“Politehnica” University of Timisoara
Faculty of Automation and Computers

Handover Algorithm Design and Simulation in Heterogeneous Wireless Networks

Bachelors Thesis

Madalina Fiterau-Brostean

SCIENTIFIC COORDINATORS:
Prof. Dr. Ioan Jurca
“Politehnica” University of Timisoara

Dr. Gabriel-Miro Muntean
Dublin City University

July 2009

Table of Contents

1 Introduction

1.1 Context

In future wireless networks, it is expected that users will have the advantage of multi-mode wireless devices, enabling the choice of connectivity to multiple access networks, with different characteristics according to the radio access technology (RAT) they employ. This choice involves mobile terminals transferring their communication sessions from one radio access network (RAN) to another, process known as handover. A network selection decision is required as part of handover to make a choice between available networks. A good network selection decision should take into account all candidate RANs' characteristics, application requirements and device capabilities as well as user preferences [1].

A wide variety of access technologies has become available, as well as mechanisms to switch from cellular to wireless local area networks and metropolitan area networks. According to this survey from 2007 [23], the deployment of WiMAX technology is on an exponentially rising curve, with a predicted 5-fold increase in North America and Europe, and a 15-fold increase in the Asia Pacific and Africa/Middle East regions by 2012. City-wide WiFi coverage has also become a common phenomenon [24].

At the same time, a wide range of services is available to the users through the Internet, each having different specific requirements, which need to be fulfilled by the wireless access networks [3]. An overwhelming percentage of software developers, 94 %, expect the development of wireless internet applications to maintain the current trend or increase in the following two years, according to this survey [25]. With Windows Mobile and iPhone users seemingly content with their devices [26], an expected 500 million viewers of Mobile TV by 2013 [25], and a increased use of WiFi services – British Telecom reported WiFi traffic has doubled in 2008 [28], it is fair to say that the prospects of wireless service-oriented technologies look as bright as ever.

In this context, efforts are being directed towards ensuring user mobility through seamless connectivity, an example of which is the recently approved IEEE 802.21 Media Independent Handover (MIH) standard [2].

1.2 Motivation

With the increase in the popularity of the multi-mode mobile devices, switching between networks will occur regularly and multiple users performing simultaneous handover will happen more often. This is either as the users move away from an access point or because the current network becomes congested, but in both situations a better network becomes available. Consequently, a good network selection algorithm must account for the possibility of multiple users simultaneously performing handover in the same area.

Scenarios that involve groups of similarly profiled people moving in the same region at the same time, while accessing similar applications, include the example of students leaving a lecture theatre and walking to their next class while accessing similar data such as online lecture notes or the latest music video clip. The students all have similar profile with regard to user preferences (cheaper, high-bandwidth content). Other examples include delegates at a conference on a short break checking email, and commuters watching news clips.

The mobile devices that perform handover in groups needn't even be controlled by people. Interesting applications benefiting from this type of research are in the field of infrastructure-based sensor networks, whether the sensors in the cluster pick up the same type of signal (vibrations, temperature or humidity) over an area or different signals (video and audio).

1.3 Problem statement

Given the above considerations, it has become paramount that a mobile device connects, at a given time, to the ***network that best fits the service*** that runs on the device at that time, with an ***awareness of other devices*** operating within the same region.

1.4 Objectives

The objectives of this thesis are as follows:

- To develop an effective handover policy that
  - Provides good quality of service to the single user
  - Outperforms alternative approaches for multiple-user scenario
  - Achieves a balance between algorithm complexity and simple computation
- To create an environment to simulate the handover scenarios and assess performance
- To establish realistic and relevant scenarios that are to be simulated in the environment
- To implement the algorithm, compare results and draw conclusions

1.5 Contributions

1.5.1 Simulation framework

The simulation framework used, based on the standard NS 2 and NS 2 add-ons, has undergone a continuous improvement and extension process throughout the development of this thesis. Attempts to test the presented algorithms have revealed that, currently, with the widely-used NS2, simulating handover algorithms is cumbersome, requiring detailed knowledge of the internal workings of the simulator, adding to its code and rebuilding, to make it aware of the algorithm. Therefore, alternative mechanisms are required to make simulating handover algorithms, as well as extracting, storing and comparing simulation results.

1.5.2 Quality provisioning

This thesis introduces the Quantified Adaptive Delay Selection (QADS), a novel multi-user-aware handover algorithm which makes the best network selection decisions in order to maintain the best possible overall Quality of Service (QoS) for a user, while taking into consideration the impact of other users operating in the same area. QADS includes an innovative solution to increase user QoS even in the case when multiple mobile users consider the same RAN to be best and intend to perform handover simultaneously. QADS considers terminal-controlled network selection decisions in infrastructure-based networks and focuses on the performance of handover multiple users as they move en masse from one RAN to another.

1.5.3 Focus on multiple users

Notably, the two focal points of the algorithm are the contention of handover based on a quality of service function that prevents nodes from handing over simultaneously and a random factor that distributes nodes across multiple networks when there is no other discriminator. Thus, in order to provide good connectivity, the network selection algorithm will adapt according to the network conditions, with a constant awareness that other users' decisions impact these conditions.

2 Background knowledge

In order to introduce the handover solutions, a detailed explanation of the notion of handover an its implications must be given. For the solutions to be considered acceptable, they must adhere to ratified standards, which are briefly described. Last but not least, an appropriate testing environment must be selected to estimate the performance of the proposed approaches.

2.1 Fundamental concepts

"Handover - The process by which a mobile node obtains facilities and preserves traffic flows upon occurrence of a link switch event. The mechanisms and protocol layers involved in the handover can vary with the type of the link switch event (i.e., with the type of the serving and target point of attachment and the respective subnet associations).
Vertical handover - A handover where the mobile node moves between points of attachment of different link types, such as from UMTS to WLAN." [2]

2.1.1 The need for handover

As the heterogeneous environment is shifting from the realm of expectation to that of reality, users will need to choose the network which offers the best quality of service. It is apparent that a single such choice is not enough as terminals achieve mobility – perhaps even high mobility – and enter and exit the coverage areas of networks with increasing speeds. Also, the quality of networks is far from constant, mainly because of the unpredictability of the wireless physical medium for instance if a large obstacle suddenly blocks the waves -, but also given the dynamic decisions of other devices operating in the proximity of the network – such as a terminal starting to transfer HDTV traffic that imposes a considerable strain on the network. Moreover, the applications on the node itself change periodically, inducing a corresponding change in the requirements for the network parameters. In any of these scenarios, the solution is obvious: the terminal must perform seamless handover – handover that is transparent to the user.

2.1.2 Stages of handover

The handover process occurs in three steps:

- Network discovery – responsible with identifying networks that are in range of the device. It is usually done by scanning the medium for beacons – scanning is different from simple receipt of broadcast packets in that the power of the receiver is boosted to intercept even the weakest beacons.
- Network selection – a choice is made, by the terminal or the network, from the available networks, of a subset to which the terminal can be connected to at the same time – usually, it is just one, though more networks can be selected if more than one interface can be used at one time.
- Handover – the actual process of switching the data session from one network to another, which is highly dependent on technology. For instance, the 3GPP UMA/ GAN standard defines handover between cellular technologies UMTS/GSM and WiFi (see descriptions below). The IEEE 802.21 standard [2] is an attempt to provide a uniform interface to the handover process.

2.1.3 Handoff protocols

Whether horizontal or vertical, handover ultimately consists of selecting the network and establishing the connection itself. According to where the operations take place, the handover can be network-controlled, mobile-assisted or mobile-controlled. [1] Decentralization reduces the handover delay, though the amount of information available for network selection is also decreased. Network controlled handover is a protocol in which the network both makes the handover decision and performs the switch. The signal from the mobile device is measured both in the cell associated with the device and in adjacent cells. If the signal is strongest in a different cell than the one with the associated BS, the network changes the BS associated with the mobile node. Although network-wide status is available to make the decision, the perceived quality of service at the terminal is difficult to estimate. Also, while this scheme might have worked for cellular technology, it tends to perform poorly when non-cellular networks are in range of the mobile device, since inter-network communication is not normally implemented. The disruptive length of the delay is another disadvantage, making this protocol unfit for dynamic environments. To reduce the delay, mobile-assisted handover shifts the measurements to the terminal, and the handover decision to the mobile switch center. One of the problems occurring in network-performed handover is the loss of connectivity after the connection to the old network has been terminated and before the one to the new network is established. The solution to this comes with the so-called "soft" handover. The connection to the old network is maintained until after the terminal is connected to the new network – mechanism know as "make before break". The obvious drawback is the overhead of keeping both connections. Mobile-controlled handover leaves the measuring, the decision and the handover itself to the mobile terminal, achieving the best performance in terms of handover speed. However, the selection will be based solely on the data picked up by the terminal, usually lacking network-wide information, such as coverage, number and placement of users. Highly decentralized, terminal-controlled handover has wide applicability due to its performance and scalability.

2.1.4 Handover solutions

Up to this point in this thesis, no mention has been made concerning the level at which handover occurs. That is because it may occur at any layer on in the IP protocol stack. Handover at each layer has specific characteristics [21], illustrated in the table below, which is constructed following the IETF RFC 1122 standard [20].

Application Layer DHCP, DNS, FTP, HTTP, IMAP, POP, SMTP, SSH	• Application layer handover is performed in the end-to-end session and application layer, "without the intervention of the intermediate network agents" [21] • All handover semantics are kept only by the end terminals • Network agent, such as the tunneling, support is not required • No difference between vertical and horizontal handover
Transport Layer TCP, UDP, DCCP, SCTP, ECN, RSVP	Transport layer handover was introduced as a result of such shortcomings of Mobile IP such as conflict with network security solutions and inefficient routing S. Fu et. al. [22] propose a 5-step TraSH (Transport layer Seamless Handover protocol) with SCTP, occurring in the following way: • A new IP address is obtained (either through DHCP or MIP) • The address is added to the SMTP association • Data packets are redirected to the new IP – by setting the new address as primary • The location manager is updated • Finally, the obsolete IP address is deactivated or removed The considered security issues are cooperation with firewalls and ingress filtering and dynamic address reconfiguration.

	The main issue is the lack of readily-available link layer parameters such as SNR (Signal Noise Ratio) and BER (Bit Error Rate).
Internet Layer IPv4, IPv6, ICMP, ICMPv6, IPsec	Network layer handover provides transparency to the upper layers, "irrespective of the movement of a User Equipment". Some difficulties appear in applying implemented horizontal handover mechanisms to vertical handover, for instance network applications may assume a single technology and a single interface is used at once. Approaches to providing Mobile IP for vertical handover [21]: • "modifying Mobile IPv6 code to support heterogeneous access technologies and the multiple network interfaces at the same time" • "using virtual network interface to hide the existence of multiple network interfaces" An additional issue is the (lack of) compatibility between MIPv6 and MIPv6.
Link Layer ARP, RARP, NDP, OSPF, L2TP, PPP, MAC (Ethernet, ISDN, DSL)	"Link layer handover is transparent to network layer and upper layers. In these types of handover schemes, a UE changes its point of attachment by using only link layer operations." [21] Not suited to vertical handover, because it would imply introducing into one link layer technology details about all others it needs to perform handover with. Unless a cross-layer architecture is devised, Link Layer handover is only appropriate for horizontal handover.

2.1.5 Factors in handover

Depending on the layer at which it is implemented, the efficiency of handover depends on different sets of factors. Nevertheless, several of these factors are of the essence independent on the level at which handover is performed. First of all, the availability of information about a network is crucial in the handover decision. Although, traditionally, the network selection is based on RSS (Received Signal Strength), SNR (Signal Noise Ratio) and BER (Bit Error Rate), more realistic approaches consider a number of parameters, most of which are only available at MAC level. The speed and location of the mobile device are also matters to be considered. If a device is moving fast, then the stability of the selection might take priority, for instance a WiMAX might be a better choice than a WLAN because of its wider range. The type of application running on the terminal also needs to be taken into account, since different applications have different requirements. The capabilities of the mobile terminal, such as the available energy, the number of interfaces and the number of channels it has also influence the handover process.

2.1.6 Performance of handover

After handover is performed, the benefits must be assessed and compared with the costs. Benefits of handover are represented by improved network parameters for limited amounts of time. The gain in perceived quality of service is difficult to estimate, but can be done so by a utility function that accounts for the most relevant of link parameters. Also, rather than measuring the link parameters, certain traffic types allow the comparison of the final result after handover with an estimation of what would have occurred if the handover had not been performed. For instance, for a file transfer application, the total time it took to download a file can be compared with the estimated time prior to handover. The costs of the handover are represented by the power consumed during neighbor discovery and handover and the data loss that inevitably occurs during handover. Other costs, which might be less apparent, are connected with the impact of the mobile device on the network. For instance, a device that transfers heavy traffic and is close to the AP might make it impossible for other devices that are further away to get good connectivity.

2.2 Standards

Taking standards into account is not only important to the accuracy of the simulations, but it also ensures that relevant scenarios are taken into account – i.e. WiFi and WiMAX were standardized, so it is expected that networks conforming to these standards are frequently encountered. Also, in the event of actually building a physical model to match the simulated mode, standard conformance makes the present work easier to implement.
Several types of standards were referenced; the standards for wireless data transport refer to the physical characteristics of networks, therefore the typical values for network parameters. The G1010 recommendation talks about the specific parameter values required by different applications to ensure high QoS. The draft handover standard presented ensures transparent handover at network level.

2.2.1 Standards for wireless data transport

2.2.1.1 Local and Metropolitan Wireless Networks

The standards for the WLANs (Wireless Local Area Networks) and WMANs (Wireless Metropolitan Area Networks) are part of the IEEE family of standards – IEEE 802.11 and 802.16 respectively. Other standards belonging to this family, such as 802.3 (for Ethernet) and 802.15 (Wireless Private Area Networks) were not presented for obvious reasons.

2.2.1.1.1 IEEE 802.11 - WiFi

- General characteristics

According to the standard specifications, there are significant differences at the physical layer between wired networks and 802.11 WLANs. First, there are no boundaries to the medium and the limit after which frames can no longer be received is unknown – meaning that handover may have to be performed long before the mobile node exits the network. Less reliable and asymmetrical, the medium carries all signals, making interference between signals with compatible frequency bands a serious issue – so an increase of the number of users on a network burdens the network not only in terms of throughput, but also makes exchange of information more difficult through interference. Security is also a problem, critical packets requiring encryption – AES and TKIP are examples of such protocols. The topologies are dynamic, lacking full connectivity. Given all of these characteristics, guaranteeing a given level in QoS is not possible, with the exception of environments under very strict control. Under normal propagation conditions (for instance if no obstacles are present), service degrades as the distance between the access point and the mobile device increases. Because the topologies tend to change, the addressable unit is a station rather than a physical location.

- Advantages

The obvious advantages are mobility – devices are no confined to given locations – and portability – the network itself fits in most environments – making the WLAN a convenient alternative to wired networks. Easily deployable, the wireless network only needs the setup of an access point to function, making the cost low enough for such networks to be virtually expandable.

- Limitations

On the other hand, the wireless networks are limited in several respects. Aside from the already-mentioned drawbacks such as security and reliability, there is also the problem of speed. A typical wireless network has a data rate 100 times slower than an Ethernet: 1-100Mbit/s compared to 100Mbit/s – several Gbit/s. Also, the WLANs are low-capacity networks, build to accommodate, in practice, only a limited number of users.

- 802.11b

The extension of 802.11 in the 2.4 GHz band, known as 802.11b, allows nominal data rates of 5Mpbs to 11Mbps. The standard dictates the use of either direct sequence spread spectrum or frequency-hopping spread-spectrum at the PHY layer, making the range of such networks is usually in the tens of meters. The most common of 802.11 variations, it was considered representative enough to be used in the simulations.

2.2.1.1.2 IEEE 802.16 WiMAX

- General characteristics

The WiMAX specifications place the operational frequency of the physical layer in the 10-66 GHz range, using Orthogonal Frequency Division Multiplexing. The ranges are in the order of kilometers and tens of Mbit/s.

- Mobile WiMAX

Mobile WiMAX (802.16e) introduces SOFTMA (scalable orthogonal frequency-division multiple access). It also brings Multiple Antenna Support through Multiple Input-Multiple Output, obtaining frequency reuse and bandwidth efficiency.

- Comparison with WiFi

WiMAX is often compared with WiFi because they are both standards referring to wireless technologies and both from the 802 family of standards. WiMAX networks are different from WLAN networks in that they range for kilometers rather than hundreds of meters. Typically, WiMAX networks use licensed spectrum, whereas WLAN networks use unlicensed spectrum. The two technologies have different Quality of Service mechanisms. In the WiMAX, scheduling is used to reserve slots so that quality of service can be guaranteed throughout the interval the connection is maintained. The slots can be enlarged or reduced, but other subscribers cannot do it. WLAN uses contention and packet priorities, which means QoS is relative between packets, not guaranteed. Because of the slot system, WiMAX networks are more stable in the event of overload and bandwidth-efficient in comparison with WiFi. However, they can only offer service to a limited number of users.

- Limitations

According to the WiMAX specification, such networks deliver 70 Mbit/s and have ranges of over 50 kilometers. However, these do not happen simultaneously, because at that distance, the Bit Error Rate increases and therefore a lower bit rate must be used. Fixed WiMAX have industrial-style directional antennas, in contrast with Mobile WiMAX that typically use an omni-directional antenna which offer a lower throughput than a directional one. Available bandwidth is shared between users in a radio sector, so if many users are in a single sector the performance could deteriorate.

