 seconds, respectively. Furthermore, the requested throughput across QoS requests and the session durations were assumed to be exponentially distributed with mean, $E[\mathcal{R}] = 64$ kB/s and 120 seconds, respectively.

Contrary to the previous chapter, the subsequent study examines the cost-effectiveness of the proposed overlay policies under a sending cost, C_s, which was made to vary between 0 and 200 [unit cost], according to the level of congestion being experienced in the network. This varying value of C_s is expressed in relation to the value of the holding cost normalised by the average requested throughput: $E[\mathcal{R}_x]E[\omega_i]C_h$, which has a value of unity, and remains constant throughout the simulation. Similarly, the tear-down cost C_t is set to one-tenth of the value of C_s due to the lower amount of resources required in processing a TD message in the network. The cost of under-utilisation C_u was set to twice the value of C_h, while the reneging cost was kept equal in value to C_h. Since only the C-policy was required in controlling the aggregation of messages in the TD queue, the overall cost incurred by that queue was not included in the results.

The network used in the simulations was assumed to have a throughput capacity of 11 Mb/s. The simulation of each policy was carried out for 500,000 requests, and the reneging time across all requests was normally distributed with a mean and standard deviation of 20 seconds[1] and 5 seconds, respectively.

4.6 Performance Evaluation

This section presents the simulation results obtained for both the S-policy and D-policy, as well as the case in which no overlay QoS aggregation policy is used (i.e. which is equivalent to using the S-policy with $\beta = 100\%$). In each of these three cases, the C-policy is used if no other congestion policy is being used.

4.6.1 Optimal S-Policy Lower Congestion Threshold

The optimal lower congestion threshold β^* for the S-policy was determined by simulation[2]. The simulation was executed for each value of P in the range, $1 \leq P \leq 15$, and over a range of values of β. Due to the time required to carry out a single run of the simulation (i.e. for each integer value of P and β [%]), β was initially varied in coarse 10%-intervals, which was used to estimate the threshold region in which the overall cost is minimised. This was then used to trigger a finer simulation run focussing on obtaining results in 2%-intervals up to 10% either side of the minimum, thereby allowing a more accurate minimum and associated lower congestion threshold to be obtained. Figure 4.4 shows an example expected cost-rate curve obtained for an MMPP-model with $P = 10$.

The minimum expected cost rate was plotted against its associated lower congestion threshold percentage, which is given in Figure 4.5. The error bars shown in the plot represent the range of percentages for which the expected cost rate remains

[1]The value of 20 seconds was chosen based on studies carried out by British Telecom (BT) on the mean time for which users are likely to wait to be connected before reneging. This information was obtained verbally from Stewart Fallis of BT, but no concrete citation can be provided.

[2]The simulation used in this particular part of the study was carried out over ten processing cores to reduce simulation time. The MULTICORE tool for MATLAB developed by Buehren [59] was used to fulfil this purpose

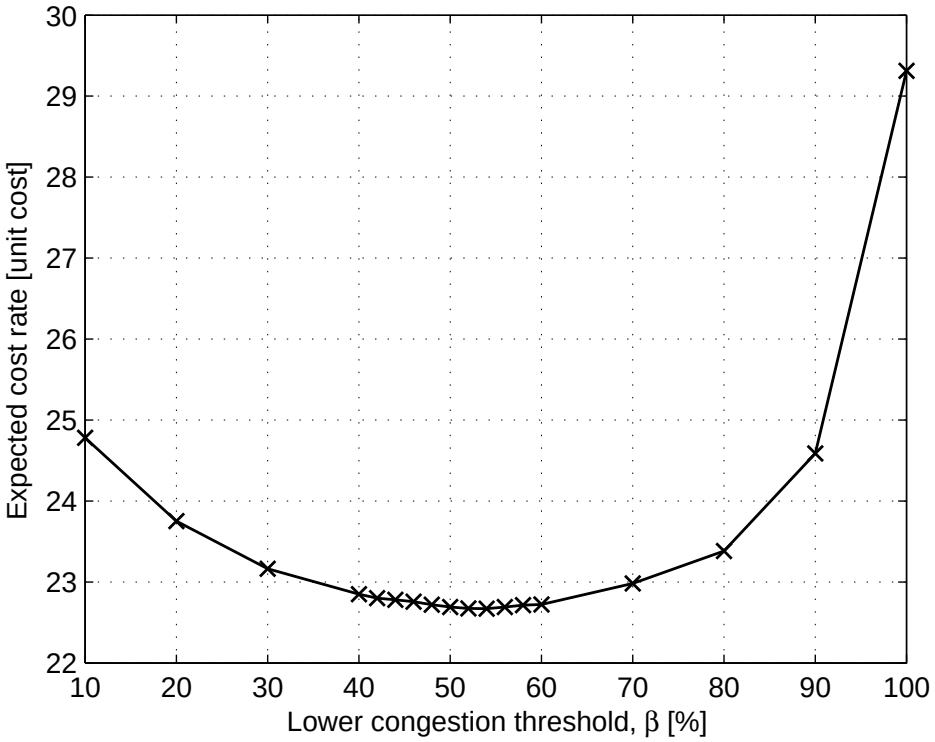


Figure 4.4: Expected cost rate of the S-policy for $P = 10$ and for values of β in the range, $10 \leq \beta \leq 100$.

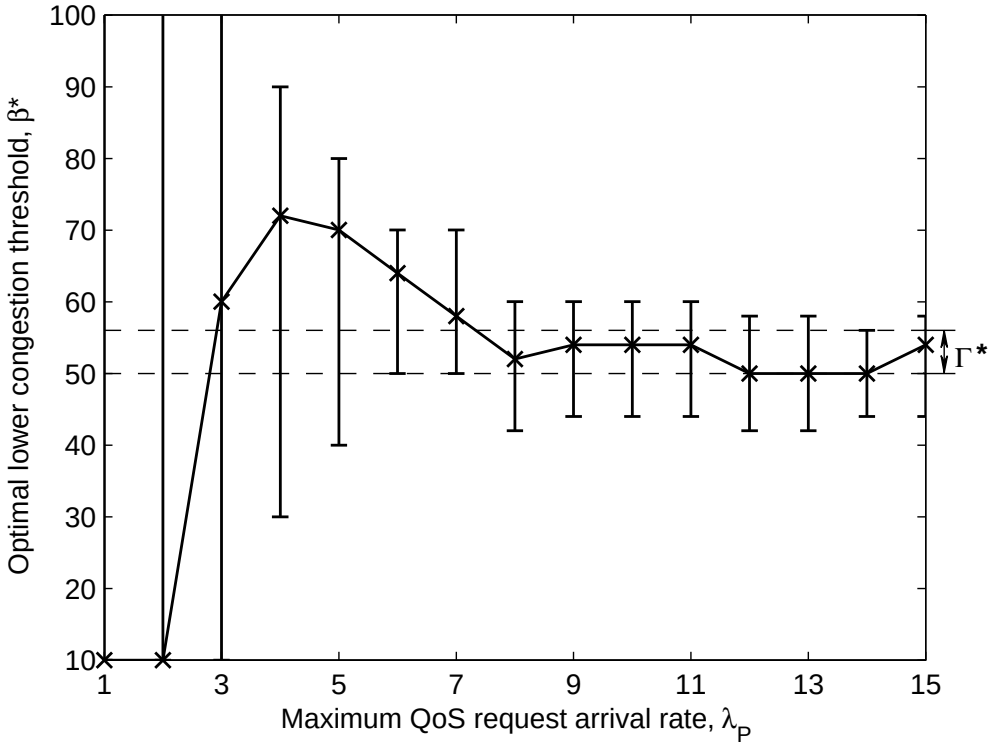


Figure 4.5: Variation of the optimal lower congestion threshold with the maximum QoS request arrival rate, λ_P.

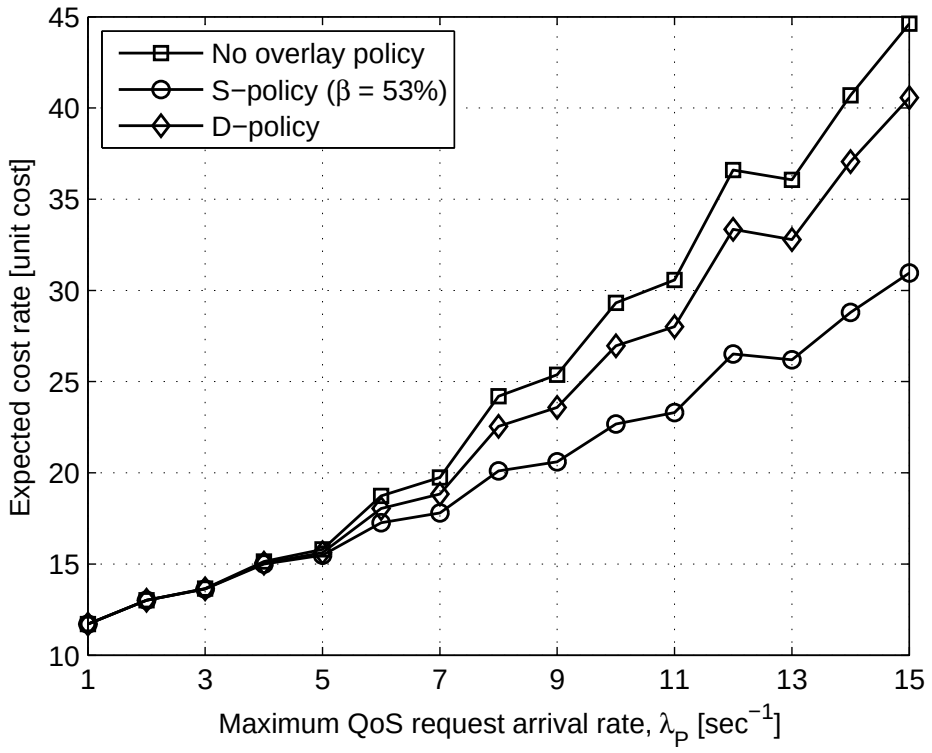


Figure 4.6: Relation between the expected cost rate and the maximum QoS request arrival rate, λ_P

to within 0.5% of the minimum cost. In this study, it is assumed that the value of P (i.e. the maximum arrival rate) is unknown. Therefore, the optimum lower congestion threshold was taken from the set of percentages obtained by intersecting the percentages of each value of P. In other words, if the set of percentages for a given P lie within the set, Γ_P, then the range of possible values of β^* will lie within the set Γ^*, where

$$\Gamma^* \in \bigcap_{p=1}^{P} \Gamma_v. \tag{4.6}$$

The actual value of β^* used in the simulations was the median value of Γ^* (53%), which allows the most room for error in congestion measurements.

4.6.2 Operator Cost

Figure 4.6 shows the expected cost rate for both the S- and D-policies, as well as the case in which no congestion policy is applied. It can be seen from the figure that the S- and D-policies both out-perform the no-overlay-policy case by margins that

grow with increasing values of λ_P. However, the cost-saving margin of the S-policy is significantly greater than that of the D-policy. This can be attributed to the fact that the D-policy aims only for local optimality (i.e. optimality taking into account only the current aggregation cycle), whereas the S-policy, which although uses a static parameter, is able to achieve global optimality (taking into account all aggregation cycles over a large window of time). In quantitative terms, the maximum potential cost-saving of the D-policy relative to the no-policy case is 9.1% with an average saving of 5.1% over the range of values of λ_P that was simulated. On the other hand, the S-policy is able to achieve a maximum potential cost saving of 23.7%, and an average 10.3% over the range $1 \leq \lambda_P \leq 15$. For $\lambda_P \leq 3$, the expected cost rate is observed to be the same for all policies. This is due to the fact that congestion does not occur at low values of λ, which results in only the C-policy ever being used.

