QoS-oriented Framework for Link Selection in Heterogeneous Wireless Environments

Thesis by

Ashton Wilson

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

Kingston University

Kingston-Upon-Thames, Surrey

October 2008

Abstract

Wireless is now common for access to multimedia services, with many different devices and choice of access technology. Access methods have become varied and with more types of services, which requires more consideration for coordinating existing protocols and quality of service (QoS). Increasingly, new wireless access technologies co-exist on the same devices, for example, smartphones already have third-generation cellular and WiFi. Devices with multiple links are described under the umbrella term *heterogeneous environments*. A trend towards heterogeneous wireless environments and varied types of media services requires that QoS and user satisfaction are prominent in next-generation networks. The problems in next-generation heterogeneous wireless environments include many levels of complexity; from link coexistence to user-centric policies and contexts. This thesis explores the issue of QoS in interface selection for devices with more than one wireless access link. A solution that provides link selection for QoS policy is investigated using analytical and simulation techniques.

Different wireless networks have capabilities and limitations, determined by radio technology and network conditions. The research focused on an approach to improve QoS by leveraging the differences in wireless networks. However, it is complicated by issues such as: different protocols, physical device co-existence, mobility, and application QoS requirements. Following a review of artificial intelligence (AI) techniques, finite-state machines (FSMs) and fuzzy decision-making (FDM) are proposed as a solution approach. An agent-based prototype is used to combine FSMs and FDM for automating link selection, determined by user and QoS policy.

Prototype evaluation was performed using sensitivity analysis for FDM, and discrete-event simulation for generating QoS metrics in wireless environments. The results are comparisons of FDM prototypes using different parameters; different agent prototypes were run with different QoS conditions for comparing points of handover between UMTS and WLAN networks for one service type. The research has shown an agent model can reduce the complexity for a user in wireless interface selection, while including QoS metrics and user preferences into

the decision process. Core decision-making techniques in the design are relevant for emerging standardisation frameworks such as 802.21, and the next-generation of wireless networks to support heterogeneous access.

Acknowledgements

Firstly, I thank my supervisor Dr Andrew Lenaghan for his unwavering support and dedication, fascinating discussions, and attention to detail in feedback. Thanks to Dr Ron Malyan for introducing me to wireless data communications and the chance to pursue such a challenging topic. Thanks to Dr Martin Tunnicliffe for his advice and contributions during redrafting and submission. Also, thanks to the members of the Networking and Communications Group and the Faculty of Technology at Kingston University for the opportunity to pursue the studentship full-time.

I would like to thank Cyril Onwubiko, a fellow PhD student at Kingston for his research discussions and advice on mathematical terminology. Thanks to Bippin Makoond and the other researchers in the faculty (apologies for those names I forgot) for their discussions on research and spurring my creativity.

I would like express my gratitude to members of my immediate and extended family for their support. Thanks to Rosemarie, Joe and Paul, for their advice and assistance in the redrafting process, and positive encouragement throughout.

Ashton Wilson

Kingston University, London

October 2008

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

2G	Second Generation
3G	Third Generation
3GPP	Third-Generation Partnership Project
4G	Fourth Generation
ABC	Always Best Connected
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
ANN	Artificial Neural Networks
AP	Access Point
ATM	Asynchronous Transfer Mode
BOD	Behaviour Orientated Design
BS	Base Station
BSC	Base Station Controller
CE-FSM	Communicating Extended Finite State Machine
DAR	Dynamic Address Reconfiguration
DCF	Distributed Co-ordination Function
DiffServ	Differentiated Services
DST	Dempster-Shafer Theory
E-FSM	Extended Finite State Machine
EDCA	Enhanced Distribution Coordination Access
EDGE	Enhanced Data-rates for GSM Evolution
EPS	Evolved Packet System
ETSI	European Telecommunications Standards Institute
EVDO	EVolution Data Optimisation
FCM	Fuzzy Cognitive Map

FDM	Fuzzy Decision-Making
FLC	Fuzzy Logic Controller
FMC	Fixed Mobile Convergence
FSM	Finite State Machine
GAN	Generic Access Network
GGSN	Gateway GPRS Support Node
GPRS	General Packetised Radio Service
GSM	Global System for Mobile
GWAO	Generalised Weighted Averaging Operator
HAL	Handover Agent Layer
HCCA	HCF Controlled Channel Access
HDTV	High-Definition Television
HLR	Home Location Register
HSCSD	High Speed Circuit-Switched Data
HSPA	High Speed Packet Access
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMT-2000	International Mobile Telecommunications-2000
IntServ	Integrated Services
IPPM	IP Performance Metrics
ITU	International Telecommunication Union
IWU	Inter-Working Unit
LTE	Long Term Evolution
MAC	Medium Access Control
MADM	Multiple Attribute Decision-Making
MAN	Metropolitan Area Networks
MCDM	Multi-Criteria Decision Making
MF	Membership Function
MIH	Media-Independent Handover
MIP	Mobile IP
MN	Mobile Node
MSC	Mobile Switching Centre

NS-2	Network Simulator 2
PCF	Point Co-ordination Function
PDA	Personal Digital Assistant
PLR	Packet-Loss Ratio
PoA	Point of Attachment
PSTN	Public-Switched Telephone Network
QoS	Quality of Service
RAB	Radio Access Bearer
RAN	Radio Access Network
RNC	Radio Network Controller
RSVP	Resource reSerVation Protocol
RTP	Real-time Transfer Protocol
SAP	Service Access Point
SCTP	Stream Control Transmission Protocol
SGSN	Service GPRS Support Node
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SNR	Signal Noise Ratio
TDMA	Time Division Multiple Access
TE	Terminal Equipment
ToS	Type of Service
UE	User Equipment
UMA	Unlicensed Mobile Access
UMB	Ultra Mobile Broadband
UMTS	Universal Mobile Telecommunications Service
UTRAN	UMTS Terrestrial Radio Access Network
UWB	Ultra Wide-Band
VoIP	Voice over IP
W-CDMA	Wideband Code Division Multiple Access
WiFi	Wireless Fidelity
WiMAX	Worldwide Inter-operability for Microwave Access
WLAN	Wireless Local Area Network

WMM	WiFi Multimedia
WPAN	Wireless Personal Area Network
WSM	Weighted Sum Model
WWAN	Wireless Wide Area Network

Chapter 1

Introduction

Fixed-line broadband to the home is now common, with more options for high-speed wireless emerging. Third-generation cellular (3G) provides video and fast web-browsing on the move, and WiFi is available in offices and public hot spots. New devices will be *heterogeneous*, supporting more methods for wireless connectivity choice. Devices with multiple wireless interfaces will require support from next-generation networks for inter-access roaming, without limiting user experience. This thesis investigates handover strategies in multiple wireless access environments. It proposes a new hybrid framework for handover automation that uses a combination of artificial intelligence (AI) methods. The framework shows that handover between different wireless links can be controlled and automated by quality of service (QoS) policies and user preferences.

1.1 A Wireless Shift

The trend towards 3G deployments and beyond, allows more types of services to be accessed on the move. As discussed by Lehr & McKnight (2003), the future wireless environment will be a heterogeneous mix of wireless access networks. Other technologies, such as wireless local-area networks (WLAN) have provided more flexibility to access online services in offices and public 'hot spots'. Emerging wireless technologies such as WiMAX, will provide high-speed mobile access over several kilometres. While cellular 3G continues to provide the wide area coverage for more types of services. Mobile subscribers have continued to out-grow fixed-line Internet subscriptions (figure 1.1). This shows the potential of the sector for mobile applications, with growth of over 1.5 billion more than fixed-line subscribers.

Services accessed over wireless have also become varied. From voice only cellular, extensions were provided to transport data and increased downlink speeds (figure 1.2 on page 3).

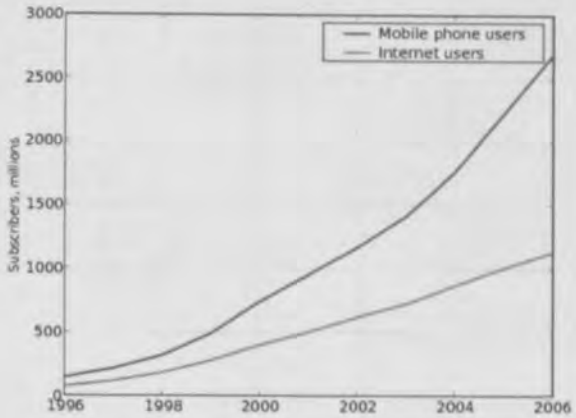


Figure 1.1: Worldwide mobile phone and Internet subscribers (*Source*: ITU (2008))

Since the second-generation (2G) of digital cellular technologies, standards releases by the Third-Generation Partnership Project (3GPP) and 3GPP2 (for cdma2000) have shown increased downlink (network to terminal) data-rates. Developments for 3G continue, such as High Speed Packet Access (HSPA) and EVolution Data Optimised (EV-DO), towards forth-generation (4G) networks. 4G networks have been set out by the 3GPP Long Term Evolution (LTE) and 3GPP2 Ultra Mobile Broadband (UMB) to support multimedia applications. Multimedia is the generic term for applications content, such as voice, web-browsing, video, and instant-messaging. The varied types of media and networks has led to a shift towards *converged services*. Previously separate services like television, voice, and the web, are now just forms of digital media (Economist, 2006). The trend in 4G will have these services in the all-IP core networks that support different types of radio access networks.