2.2.1.2 Cellular technologies

Cellular networks are made of a number of radio cells, each with at least one transceiver, with areas of coverage kilometers-wide. Frequency Division Multiple Access or Code Division Multiple Access is used to identify signals coming from different transmitters. Horizontal handover between cells of the same network is performed based on SNR. A number of such technologies exist: Global System for Mobile Communication (GSM), General Packet Radio Service (GPRS), Code Division Multiple Access (CDMA), Enhanced Data Rates for GSM Evolution (EDGE) and UMTS (Universal Mobile Telecommunications System).

2.2.2 Standard for handover: IEEE 802.21

The IEEE 802.21 offers specifications for aspects of handover such as quality of service and service continuity. It states how operations such as network discovery and network selection should occur. It facilitates handovers by offering information to the network selection entity. The 802.21 standard offers three services:

- Media Independent Event Service

MIH events are raised when changes occur at the physical layer. Either the link parameters change or new networks are detected or a link is established or the connection has been interrupted.

- Media Independent Command Service

The command service allows higher levels to control the link layer – to reconfigure a link or select an appropriate link. When a MIH command is issued, it is always executed.

- Media Independent Information Service

The information service offers an interface for the handover policy to gather data about the neighboring networks. The standard specifies the data structures used for link parameter value retrieval.

2.2.3 Connection parameters – the ITU-T G1010 recommendation

The G1010 recommendation [3] defines "a model for multimedia Quality of Service (QoS) categories from an end-user viewpoint". Eight QoS classes are defined by establishing certain expectations for network parameters for different multimedia applications. In this thesis, a point is made that the applications that run on the mobile terminal have just as much impact on the perceived QoS as the network parameters. For instance, in a real time application, the steadiness in quality of service is important because a 30 seconds delay in a call or a video transmission is usually unacceptable. In contrast, file transfer applications can cope with variations in throughput, as long as the average value of the throughput ensures that the file is transferred within a reasonable amount of time. On the other hand, in a file transfer the information loss should be zero, whereas a real time application can recover from a Bit Error Rate of up to 10^{-6}.

The G1010 parameters can be used in the network selection process, provided that estimations of the link parameters – through an MIH implementation or other mechanisms – are accurate.

Alternatively, the G1010 parameters also provide a simple solution to assessing handover performance: perform handover according to different criteria but rank the performance of the handover according to how well the targets were reached.

2.3 Modeling and simulation

The handover algorithms presented in his thesis need to be validated. This can be done through either simulation or emulation. Each alternative is presented, together with its advantages and drawbacks.

2.3.1 Emulation

Because the problem in question is a practical one, emulation is the most accurate way in which the algorithms could be validated. However, there are several difficulties involved.

First of all, although multi-face mobile devices are common, implementing the handover algorithm in the mobile device accurately is somewhat problematic, mostly because implementations of the MIH function are not yet available. Also, WiMAX or UMTS networks cannot be easily setup and controlled without industrial support. Moreover, it would not be cost effective to invest in the emulation of an algorithm without prior analysis of simulation results.

2.3.2 Simulation

2.3.2.1 Assessment of available simulation environments

Since emulation poses insurmountable difficulties, the remaining alternative is simulation. The simulation environment needs to fulfill a series of requirements. First of all, it must allow simulation of both wired and wireless networks (the wired connections are necessary to link the wireless networks). Because the handover algorithms need to be assessed for both UDP and TCP traffic, both of these protocols must be implemented. Also, different applications and scenarios must be easily modeled.

- A wide range of simulators is available, but most of them do not fulfill all the requirements. The QualNet simulator offers a scenario designer, an animator, an analyzer and a packet tracer. Different packets were developed for UMTS and WiMAX simulation and handover can be simulated as well. However, it is not open source or even free for that matter.
- GloMoSim is another simulation environment that fulfils the transport and application layer requirements. It is also available free of charge, but it cannot simulate WiMAX or UMTS networks – in fact it cannot be used to simulate vertical handover.
- OPNET is yet another commercial simulation product, which contains the required link, network and transport layer protocols and can be easily extended. However, the handover protocol would have to be written from scratch.

2.3.2.2 The network simulator (NS2)

Network Simulator version 2.33 (NS2) contains the necessary features for handover algorithm simulation. Among its relevant features are, aside from wired and wireless network simulation capabilities, the possibilities to easily simulate applications based on TCP/UDP/SCTP traffic and easily creating topologies through scripts. Highly extensible, the standard NS 2.33 contains an implementation for the IEEE 802.11 standard.

What the standard version lacks are implementations of the WiMAX and UMTS technologies, as well as an IEEE 802.21 implementation. Also, the WLAN simulations are not accurate and had to be revised.

Nevertheless, add-ons that resolve most of the illustrated issues are available online:

- The NIST WiMAX implementation
- The NIST Neighbor Discovery module
- The NIST mobility add-on.
- The Marco Fiore patch

3 State of the art

3.1 Directions in vertical handover research

Research in the field of handover has been conducted for more than a decade [15]. Nevertheless, in past years, new directions have begun to emerge. While previously, in cellular networks, terminal-controlled handover was deemed inefficient, now it has become common for the terminal to make the decision and even control the handover process. Handover in ad-hoc networks has also received increased interest [4], [5], because such research can be readily applied in sensor networks. The complexity of the decisions also varies from simple policies (such as "Always Cheapest Network" selection [1]) to link parameter-based utility functions [4], [15] to complex approaches involving game theory [14].

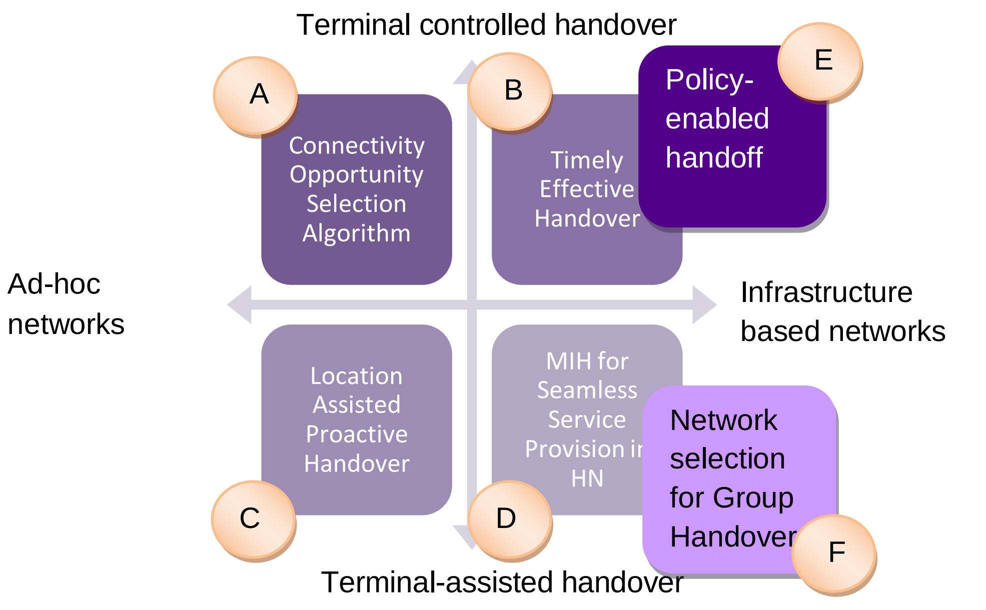

Figure 3.1 Directions in vertical handover research

3.2 Related work

The QADS implementation makes use of the Media Independent Handover, IEEE 802.21 standard [2]. The purpose of MIH is to enable the handover of IP sessions from one access technology to another, thus achieving mobility of end user devices. Recently this area of handover in heterogeneous wireless networks has seen increased interest.

A. D. Calvacanti et al. [4] propose selecting the best connection through a Connectivity opportunity Selection Algorithm (CSA), using network state information and a mobile profile, based on application requirements. As expected, the efficiency of the selection procedure depends on the availability and accuracy of the network state information. The concept of connectivity opportunity is defined by: Network Information (type, mode, authentication and cost), Routing Information (Gateway, Hops and Gateway Cost) and Network Condition (rate, throughput, delay and avg. packet loss rate).

B. S. Yoo et al. [6] make use of neighbor network information to generate proactive handover triggers in time to allow seamless handover. However, their algorithm does not take into account selection criteria such as cost, user preference, or application type.

They present a neighbor information-based predictive handover architecture that would place a handover decision engine between the MIH function and the layer three mobility protocol, with a handover trigger that can be configured to act on any quality metric.

C. In [5], A. Dutta et al. establish an algorithm that takes advantage of the mobile's location to perform proactive handover. While their algorithm does improve handover performance compared with a signal-noise-ratio (SNR) based selection, it involves considerable overhead by tracking all RANs and using transient tunnels to connect to several networks at once. The proposed architecture makes use of mobile assisted higher layer authentication, authorization and handover. An Application Information Service (AIS) offers information about the RANs available to the node. GPS positioning is used as a selection criteria and the terminal is aware of the coverage of the networks.

D. An 802.21-based architecture has been proposed in [23] which will satisfy all considered handover principles, using the information, commands and events provided by the 801.21 standard. All the RANs connect to the evolved packet core (EPC). The GPRS networks connect through an intermediary Service GPRS Support Node (SGSN), while WLANs are accessible through the evolved packet data gateway (ePDG). IP-based networks and 3GPP networks are placed under different gateways, which are in turn linked to a packet data network gateway (P-GW) that incorporates functionality such as packet filtering, interception and charging, IP address allocation and routing of traffic to the operator's network. Additionally, the EPC contains the following network servers:
   - Authentication, Authorization and Accounting server
   - Home Subscriber server
   - Mobility Management Entity
   - Policy Control Rating Function

E. A well-known utility function, introduced by Wang et al. in [15], employs weights and uses the logarithmic scale to serve as normalization. Out of the three parameters considered in thesis, namely bandwidth, cost and security, the last one is not within the responsibilities of the IEEE 802.21 standard, and therefore unavailable for use in the case of the proposed algorithm. An alternative quality function involving heterogeneous criteria such as different RAN parameters or a combination of network characteristics, user preferences and terminal characteristics were proposed in [4].

F. Of the works that address the terminal-controlled network selection algorithm most do not consider the impact of multiple users employing the same algorithm. One exception is the work of Cai and Liu [14], in which the Group Handover problem is approached with Game Theory. The authors propose three RAN selection algorithms all of which involve some knowledge of all other user's traffic load and the RANs capacity, or user's broadcasting their handover choices and traffic load. In this thesis, it is considered that the terminals do not have this knowledge and work either on link parameter values detected at MAC level or on predicted values based on historic data.

In the present work, a weighted sum function was used, with both upper and lower bounds defined for all parameters, the parameter values are normalized with respect to their bounds, a different mechanism than the logarithm-aided normalization in [15]. As opposed to [4], where the formulas differ according to the type of application and the low/medium/high mobility involved, a generalized formula is presented which accounts for competing users simultaneously choosing the same target handover network.

4 Requirements analysis

This sub-section discusses the necessary improvements to the NS2 environments to successfully and efficiently simulate the designed handover algorithms. The integration of the NIST add-on to the 2.33 version of the simulator, the modifications required for it to provide accurate simulations and the extensions proposed to the handover framework are all discussed here.

4.1 Additions brought to simulator

4.1.1.1 Evaluation and integration of the NIST add-on

As stated in a previous section, NS2 offers a wide range of features for wireless network simulation. However, little is facilitated in the way of handover simulation. Previous efforts [1] were made to simulate handover by overlapping nodes and starting/stopping traffic from the nodes corresponding to each interface. Although this type of simulation is an appropriate model when the loss of packets and the delay caused by handover is negligible, it is not sufficiently accurate to illustrate the difference in performance of the algorithms presented in this thesis. This is mainly because the ping-pong effect is studied, and the number of handovers occurring can be as high as 15 per node, which, in a real-life scenario impacts the overall traffic, whereas with the previously mentioned implementation, no traffic is lost because of handover. Also, multiple users are studied, so the fact that handover does not happen instantly affects the perceived link parameters. Moreover, to implement the algorithms presented in this thesis, an 802.21-conformant implementation is necessary.

The NIST add-on to the Network simulator [10] offers a more accurate implementation of handover, actually redirecting the data flows between interfaces. The handover model, based on the 802.21 standard, although incomplete, is sufficient to approximate handover effects on traffic. The neighbor discovery module is used to provide network information to the handover implementation. Also used for this thesis was the implementation of the 802.16 standard, comprising of OFDM for the physical layer, Time Division duplexing as multiplexing strategy and round-robin as the default scheduler. Scanning and handover features are available for 802.16e. Quality of Service scheduling is not available, which is not a problem since the objective is to study the effectiveness of the algorithms. A notable inaccuracy is the lack of error correction capabilities, which means that possibly recoverable packets are cast out.

4.1.1.2 Adaptation and correction for standard conformance

The NIST add-on is made for NS versions 2.28 and 2.29, both of which contain less accurate implementations of the 802.11 standard, the TCP and SCTP protocols than NS 2.33. Consequently, the patch had to be integrated in NS 2.33. A number of other differences, some affecting packets and classifiers, are relevant to the present work. An 802.11 implementation issue which is not solved even in 2.33 is related to the channel propagation, and a different patch [10] was applied to fix that error along with several others. The add-on itself is not error-free, when simulating handover for more than one node, an error occurs. Correcting it was necessary for the simulations to work properly.

A different set of inconsistencies is directly related to the implementation of the MIH function. First of all, only part of the events and commands are implemented in the add-on, as follows:

MIH Events	WLAN	WMAN	UMTS
Link Up	✓	✓	✓
Link Down	✓	✓	✓
Link Going Down	✓	✓	✗
Link Detected	✓	✓	✗
Link Event Rollback	✓	✓	✗
Link Parameter Report	✓	✓	✗
Link Handover Imminent	✓	✓	✗
Link Handover Complete	✓	✓	✗

MIH Commands	WLAN	WMAN	UMTS
Link Event Subscribe	✓	✓	✓
Link Event Unsubscribe	✓	✓	✓
Link Config Threshold	✓	✓	✗
Link Get Parameters	✓	✓	✗

As illustrated in the table, the UMTS implementation is the most lacking, that is why the simulations presented in this thesis are focused on handover between WLAN and WMAN.

Table1: 802.21 Commands and Events implemented in the NIST add-on

As the add-on was originally implemented, The Link Parameter Report event was never raised for WiMAX networks, because only packets without absolutely any errors were considered in the link measurements, and basic error correction of 802.16 frames was not implemented.

4.1.1.3 Handover framework extensions

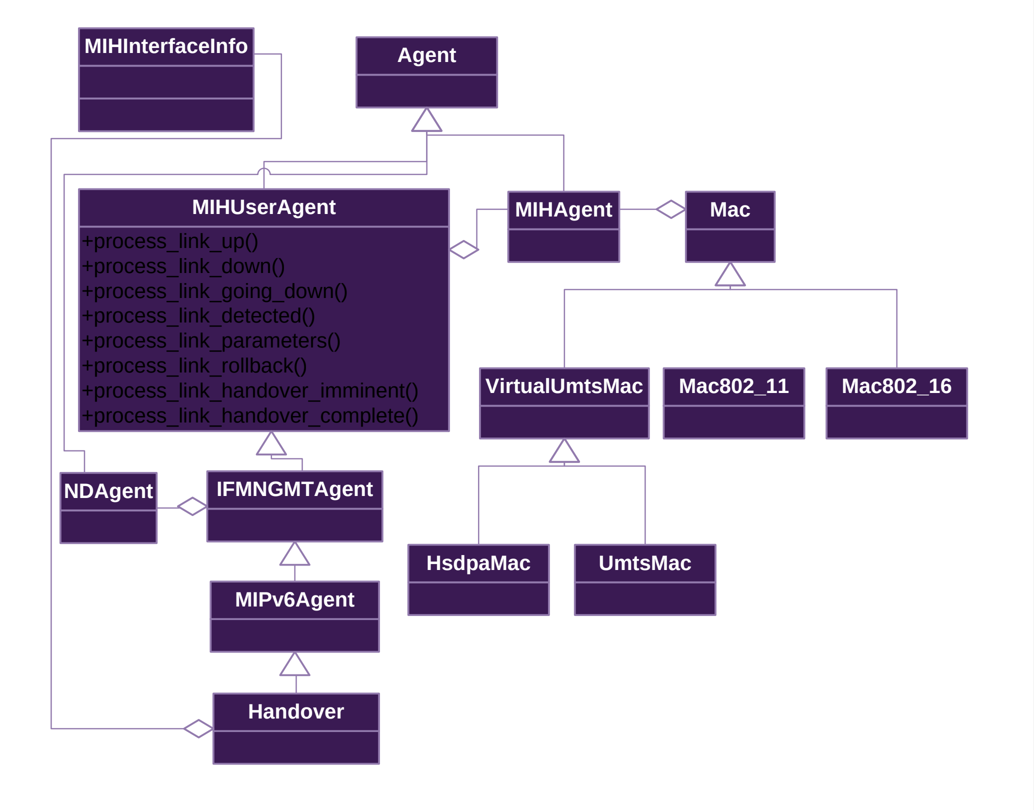

Figure 4.1 The NIST handover framework

The add-on provides a framework for implementing handover algorithms. The diagram below shows the classes introduced by the add-on. MIHAgent is the implementation of the MIHFunction. Its responsibility is receiving events from the MAC layer and passing them on to the MIH user, ultimately the handover policy– if the MIH is at BS level it relays the event to its remote users. Also, it receives commands from the handover implementation and passes them on to the MAC layer. MIHUserAgent defines virtual functions for intercepting the MIH messages and events – these are called from the MIH to which the MIH user is registered. IFMNGMTAgent adds flow management and receives events from the Neighbor Discovery Module. MIPV6Agent, based on MIPv6, implements the mobility protocol, including redirection messages. Finally, Handover is the abstract class that adds functionality common to all handover implementations. To implement a new handover algorithm, one must subclass Handover and the appropriate implementations for, at the very least, the methods shown in the diagram must be provided – for a full list of methods that are implemented in the algorithms that have been successfully tested, please see code annex [reference needed] or the MIH-related documentation in [10].