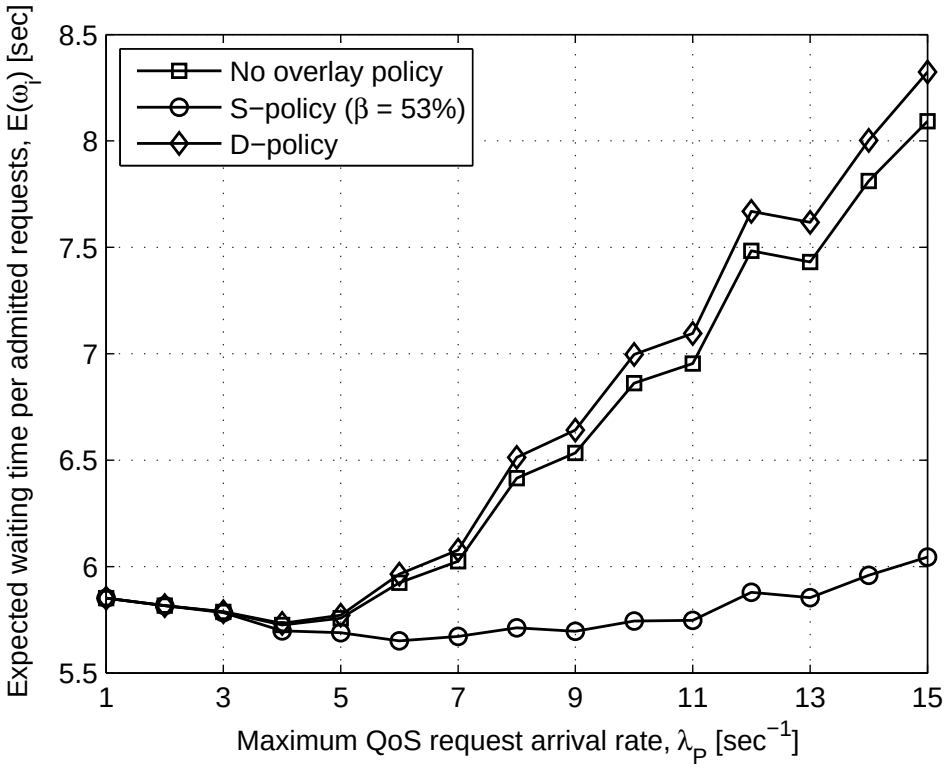


Figure 4.7: Relation between the expected waiting time of a QoS request and the maximum QoS request arrival rate, λ_P

4.6.3 User Waiting Time

The expectation and variance of the user waiting time of *admitted* requests for each of the simulated policies are shown in Figures 4.7 and 4.8, respectively. From Figure 4.7, the expected waiting time of the D-policy can be seen to be marginally greater than the no-overlay-policy case, with a maximum potential waiting time increase of 2.8% (0.23 seconds) and an average of 1.3% (0.1 seconds). On the other hand, the S-policy reduces the expected waiting time over the no-overlay-policy case by a maximum potential waiting time of 13.0% (2.0 seconds) and an average of 10.1% (0.8 seconds), again indicating the ability of the S-policy to achieve a performance that is based on a global optimum rather than a local one.

A similar trend as that given by the expected waiting time can also be seen with the variance of waiting time, which is plotted in Figure 4.8. The D-policy is found to have a maximum potential waiting time variance of 40.0 seconds, which represents a 2.4% (1.0 seconds) increase over the case in which no overlay policy is used. By comparison, the S-policy yields a significantly lower waiting time variance. The maximum variance was found to be 27.1 seconds, representing a peak reduction of 30.5% (11.9 seconds) over the no-overlay-policy case.

4.6.4 Admittance Percentage

One of the direct impacts of hysteretically holding requests at the MR is that requests could be forced to wait significantly longer than the user is willing. This leads to an increase in the percentage of users that renege on their request(s), or conversely, a reduction in the percentage admittance of requests. Figure 4.9 shows the admittance percentage of requests for each of the simulated policies. It can be observed that the percentage reduction of admitted requests relative to the no-overlay-policy case reduces significantly with increasing maximum QoS request arrival rate. In absolute terms, this reduction equates to an average of 6.42% fewer admitted requests than the no-overlay-policy case, with a maximum potential reduction of 12.8%. In

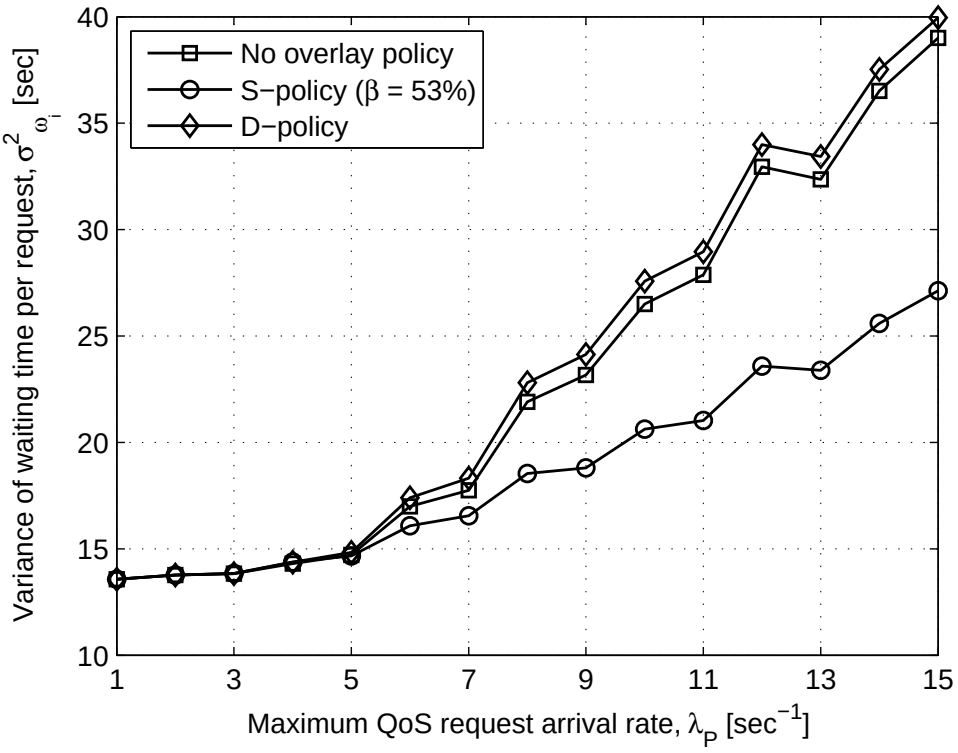


Figure 4.8: Relation between the variance of waiting time of QoS requests and the maximum QoS request arrival rate, λ_P

contrast, the D-policy reduces the percentage of admitted requests by only 0.3% over the no-policy case.

4.7 Discussion and Summary

The principle of aggregating QoS requests have been shown in the previous chapter to improve cost efficiency (and hence operational efficiency) for the network operator. However, when the network is congested, the efficiency of QoS aggregation becomes significantly compromised as the number of requests that can be aggregated is limited by the remaining available network resources. This chapter has therefore proposed two potential aggregation policies (the S-policy and the D-policy) that could be used to increase operational efficiency when the network becomes congested by preventing new aggregations until a particular parameter value is reached. In the case of the D-policy, this parameter is the aggregation utility of the combined

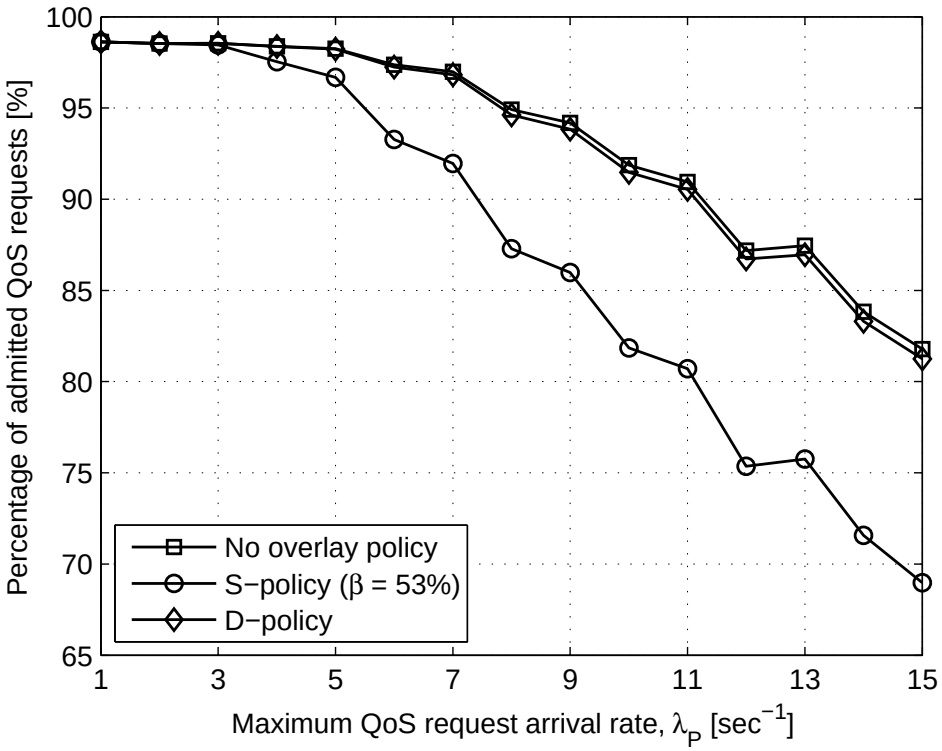


Figure 4.9: Relation between the admittance percentage of QoS requests and the maximum QoS request arrival rate, λ_P

holding and reneging costs reaching unity, while in the S-policy, it is the network congestion dropping to a particular level.

Comparing the cost saving of the S- and D-policies against the case in which no overlay policy is applied, it was found that the S-policy led to significantly reduced costs and user waiting times, while the D-policy only marginally improved operator cost and increased user waiting times. Most notably, there was a 23.7% maximum potential cost saving over the no-overlay-policy case, with 13.0% and 30.5% maximum potential reductions in the expectation and variance of waiting times respectively. This compares with the D-policy's 9.1% maximum potential improvement in cost, and 2.8% and 2.4% increases in the expectation and variance of users' waiting time respectively. The S-policy's ability to reduce costs stems from the fact that although only a static parameter is used (for which the optimal value was determined through simulation), its optimisation of costs is based not only a single cycle, but over many cycles collectively. In other words, the S-policy can be said to aim for

global optimality, whereas the D-policy, only local. However, in order to gain such significant cost-savings, the percentage of admitted requests was reduced over the no-overlay-policy case by a maximum potential of 12.8%.

While the study in this chapter has considered only a particular scenario of input parameters such as the rates of QoS requests and reneging behaviour of users, the simulation framework used can easily be extended to incorporate "real" input data collected from live moving-network deployments. This will allow for better approximation of optimal policy parameter values. Further research in this area may also consider a more dynamic technique of optimising costs when the network is congested, for example, by considering the request rate experienced over a moving window of time, such that cost optimisation could be achieved at run-time for any set of input parameters.

Future research in this area may also consider an element of user satisfaction, and the way in which the aggregation policy can be tailored to achieve balancing the cost of signalling and the satisfaction of users. This may involve providing estimates of the expected waiting time to users, which studies have shown can improve the overall satisfaction of users and their tolerance to waiting [60]. Another important area for consideration in future studies in this area is is on the parameters that influence the lower congestion threshold of the S-Policy, and the possibility that the relation between this threshold and other system parameters be represented formulaically, such that optimisation could be achieved under different and/or dynamic system scenarios.

Chapter 5

Seamless and QoS-Enabled Mobility Management

5.1 Introduction

The previous chapters have focussed on methods to improve the efficiency of *pre-session* signalling, whereby the aim was to ensure cost-effectiveness for the network operator, without causing users to wait an unreasonable amount of time to establish a QoS-enabled session with the fixed network. However, with an increasing number of applications becoming more delay-intolerant, there is a strong need to ensure that *in-session* signalling is also made to be efficient, such that the occurrence of a network-layer handover does not affect the seamlessness and agreed QoS of sessions that already have a connection established with the network.

Vehicular networks are at an advantage over self-managed mobile terminals when it comes to addressing the physical layer problems that occur due to travelling at high velocity. For example, the space afforded by vehicles makes it possible to deploy multiple antennas to mitigate the effects of multi-path fading, and the virtually limitless availability of power makes the use of sophisticated filters to overcome the effects of Doppler shifts more feasible. However, at a networking level, travelling

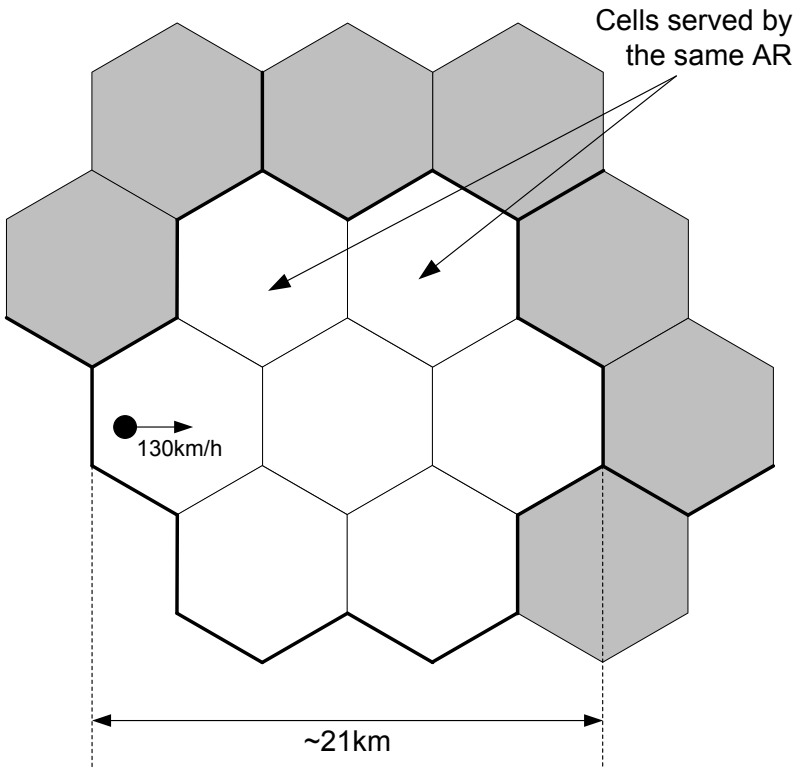


Figure 5.1: Illustration of the reason for high handover frequency of moving networks communicating through terrestrial.

at high velocity can present its own set of problems that no amount of space or power can address. For instance, by using wide-area terrestrial technologies such as High-Speed Downlink Packet Access (HSDPA) and Worldwide Inter-operability for Microwave Access (WiMAX) 802.16e, a vehicle travelling at high velocity will be required to perform an IP handover very frequently. To illustrate, consider the cell topology of an 802.16e network shown in Figure 5.1, in which each cell has an average radius of four kilometres, and each cluster of seven cells is controlled by its own AR. Assuming the best case scenario in which a PTV traverses a cell cluster through its centre cell, and an average travelling speed of 130km/h, a handover would need to be performed approximately every ten minutes.