Supporting a further range of services requires more sophisticated handsets. Such devices now have sufficient processing power to decode video, with better resolution and touch-sensitive screens, providing richer interaction. Handsets that blend the mobile phone with personal digital assistants (PDAs) are commonly referred to as *smartphones*. The initial market for smartphones has been business users, particularly for email (Economist, 2006). Although it is possible to have devices with many capabilities, commercial activities are more cautious; sometimes favouring more modest hybrid devices (Economist, 2006). Current smartphones already have WiFi and cellular 3G capabilities, that provide connectivity for voice, email, and web applications. With these features, many different usage scenarios are now possible.

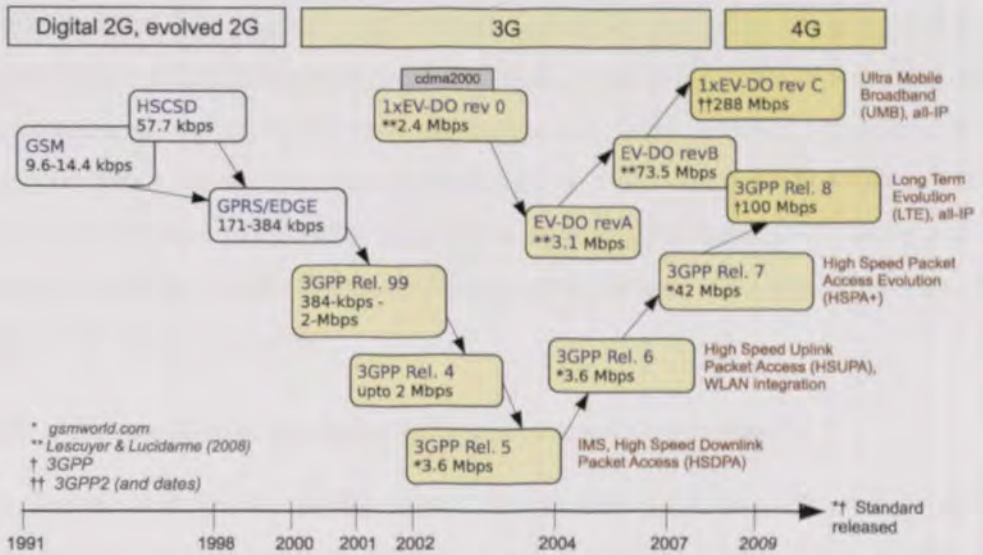


Figure 1.2: Evolution of cellular networks standards, including theoretical data-rates and real ease dates (*adapted from*: Prasad & Ruggieri (2003, p.3)).

1.2 Issues for Next Generation Wireless

More pervasive heterogeneous wireless environments exposes technical complications within networks. Dissimilar wireless networks, such as 3G cellular and WLANs have complementary benefits for the user and the mobile operator (Lehr & McKnight, 2003), with inter-working addressed by the 3GPP for 3G-WLAN inter-working (3GPP, 2004a). Also, other emerging wireless technologies could compete with each other, such as 3G and WiMAX, although WLAN integration is likely to complement wide area access networks (Ahmavaara et al., 2003; Lehr & McKnight, 2003; Gunasekaran & Harmantzis, 2008). A more 'intelligent' terminal device that can adapt to wireless access heterogeneity will be investigated in this thesis.

More sophisticated devices supporting many types of applications and heterogeneous wireless access, provide opportunities and additional complexity. Approaches that manage the complexity have generated concepts such as, *user-centric computing* and *context-awareness*. This thesis follows the user-centric concept to explore the adaptable link selection that is centred around QoS and user policy. The argument for user-centric computing places emphasis on user activities and choice (Kanter, 2003) and visions for wireless adaptation around user activities (Prasad & Ruggieri, 2003, p.269). Context-awareness is a concept that allows mobile devices to interact and adapt to their environment with little or no human interference. In networks and handovers, this can be additional network entities and protocols to assist the mobile device

(Wei et al., 2006). Reconfigurable and adaptable connectivity using context and network information is explored in a European Framework project, Ambient Networks (IST-507134, 2007). But context-aware computing has been criticised as a deceptive metaphor by Erickson (2002) by the subtleties of human awareness that is difficult to attain by computer 'intelligence'. However, these ideas have spurred other research into managing the challenges of next-generation mobile networks, such as handovers, session continuity, and QoS (Chin et al., 2003; Hawick & James, 2003; Mishra et al., 2004).

1.2.1 The problem of handover in heterogeneous environments

This thesis reviews the scope and QoS issues in vertical handover scenarios that could involve more than one wireless interface. Subsequent use of *wireless interface* refers to those protocols below the network layer, consisting of link and a physical layer protocols. Managing multiple interfaces raises additional problems. The wireless channel and handovers are disruptive to QoS and security, which can vary between domains of control. Network addressing requires additional protocols and negotiation; handover and management across interfaces become complex. To answer the question *"which access interface is the best?"*, depends on many factors: QoS, user preference, application types, interface availability, power consumption, and economic cost. Previous research has developed solutions that focus on specific pairs of access technologies, such as WLAN and cellular (Ylianttila et al., 2001; Ma et al., 2004; Chakravorty et al., 2004; Calvagna & Modica, 2005). These approaches reduce the problem space of problem variables to an engineered solution for two interface devices. A significant solution would develop a cross-technology approach for managing vertical handover scenarios.

The concept of *always best connected* (ABC) suggests solution components that include access discovery, access selection, and mobility management (Gustafsson & Jonsson, 2003). These components are required to manage handovers in heterogeneous access environments. This is mainly between links and networks of different types *(vertical handovers)*, rather than same domain or technology *(horizontal handover)* (Stemm & Katz, 1998). Other efforts for coordinating different interfaces is ongoing in the IEEE 802.21 working group (IEEE, 2006). The proposed 802.21 framework provides additional support in the protocol stack, but it does not address the selection of interface, which implementations must address.

1.2.2 Optimising interface selection for QoS

One of the components required for an ABC device–access selection–is the motivation for this thesis. The hypothesis is that devices with multiple interfaces can provide a system for improving QoS experience. Enabling choices and preferences in heterogeneous wireless environments requires new approaches that are more adaptable and can be optimised based on multiple sources of performance information. Research in handover strategies have focused on a single optimisation method (Zhang, 2004; Zhu & McNair, 2006), although there have been some that use a combination of techniques (Song & Jamalipour, 2005).

This thesis poses the question: *to what extent can AI strategies optimise link selection for different QoS policies?* Thus, the aim of the research:

> *This thesis proposes a hybrid approach using a combination of different AI techniques for optimising QoS policy in handovers in heterogeneous wireless environments. The scope is defined as handover selection and decision, rather than mobility execution and management. It places emphasis on user-centric and application specific QoS descriptions for interface selection.*

Assumptions and constraints are placed on the scope:

- Link selection can be terminal based, network based, or manually controlled by the user (Gustafsson & Jonsson, 2003, p.53). A terminal-based approach is used, where the terminal decides the handover points.
- The approach is constrained to one application type at a time, such as a voice session; no concurrent applications.

The scope of the research is restricted to wireless interface selection, where the problem space includes mobility management. Although handovers are part of the process, it is seen as a secondary procedure that could be introduced from further research. Once the link is selected it is a technical issue to perform the handover, rather than negotiating variables for selection. It is the QoS policy that is the main driver for interface selection. Also, the framework uses local information from the device status, but it can also be extended to allow other sources from the network to make selection decisions.

1.3 Contributions

The following are an overview of the main contributions provided by this research.

- **A reference framework for QoS related issues of complexity in heterogeneous wireless environments.** The trends of wireless networks are directed to more types of access methods and integration of core networks. Co-existence of interface devices and interaction between network types introduce problems for QoS and mobility. Different types of interface have different QoS characteristics. These differences in capabilities can be exploited at points in a session to improve QoS experience. For QoS policy there may be times when high throughput is preferable, or low latency for interactive voice. A solution model is conceived from different levels of these factors, such as application QoS classes, mobility, and capabilities of interfaces, simultaneous applications (trade-off and granularity). Although not exhaustive, it provides a sub-division of the complexity heterogeneous environments for possible solutions.

- **Comparison and performance evaluation of decision-making techniques.** A review of handover strategies in Kassar et al. (2008) has shown decision theory and multiple attribute decision-making (MADM) methods that have been applied in the context of vertical handovers optimisation. Different interfaces are described as options, and then compared using multiple performance metrics and criteria. Metrics and criteria are sometimes not easily or directly comparable, such as comparing cost to throughput; while some are vague, such as "low-cost". To handle these issues, fuzzy concepts have been used in similar problems to provide more user-centric representation of comparisons (Chan et al., 2002; Zhang, 2004).