When it comes to actually implementing the algorithms, several issues have arisen due to this approach. First of all, most handover algorithms tend to have more in common than the interface provided in the Handover class. For instance, the MIH registration, the packet receipt and the flow redirection all happen in the same manner, so there is the risk of code duplication. Even more burdensome is the need to rebuild the simulator every time a new handover algorithm is introduced. Also, because low-level details which are not necessarily useful in the handover decision can easily be overlooked, the application tends to crash before all bugs are ironed out. Moreover, debugging is tedious.

Therefore, it would be useful if handover algorithms could be implemented with no changes to the simulator, keeping the user from knowing the internal workings of NS2 and with a minimal guarantee that the system will not crash – even if no implementations are provided for the event handlers. The mechanism proposed here is basically to extend the handover class with one that provides a minimal set of handover operations and handlers, and which can be then linked with code written in a higher-level language that is outside the simulator itself. The connection between the class and the outside code should be dynamic, so that handover algorithms are interchangeable. Although NS is already linked to Tcl/Tk, extending the handover class with one written in Tcl is unfeasible, because the data structures used in the handover modules have no direct correspondents in Tcl. For them to be accessible, they would have to each be explicitly exported with a binding class. On the other hand, Python, a commonly-used interpreted language has various ways of being linked with C++, and, at least in theory, obtains performance comparable to that of native C code.

4.2 The handover algorithms

4.2.1.1 Quantified Adaptive Delay Selection (QADS)

In previous recent works on network selection in infrastructure-based networks, the main weakness is that no allowances have been made by these terminal-based algorithms concerning the actions of other users operating in the same environment. Moreover, if the user terminal is in a crowded area, a given network might be considered to offer the best RAN choice by a number of users' terminals – in which case these users could end up selecting and handing over to the same network in the same timeframe.

In QADS, to account for the chance that the same RAN is chosen by a number of users in the same time interval, the connection to the best candidate network is delayed in each mobile node. This delay is set inversely proportional to the benefit that the selected network brings to the user. When the delay has elapsed, the mobile node will re-compute the QoS to ensure that the network choice is still the best for this node.

In the case where a number of nodes have the same preferences, the computed delay could be the same. This would result in all these nodes connecting to the same RAN in the same timeframe, possibly overloading the new network. In QADS, this problem is solved by monitoring the QoS before, during and after handover. In situations when the new QoS is far less than the expected QoS, QADS infers that a large number of nodes may have simultaneously handed into the same RAN. This is resolved by using a random decision to stay with the new RAN or to handover to an alternative network, as opposed to all nodes switching networks with the possibility of ending up in an unstable ping-pong state.

While QADS uses a weighted sum function for calculating the candidate network score, minimum and maximum thresholds are used to eliminate candidate RANs that fail minimum criteria. This is done so that for example a very low network cost does not outweigh a poor, unacceptable throughput rate.

4.2.1.2 Polled Network Quality of service Selection (PNQS)

4.3 The simulation scripts

With NS-2, network topologies are simulated through user-created Tcl scripts. First off, the necessary classes are written in C++ and explicitly exported through a binding mechanism. Such classes include Agents, such as TCP or UDP, classes that correspond to links, classes corresponding to the PHY layer, the MAC layer, applications – CBR traffic, FTP –, routing strategies, classifiers and, of course, the nodes representing wired stations, access points for hot spots or mobile devices. The binding mechanism allows instantiating instances of the classes from Tcl, setting attributes of the classes or objects and calling methods of the objects. Aside from calling methods of the C++ objects and the standard operations allowed by Tcl, simulator specific commands must be issued. Creating an instance object of the Simulator class is mandatory, because to start and stop the simulation, the “start” and “stop” commands must be scheduled through this object. Other commands needed for the scripts presented here are “traffic start” – applied to a traffic application and “set destination” – necessary to schedule node movement. When scheduling a command, a time must be specified, and when the simulation clock reaches that time, the command will be issued. After the script is written, it is passed as a command line argument to the NS executable, which acts as an interpreter.

There are several limitations to this mechanism. Aside from the previously-mentioned necessity of explicitly exporting classes, there are several drawbacks, some of which apply to most scripts and others that affect only the handover scripts.

- In the first category, a main disadvantage is the lack of pre-run check of the scripts, the inability to simulate a non-smooth topography with obstacles and different propagation models, and the difficulties involved when a node movement other than linear is required.
- The limitations strictly related to handover are the impossibility to control or monitor power consumption in the terminal – it is worthwhile to point out that this is an issue in soft versus hard handover and when ping-pong is involved (constantly switching from one network to another, although maximizing QoS, boosts power consumption, sometimes cancelling the benefits). Other characteristics of the handover – such as duration – cannot be controlled in the current distribution.

The scripts presented in the upcoming sections illustrate important aspects of handover involving multiple users. That is why the requirements for them are very specific. First of all, a heterogeneous environment is required. The ideal combination would involve both 802 and cellular networks. However, because not all MIH events necessary to the QADS algorithm are implemented for UMTS, the use of a WiFi and a WiMAX network was deemed sufficient for the proof of concept. The AP and BS attributes must be set according to the 802.11b and

802.11e standards respectively, meaning two-ray ground propagation, Omni antenna, and OFDM modulation for WiMAX. The configurable terminal capabilities are limited to the number, type and settings of the interfaces. In this case, for each device a multi-face node is created and two wireless interfaces are added to it. The WiMAX interface uses the SS scheduler and the SDU classifier. The multi-face node is responsible for directing the traffic to the appropriate interface during handover.

Because the current thesis focuses on handover involving a group of users, their number and relative placement is critical, as it influences their interference in communicating with the network. The research in this thesis is aimed at relatively small groups of users, as can be inferred from the scenarios presented in the motivation, ranging from several users (3-5) to a larger group (10-15). It is also important for the placement to vary, from a regular structure – a line or a matrix to a disorderly pack. Because one of the algorithms is non-deterministic, repeated runs of the simulations are in order and an average of the results is to be compared with the deterministic algorithms.

The scenario itself is relatively straightforward, in its simplest version: because of the difference in range between the two networks, the WLAN can actually be contained in the WiMAX, facing the nodes with at least two decisions, as they come in the range of the WLAN while staying in the range of the WiMAX at all times.

Several remarks must be made regarding potential problems with the simulations. First, the network ranges are not explicitly stated, they must be calculated based on the RX and CS thresholds, which is not always straightforward. Caution is necessary when scheduling the traffic start between the mobile nodes and the BS, as collision is influenced by time-wise proximity (i.e. if all the nodes start traffic at the same time, the BS will not be able to respond to all requests, so a displacement will be introduced between the nodes). Collision is also influenced by physical distance between nodes and NS is known to crash with certain combinations of traffic scheduling, node placement and network characteristics.

4.4 Automatic data extraction tools

The outputs of a simulation are, typically, a trace file that logs the packets that are sent or received at MAC level, the movement of nodes and the routing of packets, a file used to visualize the simulation using the network animator (nam) and all other messages that are directed at the standard output.

To compare handover algorithms, it is useful to keep track of the traffic at each moment. Because traffic is not continuous, it is enough to calculate the average throughput at intervals specified through a command line parameter. Other parameters, such as data loss and jitter will need to be obtained in a similar manner. It is also relevant to calculate packet loss at the base stations at different intervals - as an indicator of how the load on the networks shifts. The average overall values of parameters (throughput, jitter) show the quality of service offered by the handover policy. The average standard deviation of parameters over the nodes shows the fairness with which the algorithm distributes the QoS among the nodes. The total traffic is an indication of how efficient the handover policy is in reducing network congestion. Because when less data gets transmitted, the data loss is correspondingly low, it is valid to compute its percentage of the total traffic.

A MAC trace entry in the trace file has the following format:

```
S –t 0.006223523 –Hs 13 –Hd -2 –Ni 13 –Nx 250.00 -Ny 300.00 -Nz 0.00 -Ne -
1.000000 -Nl MAC -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0
```

In the above trace, s is an indicator that the entry refers to a sent packet, -t pinpoints the simulation time at which the event happened, -Hs shows the source MAC id of the packet, and -Hd shows the destination MAC id, in this case -2, meaning that the packet is broadcast, which makes sense since it is a beacon. The -Ni parameter is used when routing is involved, representing the last hop in the packet's trajectory. The position at which the source node was when it sent the packet is given by -Nx -Ny and -Nz, while -Ne is supposed to represent the "energy" of the packet (an indication of the signal strength), which, for sent packet traces, is always 1. The trace level (application, transport or mac) is marked by –Nl. The rest of the parameters were not used in data extraction.

Because extracting data after the simulation is over is space-consuming – trace files were as large as a few Gb for the most demanding simulations displayed in the present work, it is expected that the extraction tools will be integrated with NS so that the results are readily-available at the end of the simulation. Consequently, the computations must happen fast enough so that data can be fed in real time and the recursive formulas for the mean and standard deviation must be used:

$$\overline{x}_k = \overline{x}_{k-1} + \frac{x_k - \overline{x}_{k-1}}{k}$$

$$s_k^2 = \left(1 - \frac{1}{k-1}\right) s_{k-1}^2 + k(\overline{x}_k - \overline{x}_{k-1})^2$$

Upon attempting to pass a nam trace to the animator, a problem was found: the topography is not displayed correctly because the trace does not have the format expected by the animator. The problem was identified and an executable that corrects the trace file was made.

5 Design

5.1 Adaptation and integration of the NIST add-on in NS 2.33

The first step in creating the required simulation environment is to integrate the NIST add-on on the 2.33 version of the simulator. Figure 5.1 shows the classes that have been modified in the version 2.33 of the simulator prior the start of the present work. The classes that were added to extend and use existing classes were not represented on the diagram, since they did not have a direct impact on the classes added or changed by the NIST patch.
The TCP traffic implementation was modified in order to increase its accuracy.

- In TcpAgent – the class that models a TCP traffic server
  - the slow start and quick start calculations are changed in method limited_slows_start
  - for the timeout, the initial window is not increased if the first packet is dropped in method processQuickStart;
  - the connection is aborted in method output;
  - the timeout calculation in method rtt_timeout is rounded to the nearest larger multiple of a predefined interval characteristic to the traffic.
- TcpSink – the class that models a TCP client – is adapted to allow Explicit Congestion Notification for synchronization and acknowledgement packets.
- In FullTcpAgent – which models a full-duplex connection – the ECN sanity check is moved and the sticky bit after experiencing congestion is set in method recv.

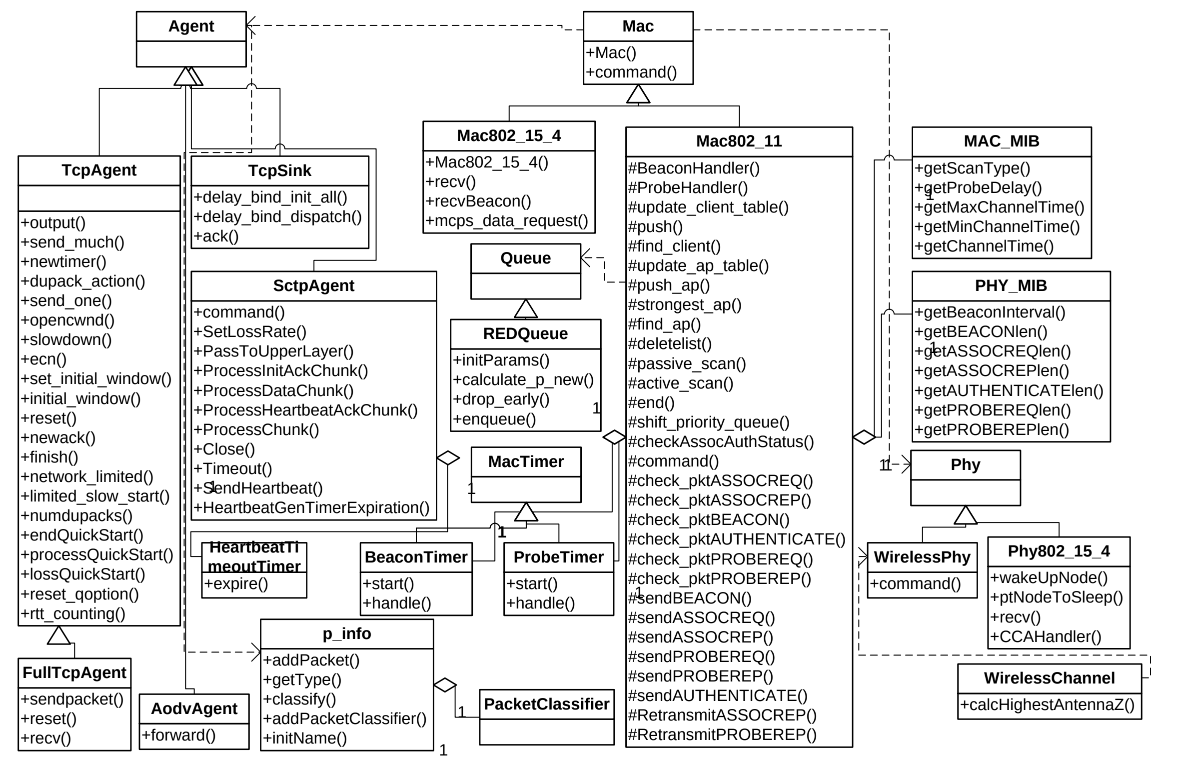

Figure 5.1 Most significant changes in NS classes from version 2.29 to 2.33 (prior to integration)

*Only the changed/added methods in existing classes that have an influence on the presented simulations are represented.

For AODV – routing that is modeled by the class AodvAgent - the jitter is adjusted for broadcast packets.

In SctpAgent – which models the SCTP server – modifications are made to only send heartbeat packets to confirmed addresses. The packet classifier is now referenced in the packet information.

The 802.11 implementation – represented by the Mac802_11 class – is enriched with beacons and probes, association and authentication are implemented.

Additional data structures were used in method calls – mostly to represent the structure of packets and different information stored in them.

Below, the most relevant additions brought by the NIST add-on are described, as implemented unto NS2.33 specifically to make the research in this thesis possible.

The NIST add-on brings significant changes to the 802.11 implementation.

- Beacons, as well as associations are introduced – this is actually a significant issue, because the same features are introduced in NS 2.33.
- Handover-specific information is added, in the form of client and BSS descriptions and bandwidth statistics,
- The min and max scan timers are used in scanning the medium for hot spots.

Changes were made to the DSDV implementation, the addition of the update_address method in the DsdvAgent class facilitating mobile IP.

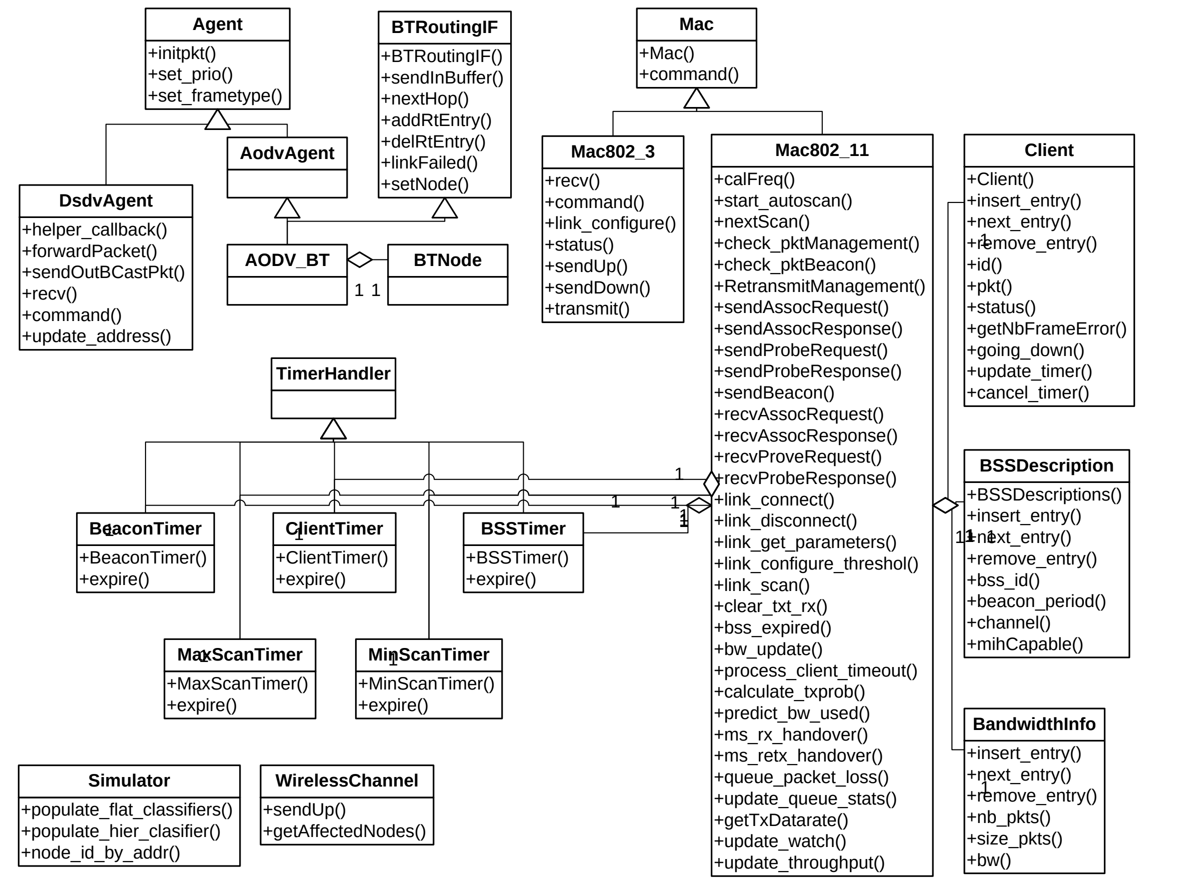

Figure 5.2 Most significant changes in NS classes brought by the NIST add-on.