Some providers of broadband for public transport have taken the approach of using satellite technology as the main data carrier, with terrestrial technology used only as a so-called *gap-filler* when satellite reception is not available. Although satellite

can support high bandwidth and eliminate the need for handovers, it suffers from three major disadvantages. First, real-time applications cannot be supported due to large propagation delay, thus for such applications at least, terrestrial technology would need to be used. Second, satellite requires a Line-of-Sight (LOS) path, making it unsuitable for routes with signal obstructions such as tunnels. In the worst case, a situation could arise in which a vehicle unstably hands over to and from terrestrial as it experiences fluctuating satellite reception. Finally, satellite has much greater operational expenditure than terrestrial, requiring either passengers to pay a premium for the service, or the PTV operator to subsidise it.

There is therefore a clear need for terrestrial in cases where satellite is technically unfeasible or economically unviable. However, in order to use terrestrial with satisfactory performance, a method is required for ensuring more seamless connectivity to users as a vehicle roams within and between ANs. In traditional host-mobility in which terminals manage their own mobility, seamlessness is most commonly achieved through the employment of a micro-mobility protocol which aims to minimise the average depth with which handover-related control signalling propagates into the network. However, applying such protocols to a network mobility scenario can lead to a number of operational problems that can degrade the QoS experienced by users.

This chapter therefore presents a novel, patent-pending mechanism [61, 62] called QENEMO, designed to ensure the continuity of all sessions handled by an MR as it roams and performs handovers between networks. The following section illustrates in greater depth the problem of seamless service provision in moving networks that we are addressing. Following this, Section 5.3 details our approach to the problem, with the supporting functionalities required by QENEMO expanded upon in Section 5.4. Section 5.5 then sets out an implementation of QENEMO within the NSIS protocol framework, detailing the node architecture, and proposed message and object formats. Finally, the chapter is concluded in Section 5.6.

5.2 Problem Description

The NEMO Basic Support (NBS) protocol is able to bring significant scalability improvements through its employment of the PSBU in updating the location of a moving network and all of the terminals that it serves. However, the process of performing a binding can introduce unwanted delays and lost packets, leading to performance degradations from the users' perspective. As the NBS protocol uses at its core the same mechanism as that of Mobile IP (MIP), the handover latency of NBS is comparable to that of MIP, which can range from one to three seconds depending on the configuration of the protocol [63].

The reduction in handover latency offered by tunnel-based micro-mobility protocols underlines their importance in the goal towards achieving seamless communications. However, one of the main weaknesses of such protocols is their centralised nature which, by virtue of their operation, can lead to bottleneck congestion occurring within the AN [64]. The HMIPv6 protocol [27], for example, uses so-called MAPs to track the location of a terminal as it moves between ARs within the same AN.

Under HMIPv6, every terminal is allocated two IPv6 addresses besides its home address: a regional one provided by the MAP, which the terminal registers with its HA in the usual way, and a local one provided by each AR at every handover, which the terminal need register only with its serving MAP. While this significantly reduces the frequency with which a terminal need contact its main HA (and thereby reducing the average handover execution time), the two-tier addressing mechanism brought about by HMIPv6 forces the packets of all sessions being served by a MAP to flow through it, despite the likely existence of less-congested routes within the AN.

In moving networks, the bottleneck problem that micro-mobility protocols induce is significantly magnified. This is due to the effect of the high volume of traffic that moving networks transfer, coupled with the further limitation imposed by the NBS

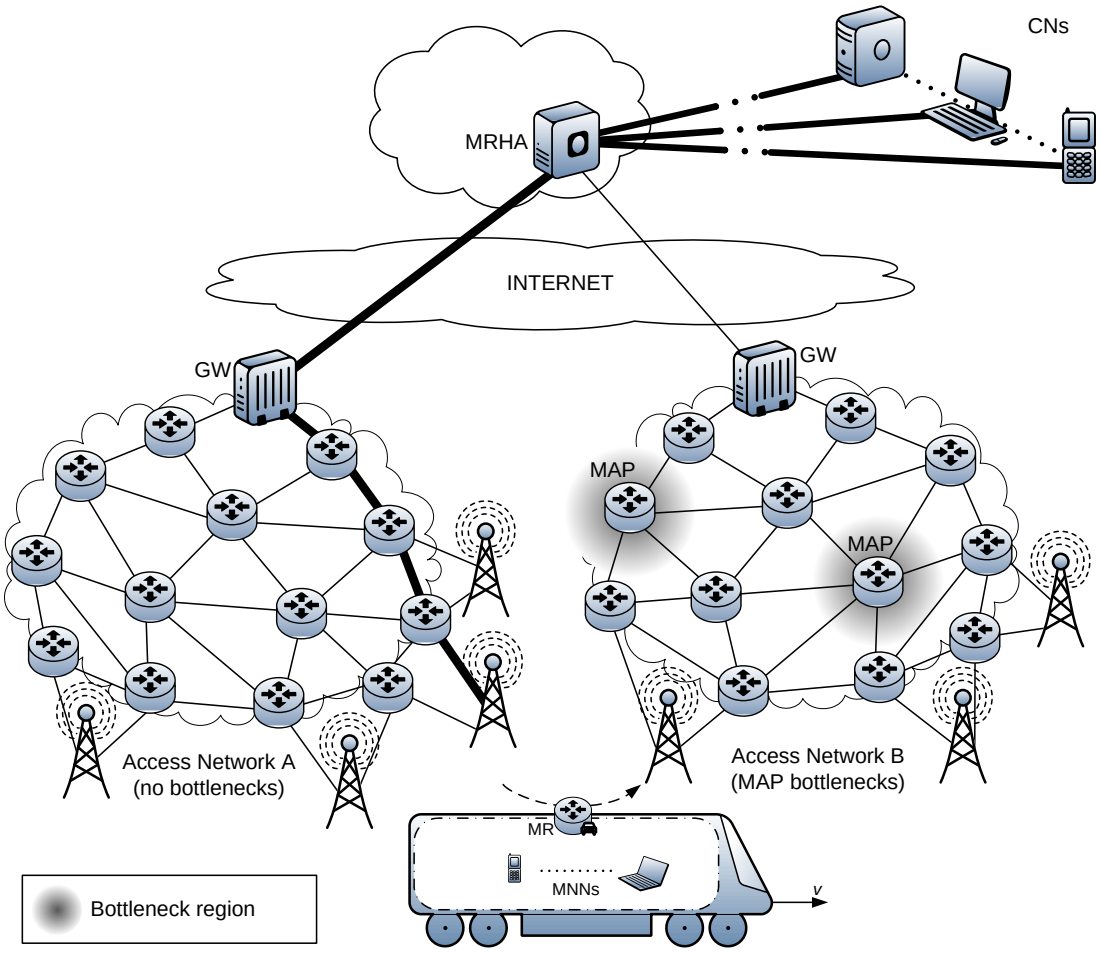


Figure 5.2: Scenario of an inter-AN handover.

protocol on the route along which traffic must flow. In essence, the NBS protocol is an extension of the MIPv6 and HMIPv6 protocols, introducing an additional tier of addressing to make mobility transparent to the MNN on the vehicle. As a result, all traffic destined to the moving network must first flow through the MRHA, before then being encapsulated and forwarded to the MAP, and onwards to the MR and then to the MNNs, all through a number of nested IP tunnels. The resultant effect is the possibility that no single MAP within the new network would have the capacity to support the entire resource requirements of the moving network when the MR performs a handover, as illustrated in Figure 5.2. This will lead to the target MAP having to reject the resource request of the moving network, causing the MR to either:-

1. blindly seek alternative target MAPs with which to make a resource reservation, or
2. negotiate a lower resource reservation with the MAP, with the intention of either dropping the sessions of some users, or proportionally reducing the QoS provided to the sessions of all users.

In both of these situations, handover latency will be significantly increased as the MR attempts to establish a binding with another MAP that can accommodate the resource requirements of the moving network, which itself may not even have the resources available to support the requirements of the moving network. In turn, this leads to a degradation of QoS, as increased handover latency will inevitably lead to increased packet loss and delay. The next section therefore details the approach that was taken to combine mobility and QoS signalling to minimise the handover delays and packet losses caused by the traditionally sequential and independent phases of mobility and QoS establishment.

5.3 QoS-Enabled Handover Mechanism

The proposed QENEMO mechanism aims to facilitate the establishment of QoS and mobility states of a moving network by combining the signalling procedures of each using a single, co-operative approach. Therefore, if a single MAP is unable to support the aggregate resource requirements of a moving network, QoS-enabled data paths can be efficiently set-up across multiple MAPs with minimal disruption to running sessions. This also avoids the need for sessions to be dropped or for resources allocated to established sessions to be reduced as a result of insufficient resources at any one MAP.

The architecture assumed for this work is an AN containing two or more special entities known as Enhanced Nodes (ENs) [65] that subsume the functionalities of MAPs. To support the aims of the QENEMO mechanism, the ENs also carry out additional functionality to facilitate the handover of the MR. These shall be elab-

orated on later in this section. The network itself is assumed to use a generic QoS protocol that provides support for both proxy-based receiver- and sender-initiated resource reservations. However, to this end, an alternative mechanism is also proposed in Section 5.3.4 for situations in which only a conventional QoS protocol such as IntServ is supported by the network. NB. Section 5.5 proposes an example implementation of the handover mechanism using the NSIS protocol framework [42], including the content and format of the protocol messages.

The main contributions of the QENEMO mechanism are embodied within two main scenarios: inter- and intra-AN handovers. In both scenarios, both uplink and downlink traffic is selectively distributed at the MR and MRHA, respectively, across the different established paths in the network. The decision of which packets to send on each QoS-path as QoS and handover states are being established is carried out by the specific splitting algorithm installed at the MR and MRHA, which is discussed later in this chapter. For scenarios in which it is not possible to accommodate the entire resource requirements of the moving network across the available ENs, a fallback mechanism is proposed in Section 5.3.3 which allows for any "excess traffic" to be sent across alternative non-EN paths through the network. For both cases, only the downlink reservation toward the moving network is considered.

5.3.1 Inter-AN Handover Mechanism

Upon entering the coverage of a new AR (be it initial registration or handover), the MR will communicate with the AR to configure an LCoA. This will enable it to receive or solicit an RA from the AR, which will contain information about the ENs available in the AN. The MR will then select an EN with which to register and form (in a stateless manner) an RCoA based on the IP address prefix of the selected EN. The signalling procedure thereafter is specific to QENEMO, and is shown in Figure 5.3 for a generic (non-protocol-specific) scenario.