- **A framework that combines AI methods to plan and control vertical handovers.** A reactive finite-state machine is used to describe logic for handover control and assessment. A fuzzy decision-making component–based on research by Bellman & Zadeh (1970) and further detailed in Sousa & Kaymak (2002)–is used for comparing data-link performance, user policy, and local device conditions. These are combined by using an agent metaphor for combining inputs, data representation, and control actions. This is shown to be a flexible model of describing handover logic, whilst allowing QoS policies to influence the decision process.

- **Evaluation of settings for fuzzy MADM methods using sensitivity analysis.** The

fuzzy MADM approaches were tested using different functions to show how different criteria can be aggregated. From this aggregation, the fuzzy decision making prototypes generate a suitability index, or *rank-score* which determines ranking of alternative options. The sensitivity analysis showed the trade-off that can be applied by different aggregation functions, criteria functions, and criteria weights. It was shown through the approach by Kaymak & van Nauta Lemke (1998), that an index of optimism parameter can change the decision trade-off (risk) between satisfying good criteria (optimistic), and satisfying poor criteria (pessimistic).

- **Evaluation of a simplified agent framework, incorporating fuzzy decision making and finite-state models through wireless simulation.** The agent prototypes used finite-state machines to handle device changes by encoding the process logic as a series of states and transitions. Prototype models were simulated using a discrete-event simulation program. This program used QoS metrics traces generated from wireless network simulation in NS-2. Input data from post-simulation runs were used as input for agent programs. This decoupled approach could use trace data from other sources: empirical testing or other testbed setups. As a result, it does not fully model handover and mobility issues. However, the The NS-2 extensions used for simulation (Baldo et al., 2007) allow highly configurable protocol stacks to be used. This features cross-layer capabilities, which provide possibilities for embedding the agent prototype in a simulated multi-link device.

1.4 Chapter Organisation

Following this introduction, the chapters begin with related background topics and the challenges in multimedia, networking and wireless domains. From the problem analysis and evaluation of decision strategies, a solution framework is designed based on the agent paradigm. The latter chapters present the method and results of an evaluation approach using simulation.

Following this introduction, the domain background begins with *Chapter 2:* an overview of QoS concepts, the effects of media and link QoS, and current state of QoS architectures and protocols. *Chapter 3* introduces the different types of wireless access network, and the difficulties caused by the wireless channel, such as propagation, interference, and error correction. The chapter also describes what QoS capabilities are provided by different wireless networks.

Chapter 4 introduces the concept of heterogeneous wireless environments and shows the overall motivation for this thesis. Industry trends propose multi-service architectures and further integration in next generation wireless networks. Using handover strategies and industry standards, QoS improvements can be sought from heterogeneity. Finally, the chapter concludes by providing a model of complexity for issues in heterogeneous wireless environments. *Chapter 5* begins with defining requirements for a vertical handover strategy, and provides a review of techniques from decision-making, probability, and AI. *Chapter 6* presents a interface selection framework based on QoS requirements. A simplified behaviour-based agent is proposed using finite-state machines for control logic and fuzzy decision-making algorithms for link assessment. *Chapter 7* describes a two-phase approach to evaluate prototypes of the new agent framework. In the first evaluation phase, an analytical approach is used to explore the effects of decision-making algorithms. The second phase uses wireless simulation models to evaluate the agents behaviour in response to stimuli. The chapter defines experimental designs of scenarios and parameter setting for comparing the prototypes response. *Chapter 8* presents the results generated from the analysis and simulation experiments. The analysis discussion explains the decision-making component results and a recommended decision-making algorithm. The second part of the chapter shows simulation experiments results and provides a discussion of the main findings. *Chapter 9* states the research conclusions and discusses implications for further research.

Chapter 2

Quality of Service

Quality of service (QoS) was an inbuilt and tightly engineered solution in the public-switched telephone network (PSTN). A trend towards converged networks means different application data are sent over a common network. This requires additional facilities for QoS support.

The following discussion presents an overview of QoS concepts, measures, and standards. QoS for multimedia applications are more susceptible to changing networks conditions. Methods that manage QoS are also explained for IP networks.

2.1 Parts of Quality

Standards organisations, such as the International Telecommunication Union (ITU) defines QoS as: "the collective effect of service performance which determine the degree of satisfaction of a user of the service" (ITU-T (1994) cited in Gozdecki et al. (2003, p.1)). The European Telecommunications Standards Institute (ETSI) and the Internet Engineering Task Force (IETF) have also documented QoS recommendations. Those recommendations and standards from different organisations overlap in places of their treatment of QoS. Historically their treatment of QoS originated from focus on separate distinct services and networks.

Telecoms standard organisations, ITU and ETSI, have extensive coverage of QoS aspects. Early data networks based on IETF standards had no in-built QoS, but has subsequently introduced amendments and recommendations for QoS assessment. Different viewpoints exist between ITU and IETF orgnatisations that describe how QoS should be treated for applications and networks. The ITU places focus on the user rating QoS, with the network having static capabilities; while IETF treats user QoS as variable, and the network can be controlled for the level of service (Burgstahler et al., 2003).

The following sections explain effects of components in packet-based networks, the effects

on QoS, and some descriptions of QoS terms and measurement.

2.1.1 Data networks

The two categories of data network are circuit-switched networks (CSNs) and packet-switched networks (PSNs). In CSNs, resources are explicitly setup for the duration of a session, i.e. a voice call. Whereas PSNs data packets may not always traverse the same path due to routing algorithms.

Data packets in non-QoS PSNs are treated as best-effort and subject to effects from congestion, queueing, and limited bandwidth. Increased network utilisation begins to cause packet loss and congestion. These effects are reduced by transport protocols for delay elastic applications. For real-time applications, best-effort delivery is not sufficient to deal with the effects of loss and delays. Such delays are caused at multiple points within the network (figure 2.1):

- **Coding** delays are determined by application codecs[1]. Those that use high levels of compression require extra processing times, hence adding to coding delay.
- **Queueing** delay is introduced by bottlenecks, such as routers and switches. Routers have queueing buffers while waiting to be served to the output link. Depending on the queueing algorithm and load, delays may be variable.
- **Serialisation** delay refers to the time taken by the link interface to place all bits on the link. This starts from when the serving interface places the first bits to the last bits of the frame. Such delay is affected by media-access protocol being used, such as Ethernet or ATM.
- **Propagation** delay is determined by link bandwidth and physical properties of the media.

2.1.2 Common measures

The ITU and ETSI define two levels of QoS which relate to a general model as *perceived* and *intrinsic* (Hardy (2001), cited in Gozdecki et al. (2003, p.154)). Perceived QoS is the requirements and perceptions by the customer to what is offered and achieved by the provider. Intrinsic QoS relates to the technical properties and metrics of the network. The ITU approach focuses on perceived QoS, but also intrinsic QoS as *network performance*, whereas the IETF

[1] codec - compressor-decompressor/coder-decoder

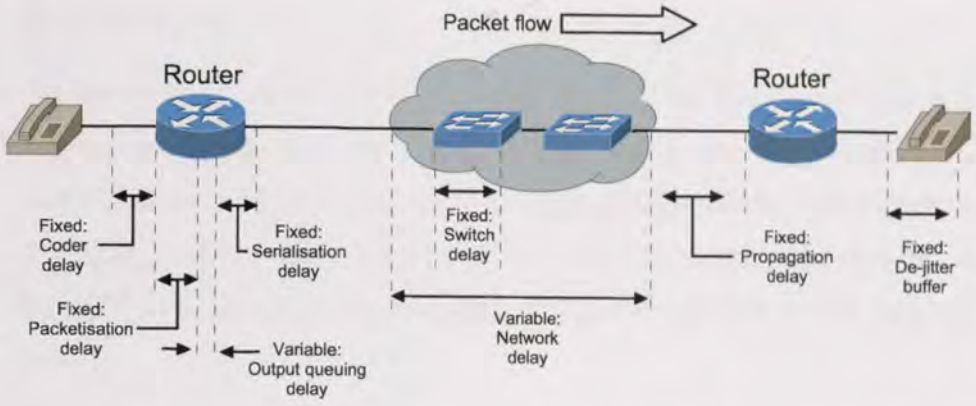


Figure 2.1: Sources of delay for voice application in the network (*Adapted from*: Cisco (2006, p.4))

focuses on intrinsic but not perceived QoS (Gozdecki et al., 2003). Above perceptive QoS is *assessed* QoS, that suggests a user makes a decision on continued use of the service; though is not addressed formally by ITU, ETSI or IETF (Gozdecki et al., 2003). These levels of QoS are used to further discuss the distinctions in applications and networks.