*Only the changed/added methods in existing classes that have an influence on the presented simulations are represented.

The methods in the Simulator class that are represented in Figure 5.2 are also changed to function correctly with mobile IP. The existing class BTNode is used in the newly-introduced BTRoutingIF hierarchy.

The design of the version used to obtain the results presented in this thesis keeps all the improvements brought by NS 2.33 while adding the functionality required for handover. This was done by applying the following rules when integrating the NIST changes:

for each class modified by the NIST add-on

if the class is not modified in NS 2.33, keep the NIST version

otherwise

- add any required attributes;
- take each modified method;
- merge modifications ;
- check consistency with the rest of the class;
- for new methods, check consistency.

A special case was that of the Mac802_11 class, which had different implementations of the same functionality (the beacons). Moreover, the class BeaconTimer appeared in NS 2.33 and NIST, though it does not appear in 2.29. Both implementations were kept, but in different classes. The 802.11 class in 2.29 is the base class for the two implementations, while the default 802.11 class used is the one in the NIST add-on. The beacon timer class corresponding to it was renamed NistBeaconTimer. To use the regular 2.33 beacons, a flag needs to be set in the Tcl script that uses the WLAN MAC.

5.2 The handover algorithms

5.2.1 QADS

5.2.1.1 Utility function

All information received from the application layer and from the MIIS, is processed using a quality of service weighted sum function, which we have named the Application-Network Match (ANM) function. ANM represents the quality of service of a network as perceived by an application on the mobile terminal and is defined in (1).

$$\mathrm{ANM}_{\mathrm{App}}(N_i) = \begin{cases} 0, & p_{kR}(N_i) < 0 \\ \sum_{k=1}^{m} w_k * p_{kR}(N_i), & p_{kR}(N_i) \geq 0 \end{cases}$$

For parameters that should be max*imised*

$$p_{kR}(N_i) = \frac{\min(p_k(N_i), p_{kU\max}) - p_{kU\min}}{(p_{kU\max} - p_{kU\min})} \qquad (1)$$

$$p_{kU\min} < p_k(N_i)$$

For parameters that should be min*imised*

$$p_{kR}(N_i) = \frac{\max(p_k(N_i), p_{kU\max}) - p_{kU\min}}{(p_{kU\max} - p_{kU\min})}$$

$$p_k(N_i) < p_{kU\min}$$

$$\sum_{k=1}^{m} w_k = 1 \qquad (2)$$

In (1), N_i represents the network i for which the ANM is computed; w_k is the weight attributed by the application/user to parameter k; $p_{kR}(N_i)$ is the normalized value of $p_k(N_i)$ relative to the parameter bounds; $p_k(N_i)$ is the value the network provides for network parameter k; p_{kUmin} and p_{kUmax} give the interval to which $p_k(N_i)$ must belong for it to be considered acceptable to the application: p_{kUmin} is the value of the parameter for which the utility provided to the user is minimum, while p_{kUmax} is the value for which the utility is considered maximum; ***m*** represents the number of parameters considered relevant for the application.

The upper and lower bound values for the interval are obtained from the application layer, and the values are specific to the application, user and device. Parameters that are better than p_{kUmax} may not add any benefit for the user or application. For example if an application requires only 0.8 Mbps average bandwidth, then a network offering a throughput of 2 Mbps brings no additional benefit over a RAN offering the required 0.8 Mbps. Equally, once a user can see no difference in the application quality on the device for additional parameter change, the upper bound has been reached.

p_{kUmin} represents the lowest acceptable bound on any parameter ***k***, e.g. in the case of video applications, the lower bound on the transfer rate p_{kUmin} represents the minimum frame rate for which the stream can be reassembled. RANs which cannot meet this lower bound should not be selected. The physical limitations of the device are also used to define the bounds on certain parameters.

Other simple additive weighting formulas, for example the work in [], which do not consider bounds on the parameters, suffer from a very strong parameter value out-weighing an unacceptably low valued utility parameter [16]. This is avoided in our work by including the bounds in our network selection parameters.

5.2.1.2 Back-off delay

If the users employ the same network selection algorithm with no randomness involved, and one of the networks is particularly good, all the mobile devices may end up selecting that network. If all the mobile nodes within a local area handover to the same WLAN within a short interval and there is not enough link capacity available, the link quality will drop considerably. This drop could cause all the users to simultaneously switch networks again, possibly resulting in a ping-pong effect between local networks. To resolve this problem, a backoff delay (***D***) is introduced to avoid all nodes connecting at the same time.

D is computed in (3) based on the estimated benefit of handing over to the selected network (N_s) over staying on the current network (N_{crt}), the bigger the difference in quality between them, the less delay there is. If the gain in quality is above a predefined, heuristically determined threshold, ***GT*** (Gain Threshold), no delay is introduced, since the handover is considered of high priority. If the ANM value of the selected network is not significantly larger than the current network, the handover will be delayed for enough time for other nodes to handover to the network. ***MHT*** (maximum handover time) is an interval sufficient for the nodes to perform a vertical handover, for example the value determined in [6].

$$D = \begin{cases} (MHT) * \left(1 - \left(\mathrm{ANM}_{\mathrm{App}}(N_s) - \mathrm{ANM}_{\mathrm{App}}(N_{crt})\right)\right), & ANM_{App} < GT \\ 0, & ANM_{App} \geq GT \end{cases} \quad (3)$$

5.2.1.3 Random decision

If the nodes have similar user profiles, run similar applications and have the same choice of RANs then the likelihood is that their D would end up being the same. In order to account for this a check is performed after handover to the new link. The ANM value of the newly attached network is checked against the expected ANM and if it is considerably worse a random decision is made on whether to stay on the newly established link or not. The probability to stay on the new network can be fixed or based on the drop of the ANM value for the network.

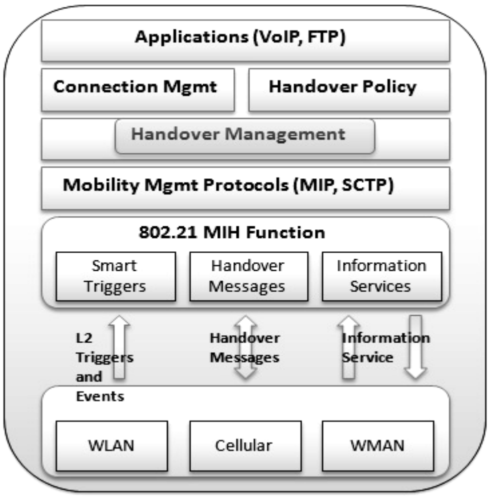

Figure 5.3 QADS Placement at Handover Management level []

5.2.1.4 *Linkage to simulation framework*

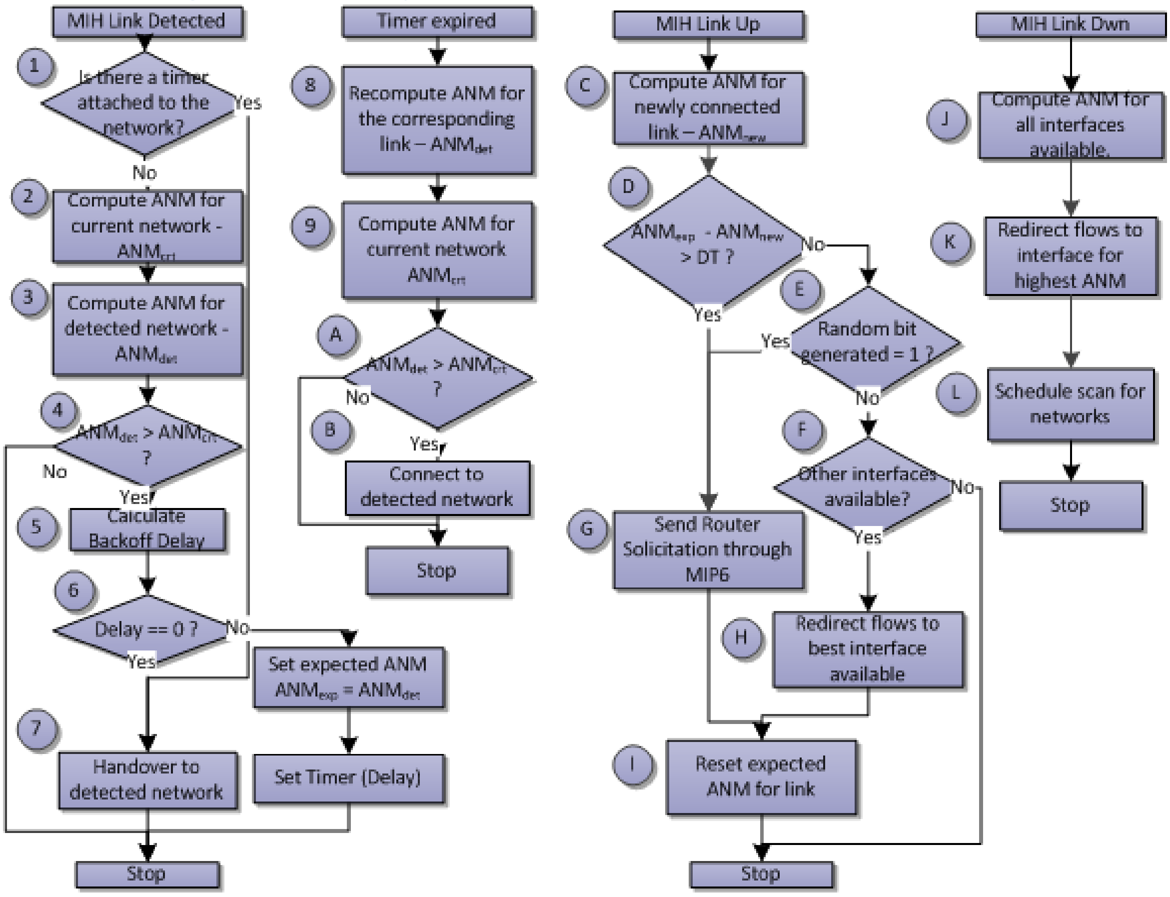

Figure 5.4 Flowchart of the QADS algorithm

The proposed strategy, QADS, is designed to work with the IEEE 802.21 MIH function and a cross-layer architecture in the terminal to get the network parameters and the application characteristics, respectively. Figure 5.4 shows how the algorithm, implemented at the handover management layer, is integrated with the MIH Function. QADS uses IEEE 802.21 signaling together with Mobile IPv6 (MIP) as the mobility management protocol. Readings of the network parameters, such as throughput, are obtained from the MIIS. The algorithm reacts to specific MIES triggers, as described below, while the MICS offers support for

network scan, link configuration and subscription to events. QADS uses minimum and maximum acceptable values for each of the network parameters. These are relevant to the application and/or to the user: such as delay for non-real time applications, jitter for real-time applications or cost for users. This thesis assumes that these values are determined in advance of the call setup. The user setup preferences may be gathered during the original application setup on the terminal or through a graphical user interface at run time [13].
The three features of the IEEE 802.21 standard are used by the algorithm. MIIS provides the network information, which is stored in a list, thus making the node aware of the networks that are available and their characteristics. MIES notifies the node of the changes in network characteristics, while MICS allows the node to react accordingly.
The QADS algorithm is implemented by providing IEEE 802.21 handles to the MIH events. Figure 2 displays the flow of the algorithm for the considered IEEE 802.21 events. Once a new link is detected, its ANM value is compared with the current network. If the new link offers a higher ANM then a handover should be performed, based on the ANM difference a backoff delay is computed, and a timer set accordingly. When the timer expires, the ANM value of the new network is checked again and if it is still higher, a connection is launched on the selected network. Once the link has been established, the link up event is triggered [2], the ANM is recomputed and compared to the expected ANM, and any loss in value is compared to the drop threshold (DT). If the link is far worse than expected, a random decision is made on whether to hand back to the original RAN or to stay on the new RAN.

5.2.2 PNQS

In this thesis, we have implemented an algorithm – which we call PNQS for Polled Network Quality-based Selection - based on a variation of the function presented by Wang et al [].The idea behind it is simple, but effective: when new networks are detected, they are added to a list of candidates and a timer triggers the reassessment of the quality of the networks every second, the network with the best quality of service being selected. If this is not the same network that the user is connected to, handover is performed.

5.2.2.1 Utility function

As in the work referenced above, PNQS uses a weighted utility function to estimate the Quality of Service. Unlike the utility function proposed by Wang et al[], this one does not take security into account, as this is not provided though the MIH and, since it depends on a lot of factors, it would be difficult to estimate. In the formula below, w_c and w_b are weights for cost and bandwidth respectively, B is the bandwidth required by the terminal and b_i is the actual bandwidth received by the terminal. Security was not considered in the simulations.

$$QoS\left(N_i\right) = w_c \ln\left(\frac{1}{C_i}\right) + w_b\, ln\left(\frac{1}{B-b_i}\right) \qquad (4)$$

There are several differences to the implementation provided here: first of all, the utility function is different in that the objective for it is to be maximized rather than minimized as is the case of the cost function in their implementation, meaning that the parameters to which logarithm is applied are inverted. Second, the advertized bandwidth is not obtainable through MIIS, so the typical rate of the network, based on its technology, was used instead. Third, because a network night be free of charge, "infinity" was approximated with a very large number, so that below a very low cost (equal 1/Very_Large_Number) the quality stays the same. However, it is unlikely that any network will ever charge 1e-15 c/Gb for their network. Since the cost is not available either, it is read from a file for each network. Additionally, re-computation of the quality of service is made right after connecting to a network, to avoid disastrous choices.

5.2.2.2 Linkage to simulation framework

PNQS is connected to the handover framework similarly to QADS, through handlers to the MIH events. It uses the MIIS and the MICS to gather information about the networks and to issue commands to the MAC layer, respectively. When a new link is detected, its cost of the network is stored, and the throughput is initialized to the typical value for the technology. The best network available from the ones associated to the interfaces is selected, and its quality of service is compared to the new one detected. If the new one offers better quality, the user connects to it. Otherwise, if the best network available is not the one currently connected to the node, flows are redirected toward that interface. Once a connection has been established (the Link Up event), re-compute the quality of service and compare to the quality of service offered by the best interface that is not connected. If the best network is not the one the device is using, the flows are redirected to that interface. When the connection goes down, and no action has been taken to initiate a new one, the network scan is started on all interfaces. The network with the highest value of the utility function is selected. When a parameter report is received for the network, the throughput information corresponding to the network is updated.

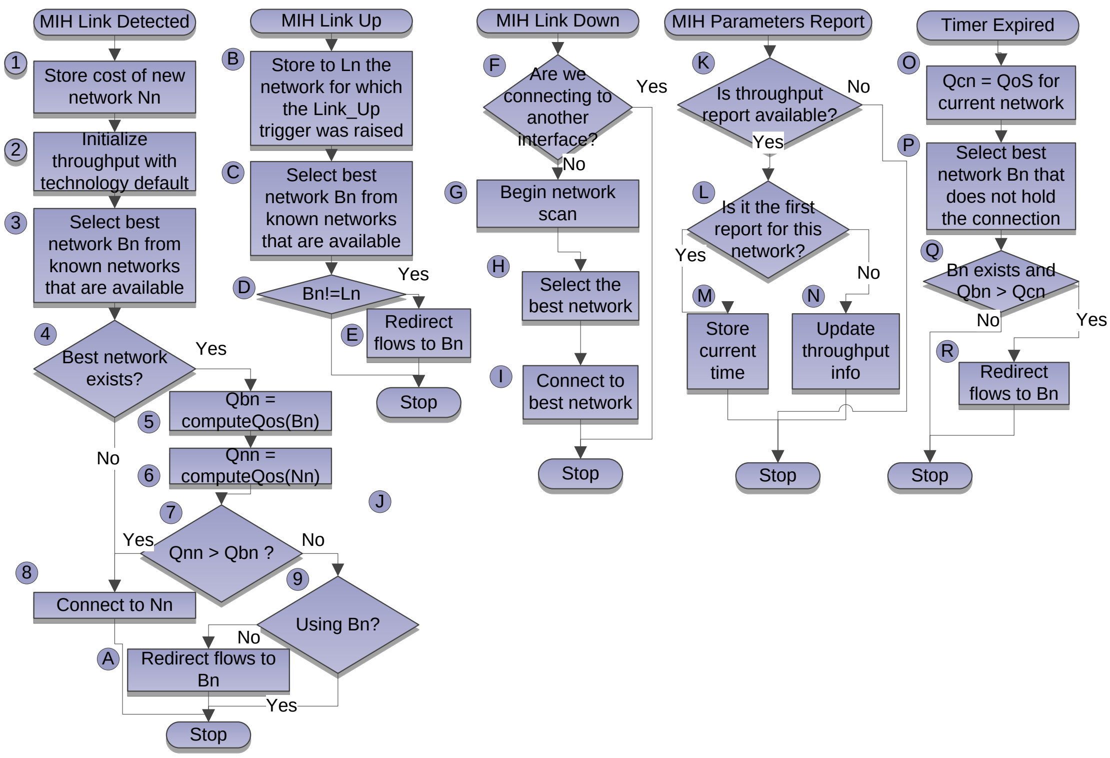

Figure 5.5 Flowchart of the PNQS algorithm

An important part of the PNQS algorithm is the timer that dictates the re-computation of the quality of service of each network. Its purpose is to ensure the best quality all the time. A drawback can be a high number of handovers. Moreover, when multiple users switch networks simultaneously, the risk of ping-pong appears, because the quality of service for the network has the same value for all the users. After all devices handover to the network, it becomes so overloaded that all the nodes switch back. The simulations in 7 prove this point.