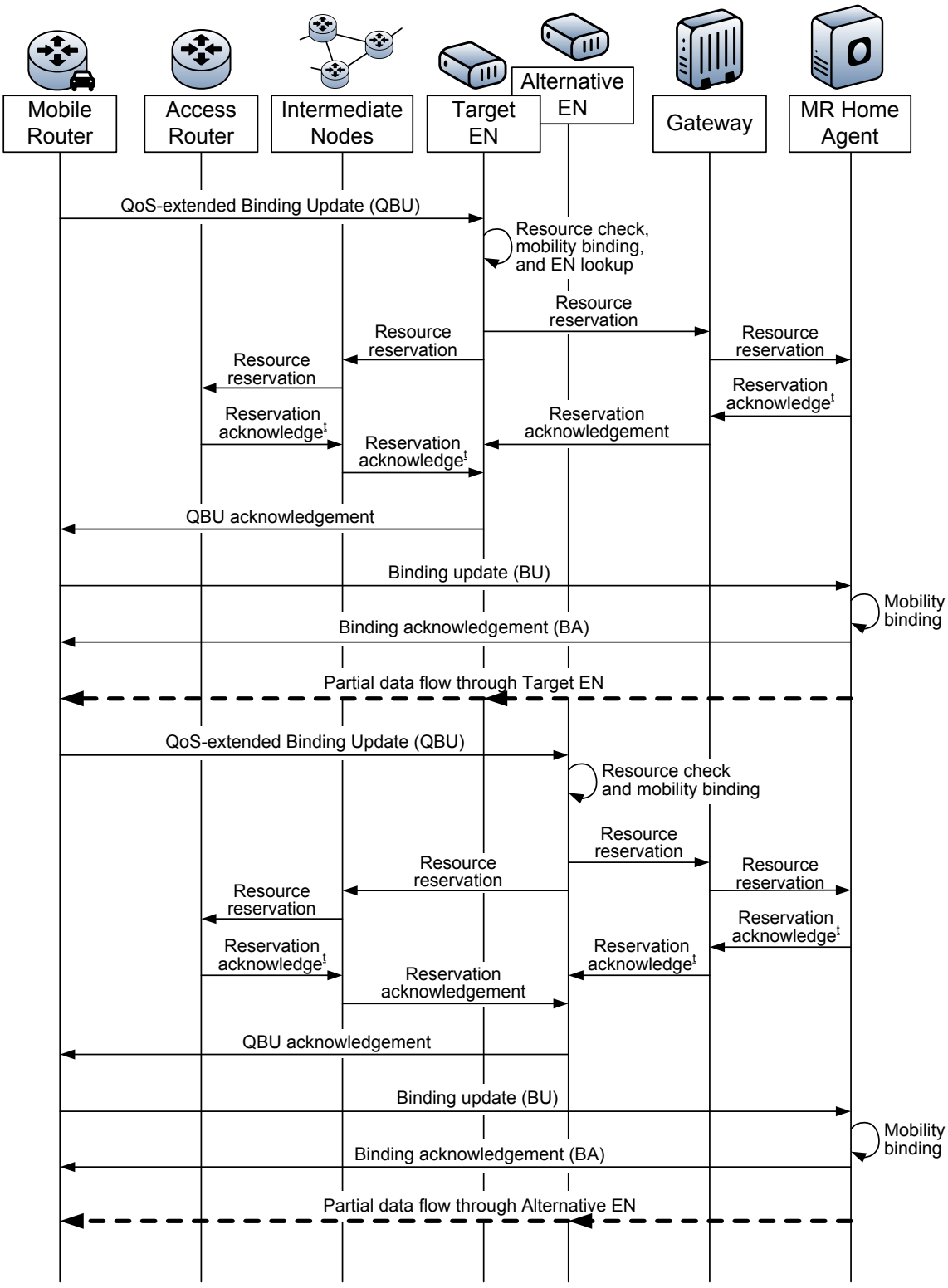


Figure 5.3: The inter-AN handover procedure of the QENEMO mechanism.

After forming an RCoA, the MR sends a QoS-Extended Binding Update (QBU) message to the EN, which contains information about the aggregate resource requirements of the moving network, and the IP addresses it wishes to bind (the LCoA and RCoA). The QoS-related information carried by the QBU is based upon the information collected by the MR from the QoS requests received from the individual terminals on the vehicle, and may include parameters such as requested bandwidth, delay and jitter, for a set number of aggregate traffic classes, and for both the uplink and downlink. The QoS-related content of a QBU message is discussed in greater detail in Section 5.4.1.

Upon receipt of the QBU message, the EN will first perform a check of its available resources, and make a decision as to how much of this it would be able to allocate to the MR. If the EN is at least able to fulfil a portion of the MR's requirements, it will proceed to create a mobility binding in its cache, and will pro-actively look up other ENs from its internally maintained table that would be able to fulfil the remaining requirements. At the same time, the EN will perform a stateful resource reservation for the resources that it *can* fulfil. This is carried out along two "legs": one from the EN up to the MRHA, and the other from the EN down to the AR to which the MR is connected. Since the reservation is essentially carried out by a proxy the downlink EN-AR path will be a sender-initiated reservation, and the EN-MRHA path, a receiver-initiated one[1].

Once the resource reservations along both legs have been acknowledged, the EN will send a QBU acknowledgement to the MR, which will give information about:

- The RCoA assigned to the MR (which may be different to the one contained in the original QBU message, due to duplicate address detection at the EN);
- The amount of resources reserved for the MR by the EN, and;

[1]In order for the reservation messages that the EN sends to the MRHA to be acknowledged successfully through the same path along which it was originally sent, the EN must first establish routing states between it and the MRHA, using a QoS-based routing protocol such as QOSPF [66]

- The amount of resources available at alternative EN(s) with which the MR may establish a QoS path.

Having been assigned an RCoA, the MR will send a binding update to the MRHA in the usual manner. Once the MR receives acknowledgement of the binding from the MRHA, a proportion of the packets will begin to flow to the MR through the EN. However, in order to ensure enough resources to meet those required by the moving network as a whole, the MR will send a QBU to each suggested alternative EN in turn. The amount of resources requested by the MR from the alternative ENs will be that which has not yet been allocated. After each new EN registration, the MR will send a BU to the MRHA, containing all RCoAs that have so far been allocated to it.

5.3.2 Intra-AN Handover Mechanism

When an MR performs a handover between ARs served by the same EN, it need only send a reduced QBU, containing only the information pertaining to its new LCoA. When the EN receives this, it will carry out a resource reservation across the new path, and tear-down reserved resources along the old path.

As an MR roams across the coverage of an AN, the data paths between the serving ENs and the MR may become too long to support the QoS requirements of the moving network. Additionally, local handovers will take longer to perform, due to the greater number of hops for which handover signalling must traverse and QoS states installed. Therefore, if alternative ENs are available that are topologically closer to the MR, the MR may, at any time, decide to handover sessions from one EN to another within the same AN, in order to ensure the continued fulfilment of the QoS requirements of those sessions. This is achieved in a similar way to that of the inter-AN handover.

Figure 5.4 shows a handover scenario in which the MR performs a handover from EN_1 to EN_3, while maintaining its association with EN_2. With reference to this scenario,

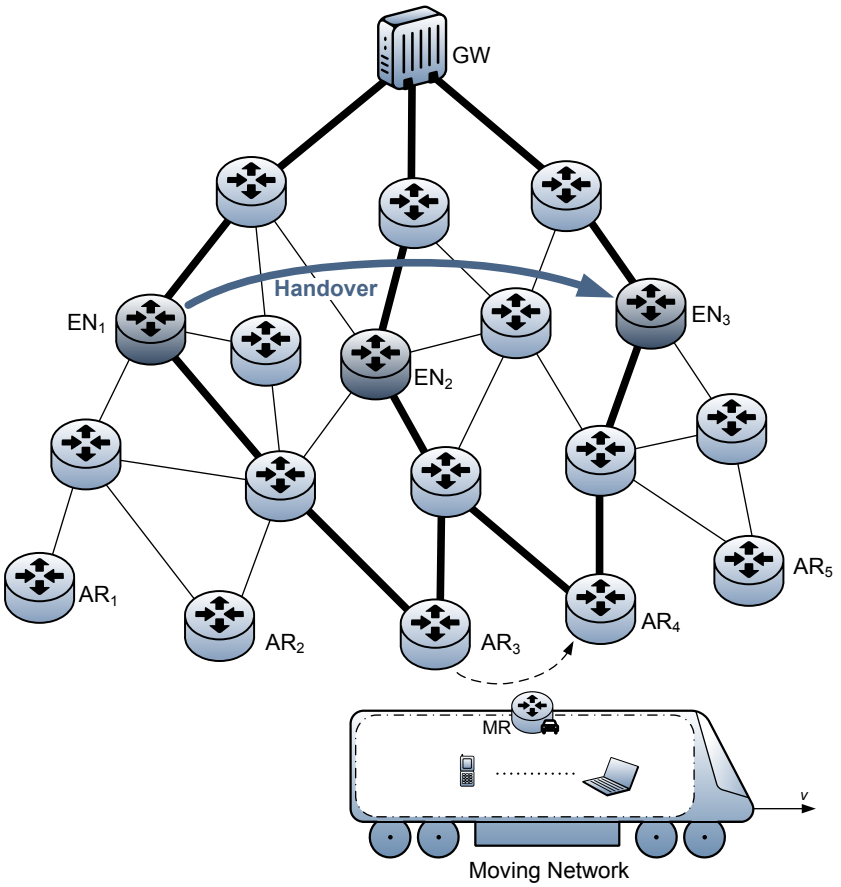


Figure 5.4: Intra-AN handover of a moving network from EN_1 to EN_3.

the MR first sends a QBU to the new EN, (EN_3), which will use the information contained in it to reserve resources on behalf of the MR subject to its available capacity. After EN_3 has acknowledged the resource reservation establishment, the MR will then send a BU to the MRHA. This will contain the RCoA obtained from EN_3, as well as its existing RCoA from EN_2: the absence of the RCoA from EN_1 will act as an implicit trigger to tear down its binding at the MRHA.

5.3.3 Fall-Back Mechanism

In cases where it is possible to communicate with only a single EN from the current AR, or if other ENs do not have sufficient resources available, the MR may establish other QoS-enabled paths directly with the MRHA (using the NEMO basic support protocol) without traversing an EN. However, utilisation of non-HMIPv6 paths will result in a greater AR-to-AR handover latency, as the binding update from the MR

must propagate up to the MRHA for every AR handover. Therefore, use of such paths should preferably be reserved for delay-tolerant applications, such as e-mail and file transfer.

Once an MR has determined to establish a path directly with the MRHA, it will first reserve resources up to the MRHA. Upon confirmation of the reservation, the MR will send a binding update message to the MRHA to bind its HoA prefix to its CoA. QoS forwarding states may then be set up by the MR along the newly established path.

5.3.4 Alternative MR-Controlled Mechanism

An alternative to the aforementioned network-controlled handover method is an MR-controlled method, which removes the restriction to use a QoS protocol supporting both sender- and receiver-initiated reservation procedures. Overall, this method is similar to the network-controlled method, except that QoS reservations are made by the MR instead of the EN.

An example of the MR-controlled method is shown in Figure 5.5. Immediately after receiving the QBU and performing the necessary resource checks and mobility bindings, the EN replies to the MR with a QBU acknowledgement, indicating the resources it can accommodate, as well as information about alternative EN(s) that can meet its remaining resource requirements. The MR will then reserve resources up to the MRHA through the first target EN, and then send a binding update to the MRHA. The remaining bindings and reservations carried out with the alternative suggested ENs follow the same procedure as with the first EN; only the first phase is shown in Figure 5.5, as the second is essentially identical to the first.

5.4 Supporting Functionalities

This section details some of the supporting functionalities that are common to the mechanisms defined in Section 5.3.

5.4.1 QoS Profiling

The resource reservation information contained in a QBU message is constructed based on the information collected by the MR from the QoS requests of individual terminals on the vehicle. Figure 5.6 shows the interaction between the pre-session and in-session QoS provisioning components inside the MR. When QoS messages from individual users arrive at the ingress interface of the MR, the parameters of the QoS request are parsed into a generic QoS message, and sent to the appropriate QoS aggregation queue. Each time an aggregate QoS message is sent from the MR, the aggregation server updates the In-Session QoS Database with details of each individual flow that was aggregated. Upon handover, the QoS Profiler will analyse

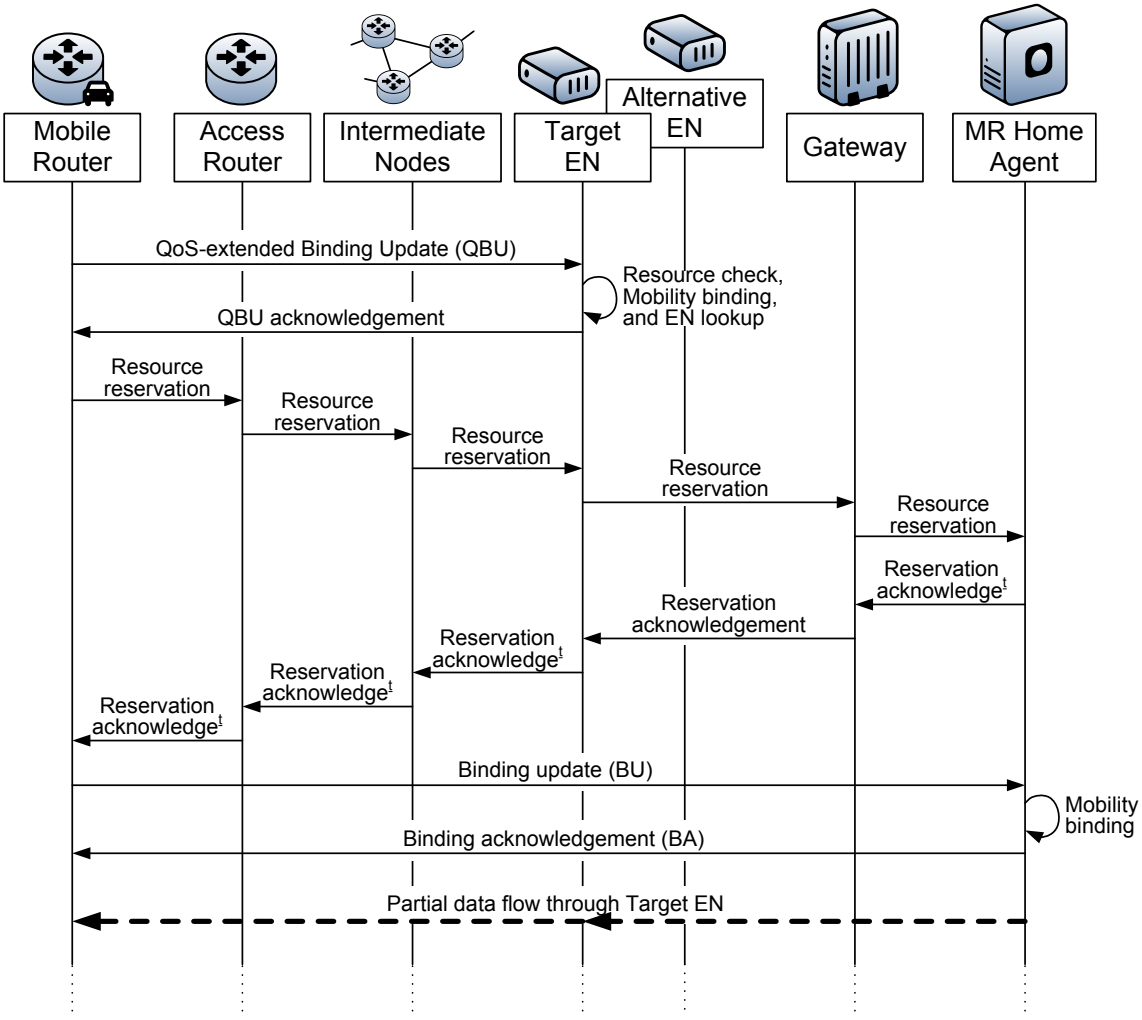


Figure 5.5: An alternative network-assisted, MR-controlled handover procedure.