A framework of applications to loss and delay is defined in ITU recommendation G.1010 (ITU-T, 2001b). The model focuses on end-user performance perception, and not network or device specific characteristics. A separation between network technology and service enables simpler network management with classes, especially for next generation all IP networks (Mustill & Willis, 2005). For QoS management, there must be performance measures for assessment. A *metric* is used as such a measure. Metrics and measures of QoS can be defined as technical and user-based terms (Chalmers & Sloman, 1999).

Technology metrics

Technical QoS measures includes metrics that are usually used by protocols and applications. The following are measures of performance in capacity, reliability, and timeliness:

- *Bandwidth* is usually used to refer to spectrum frequency ranges in signal processing, but it also refers to data-rate, or bit-rates. Related terms include: throughput as average data-rates; and goodput as the application layer data-rate.

- *Reliability*. The term packet-loss ratio (PLR) is often used to measure the ratio of lost versus received packets. According to the IETF IP packet-loss metric, one-way loss metric is preferential to round-trip loss due to asymmetric traversal of paths (Almes

et al., 1999b, p.2).

- *Timeliness.* Packet delays in IP are measured by round-trip delay metric (Almes et al., 1999c), and one-way delay metric (Almes et al., 1999a). Jitter is often defined as the variation in delay of packets, as in the IP delay variation (IPDV) metric (Demichelis & Chimento, 2002). Delay variation is also defined in the Real-time Transfer Protocol (RTP) using an exponential filter method to give a smoothed average from arriving packets.

User metrics

Requirements of QoS performance can be described by metrics such as cost, security, and perceived QoS or quality of perception (QoP) (Ghinea & Magoulas, 2001). The following are user-based QoS metrics (Chalmers & Sloman, 1999, p.4):

- *Importance.* A rating of priority between application or flows. The user may require strict levels of quality for some applications or indifferent to others.
- *Perceived QoS.* These relate to metrics at the application layer, such as: frame rate or lip-sync in video; bit-rate or sampling rate of audio; picture detail or pixel resolution; smoothness of video, determined by frame jitter.
- *Cost.* An important metric to some users, but may be less so for others. A financial cost associated with access, or a per-unit cost for bytes received.
- *Security.* The level of security is important in both corporate and personal settings. Encryption and virtual private networks (VPNs) provide solutions together with authentication and access control methods. Though, security is a general term and without quantifiable measures.

2.2 Multimedia Characteristics

Multimedia services can exhibit different end-to-end traffic flow characteristics between client and server because of protocol exchange patterns. Applications can be defined as service classes that have patterns of usage and QoS requirements (table 2.2), include: voice and video (interactive and non-interactive); data, such as images and text; messaging and presence; and

transactional. Real-time service classes are more susceptible to variations in network conditions. For others, different QoS related factors are important, such as response time for transactional class. This section explores the effects of control signalling methods, and traffic patterns of different service types and QoS requirements.

Service Class	*Application example*	*Important QoS*
Real-time	Conversational voice (VoIP), streaming audio, interactive video, streaming video	Delay, delay variation (jitter), packet loss
Non-real-time	HTTP, e-mail, FTP	Throughput, response time
Transaction based	E-commerce, mobile-banking.	Security, response time
Message based	SMS, MMS, instant messaging	Delivery success
Location based	Location-based services (LBS)	Location accuracy, response time

Table 2.2: Service classes and applications types (*Source*: Gurijala & Molina (2004, p.38))

2.2.1 Transport and signalling

Network routing and protocols have an affect on application bit-rates. Although, certain services may exhibit a "natural bit-rate" (Black, 2000, p.9), for example bursty behaviour in on-off speech, asynchronous upload and download in web browsing, and maximum bandwidth consumption in file-transfer protocol (FTP). Depending on the bursty nature of the application, the full bandwidth may not be utilised (Black, 2000); alternating between periods of under-utilisation and full-utilisation (figure 2.2). As an application increases the data-rate beyond the level supported in the network, the network begins to drop packets and the application is forced to reduce the rate. This effect is shown by transport control protocol (TCP) slow start, where transfer rate is increased until congestion is detected, either by losses or long delays (Hersent et al., 2000, p.334).

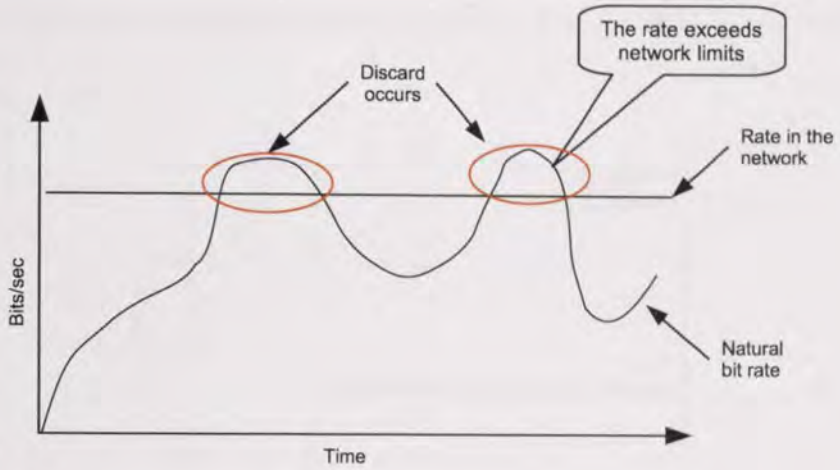


Figure 2.2: Transfer rate of a bursty or variable-rate application (*Source*: Black (2000, p.10))

Current transport layer protocols provide flow maintenance and signalling to applications. Reliable transport protocols, such as TCP provides control mechanisms to adapt or change transfer rates according to congestion and loss in the network. However, packet loss causes TCP slow start, which reduces throughput and increases end-to-end delay due to retransmissions (Hersent et al., 2000, p.296). This makes TCP unsuitable for use in real-time applications that have limits on delays. User datagram protocol (UDP) is usually used instead, as there is no flow control or re-sending missing packets. Real-time services such as voice and video require other common signalling to supplement the basics of UDP such as, call-setup, checksums, and timestamps.

The RTP (Schulzrinne et al., 2003) was developed for video and audio applications. Real-time applications use RTP over UDP to provide checksums, timestamps, and sequence numbers. This information can be used for control and adaption by the application, but RTP does not guarantee QoS. RTP uses real-time transport control protocol (RTCP) (Schulzrinne et al., 2003) for additional signalling of current status of end-points, such as multi-cast settings and unique RTP identifiers (Lloyd-Evans, 2002, p.166).

2.2.2 Conversational voice

Voice applications have traditionally been circuit-switched, with highly engineered solutions for QoS. In converged networks, application QoS can become variable as voice packets are bundled with other types of traffic. Interactive voice is particularly sensitive to delays and packet loss. The effects of delays depend upon the coding scheme used, but usually delays of

more than 400ms are undesirable for voice and leads to user confusion as end-points become out-of-sync.

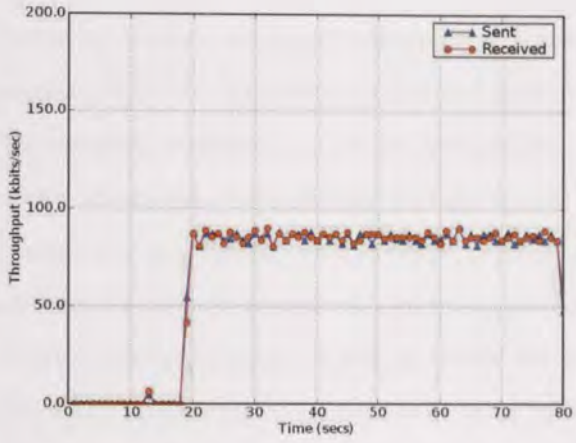


Figure 2.3: A traffic pattern for voice over IP using G.711 codec.

Voice applications tend to have symmetrical flow requirements (though it depends on encoding method). Figure 2.3 shows the throughput of bi-directional flows captured[2] from a voice call. In this codec, the throughput is mainly symmetrical; both directions have a bandwidth requirement of 64-kbps. Bandwidth requirements of other codecs vary depending on coding, compression, or silence suppression used. For example, the ITU-T G.711 codec has a data-rate of 64-kbps[3] in both directions, whereas G.729A uses 8-kbps[4]. This does not include RTP and other protocol headers. For the G.711 voice call over Ethernet (including layer 2 headers) the data-rate rises to 93-kbps. Therefore, demands of voice applications depends on the coding and type of technology used for transport.

2.2.3 Interactive video

Interactive video, or video conferencing, has similar requirements in delay and delay variation to conversational voice, along with additional bandwidth for moving images. In much the same way as real-time voice, the choice of codec determines the perceptive quality, QoS requirements, and traffic pattern or burstiness of packet transmissions. Some video codecs can operate at a low bandwidth of 10-kbps, such as H.264, while others require data-rates greater

[2]Captured using tcpdump. Details of calculation and plot data can be found at `http://intstack.wikidot.com/empirical`.

[3]For 160 byte payload (20 ms sample) at a send rate of 50 packets per second.

[4]For 20 byte payload (20 ms sample) at a send rate of 50 packets per second.

than 100-kbps to over 1-Mbps (Morrison, 2005, p.33).