5.3 Simulation scripts

In order to test the handover algorithms, simulation scripts were created. These scripts involve mobile nodes downloading MPEG traffic from a router or running an FTP application. Figure 5.6 shows how a connection between the multi-face node and the router is established, how the handover management object is connected to the wireless interfaces and how the traffic can get from the router to the mobile device. The wired setup is simple, the two routers are connected, and each of them is connected to a wireless base station, one of which is the access point for a WLAN network. Each of the two base stations is modeled by a Node instance, which is connected to an MAC object. The mobile node is represented by three nodes: two interface nodes and a multi-face node that binds them together. Each of the two interfaces has a medium access control object (marked as Terminal MAC on the image), which is connected to the corresponding base station's MAC. The multi-face node has a MIHAgent installed, which is connected to the interfaces, as well as the MACs. The flow management goes through the Handover object, which is connected to the MIHAgent. To get the traffic from the router to the mobile node, a UDP agent is connected to the router, and a Null Agent is connected to the multi-face node. A duplex link is set from the UDPAgent to the NullAgent. The CBR application runs over the UDP agent. The flow is first set to go through the WiMAX interface, but can be later switched to the WLAN.

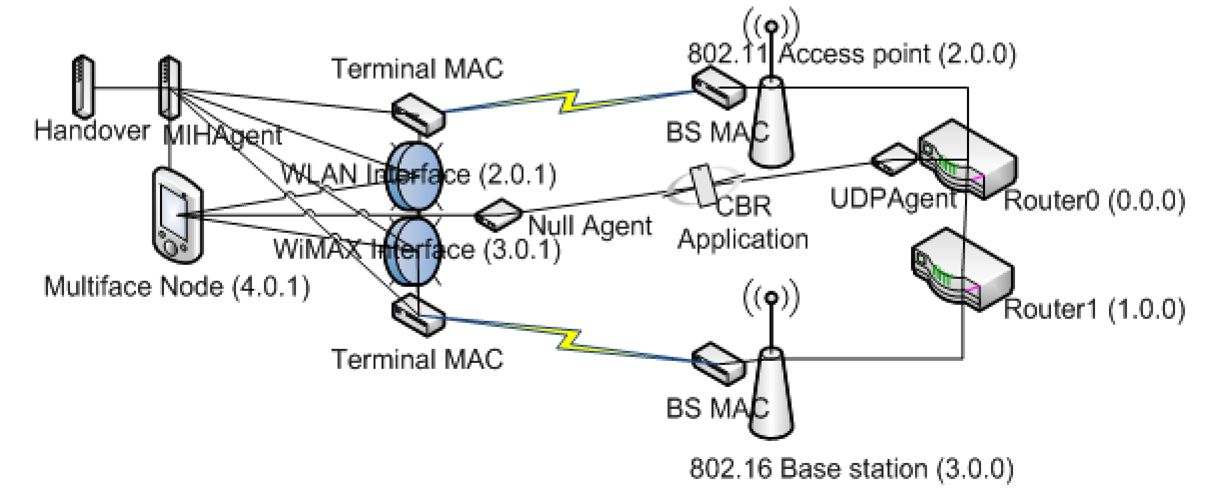

Figure 5.6 Communication mechanisms in the simulation scenarios

5.4 Automatic data extraction tools

As seen in the previous section, the simulation does not yield results instantly so that additional programs are required to obtain the viewable results. The data extraction process is displayed in Figure 5.7. It all starts from the Tcl Scipts, which are ran by the simulator to obtain a trace file with the format descrbed in section 4.3. This trace needs to be processed line by line, and the relevant parameters of the traffic between a given source node and a given destination node need to be computed. In the current thesis, two aspects are important: the throughput between each mobile interface and its associated base station throughout the simulation and the packet loss at each base station. Perl scripts were written to calculate the throughput between an interval and a base station at equally distributed intervals by filtering only the data packets with the appropriate source and destination MAC id – the files obtained, called interface trace files, hold a timestamp and the values for the parameters calculated at that time on each line. The data loss at base stations is obtained by filtering only the dropped packets with the right destination. Although the traffic for each interface can now be plotted as a graph, some statistics on the data are also required. The distilled takes the interface trace files and calculates the average, standard deviation and the total traffic for each parameter for all mobiles nodes as well as the total data loss. Because a node has more interfaces, which might even be used simultaneously, these were added before the statistics are computed. Also, the program can run with runtime traces, since recursive formulas were used for the average and standard deviation and therefore integration with the simulator was possible.

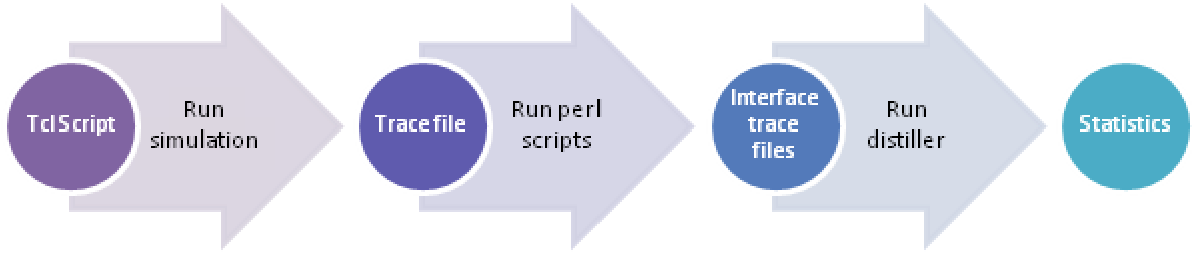

Figure 5.7 Data extraction process

6 Implementation

6.1 Changes to the simulator code

As illustrated in the previous section, the NIST add-on could not be applied directly to the NS 2.29 code. While most of the methods, even classes, could be kept unchanged from one set of sources or another, some were not as easily dealt with. Several illustrative examples of these are presented in the merged code section. The bug fixes section documents the adjustments made to acquire better standard conformance or because errors or incorrect functionality was encountered at program runtime.

6.1.1 Merged code

6.1.1.1 Alternative 802.11 implementations

Regarding the 802.11 implementation, for which duplicate functionality was introduced with the patch and the 2.33 version of the simulator, a design decision was made to keep both implementations when merging the two. The reason why they were not merged into one class was that the 802.11 implementation provided in version 2.33 does not offer a better model. On the other hand, the implementation in the NIST add-on is connected to the MIH framework, so it cannot be used on its own, which might not correspond with the user requirements. Therefore, the module created for this thesis allows the use of objects of both classes (even in the same script). The default implementation is the one in the NIST patch, since the purpose of this thesis is, after all, the study of handover. A factory class was added to allow the users the creation of an MIH-capable or MIH-independent 802.11 MAC agent.

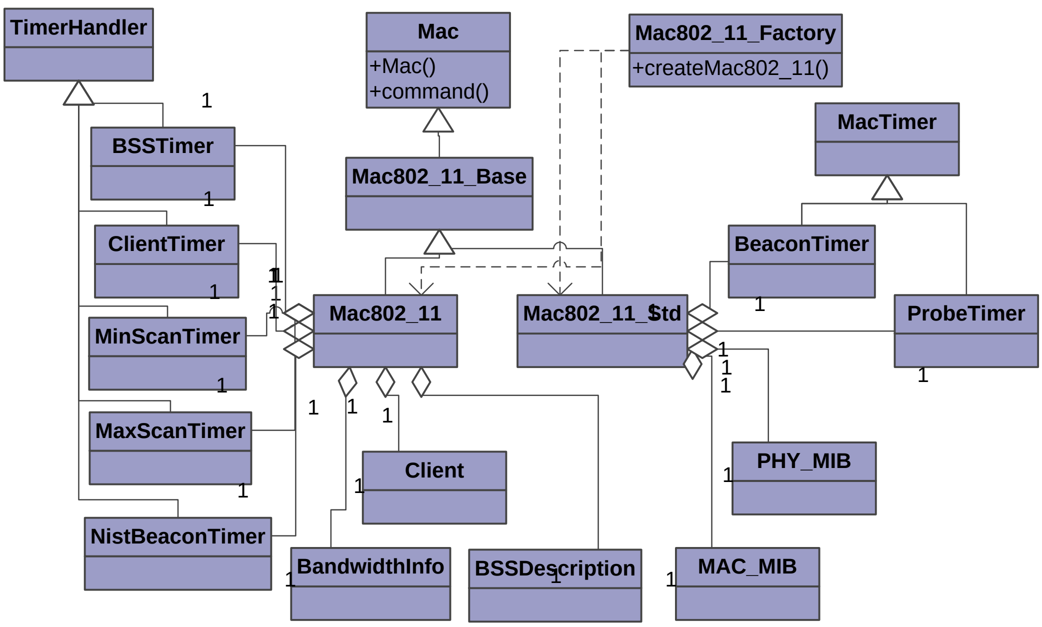

Figure 6.1 Integration of the two distinct implementations of the 802.11 standard

```
// Implementation of the factory class
class Mac802_11_Factory{
private:
       bool _mih_enabled;
public:
```

```
	Mac802_11_Factory(){
		_mih_enabled = false;
		bind_bool("mih_enabled_", &_mih_enabled);
	}
	Mac802_11_Base *createMac802_11(){
		return _mih_enabled? new Mac802_11() : new Mac802_11_Std();
	}
};
// Binding of the factory class
static class Mac802_11_FactoryClass : public TclClass {
public:
	Mac802_11_BaseClass() : TclClass("Mac/802_11/Factory") {}
	TclObject* create(int, const char*const*) {
		return (new Mac802_11_Factory());
	}
} class_Mac802_11_Factory;
```

```
# Tcl code that creates a MIH-capable 802.11 MAC object
set factory_ [new Mac/802_11/Factory]
$factory_ set mih_enabled_ true
set mac $factory createMac802_11
# or simply
set mac [new Mac/802_11]
```

```
# Tcl code that creates a MIH-capable 802.11 MAC object
set factory_ [new Mac/802_11/Factory]
set mac $factory createMac802_11
```

6.1.1.2 Wireless channel updates

The wireless channel implementation in the integrated module made for this thesis combines the corrections brought by both versions of the simulator. The calcHighestAntennaZ method in the WirelessChannel class selects the highest point at which an antenna can send/receive signal, taking into account a set of given physical models. In version 2.33, this method is changed so as to perform the same operations for the extended wireless physical models as for the standard wireless ones. This is done by dynamic casting and type checking– a questionable design, which will perhaps require refactoring in future versions, but that fulfills its purpose for now. The method sendUp in the same class sends a packet to all the mobile nodes that are in position, given the topology, to receive it. It relies on the antenna capabilities computed in the previously mentioned function. It was modified in the NIST add-on to take interference into account if the flag is set, and to filter by frequency otherwise. The method that returns the nodes that should react to a certain transmission, getAffectedNodes, is modified to filter nodes according to their coordinates in one pass. The code below is the result of the merge:

```
class WirelessChannel{
//…
public:
	void calcHighestAntennaZ(Phy *tifp) {
		double highestZ = 0;
		Phy *n;
		for(n = ifhead_.lh_first; n; n = n->nextchnl())
			if(dynamic_cast<WirelessPhyExt*>(n))
				//…
			else
				if (dynamic_cast<WirelessPhy*>(n))
					//…
				else highestZ = 0;
		highestAntennaZ_ = highestZ;
		if (dynamic_cast<WirelessPhyExt*>(tifp))
```

```
                    //…
            else
                    if (dynamic_cast<WirelessPhy*>(tifp))
                            //…
                    else  distCST_ = DBL_MAX;
    }

    void sendUp(Packet* p, Phy *tifp) {
            Scheduler &s = Scheduler::instance();
            Phy *rifp = ifhead_.lh_first;
            Node *tnode = tifp->node();
            Node *rnode = 0;
            Packet *newp;
            double propdelay = 0.0;
            struct hdr_cmn *hdr = HDR_CMN(p);
            /* list-based improvement */
            if(highestAntennaZ_ == -1) {
                    calcHighestAntennaZ(tifp);
            }
            hdr->direction() = hdr_cmn::UP
            if (GridKeeper::instance()) {
                    //...
            }
            else {
                    //…
                    for (i=0; i < numAffectedNodes; i++) {
                            rnode = affectedNodes[i];
                            rifp = (rnode->ifhead()).lh_first;
                            if(rnode != tnode){
                                    WirelessPhy *wrifp = (WirelessPhy *)rifp;
                                    if (!wrifp->isInterferenceEnabled()) {
                                            WirelessPhy *wtifp = (WirelessPhy *)tifp;
                                            if(wtifp->getFreq() !=  wrifp->getFreq())
                                                    continue;
                                    }
                                    newp = p->copy();
                                    propdelay = get_pdelay(tnode, rnode);
                                    rifp = (rnode->ifhead()).lh_first;
                                    for(; rifp; rifp = rifp->nextnode()){
                                            s.schedule(rifp, newp, propdelay);
                                    }
                            }
                    }
                    //…
            }
            //…
    }

    MobileNode ** getAffectedNodes(MobileNode *mn, double radius, int *numAffectedNodes) {
            double xmin, xmax, ymin, ymax;
            int n = 0;
            MobileNode *tmp, **list, **tmpList;
            if (xListHead_ == NULL)
                    //…
            xmin = mn->X() - radius;
            xmax = mn->X() + radius;
            ymin = mn->Y() - radius;
            ymax = mn->Y() + radius;
            tmpList = new MobileNode*[numNodes_];
            for(tmp = xListHead_; tmp != NULL; tmp = tmp->nextX_) {
                    tmpList[n++] = tmp;
            }
            for(int i = 0; i < n; ++i)
                    //…
            n=0;
            for(tmp = xListHead_; tmp != NULL; tmp = tmp->nextX_)
                    if(tmp->Y() >= ymin && tmp->Y() <= ymax && tmp->X() >= xmin &&
                    tmp->X() <= xmax)
                            tmpList[n++] = tmp;
            list = new MobileNode*[n];
            memcpy(list, tmpList, n * sizeof(MobileNode *));
            delete [] tmpList;
            *numAffectedNodes = n;
            return list;
    }
}
```

```
Significance of colors:
Code that remained unchanged since version 2.29
Code that changed in the NIST add-on
Code that changed in version 2.33 of the simulator
Code impacted by both changes
```

6.1.2 Bug fixes

6.1.2.1 Multiple node handover runtime error

The handover model implemented in the NIST add-on was supposed to work well for any number of mobile nodes. However, when a script involving more mobile nodes was run, the simulator crashed with the following error:

```
--- Classfier::no-slot{} default handler (tcl/lib/ns-lib.tcl) ---
        _oXX: no target for slot 1023
        _oXX type: Classifier/Addr
        content dump:
        classifier _oXX
                0 offset
                22 shift
                1023 mask
                1 slots
```

After searching for the cause of the error, it was concluded that the MIH Agent was programmed to respond to all capability discovery requests. Base stations do not send such messages, and if they receive them, they can send replies successfully. Mobile nodes, on the other hand, should not reply to these messages. In the NIST add-on, no differentiation is made between the MIH agent of a mobile device and the one belonging to an access point, so the mobile device tries to respond to a broadcasted capability discovery of another mobile device. However, because at start of the simulation the node is not connected to a network, it cannot reply to its peer, so an error occurs. When a single node is used, this scenario is not possible because the node never receives the request.

To correct this problem, a boolean attribute was added to the state of the MIHAgent. This attribute determines if the node responds to capability requests, and it is true for base station MIH agents and false for mobile node MIH agents. The flag is tested in the method `MIHAgent::recv_cap_disc_req`, thus the capability discovery reply is only sent by BSs.

6.1.2.2 Missing parameter report

The handover algorithms presented in this thesis, QADS and PNQS require an estimation of the throughput offered by each network. The NIST patch offers statistics that reach the handover agent in the form of link parameter reports. In theory, when a mobile device is connected to a network, it should periodically receive throughput information from the MAC layer. However, because error correction is not implemented in the NIST add-on, none of the received packets are counted towards the total traffic statistic. The WLAN MAC layer – in this case the Mac802_11 class – must be changed so that data packets that are received while the interface is active, and that contain actual data are counted. In the code below, right after receipt of a packet, the newly added method update_data_throughput is called. This method filters valid data packets, updates the statistics, creates a new parameter report and sends it to the MIH agent by calling send_link_parameters_report.

```
void Mac802_11::recv(Packet *p, Handler *h)
{
        struct hdr_cmn *hdr = HDR_CMN(p);
        assert(initialized());
        update_data_throughput(p);
        //… do everything else
}
void Mac802_11::update_data_throughput(Packet *p)
```

```
{
        if(!tx_active_) {
                hdr_cmn *ch = HDR_CMN(p);
                hdr_mac802_11 *mh = HDR_MAC802_11(p);
                u_int8_t  type = mh->dh_fc.fc_type;
                double size = ch->size()*byte_size;
                int dst = ETHER_ADDR(mh->dh_ra);
                if((type == MAC_Type_Data) && (size<max_data_packet_length) &&
                         (size<min_data_packet_length) && (dst == addr())){
                        threshold_action_t action = wlan_th_watch_.update (size, NOW);

                        link_parameter_type_s param;
                        union param_value old_v, new_v;
                        param.link_type = LINK_802_11;
                        param.parameter_type = LINK_GEN_RX_DATA_THROUGHPUT;
                        old_v.data_d = wlan_th_watch_counter;
                        wlan_th_watch_counter++;
                        new_v.data_d = wlan_th_watch_counter;
                        if(wlan_th_watch_required<=   wlan_th_watch_counter){
                                send_link_parameters_report  (addr(),  bss_id_,  param,  old_v,
new_v);
                        }
                }
        }
}
Significance of colors:
Unchanged code
Added code
```

6.1.3 Extensions to the handover framework

This subsection describes the extensions that were applied to the handover framework – depicted in Figure 4.1 for the simulator to run scenarios that use the QADS and PNQS algorithms. Figure 6.2 shows a diagram of the extended framework. The abstract class Handover models the generic concept of MIH-assisted handover. The MIHScan class is responsible for wrapping the scanning process – which occurs at the MAC layer – so that it appears transparent to the handover management layer. MIHInterfaceInfo stores information related to the networks associated with the interfaces of the mobile device. NDAgent is responsible for neighbor discovery. The classes HandoverQad and HandoverPnq model the QADS and PNQS algorithms. The class HandoverPnq depends on the implementation of the timer used for re-evaluation of the quality of service brought by the networks, which is modeled bythe RecalculateTimerPnq class. On the other hand, HandoverQad relies on a series of classes that model the application requirements and user preference and the network data statistics. MnApp holds information about both the application requirements and user preferences – in this case, the information consists in the weights attributed to each network parameter and the quality of service function attributed to each network. Its subclasses are the particular types of applications that were used in the scenarios. The network data class stores the network information as it comes from the MAC layer, and passes it on to the QADS algorithm on request. The internal working of the handover classes is further described in 6.2.