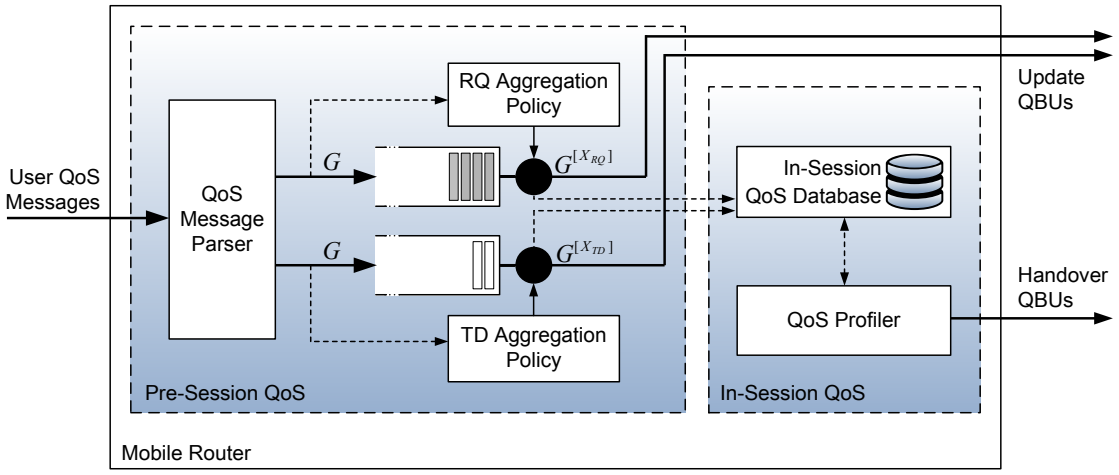


Figure 5.6: Profiling of QoS Messages inside the MR

the entries in the In-Session QoS Database, and use it to determine the optimal number of traffic classes for which to reserve resources along the new path in the network upon handover. Each traffic class will be represented by a QoS profile, which contains the aggregate reservation information (such as total throughput and maximum delay and jitter) for that traffic class. These QoS profiles are then placed within a QBU message together with the mobility information, and sent to the EN to be processed.

The handling of the QoS profiles at the EN will be dependent on the QoS architecture deployed in the network. If the QoS architecture supports NEMOR-style reservations (DiffServ queues inside a virtual IntServ tunnel), then the EN will aggregate the requirements of each QoS profile according to what it can accommodate, and initiate the EN-AR and EN-MRHA reservations accordingly. If the network supports only IntServ-style reservations, then the EN will send a resource reservation for each QoS profile, adjusted to the amount of resources that the EN can accommodate.

Figure 5.6 differentiates between two types of QBU messages: Update QBU messages and Handover QBU messages. Update QBU messages are used to either increase or decrease the amount of resources allocated to one or more traffic classes for the new sessions that have been admitted. As this message is updating only existing resource reservations along paths for which a CoA binding already exists, the message need

not contain any handover-related information such as BU messages. On the other hand, Handover QBU messages are used to reserve resources upon a handover (both inter- and intra-AN) for entire blocks of resources, and are therefore required to contain the appropriate handover messages. In order to ensure that reservations are not overloaded by extraneous flows (particularly when only a portion of the flows have been accommodated within a new network), both the Update QBU and Handover QBU must contain the *number* of flows being admitted, in addition to the reservation parameters of each QoS profile to allow the MRHA to split the MR-destined traffic with the correct ratio.

5.4.2 EN Resource Information Exchange

Each EN should, as far as is practical, contain up-to-date information about neighbouring ENs, to allow it to suggest alternative ENs to the MR when it cannot meet the aggregate QoS requirements of the moving network. To facilitate the exchange of such information, one possibility is for a Bandwidth Broker (BB) to be located in each access network which acts as a common point for all ENs in the access network to update their resource availability, as shown in Figure 5.7. ENs must ensure that the BB is kept up-to-date about their resource availability. This may be done in response to an event which, for example, caused the resource availability of a par-

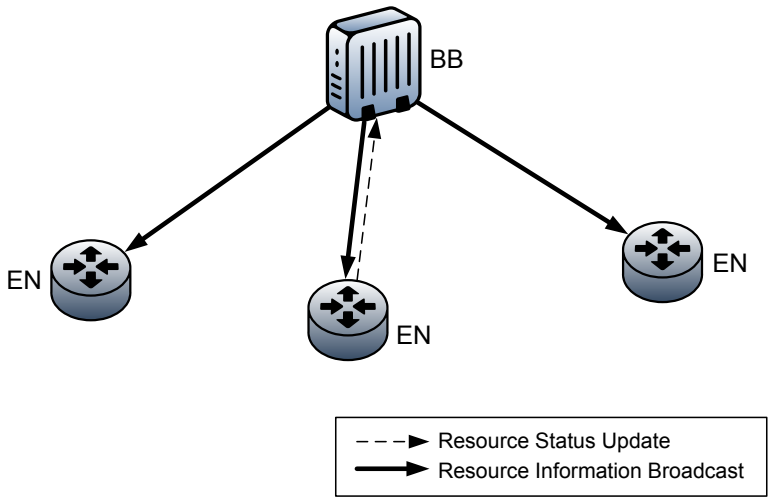


Figure 5.7: Bandwidth broker approach to maintaining up-to-date resource information amongst ENs.

ticular EN to increase or decrease by more than a certain percentage. The BB will then, either periodically or otherwise, broadcast EN resource information to all ENs within the AN, to enable each to make decisions about alternative ENs. Existing protocols such as the Simple Network Management Protocol (SNMP) [67] can be used to facilitate the exchange of such information between the BB and ENs. An alternative is for the AN to operate an intra-domain link-state routing algorithm such as that used by QoS Open Shortest Path First (QOSPF), whereby each router in the AN builds and stores a routing table with the link cost to every other destination within the administrative domain of the network. In this way, each EN will be able to make an admission decision when a QBU message is received from an MR based on the routing table information.

5.4.3 Selective Traffic Splitting Algorithm

When the EN launches a resource reservation in response to its receipt of a QBU message, the reservation north of the EN must reach all the way to the MRHA, so as to provide it with knowledge of the proportion of packets that need to be sent across a given path. Even if intermediate routers *en route* to the MRHA do not provide QoS support, such routers should transparently forward the QoS message to the next hop toward the MRHA. Thus, the mechanism of splitting traffic across different established paths is done in accordance with the resources reserved across each. Similarly for uplink traffic, the MR must split traffic according to the amount of resources confirmed by the QBU acknowledgement message.

The decision of which traffic to send over a particular path is determined by a splitting algorithm that runs at both the MRHA and MR; the so-called *splitting nodes*. When the resource requirements of the moving network are only partially fulfilled, the splitting algorithm plays a particularly important role, as it must decide which traffic to prioritise while other QoS-enabled paths are being established for the remaining resources. Once a flow has been assigned to a particular path, the splitting nodes should (independently) maintain a QoS binding cache recording the

association of particular flows to an RCoA or LCoA, depending on whether HMIPv6 or MIPv6, respectively, is being used along that path. This ensures that a QoS path is not being used to carry more flows than it can accommodate. It also avoids a situation in which packets belonging to the same flow are split across multiple paths (once other paths have become established), which could lead to packets arriving out-of-order at the destination.

5.4.4 Tear-Down Procedure

When an MR no longer requires the services of a particular EN, the resources allocated to the moving network both at the EN and at associated intermediate routers must be torn-down to allow them to be used for other potential network users. This may be achieved by either letting the resource reservation time out, or by explicitly signalling to the EN to tear down the resources immediately. The recommendation given by Chaskar [68] is the latter option as it ensures minimum resource wastage and can save money for the PTV operator in cases when the reservation is associated with an accounting record. The problem of not getting the chance to send an explicit tear-down message because of a loss of link-layer connectivity with the old router is not an issue with QENEMO, as the tear-down message generated by the MR need not travel along the route of the reservation; as long as it is able to reach the EN, the EN will handle the actual tear-down procedure.

5.5 NSIS Implementation Considerations

The mechanism presented in the previous sections was described only generically, without giving specific details of the way in which the messages are constructed and processed, nor the mechanism by which they are transported between network nodes. This section aims to fill some of these gaps by considering the way in which QENEMO may be practically implemented within the IETF NSIS protocol framework. The choice of using the NSIS framework to realise the QENEMO mechanism

was influenced by its support for proxy-based QoS reservations as well as its general extensibility, allowing for custom NSLP implementations where required.

The following subsections outline the NSIS implementation of the various components of QENEMO. Consideration is first given to the aspects relating to the generation and processing of the QBU message in both the initiation and acknowledgement phases. Following this, Section 5.5.2 discusses some implementation considerations for the resource reservation procedures made between the EN and both the AR and MRHA. Finally, Section 5.5.3 touches upon some of the security issues that would need to be taken into account in any future realisation of the mechanism.

5.5.1 QoS-Extended Binding Update

Two approaches can be taken in the implementation of the QBU signalling mechanism between the MR and EN. The first approach is to extend the QoS NSLP with mobility functionality, allowing for a local BU object to either be added to the NSLP message, or to be embedded within the QSPEC object [44]. However, this may somewhat convolute the scope and aims of the QoS NSLP, which Manner *et al.* [44] define as being to *establish and maintain state at nodes along the path of a data flow for the purpose of providing some forwarding resources for that flow.* Furthermore, the information contained in a QBU and QBU-acknowledgement is required to manipulate state in only the receiver and initiator of the message, respectively, and not the intermediate nodes between them. Therefore, a more favourable approach would be to define a new NSLP for the purpose of carrying both mobility and QoS signalling directly to and from a proxy node—in this case the EN—and to act as an interface to the IP mobility and QoS NSLP daemons to carry out their respective functions. This subsection therefore proposes a basic framework with which such an NSLP, herein referred to as simply the "QBU NSLP," can be implemented using NSIS, based on the guidelines set out in [43] and [69].