2.2.4 Streaming media

Non-interactive, real-time applications are usually referred to as streamed media, such as audio (radio, voice messages) or video. Streaming applications send most of their data in one direction, in which the upstream throughput is low for initialisation and finalisation control signalling. The download throughput continues at the coded rate until stopped; though it may be variable if some interactivity is supported, such as pause, seeking, and resuming playback. This effect can be seen from the captured packets of a streaming audio application (RealPlayer 10[5]) in figure 2.4. A small number of bytes are sent to initiate the session after 10 seconds (triangles). The streamed data is downloaded (circles) at a rate of 50-110 kbps until a request is made to end after 58 seconds

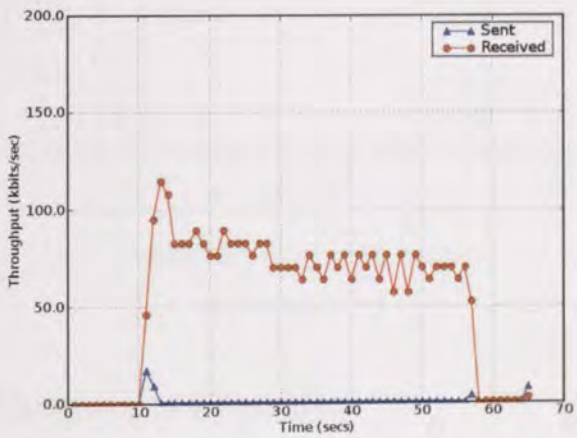


Figure 2.4: RealPlayer UDP traffic of streamed audio.

Streaming audio applications can tolerate moderate delay and variations in delay better than interactive voice. Buffering is used to limit the affect of delay and delay variation. Audio applications can benefit from higher bandwidth to support higher quality sound codecs. Generally, video streaming requires less stringent delay requirements and QoS than video conferencing. Streamed video traffic is similar to streamed audio, but usually involves larger data files, and hence is similar to an FTP download. Delay and delay variation effects are reduced, if not eliminated by buffers in the application.

[5]Linux version, from `http://www.realnetworks.com`.

2.2.5 Non-real-time and other data

Applications for email, web, and file transfer use reliable protocols (TCP) for recovery from errors and effects of congestion, as data integrity is most important. The traffic is asymmetrical, with small requests in one direction and larger downloads in another; they are throughput asymmetric. Figure 2.5 is a capture from a web-browsing session. Page requests (or upload) are shown as sent bits (blue triangles), which has a lower data-rate than the received bits (red circles).

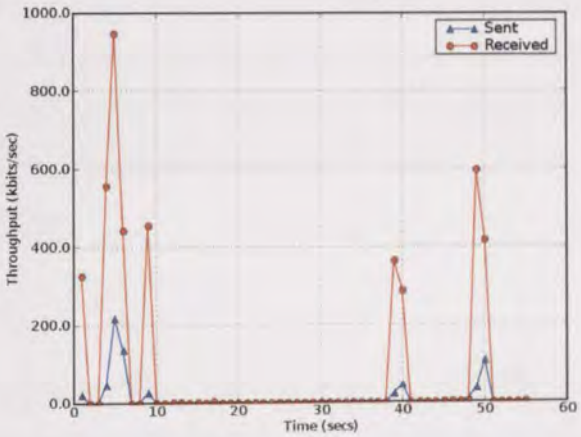


Figure 2.5: HTTP traffic pattern.

2.3 Traffic Management Concepts

In non-QoS networks, different types of traffic are treated the same and suffer the effects of congestion. These effects can be reduced using static or dynamic traffic management techniques (Chalmers & Sloman, 1999). Static management includes specification of requirements parameters and negotiating levels of service based on initial network conditions or resources. Dynamic management determines the response to changes in network conditions, at points in the network. This section explains the terminology of QoS specification, monitoring, and scheduling.

2.3.1 Specification

In trends towards all-IP networks, applications that were previously separate share a common network core. It follows that data flows must be identified to determine service levels; providing

agreements at the application level and QoS controls in the network (Gurijala & Molina, 2004). What a service provider domain should provide the customer network are defined in *service level agreements* (SLAs). The ITU-T define an SLA as:

> "a negotiated agreement between a customer and the service provider on levels of service characteristics and the associated set of metrics. The content of SLA varies depending on the service offering and includes the attributes required for negotiated agreement" (ITU-T Y.1241, cited in Gozdecki et al. (2003, p.156)).

Specifying QoS requires a process of capturing application requirements and policies. Specification may be divided across protocols and layers of abstraction (Aurrecoechea et al., 1998). These include end-to-end policies defined by application requirements, and network based policy that determines how packets are treated by QoS-enabled devices. Specific protocols are discussed in section 2.4.

2.3.2 Monitoring

QoS monitoring techniques can be either active or passive. Active monitoring approaches attempt to inject measurement packets that are tracked to assess QoS performance. This requires a trade-off on the level of monitoring accuracy and adversely affecting existing traffic from monitor packets (Schormans & Pitts, 2004). In passive monitoring, existing packets are measured at points in the network and statistics calculated. Though less intrusive, passive approaches require access to network components (such as routers), which may not always be possible (Schormans & Pitts, 2004).

An approach by the IETF defines QoS metrics measurement in IP Performance Metrics (IPPM) framework (Paxson et al., 1998). The framework defines common issues and provides quantitative methods for performance monitoring. Common QoS metrics are defined for calculations and measurements, as part of the IPPM framework (table 2.3).

Metric	*RFC*	*Summary*
A One-way Delay	RFC 2679 (Almes et al., 1999a)	A one-way path measurement is motivated by different routes that packets may take from both directions. Even if paths are symmetric, there maybe effects from queueing. Also, QoS requirements can be different depending on direction of the flow.
A One-way Packet Loss	RFC 2680 (Almes et al., 1999b)	Necessary for real-time applications. The same motivation for one-way delay measurements.
A Round-trip Delay	RFC 2681 (Almes et al., 1999c)	An indication of congestion is provided when for values above the minimum (based only on propagation and transmission delays). However, some applications require QoS that is different depending on path direction.
IP Packet Delay Variation	RFC 3393 (Demichelis & Chimento, 2002)	A clarification on the ambiguous term 'jitter': the difference in one-way delay. In a stream, the difference in delay between two packets are used to calculate IPDV. An alternate calculation is used for RTP (RFC 3550, Schulzrinne et al. (2003)): consecutive packets are selected and calculated by an *exponential filter*.

Table 2.3: Relevant metrics for IPPM framework

2.3.3 Scheduling operations

Routing algorithms determine the path of packets in the network, so consecutive packet transmissions may experience different delays. To provide QoS, there is a need for packets to be managed at points in the network. The main techniques for QoS controls in the network are queueing and dropping algorithms. In QoS terminology, a combination of techniques are used for *traffic conditioning*, or policing packet flows. Queueing and dropping methods are used to schedule packets for transmission and manage congestion.

Queueing

Present in the network layer and below, first-in-first-out (FIFO) queues are the simplest form of queueing, where each packet is served (transmitted) based on order of arrival. Class-based approaches schedule transmissions based on the type of packet; a voice packet would have priority over web traffic. Other weighted or hybrid queueing methods aim to provide fairer treatment of packets.

Priority queueing, or class-based queueing (CBQ) techniques attempt to prioritise based on class of service (Hersent et al., 2000, p.299). Packets can be marked with priority settings using the type-of-service (ToS) field. A classification process places packets in separate queues, and a scheduling process removes packets from the high priority queues first.

Weighted queueing (WQ) approaches assign percentages to traffic-classes. These values gives the minimum share to the traffic-class under congested conditions. Under congestion all queues have the opportunity to transmit. Though, in a weighted fair queueing (WFQ) approach, scheduling is done based on time proportional to the weight of the queue.

Dropping

When packets arrive faster than they are scheduled, conventional tail-drop or FIFO queues will overflow and packets are dropped. When this happens, applications that use TCP can backoff to relieve congestion. Though this process can be managed by network entities (routers) using random early detection (RED). The RED approach randomly drops packets at different times. The effect is that those TCP flows detecting losses will backoff at different times, thus preventing synchronisation of TCP backoffs for many flows (Hersent et al., 2000, p.335). A weighted variation of random early detection (WRED), assigns priorities to different queues. So a low priority queue will drop packets before those in the medium queue, but before the high priority queue.

2.4 Protocol Support in IP Networks

IP-based networks are "best-effort" in terms of how packets are routed and delivered to end-hosts. This is not usually a problem for delay-elastic service types, such as FTP, e-mail, and web-based applications. However, the requirements for real-time applications have constraints on delay and delay variation. For queueing and dynamic routing in IP networks, there is no

certainty that packets for a particular service types can be delivered within required constraints.

Two main approaches for network resource management have evolved: over-provisioning and explicit resource management (Gozdecki et al., 2003). Over-provisioning attempts to provide more resources than the level of demand, such as the PSTN where a dial-tone can be always found. Explicit resource management attempts to reserve resources or prioritise flows. This section explains the common protocols for QoS support in IP-based networks.