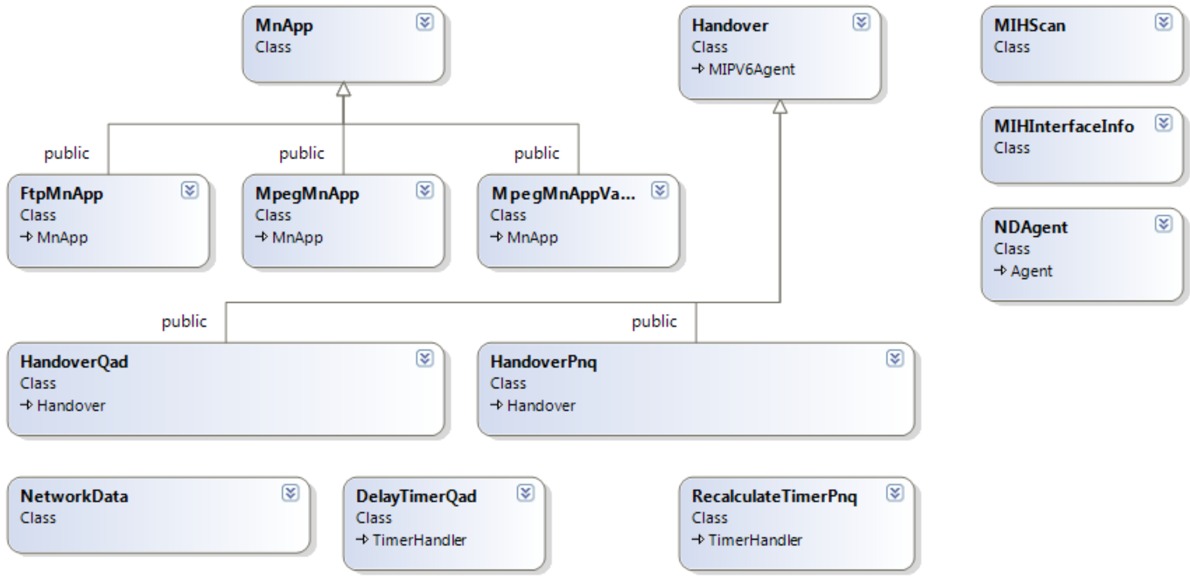

Figure 6.2 Extensions brought to the handover framework for QADS and PNQS

6.2 Implementation of the handover algorithms

So far, the design of the QADS and PNQS algorithms, as well as their placement in the handover framework provided by the NIST add-on was specified. It is time to delve deeper into the inner-workings and interoperability of the classes displayed in Figure 6.2, and see how the operations shown in the flowcharts of figures 5.4 and 5.5 translate to actual code.

6.2.1 Quantified adaptive delay selection

Starting off with QADS, as it is the main focus of this work, the sequence diagram below shows how the handover management layer reacts to a Link_Detected event. This particular event was selected to be expanded because it involves all of the relevant objects that make up the internal structure of an implementation of the QADS handover algorithm. To explain the objects in the diagram, mih_ is a reference to the MIH Agent, through which the MIES events are received and through which the MICS commands are issued. When a parameter report is received, the information is stored into the netData object, to be used later on. netData simply maps network data to its corresponding list of parameter values. Its main purpose is to provide an additional abstraction, because not all data about the networks is available through MIH. MnApp stores user preferences and application requirements. This object is the one that actually computes the ANM value for a given set of network parameter values, and so contains the implementation of the ANM function. The pending_timers object maps a list of link identifiers to their respective delay timers (a delay timer is the one that dictates how fast the handover to a given network must occur, based on its expected quality of service provision).

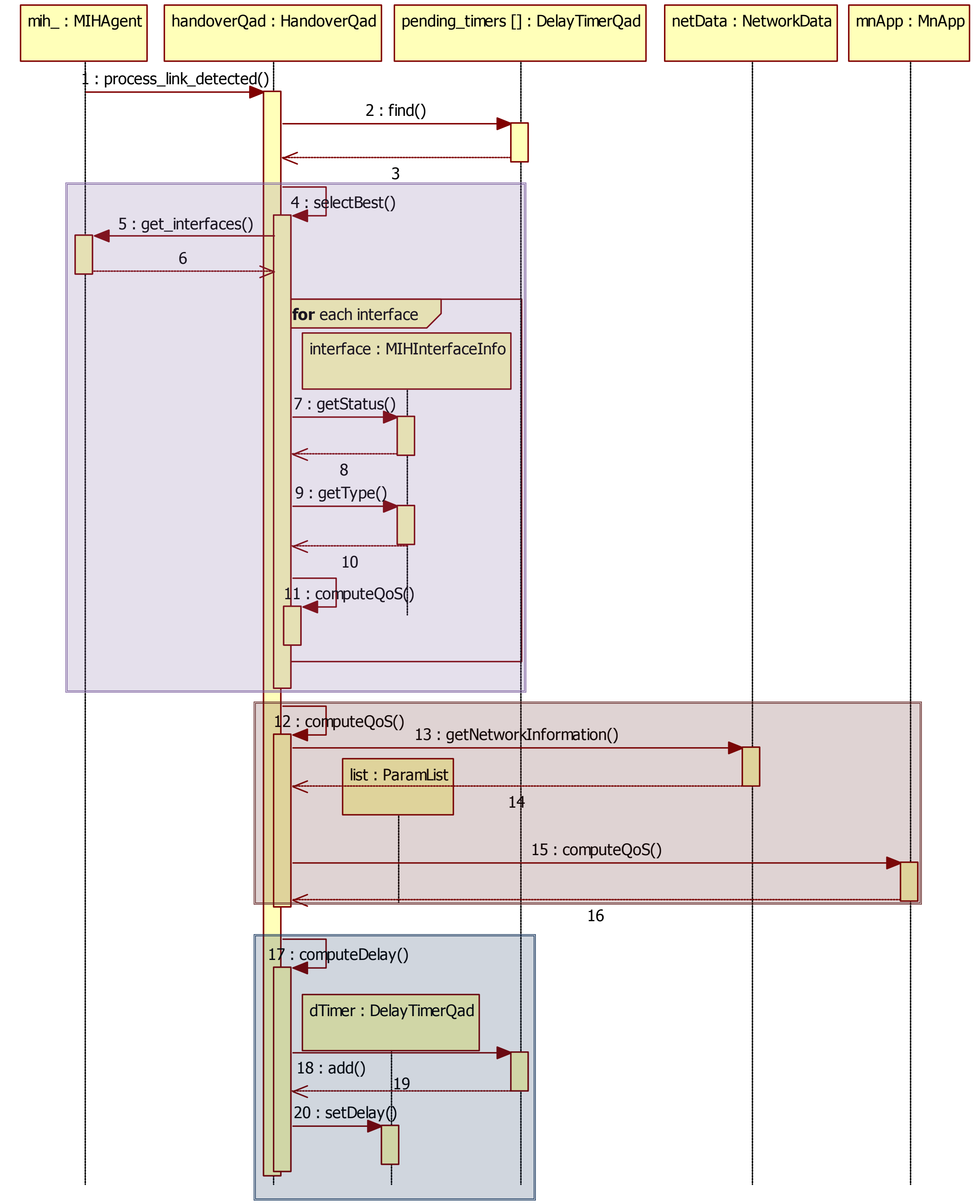

Figure 6.3 Sequence diagram of the MIH Link Detected event handler introduced by QADS

selectBest() – chooses the best interface from the already existing ones

computeQos() – calculates the estimated quality of service for a given link identifier

computeDelay() – calculates the delay in handover based on the value of the quality of service of a link

When a new link is detected, the first thing that the algorithm does is to check if there is a timer associated with that link. If there is not, the best network that is available is selected, to be compared with the newly detected one. For each interface provided by mih_, the quality of service is computed and the one with the highest value is selected. The quality of service of the new link is calculated by mnApp, based on the parameters received from netData. In this example, the quality offered by the new network is higher than the existing one, but not high enough to induce instant connect, so a delay timer is added for the link identifier in question, and added to pending_timers.

No.	Pseudo-Code
	MIH_Link_Detected (event)
1	if pending_timers.find(event.MAC_TERMINAL_ID) then
2	{best, qos} = selectBest()
3	q = computeQos(event.LINK_IDENTIFIER)
4	if q>qos then
5	d = computeDelay(q,event.LINK_IDENTIFIER)
6	if d!=0 then qos_expected[event.LINK_IDENTIFIER] = q
7	else connect(event.LINK_IDENTIFIER)
	Timer_Expired (expiredTimer)
8	q = computeQoS(expiredTimer.link_id)
9	{best, qos} = selectBest()
A	if q>qos then
B	connect(expiredTimer.link_id) pending_timers.remove(expiredTimer)
	MIH_Link_Up (event)
C	q = computeQoS(event.LINK_IDENTIFIER)
D	if qos_expected[event.LINK_IDENTIFIER]-q>drop_threshold then
E	if qadRandom.generate==true
F	if {best,qos} = selectBest then
G	redirectFlows(best)
H	else send_rs(mih_.getMac(event.MAC_TERMINAL_ID)
I	qos_expected.remove(event.LINK_IDENTIFIER)
	MIH_Link_Down (event)
J	{best, qos} = selectBest()
K	if best!=null then redirectFlows(best)
L	mih_.startScan()

Table 6.1 Pseudo-code of the QADS algorithm (numbers correspond to the blocks in Figure 5.4)

6.2.2 Polled Network Quality of service Selection

Moving on to PNQS, the sequence diagram in Figure 6.4 shows the handler to the MIH link detected event. Unlike the QADS handover, PNQS uses two objects that store the cost and throughput information. The cost information is stored when the network is first detected, as it is assumed that the cost is not likely to change during a simulation. When a parameters report is received at the handover layer, the information is added to throughputInfo. When a new link is detected, the cost information is stored and the throughput information is set to its default value for the technology of the network in question. The best network available is selected, and the quality of service for the newly detected network is calculated – the necessary information is taken from costInfo and throughputInfo. In this scenario, the newly detected network is perceived as better than the existing one, so the device connects to it. The interface in the mobile device that is associated with the selected network is passed to the mih_ layer, so that the link can be configured.

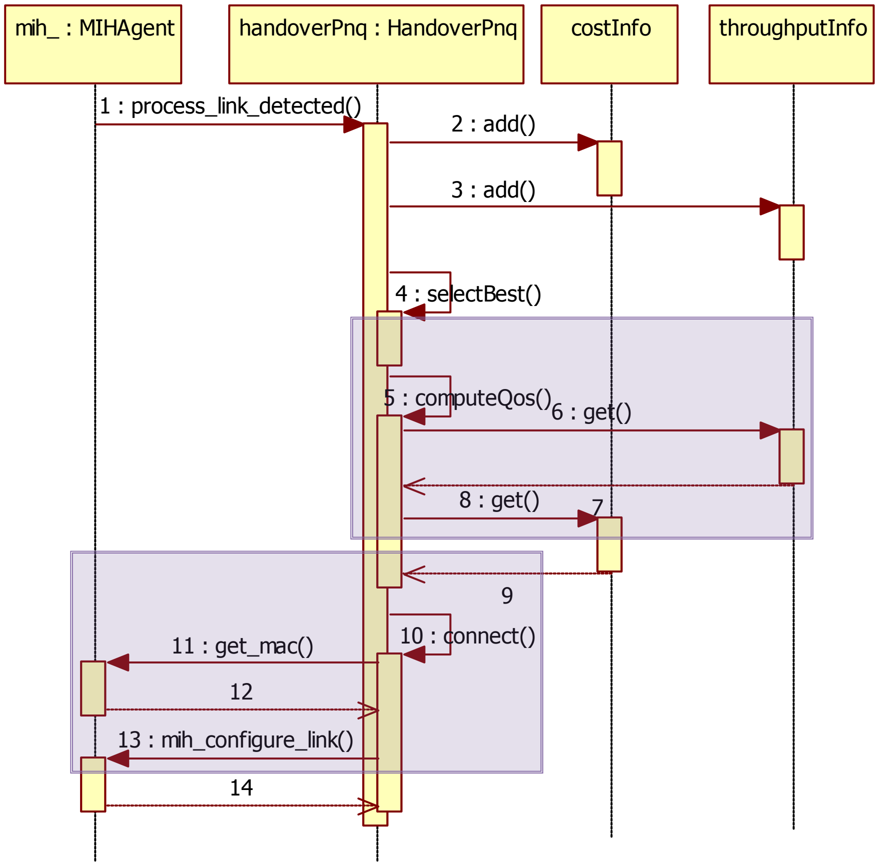

Figure 6.4 Sequence diagram of the MIH Link Detected event handler introduced by PNQS

The table below shows the pseudo-code corresponding to the methods corresponding to the PNQS handover algorithm.

No.	Pseudo-Code
	MIH_Link_Detected (event)
1	costInfo[event.LINK_IDENTIFIER] = getCostOfNetwork(LINK_IDENTIFIER)
2	throughputInfo[event.LINK_IDENTIFIER]=getDefThroughput(event.LINK_TYPE)
3	best = selectBest()
4	if best then
5	qbn = computeQos(best.LINK_IDENTIFIER)
6	qnn = computeQos(event.LINK_IDENTIFIER)
7	if qnn>qbn then
8	connect(event.LINK_IDENTIFIER)
9	else if crt.LINK_IDENTIFIER == best.LINK_IDENTIFIER
A	redirectFlows(best.LINK_IDENTIFIER)
	MIH_Link_Up (event)
B	ln = event.LINK_IDENTIFIER
C	best = selectBest()
D	if ln!=best.LINK_IDENTIFIER then
E	redirectFlows(ln)
	MIH_Link_Down (event)
F	if connectingMac_==null then
G	mih_.startScan()

H	`    best = selectBest()`
I	`    Connect(best.LINK_IDENTIFIER)`
	`MIH_Parameters_Report (report)`
K	`if report.has(PKT_NUM) then`
L	`    if throughputInfo[report.LINK_IDENTIFIER]!=null then`
M	`throughputInfo[report.LINK_IDENTIFIER]=report.get(PKT_NUM)*SIZE (CRT_TIME-get_time_reference())`
N	`    else store_time_reference(CRT_TIME)`
	`Timer_expired (report)`
O	`qcn = computeQos(crt.LINK_IDENTIFIER)`
P	`best = selectBest()` `qbn = computeQoS(best.LINK_IDENTIFIER)`
Q	`if qbn>qcn then`
R	`    redirectFlows(best.LINK_IDENTIFIER)`

Table 6.1 Pseudo-code of the PNQS algorithm (numbers correspond to the blocks in Figure 5.5)

6.3 Simulation scripts

The Tcl scripts used for the present thesis simulate a scenario where the mobile nodes are connected to a WiMAX network, and move across its area of coverage and at some point they enter the coverage area of a WLAN network. The scripts follow these steps:

- Set the default parameters for 802.11, 802.16 and the handover
- Create the simulator and the topography
- Setup the hierarchical routing
- Creation of the multi-face nodes
- Creation of the AP and the 802.11 interfaces
- Creation of the BS and the 802.16 interfaces, the classifiers and the schedulers
- Adding the interface nodes to the multi-face nodes
- Install neighbor discovery on the base stations and the interfaces
- Install interface managers (the handover objects) into multi-interface nodes
- Install the MIH agent in the multi-face nodes
- Connect the MIH agent with the handover manager
- Connection of the MIH agents to the MAC agents
- The wired routing is created
- The nodes are positioned the movement is scheduled
- Creation of the traffic, adding of the flow to the handover manager

The code for the handover-specific operations is written below:

Operation	Tcl Code
Creation of the multi-face nodes	`$ns_ node-config -multiIf ON for {set i 0} {$i < $nmn} {incr i} {` `set multiFaceNode($i) [$ns_ node 4.0.$i] }`

Addition of the interface nodes to the multi-face nodes	`for {set i 0} {$i < $nmn} {incr i} {`
	`$multiFaceNode($i) add-interface-node $iface11($i)`
	`$multiFaceNode($i) add-interface-node $iface16($i)`
	`}`