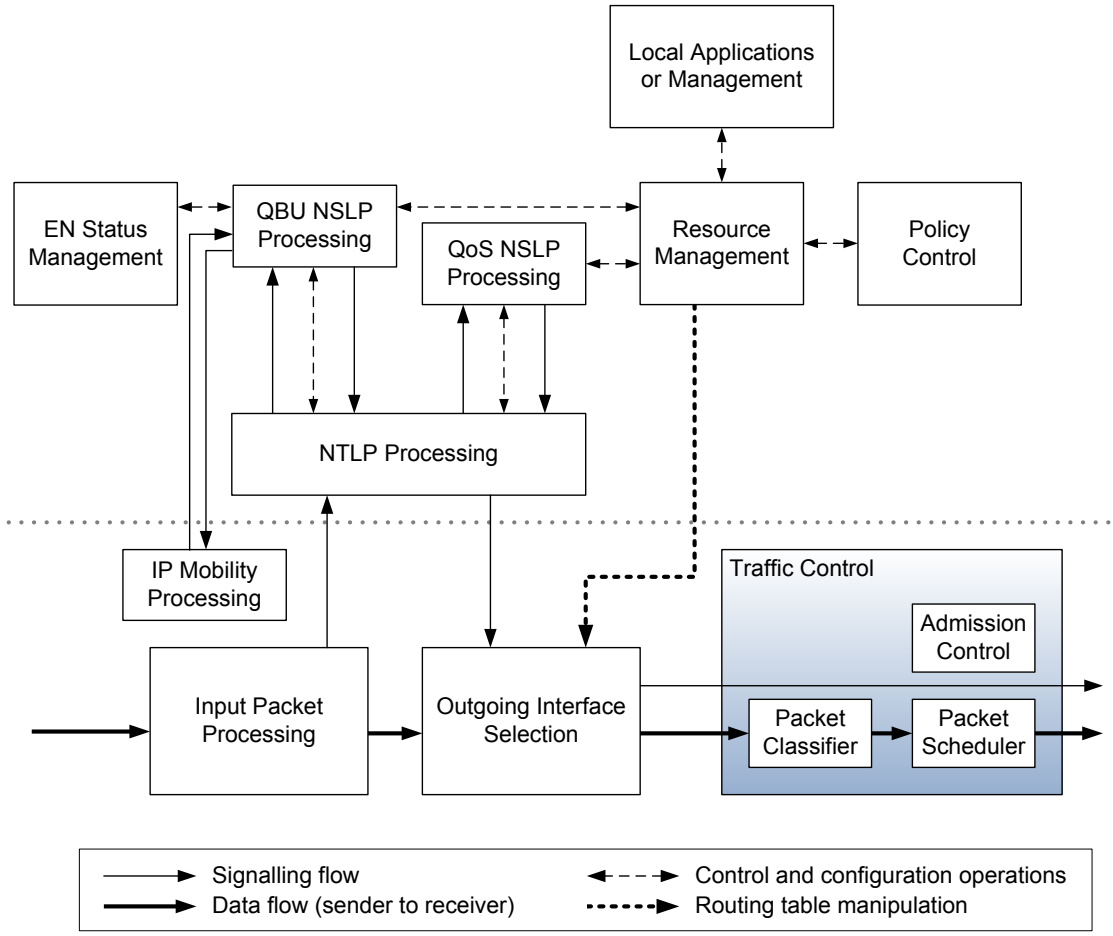


Figure 5.8: Combined QBU and QoS NSLP architecture in a node (present in the MR and ENs)

5.5.1.1 QBU NSLP Architecture

The combined node architecture of the QBU NSLP and QoS NSLP is shown in Figure 5.8, which is an extension of that given in [44] for just the QoS NSLP case.

When a QBU NSLP is generated at the MR, it is passed to the NTLP layer to send directly to the EN that has initially been selected. When the QBU message arrives at the IP layer of the EN, it will be passed to the NTLP layer, which will in turn read the NSLPID contained in the header, and accordingly pass it to the QBU NSLP. The QBU NSLP will first communicate with the Resource Management entity to determine whether it is able to meet at least a portion of the MR's resource requirements. This will lead to either a positive acknowledgement with information

about how much it can accommodate, or a negative acknowledgement indicating that it cannot meet any of the MR's resource requirements.

If the QBU NSLP receives a positive acknowledgement from the Resource Management entity, it will send the HMIPv6 BU message contained within the QBU message to the IP Mobility daemon, which will perform Duplicate Address Detection (DAD) on the suggested RCoA. This will lead to either a positive acknowledgement informing the QBU NSLP of the RCoA it has bound to the MR's LCoA, or a negative acknowledgement indicating that it has insufficient resources to provide mobility support for the MR (irrespective of the amount of network resources the MR is requesting).

Positive acknowledgements from both the Resource Management and IP Mobility entities will trigger the QBU NSLP to send two RESERVE messages to the QoS NSLP to carry out resource reservations along both the EN-AR leg and the EN-MRHA leg. This is done using the standard feature set of the QoS NSLP, but nevertheless some guidelines relating to this are given in Section 5.5.2.

The EN will contain a module called the "EN Status Management" entity, which contains up-to-date information about the resources available at other ENs in the access network. If the EN cannot fulfil any portion of the MR's requirements, the QBU NSLP in the EN communicates with this entity to obtain information to send back to the MR. This information will be placed within a QBU acknowledgement message, along with confirmation of the resources that *have* been reserved, as well as mobility-related information required by the MR to complete the establishment of mobility states.

5.5.1.2 QBU NSLP Message Format

The QBU NSLP requires two types of messages to help fulfil the goals of QENEMO: a QBU-RESERVE and a QBU-RESPONSE message. Since the QBU NSLP will ultimately be used to interact with the QoS NSLP, the objects contained in each

will only *build* on the objects of existing QoS NSLP messages, namely, the RESERVE and RESPONSE messages, respectively.

QBU-RESERVE Message

The QBU-RESERVE message will contain two non-compulsory objects in addition to the QoS NSLP RESERVE object: a MOBILITY_BU object and an EN_IGNORE object. The MOBILITY_BU is a variable-length object containing a HMIPv6 BU message, as defined in [24]. This object is needed only when the MR wishes to create or update a mobility binding at the EN. If the MR wishes only to modify an existing QoS reservation, the MOBILITY_BU object may be omitted. When the MR sends a QBU-RESERVE message to update a mobility binding at the EN without wishing to change the reserved resources, the QSPEC object may be omitted, such that the EN uses its existing knowledge of the reservation to reserve resources along the new mobility path.

The EN_IGNORE object is used by the MR to convey information to the EN about the IPv6 addresses of other ENs to ignore when the EN suggests alternatives to the MR. This is used when, for example, the MR has just communicated with a particular EN that fulfilled only a portion of its requirements or was unable to process its request for whatever reason. This object may be included only when the MOBILITY_BU is present, and the length of the object will be a multiple of 128-bits.

QBU-RESPONSE Message

The QBU-RESPONSE is based on the RESPONSE message of the QoS NSLP, with the addition of two non-compulsory objects: a MOBILITY_BA message and an ENSPEC message. The former is a variable-length object containing a standard HMIPv6 BA message generated by the IP Mobility Processing entity confirming the success or failure to bind an LCoA to an RCoA. This object is included only if a MOBILITY_BU message was included in the prior QBU-RESERVE message.

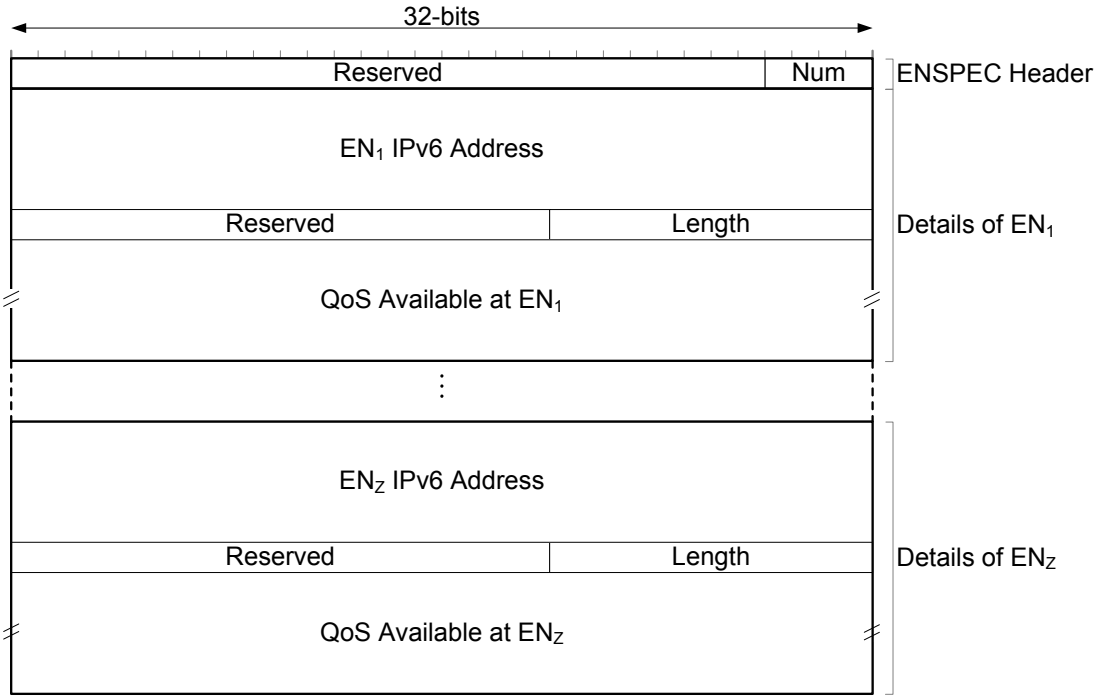


Figure 5.9: Format of the ENSPEC object contained within the QBU-RESPONSE message.

The ENSPEC message is another variable-length object containing the IPv6 addresses of alternative ENs and the resources available at each. The proposed format of the ENSPEC object is shown in Figure 5.9. The ENSPEC object has a 32-bit header with the four least significant bits of the header indicating the number of ENs that have been suggested, and thus the number of sub-objects present beneath the header. Each sub-object consists of a 128-bit field indicating the IPv6 address of the suggested EN, a field carrying parameters relating to the resources of that EN (termed, "Mini QSPEC"), and finally, sandwiched between the two is a header indicating the length of the subsequent Mini QSPEC. The Mini QSPEC is optional, and may not be included if the EN cannot accurately determine the resource availability of that alternative EN. However, the header to the Mini QSPEC is compulsory, as it must indicate, through its "Length" field, that there is no Mini QSPEC present.

The Mini QSPEC is so-named as it should follow the general template of the QSPEC parameters given in the original QSPEC object, however, it need specify only the

"QoS Available," and not the "QoS Desired," the "QoS Reserved," nor the "Min QoS."

5.5.2 Proxy-Based Resource Reservation

The resource reservation between the EN and both the AR and MRHA carried out on behalf of the MR can be achieved using the standard features of the NSIS QoS NSLP [44]. To allow the EN to reserve resources on behalf of the MR, it must be designated as a Proxy-QoS-NSLP NSIS Entity (QNE) so as to become the QoS-NSLP NSIS Initiator (QNI) of the reservation, with the AR and MRHA both assuming the role of QoS-NSLP NSIS Receiver (QNR).

The RESERVE message sent by the EN towards both the MRHA and MR must contain the following information within the respective QoS NSLP elements:

- **RII Object**

  Required to obtain a RESPONSE to the RESERVE message.

- **REFRESH_PERIOD Object**

  While this value may be omitted (leading to default value of 30 seconds), it would be prudent to set it based on the dynamics of the moving network, for instance, a proportion of the expected cell-residence time.

- **BOUND_SESSION_ID Object**

  This object must contain a unique, cryptographically random Session ID that makes it possible to alter the flow identification (which may change as a result of a handover) of existing reservation states so as to avoid the need to have to install a completely new reservation along a common path.

- **QSPEC Object**

  The parameters of the resource reservation to be established are to be contained within a QSPEC object [70], the structure and content of which will depend on the reservation model being used (e.g. IntServ reservations for each

QoS Profile, DiffServ queues within an IntServ reservation, *etc*). In any case, the QSPEC must convey to the MRHA[1] the number of flows for which the reservation is being made, so as to avoid overloading a QoS-path.

The NTLP element must contain a Message Routing Information (MRI) object [43] with the destination IP address set as the MRHA's or the AR's IP address. The source IP address should be set to the RCoA allocated to the MR, rather than the EN's own IP address, so as to enable the MRHA to associate the resource reservation with the BU message that the MR will send to it once the MR receives a QBU acknowledgement.

5.5.3 Security Considerations

The majority of security problems relating to the QENEMO protocol are addressed by the built-in security mechanisms of the NSIS protocol suite as a whole, specified in [71], and by the specific security features of the QoS NSLP mentioned in [44]. However, an issue that may affect the integrity of the QENEMO protocol is the proxy-reservations made by the EN to the MRHA. In particular, the MRHA would be required to ensure the authenticity of the reservation being made by the EN. One possible way of achieving this would be for the MR to provide a key within the QBU-RESERVE message, which the EN may include within the RESERVE message it sends to the MRHA. Another layer of security is also provided in the fact that the MR must send an NBS BU message directly to the MRHA, which will act as an implicit confirmation of the reservation made by the EN (using the MR's RCoA).

There may well be other security issues that may need to be taken into account in the design of the QENEMO protocol under NSIS, but security in general lies outside of the scope of this thesis.

[1] Information about the number of flows for a given reservation state must also be communicated to the MR, but this is achieved with the QBU acknowledgement, once QoS states across both legs of the communication path have been established.

5.6 Discussion and Summary

The benefits of micro-mobility protocols in reducing the handover latency of individual terminals and maintaining QoS have been made evident in numerous past studies. However, micro-mobility protocols have also been known to be susceptible to bottleneck congestion forming around the mobility agents that they rely upon. This makes it difficult to use such protocols to provide support to moving networks, as the high density of traffic that moving networks typically handle would lead to the mobility agent having to deny service to the moving network, or provide it with support for only a portion of its traffic requirements through lengthy resource negotiations.

To solve these problems, this chapter has proposed a novel mechanism called QENEMO which allows a moving network to exploit the mobility and QoS benefits of micro-mobility without the need for lengthy and inefficient signalling exchanges to provide allocation of resources within the network. The QENEMO mechanism is based on the use of ENs which contain both QoS and mobility functionalities, as well as knowledge of the traffic load across the access network. Together, these functionalities are able to facilitate in the seamless and QoS-enabled handover of a moving network by reserving resources on its behalf for the resources that it can accommodate, and providing information about alternative ENs that can fulfil its remaining requirements. In addition to the generically proposed mechanism, a possible implementation of QENEMO within the NSIS protocol framework was also detailed. This was based on the definition of a new *QBU* NSLP which builds upon and interacts with the QoS NSLP to provide both micro-mobility and proxy-based QoS support to the MR.