2.4.1 Integrated Services

Integrated services, or IntServ (defined in RFC 1633 (Braden et al., 1994)) is an IETF framework that aims to provide end-to-end resource controls in IP networks. Resources are reserved for particular *flows* according to application level service requests. Intended for real-time services, such as voice and video, the IntServ model must signal network devices (routers) how to treat packets. Signalling resources along the path is the role of Resource reSerVation Protocol (RSVP) (Braden et al., 1997). RSVP uses application requirements and bandwidth policy to request resources along a network path for a particular flow.

An overview of IntServ path setup is shown in figure 2.6. Routers along the path must be IntServ enabled (IntServ domain) with traffic control functions. The path is setup by the sender host by RSVP path messages to the receive host. A receiving host responds with the QoS required for the flow as RSVP reservation message. RSVP is used in one direction, so for applications where a host is both sending and receiving (interactive voice and video), two RSVP paths must be setup.

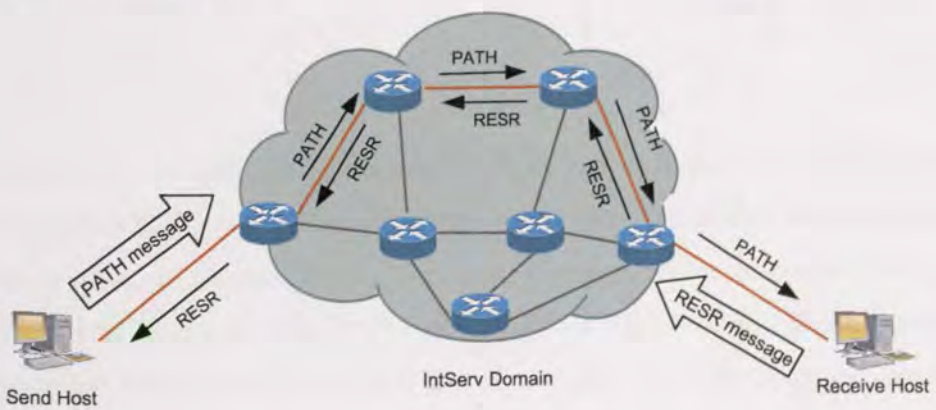


Figure 2.6: Overview of IntServ RSVP setup.

Although the IntServ approach can establish QoS for individual flows in IP networks, it does not scale in larger core networks. Devices in the network must be RSVP aware, which requires additional state monitoring and processing by routers. This limits the scalability of IntServ within the Internet and large backhaul networks (Parziale et al., 2006, p.309). A coarser-grained approach is required.

2.4.2 Differentiated Services

Differentiated Services (DiffServ) is defined in RFC 2475 (Blake et al., 1998) as a solution to the limitations of IntServ scalability. It associates classes to types of traffic flow, creating levels of priority. The prioritising process marks packets in the Differentiated Services field (DS-field[6]), which was IPv4 type-of-service (ToS) and IPv6 traffic class (Parziale et al., 2006, p.310). Codes relating to priority, or Differentiated Service CodePoint (DSCP) are used by networks nodes to determine priority of service. These details are determined by the SLA between the customer and service provider networks (figure 2.7).

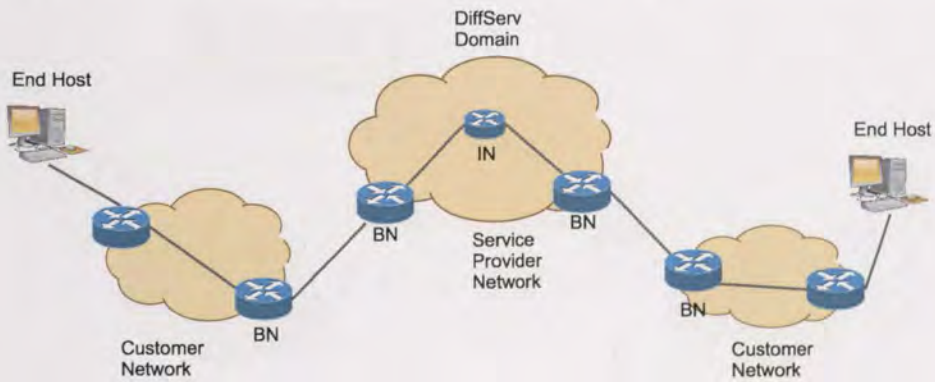


Figure 2.7: Overview of DiffServ

At the boundary nodes (BN), incoming traffic (DS ingress) is classified and conditioning is applied. Traffic conditioning can take multiple configurations that include metering operations such as, average rate, exponential weighted moving average, and token-bucket (Black, 2000, p.260). A traffic condition block (TCB) contains a traffic classifier using DSCPs to determine the next course for the packet, or *per-hop behaviour* (PHB). The PHBs determine which conditioning method is applied, and should be consistent across all routers in the DiffServ domain. Such methods include queueing type, queue parameters, and dropping algorithms (described in section 2.3.3).

[6]Defined in RFC 2474

2.4.3 Combined approaches

The limitations of IntServ is scalability in large networks, whereas DiffServ provides coarse-grained priority of packets based on traffic classes. IntServ over DiffServ aims "to provide an end-to-end, quantitative QoS, which will also allow scalability." (Parziale et al., 2006, p.319). The most likely used approach will pass IntServ messages (RSVP) over the DiffServ domain without interpretation of signalling (Parziale et al., 2006, p.320). This approach is used in Cisco IOS for router management (Cisco, 2005).

Figure 2.8 depicts an architecture for IntServ over DiffServ. The access networks are local intranets that are an IntServ domain where RSVP messaging occurs as normal. Once the messages reach the edge node (R1) they are forwarded to the border node (R2) in the DiffServ domain. The BN node uses the DSCP field to decide how it is treated in the DiffServ domain and forwarded towards host B. Upon receiving the packet from R3, edge node R4 continues with RSVP messaging. Therefore, the DiffServ domain is "virtual" to the IntServ domain, but it does require mapping of PHBs for DSCPs to IntServ flowspec (Parziale et al., 2006, p.321).

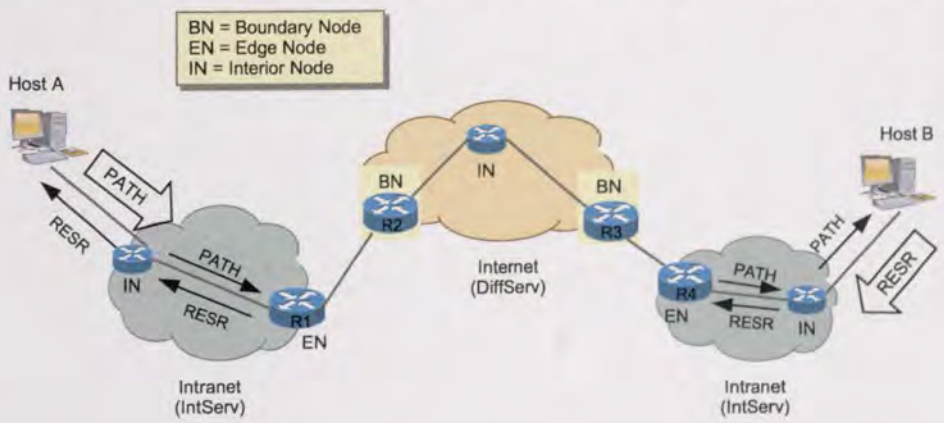


Figure 2.8: IntServ over DiffServ architecture (*Adapted from*: Parziale et al. (2006, p.320))

Concluding Remarks

In early IP networks not all QoS aspects were considered important, as loss and congestion were the main factors for reliable data transfer. Also, different services ran over different networks, such as PSTN for voice. However, IP networks now carry a wider range of services, including voice and video that require strict requirements for QoS.

A discussion of QoS has covered the network entities and application characteristics that affect provision of QoS. Methods to manage network QoS are described in specifications, classifications, and scheduling protocols. These methods have been applied to IP networks in architectures defined for end-to-end QoS reservation (IntServ) and class-based, or aggregated prioritisation (DiffServ).

This discussion has focused on generic QoS aspects and those in the wired network. In the following chapter, mobile networks are introduced that provide wireless access for different types of service, but with further complications for application QoS.

Chapter 3

The Wireless Environment

The process of transmitting bits in wired media does not suffer the same complications of those in wireless. As a shared medium, radio waves are susceptible to interference and propagation effects. Additional signal and error correction techniques are required to provide a reliable channel. Cellular networks are well-established for voice services, providing near-ubiquitous coverage when roaming. Also, wireless local area networks (WLAN) are an access technology that provide high data-rates for office, home, and public hotspot connectivity.

As wireless networks continue to develop and user demands for more services increase, access technologies and networks require quality of service (QoS) support. In this chapter, mobile network architectures are discussed, including wide-area, local-area, and personal-area access technologies. Also, issues of mobility and QoS are discussed as important factors in mobile network evolution.