Neighbor discovery installation – shown for WLAN	`set ndbs11 [$bs11 install-nd]`
	`$ndbs11 set-router TRUE`
	`$ndbs11 router-lifetime 1800`
	`for {set i 0} {$i < $nmn} {incr i} {`
	`set ndif11($i) [$iface11($i) install-nd]`
	`}`

Interface	`for {set i 0} {$i < $nmn} {incr i} {` `set handover($i) [new`

manager installation – for WLAN	`Agent/MIHUser/IFMNGMT/MIPV6/Handover/Handover2]`
	`        $multiFaceNode($i) install-ifmanager $handover($i)`
	`        $ndif11($i) set-ifmanager $handover($i)`
	`        $handover($i) nd_mac $ndif11($i) [$iface11($i) set mac_(0)]`
	`}`

MIH agent installed in multi-face nodes	`for {set i 0} {$i < $nmn} {incr i} {`
	`        set mih($i) [$multiFaceNode($i) install-mih]`
	`        $mih($i) set accept_cap_req_ 0`
	`        $handover($i) connect-mih $mih($i)`
	`}`

Connection of the MIH agent to the MACs – for WLAN	`set mih_bs11 [$bs11 install-mih]`
	`[$bs11 set mac_(0)] mih $mih_bs11`
	`$mih_bs11 add-mac [$bs11 set mac_(0)]`
	`for {set i 0} {$i < $nmn} {incr i} {`
	`        [$iface11($i) set mac_(0)] mih $mih($i)`
	`        $mih($i) add-mac [$iface11($i) set mac_(0)]`
	`}`

Flow added to the handover manager	`$handover($i) add-flow $app0($i) $udp0($i) $iface16($i) 1`
*** The operations shown only for the WLAN also apply to the WiMAX**	

6.4 Data extraction tools

6.4.1 Overview of the Distiller

The distiller is based on two types of objects, traces and statistics. A trace is responsible with holding data for a single parameter, and providing the value the parameter had at a given time. The statistics register at the traces, and are notified every time a new value becomes available. The Compute and NewValue methods of the statistics class are abstract and needs to be implemented in the subclasses. The NewValue function is called when a new value becomes available, and the Compute method is called when the statistic for all the values so far is needed. The classes SumStatistic, StdevStatistic and AverageStatistic are the concrete subclasses of the Statistic class, and are responsible with calculating the sum, standard deviation and average of a parameter. An instance of the class FileTrace gets the values of a parameter from one of the interface trace files generated by the Perl scripts, while the sum trace is responsible with the calculating the value of a parameter perceived by a mobile device, by summing the values of the parameter at a given moment from all its interfaces. The subclasses of SimTrace implement the methods Current – which returns the current value of the parameter, CurrentTime – which returns the time to which the value of the parameter corresponds, Reset – which rewinds the trace, Step – which moves forward in the trace and Synchronize – which makes sure that two traces point to the same moment in time.

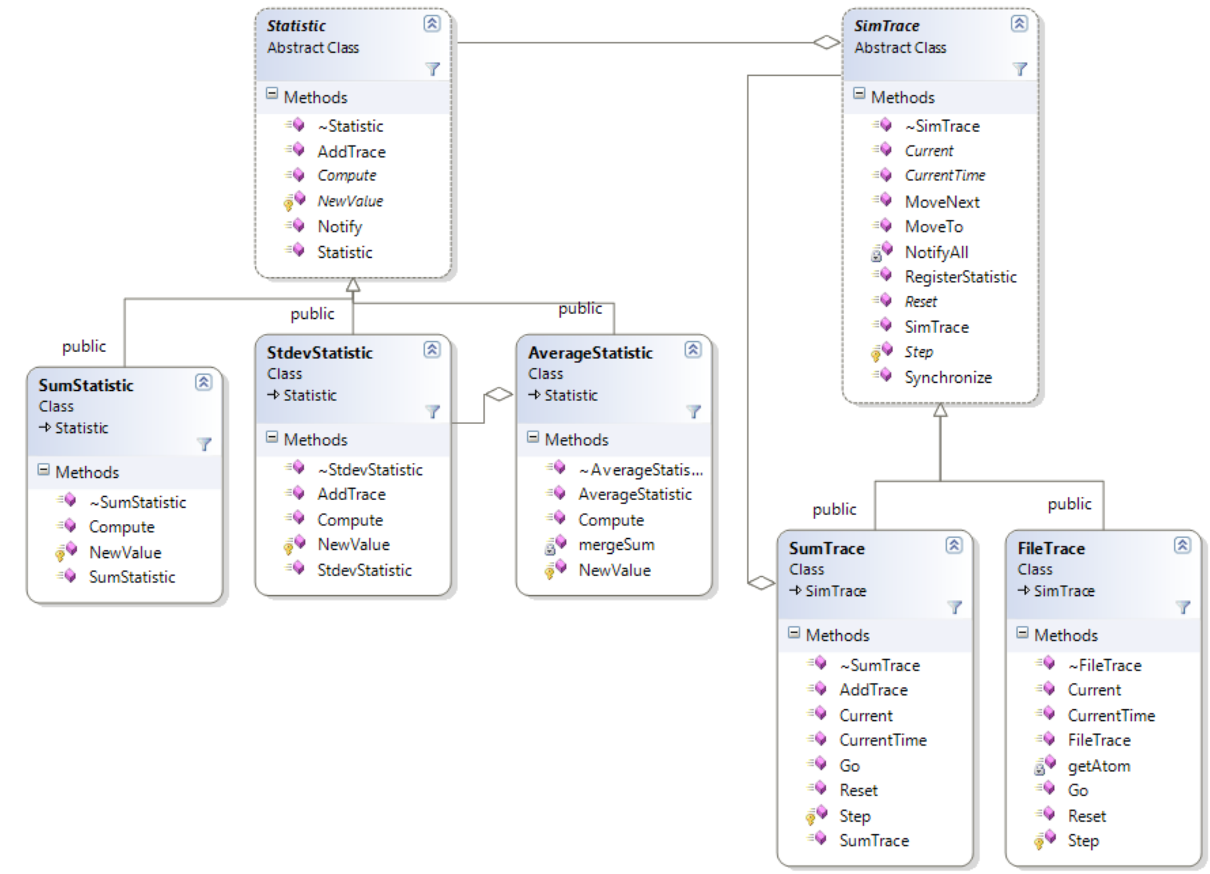

Figure 6.5 Class diagram of the Distiller

6.4.2 Parameter computation

The sequence diagram in Figure 6.6 shows how a statistic computation of data that is in two trace fies is done. The value of the parameter at a given time is made up of the sum of the two values for the parameter as fiven by the file traces at that time. A SumTrace object is used to model the final value and two FileTrace objects model the two sets of data. The statistic is computed by a StdevStatistic object that makes use of an AverageStatistic object, because the recursive formula for standard deviation at step k makes use of the average value of the k numbers in the series. As long as none of the trace files are empty, sumTrace steps through them and notifies stdev that new values are available. At that point, stdev requests the new value, so sumTrace requests the values from the file traces, adds them up and passes the result to stdev, which adds this object to the statistic calculation by calling NewValue. In this method the recursive formula is applied, but not before the new value is passed to the average statistic and the current value of the average is retrieved.

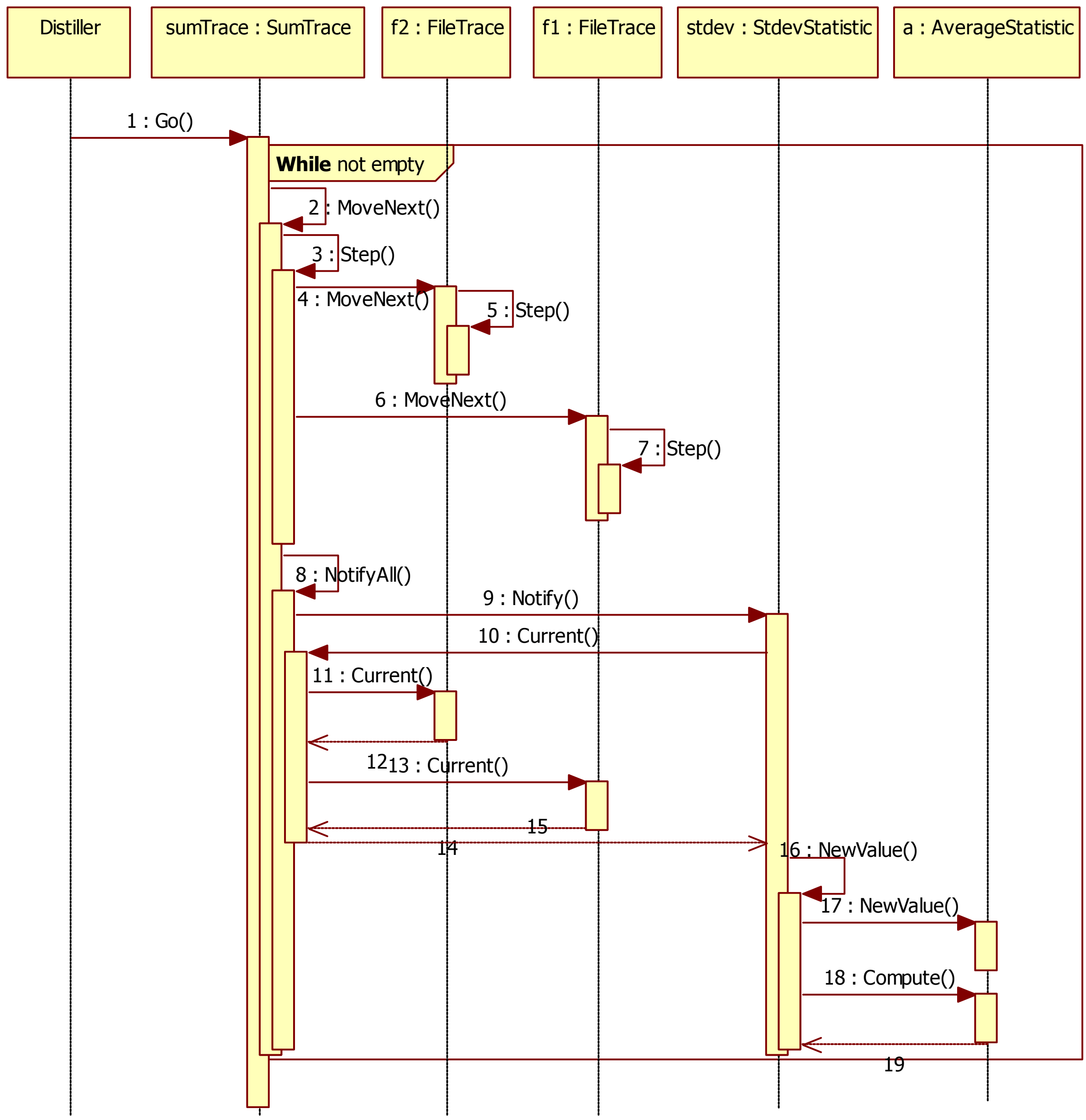

Figure 6.6 Sequence diagram of a statistic computation

7 Testing and evaluation

7.1 Configuration and special requirements

The proposed algorithm was simulated using Network Simulator version 2.29 [] with the NIST mobility add-on [10]. The simulator was further extended with an implementation of the handover management layer which supports three handover algorithms, QADS, the Always Cheapest Selection (ACS) and a third QoS-based algorithm named in this thesis as Polled Network Quality-based Selection (PNQS). The simulation uses a 0.8 Mbps Constant Bit Rate (CBR) Application to deliver MPEG 4 content over UDP [7]. Throughput and cost were considered two of the most relevant network parameters for this application, each having a 0.5 weighting in the ANM function. Since the throughput information is not readily available from the networks in the MIH NIST add-on, the throughput is predicted based on data received through the MIIS. Two other algorithms were simulated. ACS always selects the cheapest network available regardless of other characteristics, while PNQS uses a QoS function to establish which of the networks is best for a given situation.

7.2 Testing scenarios

The test scenario considered is one in which students are moving across a campus from classroom A to room B, located 400m away. The setup is shown Figure 7.1. The speed of the nodes is 1mps (a typical slow walking speed [11]) and the distance between them and direction, as they move, is kept the same. The setup contains two networks, a WLAN and a WiMAX with coverage ranges of 85m and 500m respectively (typical ranges). All of the nodes have both WLAN and WiMAX interfaces, and use MIPv6 for mobility management. The nodes start and remain within the WiMAX coverage and they cross the WLAN in their movement from point A to point B.

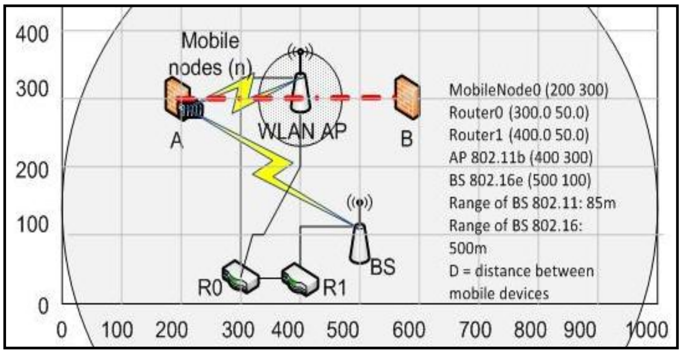

Figure 7.1 Simulated setup

In the simulations the WLAN is free to use, while 3€/Gb [8] is charged for the WiMAX. The maximum acceptable price to the user is set as 10€/Gb ($\boldsymbol{p}_{costUmin}$), the minimum cost 0 cent offers the highest utility ($\boldsymbol{p}_{costUmax}$). A maximum acceptable handover interval of 500 ms was set [6]. For throughput, $\boldsymbol{p}_{throughputUmin}$=0.064 Mbps [] and $\boldsymbol{p}_{throughputUmax}$=0.8Mbps. The other thresholds used are Gain Threshold ***GT***=0.5, Drop Threshold ***DT***=0.3, while the probability of staying on an unsuitable new network is 0.5.

Five seconds after the start of the simulation the mobile devices begin to download MPEG 4 multimedia content from router 0 (download rates vary). After 10 seconds, the nodes start to move in the direction of destination point B. From approximately second 127, the nodes gradually come into the range of the AP. Depending on the handover algorithm employed in the terminals a number of nodes may handover to the WLAN. The nodes continue towards room B crossing the WLAN cell, and at second 285 the nodes begin to leave the coverage of the AP, at which point nodes which were on the WLAN loose their WiFi connection and hand back to the BS of the WiMAX network.

7.2.1 Test-case I: five nodes in single line, ACS and QADS

In this case five users move in a single line, with a range of distances between them in different test runs (0.1m, 0.5m, 5m), as displayed by node alignment (a) in Figure 7.2. The download rates used were 0.8Mbps and 1.5Mbps in different runs. In the case of ACS, all the nodes handover from the WiMAX to the WLAN when it is available, thus placing a heavy load on the WLAN. On the other hand, when QADS is used, some nodes may remain on the WiMAX as a result of the decreased quality in the WLAN, reflected by a low value of the quality function for this network.

7.2.2 Test-Case II: 9 nodes in lines of threes, ACS and QADS

This test case considers nine nodes, moving in threes, with a distance of 4m between each triad and 0.4m between each of the nodes in a triad, as shown by node alignment (b). The application price, characteristics and requirements are the same as for the first test case, with the download rate fixed at 0.8Mbps. What differs from the first test case is the manner in which the nodes are distributed. Because of the node alignment, each set of three nodes perform their network selection decision at the same time.

When using QADS, the first row performs handover to the WLAN, Again with the ACS algorithm the nodes connect to the WLAN until it gets so overloaded that it cannot transmit acknowledgements to new users that are trying to connect, only then will these ACS users remain on WiMAX. In contrast, when using QADS, the second row of users, will all compute the same ANM score for the WLAN and the same delay and then connect to the WLAN at the same time, subsequently re-computing the ANM. On detection of the drop in expected quality (indicated by ANM difference) the random decision is triggered in each node. Depending on the random decisions (of second row of users), the WLAN may now be at full capacity. Nodes in the third row will either detect the low WLAN capacity or will all perform handover and then re-compute the ANM, which again leads to a random decision for each.

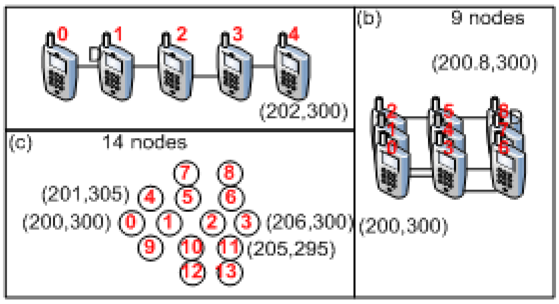

Figure 7.2

7.2.3 Test-Case III: 14 nodes in a group, ACS and QADS

This test case studies the behavior of the algorithm when a larger number of mobile users are on the move across campus. As shown in node alignment (c) the users move in a more disorganized fashion. The distances between the nodes are 1m on the x axis and 5m on the y axis. The ACS distributed nodes select and hand into the WLAN until it is completely blocked, whereas QADS manages a more even distribution of the nodes, since a terminal either connects to the WLAN to find a better quality of service or to switch back, or it doesn't connect to the WLAN at all as a result of the computed ANM function.

7.2.4 Testing of QADS in comparison with pNQS

The 5-node setup in case 1 was re-ran for PNQS and QADS with a modified version of NS 2.33 [], with an adapted version of the NIST mobility patch. The WLAN was assumed to be free of charge while the cost on the WiMAX was 0.1, corresponding to 1ç per Mb (10€/Gb) for this run. The PNQS parameters used were wc =wb = 0.5, B=0.8 Mbps, while b_i is computed at the MAC level. Because in the above examples the denominator can be 0, the constant value 1/MAX_DOUBLE was used instead. For QADS, $\boldsymbol{p}_{costUmin}$=0.8, corresponding to 8ç/Mb, $\boldsymbol{p}_{costUmax}$=0, while the rest of the parameters remained unchanged.