While the QENEMO mechanism can significantly reduce the extent to which moving network sessions are disrupted during handover, the use of micro-mobility will come at the cost of increased tunnelling overheads that can lower the utilisation of the network. This is seen as a necessary compromise, although its effect may be reduced

somewhat by employing a Dynamic Route Optimisation (DRO) mechanism such as that proposed by Pragad *et al.* [72] which dynamically adjusts the ratio of traffic receiving micro-mobility support based on the level of congestion in the network. In spite of the issue of higher tunnelling overheads, the QENEMO mechanism is able to offer an advantage in improving network efficiency through its ability to tear-down resources across old paths, even with loss of link-layer connectivity to the AR through which the reservation was initially made.

Future work on this area should entail concept-validation of the QENEMO mechanism, using, for example, the NSIS-ka software implementation [73]. This will also allow for further refinement to the mechanism, and identification of scenarios beyond those identified in this chapter.

Chapter 6

Conclusion and Future Research

6.1 Conclusion

This thesis has proposed a number of techniques for improving the efficiency of QoS provisioning in moving networks for both pre-session and in-session scenarios. Beginning with the former, Chapter 3 presented a study of existing QoS aggregation policies used to control the aggregation of users' QoS messages in order to reduce the frequency with which the resources allocated to a moving network are adjusted. However, it was shown that the parameters upon which these policies are based are dependent on the rate at which user requests arrive at the MR, making them unsuitable for the bursty request environment typically associated with frequent-stopping PTVs. Therefore in order to overcome this problem, a cost-driven QoS aggregation policy was proposed in which the decision to aggregate is based on a ratio of costs, a parameter that was proven to be practically independent of the rate of requests. It was shown through both mathematical analysis and computer-based simulation that the proposed cost-driven policy reduces the overall cost to the operator (and hence increases operational efficiency) in relation to previous aggregation policies without significantly impacting the time for which a user must wait before being granted QoS-enabled session connectivity.

In the subsequent part of the study of QoS aggregation, the assumption made in the prior work that the availability of network resources is always greater than user demand was relaxed. This led to the problem of reduced QoS provisioning efficiency when a network becomes congested, whereby the number of QoS request messages that can be aggregated is limited to what the network can accommodate, rather than to that determined by the policy itself. In addition, causing users to wait for prolonged periods while network resources become available can lead to users reneging on their request, which introduces yet another cost. Therefore, it was recognised that a separate "overlay" policy is required to ensure cost-efficiency when the network is congested. Accordingly, two such policies were proposed; a dynamic policy (D-policy) which extends the principle of cost-optimality introduced by the cost-driven policy through a modified form of the aggregation-utility function, and a static policy (S-policy) which optimises costs based on a fixed lower congestion-threshold parameter. It was found that the S-policy performed significantly better than the D-policy in reducing costs over the no-overlay-policy case, which was due primarily to the fact that the S-policy works on attaining global optimality, whereas the D-policy, only local optimality.

In undertaking this study of QoS aggregation, it was found that the subjectivity of the area made it difficult to propose hard and fast rules that could be transferred to any operating environment. For example, whilst some of the costs used in the study had a clear-cut physical and objective meaning, other costs such as that of holding QoS requests at the MR are perhaps more open to scrutiny, since it intrinsically assumes that the user will be making use of the data services for the duration of his/her journey. However, on a holistic level, the melding of somewhat subjective costs using an objective common denominator has provided a firm foundation upon which future studies of QoS aggregation policies can build. Ultimately, the value of future studies in this area will depend on the quality of the input data used, which in turn must be obtained from more targeted research of user behaviour and dynamics of moving networks, involving real scenario measurements. In addition, further costs

can be introduced into the system to help achieve other specific objectives, such as reducing long-term energy consumption.

The final part of this thesis focussed attention on the issue of in-session QoS provisioning support, which involves the re-establishment of QoS states for large amounts of resources upon handover between access routers and networks. It was shown that micro-mobility protocols which are typically used to deliver seamless network connectivity to individual hosts can be a cause of QoS provisioning inefficiency when applied to moving networks. In addition, the bottleneck characteristics that micro-mobility protocols introduce can make it difficult to accommodate the entire resource requirements of the moving network along only a single path through the network, resulting in the need for blind resource negotiations during a particularly critical time. Therefore, a novel mechanism was proposed which allows for faster and more efficient QoS provisioning through multiple paths of the same AN, by allowing for tighter co-operation and trust between the MR and ENs. Following a generic description of the proposed mechanism, a possible implementation of the mechanism within the NSIS framework was also detailed. This was based on the definition of a new *QBU* NSLP which builds upon and interacts with the QoS NSLP to provide both micro-mobility and proxy-based QoS support to the MR. Future performance evaluations of the proposed mechanism will allow for any necessary refinements to be identified and applied, and for any security issues to be addressed.

6.2 Future Research

In addition to the future work mentioned in the summaries of prior chapters, the following subsections present a number of general problems in the area of QoS provisioning for moving networks that remain open for future exploration and research.

6.2.1 Protocol Efficiency Regulation Mechanism

The QoS aggregation policies proposed in this thesis make the assumption that the networks to which the MR connects belong to the same administrative domain, which inherently gives incentive to the MR to control QoS provisioning efficiency. However, since an MR has the potential to establish and maintain connections with networks controlled by administrative domains besides its own, the incentive to want to control signalling efficiency through aggregation diminishes. Therefore, in such cases, the networks (specifically, the ARs) must take the proactive role of limiting the rate at which an MR may signal to reserve resources. One possible scheme to achieve this may be to employ a token bucket scheme between the AR and MR, whereby the MR is allocated credits at a set rate (which are either communicated explicitly to the MR, or according to an agreed algorithm), of which the MR must manage to ensure its objectives (e.g. reneging probability and/or user satisfaction) are met. Therefore, a possible area of future research would be to study the way in which the MR can efficiently use its credits such that it meets certain performance objectives. Different credit allocation schemes should be studied, including different forms of both algorithm- and protocol-based techniques.

6.2.2 QoS-Enabled Mobility Management Extensions

The QoS-enabled mobility management mechanism proposed in Chapter 5 provides an effective framework for improving the efficiency of QoS provisioning for moving networks that handover between networks at high velocity. However, a number of extensions to the mechanism are envisaged requiring further research and development. One of these is the dynamic management of queues within each aggregate QoS reservation, such that will allow for finer granularity of QoS provisioning in accordance with the experienced traffic demand. One of the particular problems of a pure-DiffServ QoS model is its coarseness of provisioning, which necessitates a degree of over-provisioning in the network to ensure that users' QoS specifications can be met. Through aggregation (and hence isolation) of the flows belonging to a mov-

ing network, it becomes possible to dynamically manage the DiffServ queues that are established within an aggregation tunnel, without affecting other flows in the network. The flexibility of the NSIS protocol suite provides a good platform upon which such a mechanism can be developed. However, in realising such a framework, further research is required into the optimal mix of traffic along a particular aggregate tunnel, so as to ensure that over-provisioning is kept to a minimum without significantly reducing the scalability of operation due to increased queue management overheads.

References

[1] BBC and Broadreach Ltd. (2004) Wi-Fi may tempt train travellers. [Online]. Available: http://news.bbc.co.uk/1/hi/technology/3729583.stm 16

[2] D. Mulvey, "HSPA," *Communications Engineer*, vol. 5, no. 1, pp. 38–41, Feb. 2007. 16

[3] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," RFC 3963 (Proposed Standard), Internet Engineering Task Force, Jan. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc3963.txt 17, 28, 34

[4] M. Tlais, H. Labiod, and N. Boukhatem, "Resource reservation for NEMO networks," draft-tlais-nemo-resource-reservation-00.txt, work in progress, Nov. 2004. 17, 20, 43

[5] M. Tlais and H. Labiod, "Resource reservation for NEMO networks," in *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on*, vol. 1, 2005, pp. 232–237. 17, 20, 43

[6] Roch Guèrin's web page. http://www.seas.upenn.edu/ guerin/overview.html. 18

[7] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," RFC 1633 (Informational), Internet Engineering Task Force, June 1994. [Online]. Available: http://www.ietf.org/rfc/rfc1633.txt 18, 38

[8] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach (4th Edition)*. Addison Wesley, Mar. 2007. 18

[9] A. Abella, V. Friderikos, and H. Aghvami, "Differentiated services versus over-provisioned best-effort for pure-IP mobile networks," in *Mobile and Wireless Communications Network, 2002. 4th International Workshop on*, 2002, pp. 450–457. 18

[10] R. Hager, A. Klemets, G. Maguire, M. Smith, and F. Reichert, "MINT - A mobile Internet router," in *Vehicular Technology Conference, 1993. VTC1993. IEEE 43rd*, May 1993, pp. 318–321. 27

[11] T. Ernst, "Network mobility support in IPv6," Ph.D. thesis, Université Joseph Fourier, Oct. 2001. [Online]. Available: http://www.inria.fr/rrrt/tu-0714.html 27

[12] IETF MEXT working group. http://www.ietf.org/html.charters/mext-charter.html. 27

[13] IETF NEMO working group. http://tools.ietf.org/wg/nemo/. 27, 37

[14] C. Ng, T. Ernst, E. Paik, and M. Bagnulo, "Analysis of Multihoming in Network Mobility Support," RFC 4980 (Informational), Internet Engineering Task Force, Oct. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4980.txt 28

[15] C. Ng, P. Thubert, M. Watari, and F. Zhao, "Network Mobility Route Optimization Problem Statement," RFC 4888 (Informational), Internet Engineering Task Force, Jul 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4888.txt 28

[16] C. Ng, F. Zhao, M. Watari, and P. Thubert, "Network Mobility Route Optimization Solution Space Analysis," RFC 4889 (Informational), Internet Engineering Task Force, Jul 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4889.txt 28

[17] T. Ernst and H.-Y. Lach, "Network Mobility Support Terminology," RFC 4885 (Informational), Internet Engineering Task Force, Jul 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4885.txt 28

[18] Ambient Networks project web page. http://www.ambient-networks.org/. 29

[19] N. Niebert, R. Hancock, H. Flinck, H. Karl, and C. Prehofer, "Ambient Networks research for communication networks beyond 3G," in *IST Mobile Summit*, June 2004. 29

[20] R. A. Calvo, M. Miozzo, J. Eisl, D. Hollós, E. Hepworth, and L. Badia, "Routing groups in ambient networking," in *Mobility '06: Proceedings of the 3rd International Conference on Mobile Technology, Applications & Systems*. New York, NY, USA: ACM, 2006, pp. 63–68. 30

[21] J.-M. Bonnin and R. Ben Rayana, *Heterogeneous Networks and Mobile Management in Intelligent Transportation Systems*. Nova Science Publishers, 2009, ch. 15. 30

[22] IBBT Tr@ins project web page. http://www.ibbt.be/en/project/trains. 30

[23] C. Perkins, "IP Mobility Support for IPv4," RFC 3344 (Proposed Standard), Internet Engineering Task Force, Aug. 2002, updated by RFC 4721. [Online]. Available: http://www.ietf.org/rfc/rfc3344.txt 31

[24] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," RFC 3775 (Proposed Standard), Internet Engineering Task Force, June 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3775.txt 31, 112

[25] I. Akyildiz, J. Xie, and S. Mohanty, "A survey of mobility management in next-generation all-IP-based wireless systems," *Wireless Communications, IEEE*, vol. 11, no. 4, pp. 16–28, Aug. 2004. 31

[26] R. Koodli, "Mobile IPv6 Fast Handovers," RFC 5568 (Proposed Standard), Internet Engineering Task Force, Jul 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5568.txt 32

[27] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier, "Hierarchical Mobile IPv6 (HMIPv6) Mobility Management," RFC 5380 (Proposed Standard), Internet Engineering Task Force, Oct. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5380.txt 33, 95

[28] E. Perera, V. Sivaraman, and A. Seneviratne, "Survey on network mobility support," *SIGMOBILE Mobile Computing and Communications Review, ACM*, vol. 8, no. 2, pp. 7–19, 2004. 34

[29] J.-Y. Hu, C.-F. Chou, M.-S. Sha, I.-C. Chang, and C.-Y. Lai, "On the design of micro-mobility for mobile network," in *Emerging Directions in Embedded and Ubiquitous Computing*, ser. Lecture Notes in Computer Science, vol. 4809, no. 401–412, 2007. 35

[30] G. Armitage, *Quality of Service in IP Networks*. Macmillan Technical Publishing, 2000. 36

[31] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Service," RFC 2475 (Informational), Internet Engineering Task Force, Dec. 1998, updated by RFC 3260. [Online]. Available: http://www.ietf.org/rfc/rfc2475.txt 38, 41

[32] S. Shenker, C. Partridge, and R. Guerin, "Specification of Guaranteed Quality of Service," RFC 2212 (Proposed Standard), Internet Engineering Task Force, Sept. 1997. [Online]. Available: http://www.ietf.org/rfc/rfc2212.txt 38

[33] J. Wroclawski, "Specification of the Controlled-Load Network Element Service," RFC 2211 (Proposed Standard), Internet Engineering Task Force, Sept. 1997. [Online]. Available: http://www.ietf.org/rfc/rfc2211.txt 38

[34] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification," RFC 2205 (Proposed Standard), Internet Engineering Task Force, Sept.