3.1 Mobile Networks

First-generation cellular (1G) was based on analog transmission to provide voice services. Digital transmission has led to second-generation (2G) networks, such as Global System for Mobile communications (GSM). With voice services established, further development of mobile networks has focused on supporting data. Third-generation cellular (3G) is the culmination of improved data-rates for data, based on a common network core for both voice and data. However, low initial subscribers and costs of implementations (frequency licences and hardware), meant 2.5G was used as a transition technology for some operators without 3G licenses to support data in circuit-switched (CS) networks (Lloyd-Evans, 2002, p.5).

The possibilities for access technologies are shown in table 3.1, and include wireless wide area networks (WWANs), wireless local area networks (WLANs), and wireless personal area

Network	*Standards*	*Data-rates*	*Frequency*	*Mobility*
WLAN	IEEE 802.11b	1, 2, 5.5, 11 Mb/s	ISM 2.4 GHz	Low
	IEEE 802.11a	Up to 54 Mb/s	ISM/UNI 5 GHz	Low
	IEEE 802.11g	Up to 54 Mb/s	ISM 2.4 GHz	Low
	IEEE 802.11n	Up to 108 Mb/s	ISM 2.4 GHz	Low
Bluetooth	IEEE 802.15.1	1 Mb/s	ISM 2.4 GHz	Low
WMAN	IEEE 802.16	134 Mb/s	10-66 GHz	N/A
	IEEE 802.16a	70 Mb/s	2-11 GHz	N/A
	IEEE 802.16e	70 Mb/s		Low/Med
2G	GSM	9.6/57.6 kb/s	900/1800/1900 MHz	High
	GPRS	115 kb/s		High
	EDGE	384 kb/s		High
3G	UMTS/W-CDMA	Up to 2 Mb/s	1900-2025 MHz	High
	Evolved UMTS/EPS	*100 Mb/s		High
	EVDO rev A	*3.1 Mb/s		High
	EVDO rev C (UMB)	288 Mb/s		High
UWB	IEEE 802.15.3	Up to 480 Mb/s	3.1-10 GHz	N/A
Sensors	IEEE 802.15.4	5-200 kb/s		Low

* Lescuyer & Lucidarme (2008)

Table 3.1: Wireless access technologies (*adapted from*: Mahonen et al. (2004))

networks (WPANs) (Prasad & Ruggieri, 2003). Most of these are considered an access technology for services, providing the last-hop between wired networks. The following sections give an overview of WWANs, WLANs, and WPANs. Subsequent sections of the chapter detail architecture and QoS aspects.

3.1.1 Wireless wide area networks

WWANs include cellular networks that provide signal coverage over several kilometres. Initially developed for voice, 2G digital cellular such as GSM, is the most common mobile implementation. In addition to voice, the digital platform provides extensions to use CS data traffic as well as packet data. However, as basic GSM only provides data-rates of 9.6-14.4 kbps (Prasad & Ruggieri, 2003), its suitability for data services is limited.

For packet extensions to cellular access for faster data rates, the General Packetised Radio Service (GPRS) is one such 2.5G technology. The use of 2.5G aims to provide faster data-rates than GSM whilst keeping much of the existing infrastructure. Other examples of 2.5G cellular are Enhanced Data-rates for GSM Evolution (EDGE) and High Speed Circuit-Switched Data (HSCSD). Operators have been using 2.5G technologies, such as GPRS or EDGE with GPRS (EGPRS) schemes as a precursor to 3G services, due to the cost of upgrading in infrastructure

costs and market risks (Lloyd-Evans, 2002).

3G cellular networks aim to support increased data-rates for a range of different multimedia services, i.e. video, voice, and data. The 3G initiative was started by the ITU in 1985, and become known as the International Mobile Telecommunications-2000 (IMT-2000) (Prasad & Ruggieri, 2003). The European implementation is the Universal Mobile Telecommunications Service (UMTS) based on Wideband Code Division Multiple Access (W-CDMA), with other cdma2000 implementations like EVolution Data Optimised (EVDO). Next generation of cellular networks of UMTS include the Evolved Packet System (EPS), which include the Long Term Evolution (LTE) for downlink rates of 100-Mbps. The equivalent cdma2000 specification is EVDO revision C.

3.1.2 Wireless local area networks

WLANs, or WiFi[1] are now commonplace in homes, offices, and public hotspots; providing data-rates of up to 100-Mbps and potential range of around a hundred meters. WLANs technologies include: IEEE 802.11 (or WiFi) and High Performance Local Area Network (HiperLAN). The IEEE has defined standards that operate in the 2.4-GHz (802.11b/g) and 5-GHz (802.11a) unlicensed band. The wireless equivalent to Ethernet, 802.11 uses carrier-sense multiple-access with collision avoidance (CSMA/CA) at the MAC layer.

Developed by the European Telecommunications Standards Institute (ETSI), HiperLAN was also developed to provide high-speed wireless. The HiperLAN/1 standard defined CSMA/CA for MAC, in the 5-GHz band for speeds of up to 20-Mbps. The successor, HiperLAN/2, was later introduced to provide speeds of up to 54-Mbps, using time division multiple access (TDMA) instead of CSMA/CA (Prasad & Munoz, 2003). HiperLAN/2 provided built-in support for QoS and improved integration for cellular. However, 802.11 was not far behind in providing QoS support through 802.11e; although not ratified until 2006.

Even though 802.11 was still behind HiperLAN/2 in some areas of QoS and protocol integration, 802.11 has caught up with QoS support and interworking standards. 802.11, or WiFi, has become the de facto choice for consumer devices and deployment hardware; such is the unavailability of HiperLAN devices in the market. Therefore, subsequent mention of WLAN will mean 802.11 based technologies.

[1] Wireless fidelity - common term derived from the WiFi Alliance.

3.1.3 Wireless personal area networks

Technologies for WPANs are usually lower range than WLANs, but have other benefits such as increased data-rates or less battery consumption. Bluetooth is a common example of WPAN that allow transmissions between many devices over short distances. Other WPANs technologies have been developed for high-speed data, such as Ultra Wide-Band (UWB). Current versions promise data-rates of up to 480 Mbps; much like wired USB. Future developments of UWB are targeted at high-definition television (HDTV) over wireless. Extensions to standards for UWB, such as IEEE 802.15.3 are being researched using the 60-GHz spectrum to provide data-rates of 2-4 Gbps (Razavi, 2008).

Ad hoc and sensor networks are also considered WPANs. Low-powered networks of devices are useful in the ubiquitous computing context. The ZigBee framework is one such example of low-powered WPANs, that operates in the 2.4-GHz band. ZigBee supports data-rates of 250-kbps, with low latency and power management; suitable for home networking of control devices (Adams & Heile, 2006).

3.2 Radio Transmissions

Radio transmissions are subject to effects from weather, propagation, multipath, frequency, modulation coding schemes, and power levels. This section introduces the wireless channel and the effects on radio design.

3.2.1 Propagation effects

Signals above 30-MHz exist as line-of-sight propagation (Stallings, 2002, p.107), common in cellular and WLANs. As a signal is transmitted the strength is reduced with distance and noise, known as attenuation. Types of noise include the thermal noise that exist through electronics; and interference, or crosstalk from other signals in the same frequency; and atmospheric absorption and foliage.

Signal errors can be caused through noise and attenuation loss. To be received without errors, a signal must be transmitted with sufficient power. Measurements of signal power over noise levels provides a useful metric, signal-to-noise ratio (SNR). The SNR can determine bit-rate because a higher bit-rate needs more power to rise above the noise. Therefore, lower SNR value increases the probability of errors in the signal.

Without obstacles, a transmitted signal strength degrades in proportion to the square of the distance to the receiving antenna, otherwise known as free space loss (Stallings, 2002, p.111). More complex propagation effects exist in cities and urban environments. For propagation effects in large cities, the empirical Okamuru-Hata model is often used (Lloyd-Evans, 2002, p.130). The wireless channel is also complicated by other factors that can interfere with signals, such as obstacles that cause refraction and multiple path fading.

3.2.2 Multipath effects

For line of sight signals, obstacles between transmitter and receiver antennas alter the signal. For direct line of sight, such as satellite and point-to-point wireless links, there is not so much of a problem. Other types of transmissions are altered by signals refracted off of buildings, causing signals received to be time-shifted or with errors. This is compounded by the mobility of antennas that introduce fading effects.

Types of fading can be taken into account and modelled, such as Rayleigh or fast fading when there is no line-of-sight path and signals are scattered. Rayleigh fading has been defined in Jakes' model (Jakes, 1974) that can account for Doppler shift. The Rician fading model describes a direct signal and other multipath signals. As such, Rician fading is useful for describing indoor and where there is a dominant path, and Rayleigh for outside or urban areas (Stallings, 2002, p.122).