Since all the nodes use the same quality of service function, they handover at the same time. Then they detect the same loss of quality. This results in subsequent handovers of all the nodes in the same direction. The second test case was also repeated using PNQS and QADS, this time the cost for the WiMAX increased to 0.4, that is 4ç per Mb, and the rest of the parameters for both of these algorithms remaining as mentioned in the above paragraph.

7.3 Experimental results

In Test-case I, since all the nodes using the ACS algorithm always handover into the WLAN, which has insufficient capacity to deal with the demand, one node is severely disadvantaged in terms of throughput, illustrated in Figure 7.3.

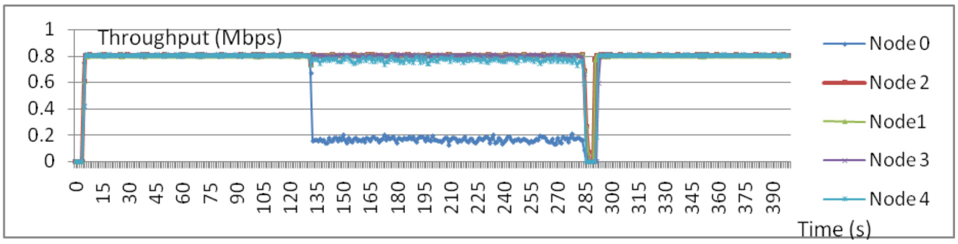

Figure 7.3 Total throughput for each of the nodes using ACS

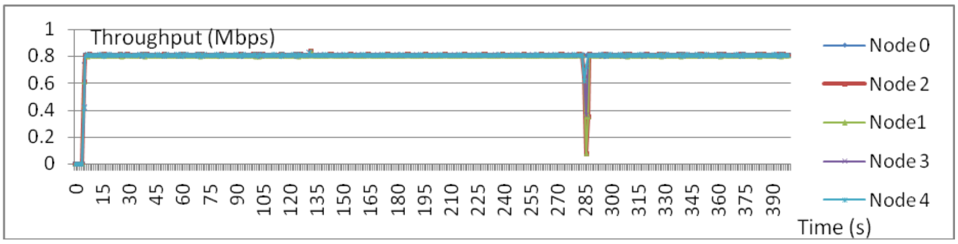

Figure 7.4 Total throughput for each node using QADS

When all the nodes employ the QADS algorithm for test-case I one node does not handover and instead stays in the WIMAX network. Although it has the disadvantage of paying more, this node has the benefit of maximum available throughput needed for the application. Also the strain on the AP is lessened with 4 nodes instead of 5, and the WLAN can provide better throughputs to all of its current users (see Table II). Figure 7.4 presents these throughputs. The advantage of QADS, in terms of packet loss is shown in Figure 7.5.

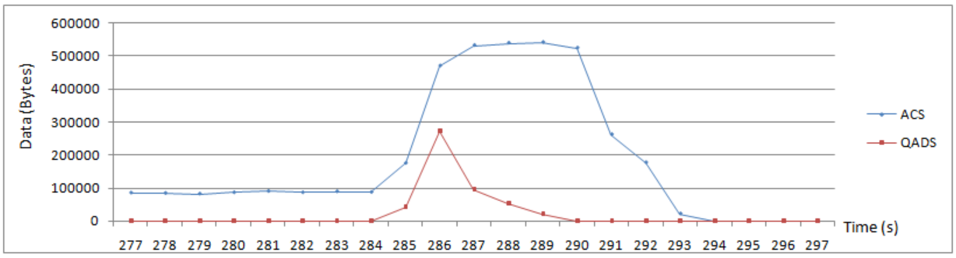

Figure 7.5 Accumulated packet loss at AP during handover from WLAN to WiMAX

Table I shows the results for Test-case I. The total throughput for the group of users offers an insight on how well the available bandwidth from the available RANs has been exploited. Total throughput and throughput variation for each node reflects the quality of the link that each mobile device receives. The standard deviation of throughput is an indicator of the distribution of achieved throughput across all the nodes and thus an indication of fairness in this scheme.

For Test-case I, QADS offers an advantage over the ACS strategy. As displayed in Table I the fairness of the throughput distribution is improved. Also, the variations in throughput are decreased for all nodes, as shown in Table II. Unfortunately one node will end up paying more for the streaming session. For example, in one run the node that remains in the WiMAX pays 0.96€, roughly 60% more than the other users that pay 0.59€. Nevertheless, the perceived quality is increased, with 47% better throughput than ACS and a huge decrease in jitter (almost 80%). In addition all the nodes on the WLAN get better quality by 1 to 4% at no extra cost to them. When the nodes move in single line the last nodes in the line will be the last to come in range of the WLAN by which time, depending on the distances between and the number of the nodes, the other nodes may have already handed over their connectivity and saturated the WLAN. Over time it is expected that things will balance out and, unless the same user is always last to reach a better network, different users will make the cost sacrifice of staying on the WiMAX, enabling other users to enjoy the benefits.

TABLE I
PERFORMANCE INDICATORS FOR ACS VERSUS QADS HANDOVER (CASE I)

Performance indicator	ACS	QADS
Average overall throughput (Mbps)	3.664	4.016
Standard deviation of average individual throughput	0.114	<0.001
Total traffic (Mb)	1461.900	1586.400
Total data loss at AP (kb)	16.434	0.489
Data loss (% of total traffic)	1.124	0.030

TABLE II
PERFORMANCE INDICATORS FOR INDIVIDUAL NODES (CASE I)

	Average throughput * (Mbps)		Standard deviation in throughput	
	ACS	QADS	ACS	QADS
Node0	0.546	0.805	0.316	0.004
Node1	0.795	0.802	0.084	0.045
Node2	0.796	0.803	0.081	0.039
Node3	0.788	0.803	0.108	0.030
Node4	0.776	0.804	0.105	0.022

* the throughput was calculated at the transport layer, so it appears to be higher than the 0.8 characteristic to the application

Table III illustrates the improvement brought by QADS for Test-case II. QADS improves the total average throughput for all users by more than 50%, while reducing loss at the WLAN AP ten-fold.

TABLE III
PERFORMANCE INDICATORS FOR ACS VERSUS QADS HANDOVER (CASE II)

Performance indicator	ACS	QADS
Average overall throughput (Mbps)	4.582	6.898
Standard deviation in overall throughput	1.984	0.859
Standard deviation of average individual throughput	0.220	0.103
Total traffic (Mb)	1832.870	2759.570
Total data loss at AP (Mb)	125.439	12.294
Data loss (% of total traffic)	6.844	0.045

Table IV shows the results for Test-case III the overall throughput combined for all users is increased by almost 20%, while a substantial decrease in packet loss is achieved. As desired, the fairness of throughput distribution across nodes is also increased.

TABLE IV
PERFORMANCE INDICATORS FOR ACS VERSUS QADS HANDOVER (CASE III)

Performance indicator	ACS	QADS
Average overall throughput (Mbps)	7.693	9.188
Standard deviation in overall throughput	1.597	1.215
Standard deviation of average individual throughput	0.290	0.212
Total traffic (Mb)	3077.200	3675.130
Total data loss at AP (Mb)	116.762	70.225
Data loss (% of total traffic)	3.794	1.910

Even more convincing, the results obtained when comparing QADS with PNQS show that using an elaborate quality of service function that balances between different network characteristic is insufficient when dealing with multiple users that travel in a group because of the ping-pong effect. Figure 7.8 illustrates the WLAN throughput of each node for the 5-node setup described in [], when PNQS is used. The ping-pong is obvious, as it is visible from the graph (and confirmed by the trace of the simulations) that all the nodes handover at the same moments, and to the same network – when the traffic with the AP goes down for one of the nodes, it goes down for all of them, which means that at that time they had handed over to the WiMAX network.

When using QADS, slight improvements in the performance indicators were detected, as shown in table V, but most importantly, the number total of handovers was reduced from 118 to 6 – though a value of 7 or 8 is also likely because of the random factor. For the 9-node setup, a similar result is obtained, with a slight improvement in perceived fairness and a dramatic reduction of handovers: from a total of 251 to 16.

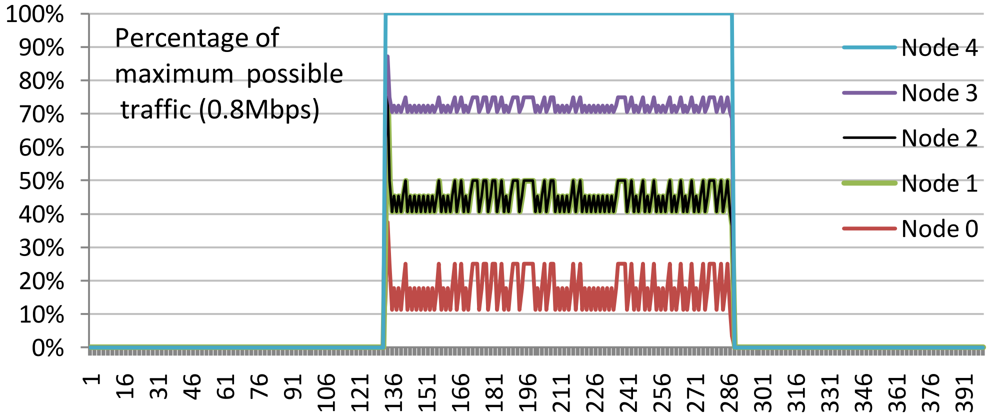

Figure 7.8 Illustration of the ping-pong effect

Figures 7.6 and 7.7 show the benefit of using QADS over ACS in terms of throughput and packet loss, compared for the three test cases.

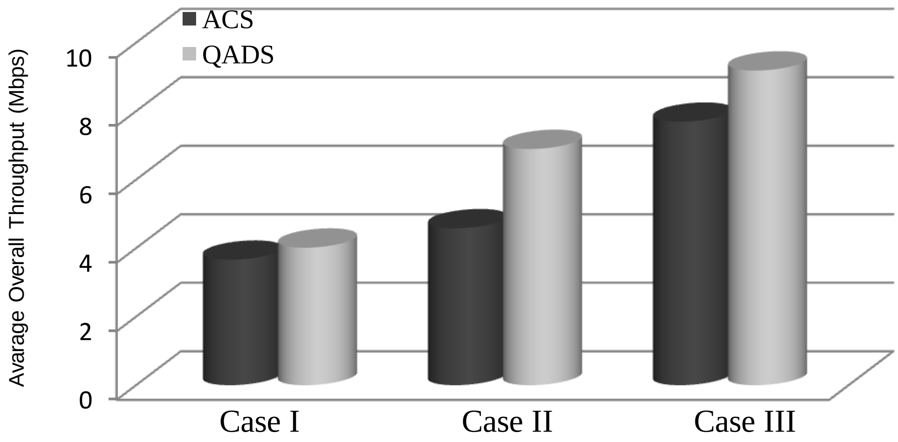

Figure 7.6 Compared results in average overall throughput

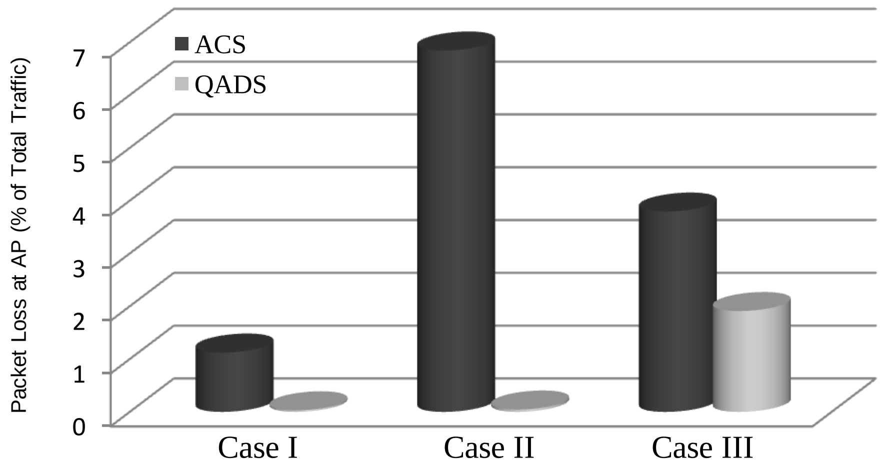

Figure 7.7 Compared results in data loss at AP

8 Conclusions

8.1 Achievements

This thesis examines the problem of terminal-controlled network selection algorithm in the context of a multiple mobile nodes moving together in a local area with similar network selection preferences, offering solutions in the form of a novel network selection algorithm, which was described, implemented and tested. The initial comparison of the proposed Quantified Adaptive Delay Selection against the simple Always Cheapest Selection algorithm showed a clear improvement, both in terms of individual and overall Quality of Service (based on group throughput and packet loss), while the comparison with the more elaborate PNQS algorithm proved that QADS is effective in minimizing ping-pong. Therefore, it is justified to say that the algorithm has fulfilled its design objective, to offer superior quality of service in situations involving multiple users, without impacting the more common case involving a single user.

8.2 Future work

Future work involves including dynamic values for instant handover and the drop threshold, and the function weights. The bandwidth prediction also needs further investigation. More tests on the social gain and the fairness, with different applications, user groups, and technologies will also be performed. Also the algorithm will be tested against other intelligent network selection algorithms with similar objectives.

9 References

[1] O. Ormond, G. Muntean, J. Murphy, "Dynamic Network Selection in Wireless LAN/MAN Heterogeneous Networks", In Mobile WiMAX: Toward Broadband Wireless Metropolitan Area Networks, CRC press (2007), ISBN: 9780849326240

[2] IEEE P802.21/Draft 11.0, http://www.ieee802.org/21/

[3] ITU-T Rec. G. 1010 (11/2001)

[4] D. Cavalcanti, N. Nandiraju, D. Nandiraju, D. P. Agrawal, and A. Kumar, "Connectivity Opportunity Selection in Heterogeneous Wireless Multi-hop Networks," Elsevier Pervasive and Mobile Computing Journal, vol. 4, no. 3, pp. 390–420, 2008.

[5] A. Dutta, S. Chakravarty, K. Taniuchi, "An Experimental Study of Location Assisted Proactive Handover", IEEE Global Telecommunications Conference (GLOBECOM), 2007.

[6] S. J. Yoo, D. Cypher, N. Golmie, "LMS Predictive Link Triggering for Seamless Handovers in Heterogeneous Wireless Networks," Proceedings of Military Communications Conference (MILCOM), 2007.

[7] T. Sikora, "The MPEG-4 video standard verification model", IEEE Trans. Circuits Syst. Video Technol.,vol.7, no.1, pp.19-31, Feb 1997

[8] WiMAX Pricing http://mybroadband.co.za/news/Broadband/340.html

[9] NS 2.29 http://www.isi.edu/nsnam/ns/

[10] The NIST Add-on http://www.antd.nist.gov/seamlessandsecure/doc.html

[11] R. Knoblauch, M. Pietrucha, M. Nitzburg, "Field Studies of Pedestrian Walking Speed and Start-Up Time" Transportation Research Record 1538, Transportation Research Board, 1996.

[12] V. Gupta, "IEEE P802.21 Tutorial", IEEE 802.21 session 15, 2006.

[13] A. Koutsorodi, E. Adamopoulou, K. Demestichas, M. Theologou, "User Profiling and Preference Modelling in 4G Terminals", IEEE Personal, Indoor and Mobile Radio Communications (PIMRC), 2006

[14] X. Cai, F. Liu, **"Network Selection for Group Handover in Multi-Access Networks",** IEEE International Conference on Communications (ICC), 2008

[15] H. J. Wang, R. H. Katz, J. Giese, "Policy-Enabled Handoffs Across Heterogeneous Wireless Networks", Mobile Computing Systems and Applications,.. (WMCSA). IEEE Workshop on (1999), pp. 51-60.

[16] O. Ormond, "User-Centric Network Selection Strategy in Heterogeneous Wireless Networks", Thesis, University College Dublin, October 2007.

[17] S. Kozlov, Adaptive scheduling of MPEG video frames during real-time wireless video streaming, IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, June 2005

[18] D. P. Anderson, Y. Osawa, R. Govindan, "A file system for continuous media". ACM Transactions on Computer Systems pp. 311-337, Nov. 1992.

[19] K. Jinzenji, H. Watanabe, S. Okada, N. Kobayashi, "MPEG-4 very low bit-rate video compression using sprite coding", Multimedia and Expo, 2001. ICME 2001. IEEE International Conference on, August 2001

[20] Internet Engineering Task Force, RFC 1122 standard: Requirements for Internet Hosts -- Communication Layers, October 1989

[21] ITU-T NGN-GSI, Handover Management Function Recommendation v.03, October 2006.

[22] S. Fu, M. Atiquzzaman, L. Ma, W. Ivancic, Y. Lee, J. Jones, S. Lu, TraSH: A Transport Layer Seamless Handover for Mobile Networks, Technical Report, January 2004.

[23] http://www.wimaxforum.org/technology/downloads/wimax_forum_wimax_forecasts_6_1_08.pdf

[24] http://www.eyeforwireless.com/radionetwp.pdf

[25] http://www.wirelessweek.com/Article-By-The-Numbers-030109.aspx

[26] http://www.windowsfordevices.com/news/NS3015364153.html

[27] http://www.iphonespies.com/iphone-reviews/iphone-user-satisfaction-highest/

[28] http://www.muniwireless.com/2008/12/03/bt-wi-fi-traffic-doubles-in-one-year/