1997, updated by RFCs 2750, 3936, 4495. [Online]. Available: http://www.ietf.org/rfc/rfc2205.txt 39

[35] J. Wroclawski, "The Use of RSVP with IETF Integrated Services," RFC 2210 (Proposed Standard), Internet Engineering Task Force, Sept. 1997. [Online]. Available: http://www.ietf.org/rfc/rfc2210.txt 39

[36] A. K. Talukdar, B. R. Badrinath, and A. Acharya, "MRSVP: A resource reservation protocol for an integrated services network with mobile hosts," *Wireless Networks*, vol. 7, no. 1, pp. 5–19, 2001. 40

[37] S.-J. Leu and R.-S. Chang, "Integrated service mobile Internet: RSVP over mobile IPv4&6," *Mobile Networks and Applications*, vol. 8, no. 6, pp. 635–642, 2003. 40

[38] M. A. Malik, S. S. Kanhere, M. Hassan, and B. Benatallah, "On-board RSVP: An extension of RSVP to support real-time services in on-board IP networks," in *Distributed Computing - IWDC 2004*, ser. Lecture Notes in Computer Science, A. S. et al., Ed., vol. 3326, 2004, pp. 264–275. 40

[39] B. Davie, A. Charny, J. Bennet, K. Benson, J. L. Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)," RFC 3246 (Proposed Standard), Internet Engineering Task Force, Mar. 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3246.txt 41

[40] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," RFC 2597 (Proposed Standard), Internet Engineering Task Force, June 1999, updated by RFC 3260. [Online]. Available: http://www.ietf.org/rfc/rfc2597.txt 41

[41] IETF NSIS working group. http://www.ietf.org/dyn/wg/charter/nsis-charter.html. 42

[42] R. Hancock, G. Karagiannis, J. Loughney, and S. V. den Bosch, "Next Steps in Signaling (NSIS): Framework," RFC 4080 (Informational), Internet Engineering

Task Force, June 2005. [Online]. Available: http://www.ietf.org/rfc/rfc4080.txt 42, 98

[43] H. Schulzrinne and R. Hancock, "GIST: General Internet Signalling Transport," draft-ietf-nsis-ntlp-20, work in progress, June 2009. 43, 109, 115

[44] J. Manner, G. Karagiannis, and A. McDonald, "NSLP for quality-of-service signaling," draft-ietf-nsis-qos-nslp-16.txt, work in progress, Feb. 2008. 43, 109, 110, 114, 115

[45] R. Ben Rayana and J.-M. Bonnin, "Mobility aware application manager for mobile networks," oct 2008, pp. 337–342. 45

[46] C. A. Taylor, O. Anicello, S. Somohano, N. Samuels, L. Whitaker, and J. A. Ramey, "A framework for understanding mobile Internet motivations and behaviors," in *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI08).* New York, NY, USA: ACM Press, 2008, pp. 2679–2684. 47

[47] G. Kamel, A. Mihailovic, P. Pangalos, and A. H. Aghvami, "Cost-optimal QoS aggregation for network mobility," in *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, Nov. 2007, pp. 5006–5010. 48

[48] G. Kamel, A. Mihailovic, and A. H. Aghvami, "Case analysis of a cost-optimal QoS aggregation policy for network mobility," *Communications Letters, IEEE*, vol. 12, no. 2, pp. 130–132, Feb. 2008.

[49] G. Kamel, A. Mihailovic, and A. H. Aghvami, "A cost-optimal QoS aggregation policy for network mobility: Analysis and performance comparisons," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 7, pp. 3547–3557, Sept. 2009. 48

[50] M. A. Malik, S. S. Kanhere, and M. Hassan, "Aggregation policies over RSVP tunnels," in *Vehicular Technology Conference, 2005. VTC-2005-Fall. 2005 IEEE 62nd*, vol. 2, 2005, pp. 1249–1253. 49, 64

[51] D. Bertsekas and R. Gallager, *Data Networks*. Prentice-Hall International, 1987. 50

[52] M. Malik, L. Libman, S. Kanhere, and M. Hassan, "Analysis of resource reservation aggregation in on-board networks," in *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, Apr. 2007, pp. 1006–1010. 51

[53] J. D. C. Little, "A proof for the queuing formula: $L = \lambda W$," *Operations Research*, vol. 9, no. 3, pp. 383–387, 1961. 52

[54] D. P. Heyman, "Optimal operating policies for M/G/1 queuing systems," *Operations Research*, vol. 16, no. 2, pp. 362–382, Mar. 1968. 52

[55] L. Kleinrock, *Queueing Systems: Theory*. John Wiley & Sons, 1975. 62

[56] W. Fischer and K. Meier-Hellstern, "The Markov-modulated Poisson process (MMPP) cookbook," *Performance Evaluation*, vol. 18, no. 2, pp. 149–171, Sept. 1993. 62

[57] The MathWorks web page. [Online]. Available: http://www.mathworks.com 64, 81

[58] L. Cavin. (2004, Oct.) TRIRND: Discrete random number generator from a triangular distribution (MATLAB tool). [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/3920 82

[59] M. Buehren. (2009, Aug.) MULTICORE: Parallel processing on multiple cores (MATLAB tool). [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/13775 84

[60] B. G. C. Dellaert and B. E. Kahn, "How tolerable is delay? consumers' evaluations of internet web sites after waiting," *Journal of Interactive Marketing*, vol. 13, pp. 41–54, 1999. 91

[61] G. Kamel, P. Pangalos, A. Mihailovic, and A. H. Aghvami, "Improvements in or relating to network mobility," UK Patent GB08 22 815.7, 2008. 94

[62] G. Kamel, P. Pangalos, A. Mihailovic, and A. H. Aghvami, "A seamless QoS-enabled mobility management mechanism for moving networks," in *Telecommunications, 2009. ICT '09. International Conference on*, May 2009, pp. 228–231. 94

[63] G. Xie, J. Chen, H. Zheng, J. Yang, and Y. Zhang, "Handover latency of MIPv6 implementation in Linux," in *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, Nov. 2007, pp. 1780–1785. 95

[64] V. Friderikos, A. Mihailovic, and A. H. Aghvami, "Analysis of cross issues between QoS routing and μ-mobility protocols," *Communications, IEE Proceedings-*, vol. 151, no. 3, pp. 258–262, 2004. 95

[65] A. Dev Pragad, G. Kamel, P. Pangalos, and A. H. Aghvami, "A combined mobility and QoS framework for delivering ubiquitous services," in *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE International Symposium on*, Sept. 2008. 97

[66] G. Apostolopoulos, S. Kama, D. Williams, R. Guerin, A. Orda, and T. Przygienda, "QoS Routing Mechanisms and OSPF Extensions," RFC 2676 (Experimental), Internet Engineering Task Force, Aug. 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2676.txt 100

[67] D. Harrington, R. Presuhn, and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks," RFC 3411 (Standard), Internet Engineering Task Force, Dec. 2002, updated by RFCs 5343, 5590. [Online]. Available: http://www.ietf.org/rfc/rfc3411.txt 107

[68] H. Chaskar, "Requirements of a Quality of Service (QoS) Solution for Mobile IP," RFC 3583 (Informational), Internet Engineering Task Force, Sept. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3583.txt 108

[69] J. Manner, R. Bless, J. Loughney, and E. B. Davies, "Using and extending the NSIS protocol family," draft-ietf-nsis-ext-04.txt, work in progress, Aug. 2009. 109

[70] G. Ash, A. Bader, C. Kappler, and D. Oran, "QoS NSLP QSPEC template," draft-ietf-nsis-qspec-21.txt, work in progress, Nov. 2008. 114

[71] H. Tschofenig and D. Kroeselberg, "Security Threats for Next Steps in Signaling (NSIS)," RFC 4081 (Informational), Internet Engineering Task Force, June 2005. [Online]. Available: http://www.ietf.org/rfc/rfc4081.txt 115

[72] A. Pragad, P. Pangalos, V. Friderikos, and A. Aghvami, "Dynamic qos aware route optimization for networks with mobility agents," in *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*, Jan. 2009, pp. 1–5. 117

[73] NSIS implementation project NSIS-ka. http://nsis-ka.org/. 117

Appendix A

Cost-Optimal K and R Thresholds

This appendix shows plots of Equations 3.23 and 3.24 against the QoS request arrival rate λ at the MR. These are shown in Figures A.1 and A.2 respectively. As with the T-policy, since K^* and R^* are both highly dependent on the value of λ, it is difficult to achieve cost optimality when requests rates are bursty using only a single parameter value.

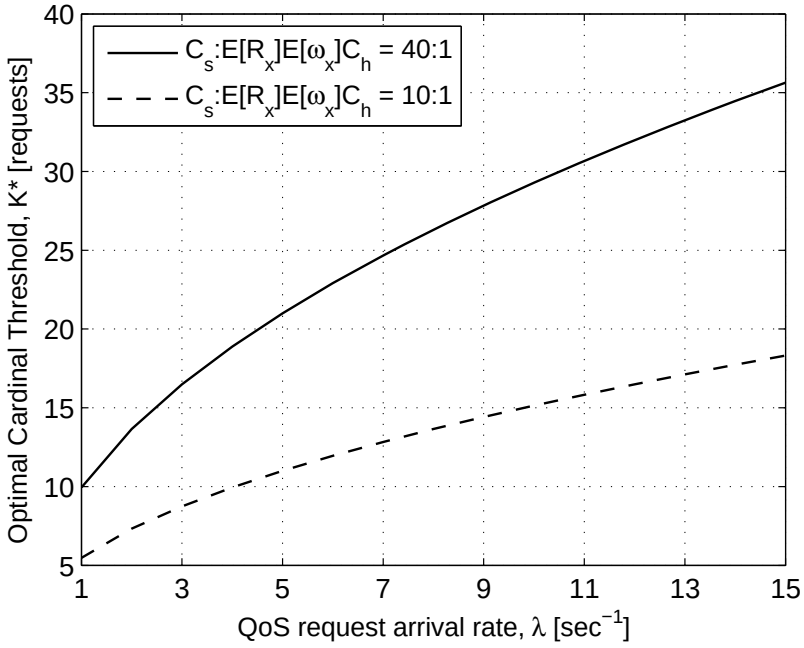


Figure A.1: Cost-optimal cardinal threshold K^* for signalling-to-holding-cost (per mean requested resource) ratios $C_s : E[\mathcal{R}_x]E[\omega_x]C_h$ of 10:1 and 40:1.

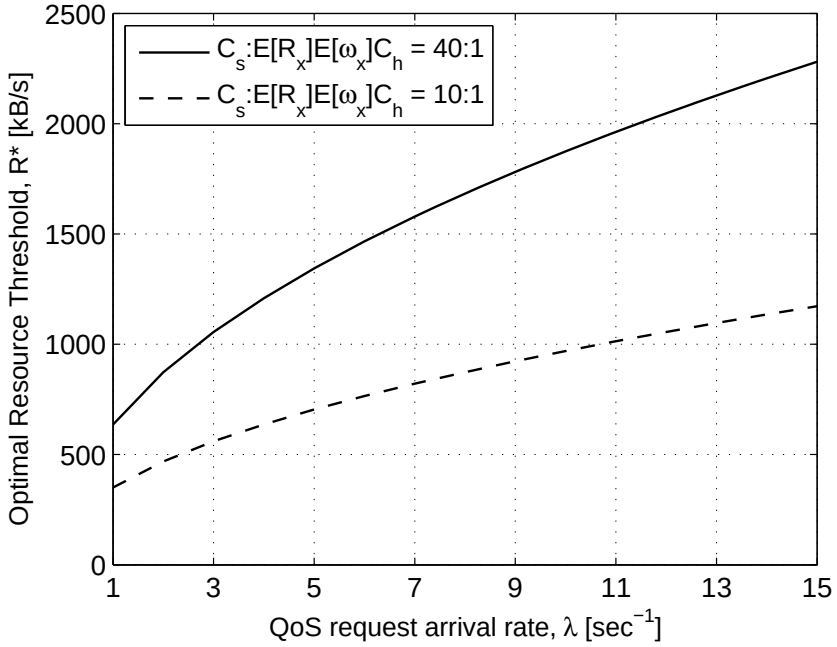


Figure A.2: Cost-optimal resource threshold R^* for signalling-to-holding-cost (per mean requested resource) ratios $C_s : E[\mathcal{R}_x]E[\omega_x]C_h$ of 10:1 and 40:1.