3.3 Mobility in IP Networks

In IP networks, the IP address is a unique identifier of the host on a network point-of-attachment (PoA). For wired networks it is unlikely that a host would need to regularly change IP address. Wireless devices allow movement such that hosts can change PoA (roaming), and thus the IP address could change. This would break the connections from upper-layers such as TCP, which rely on IP source/destination address and port pairings. Mobility solutions, such as Mobile IP were originally conceived to provide a workaround for changes in IP address. The following terminology defines different levels of mobility:

- *Transport and application mobility* protocols may use extensions to assist session setup and maintenance. They may be used to supplement lower-layer protocols for session continuity.

- *Macro-mobility* is handled at the layer of routing and addressing, where there is a change in administrative domains or subnets within domains; i.e. Mobile IP.
- *Micro-mobility* protocols provide roaming support for devices that change between APs within subnets of a domain to assist Mobile IP performance (Campbell et al., 2002). Protocols for micro-mobility support include Cellular IP, Hawaii, and Hierarchical IP (compared in Campbell et al. (2002)).

Figure 3.1 provides a reference scenario for the terms of mobility and handovers for two administrative domains. In this example, a mobile node (MN) moves between base station (BS) 1 and 2, traffic is routed via access router (AR) 1, or micro-mobility handover. Moving between AR1 and AR2 is an inter-AR handover. As the MN moves between BS 3 and 4, subnets may change and require macro-mobility handover protocols. Finally, the MN moves between domains, for BS 4 and 5. The remainder of this section presents issues and solutions relating to different levels of mobility.

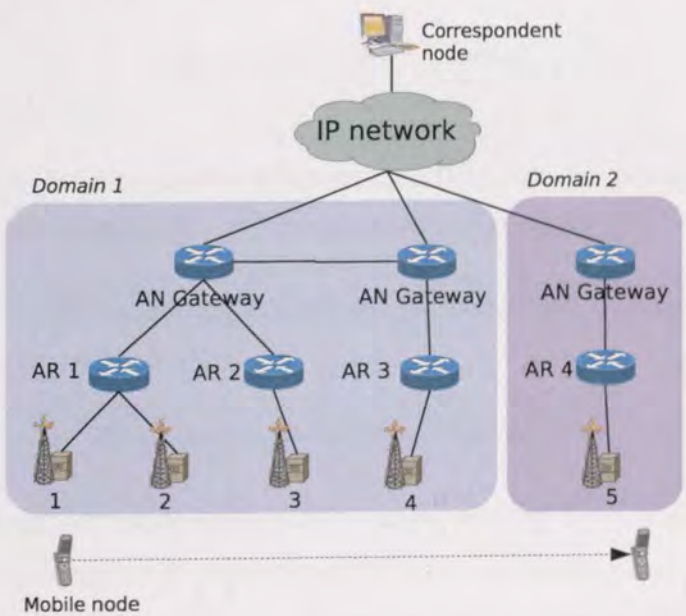


Figure 3.1: Reference scenarios for handover mobility (*Source*: Manner et al. (2002, p.148))

3.3.1 Macro-mobility support

There are two main protocols for macro-mobility in IP networks: Mobile IPv4 and Mobile IPv6. Mobile IPv4 (MIP) was developed by the IETF for mobility management in IP networks. Subsequent enhancements have been made for IPv6.

MIPv4

The protocol defined in RFC 3344 (Perkins, 2002) allows mobile hosts to roam between MIPv4 foreign networks by exchanging registration updates bindings. MIP agents in the network solicit advertisement messages, or a host may request a registration. A mobile node (MN) registers with the foreign agent, which then updates the home agent with the care-of address (CoA) whilst in the foreign network. The following describes routing of traffic in a MIP scenario (figure 3.2).

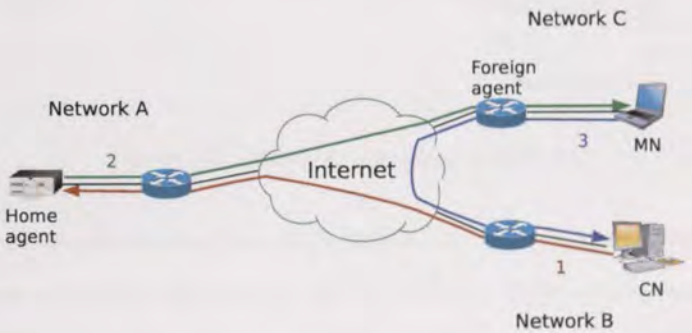


Figure 3.2: Mobile IP operation.

1. MN registers its CoA (provided by foreign agent (FA)) with the home agent (HA). Traffic from the correspondent node (CN) is sent to the home network, A.

2. The home agent intercepts packets destined for the MN and re-routes them to the MN COA in foreign network, C.

3. Packets from the MN are routed directly to the CN in network B.

Although this process solves the problem of IP-address mobility, there are performance costs. The triangular routing pattern of packets from CN to MN via the home agent, adds additional latency. A route optimisation (RO) extension was developed to remove this effect, as part of the MIPv6 implementation.

MIPv6

Mobility in IPv6, as defined in RFC 3775 (Johnson et al., 2004), integrates the RO elements. This eliminates successive packets routed via the HA by using a binding update between MN and a CN. Figure 3.3 describes the variation. When route optimisation is used, a binding update

is sent to the CN containing the CoA of the MN. All subsequent packets are sent directly to the MN at the foreign network. Any initial traffic, or those packets before a binding update is performed, may still be directed through the home agent.

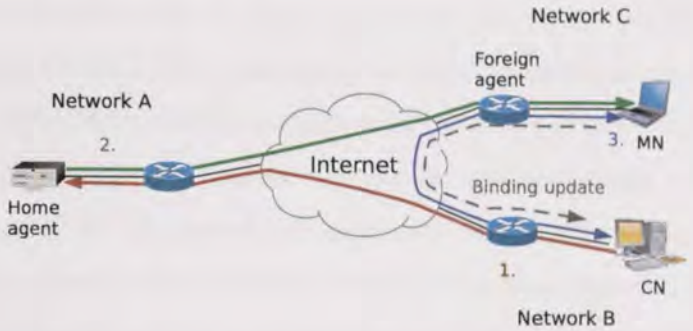


Figure 3.3: Route optimisation in MIPv6.

As networks are upgraded to support IPv6, MIPv6 will provide layer 3 mobility. Such networks will include more real-time services that are sensitive to the time between making handovers (handover latency). An extension protocol for Fast handovers of Mobile IPv6 (FMIPv6), proposed in RFC 4068 (Ref, 2005) aims to reduce this latency by eliminating configuration delay and reducing delay of binding-updates (Dunmore & Pagtzis, 2005).

With more devices supporting wireless access, suitable mobility management is needed. Security and QoS are two issues connected with mobility. MIPv6 includes features for supporting security, particularly use of IPsec to support binding-updates authentication (Prasad & Ruggieri, 2003, p.122). QoS issues relating to handover will be further discussed in chapter 4.

3.3.2 Application and transport mobility

To handle mobility at the transport layer, there must be notification of events by the lower layers, such as IP address and DHCP, so address bindings can be reconfigured (Calvagna et al., 2005). This requires session mobility built into the application. However, it is unlikely that applications would be designed to support mobility on their own (Eddy, 2004). Session initiation protocol (SIP) and stream control transmission protocol (SCTP) are transport signalling schemes to assist presence and flow management in a mobility context.

Session initiation protocol

SIP is defined in RFC 2543 (Rosenberg et al., 2002) for interactive voice and video applications requires setup and termination. It uses a unique address of call parties using the format:

user@domain. SIP performs a similar function to protocol H.245 defined in ITU-T H.323.

Stream control transmission protocol

SCTP defined in RFC 4960 (Stewart, 2007) is a reliable protocol that provides multi-homing and control of multiple media flows, used mainly for control signalling but also for data (Lloyd-Evans, 2002, p.180). SCTP defines *associations* as end-to-end connections between IP addresses which can have multiple streams. Streams are separate from the association, which means streams can be moved between associations to support multi-homing. Extensions for SCTP have been proposed to allow changing IP addresses in association for host with multiple interfaces, such as dynamic address reconfiguration (DAR) (Stewart et al., 2007),

3.4 Wireless Wide Area Networks

Wireless access has allowed seamless roaming for voice services, common to 2G cellular networks. Upgrades to cellular networks, such as 2.5G and 3G, provide higher data-rates for multimedia services. Developments in metropolitan area networks (MANs) have led to WiMAX as the main candidate for localised broadband wireless. This section describes architectures and technologies towards high-speed data and common all-IP core networks.

3.4.1 Second-generation cellular

Second-generation or 2G cellular provides the digital voice services that replaced analog systems. The most used 2G implementation is the Global System for Mobile communications (GSM). The digital system meant there was potential to extend the air-interface to support data. The first extension uses the GSM infrastructure for high-speed circuit-switched data (HSCSD). Using more time-slots, HSCSD provides data rates up to 57 kbps, but reduces channels for voice access (Prasad & Ruggieri, 2003). The impact of HSCSD to other users required an enhanced, or packetised approach.

Packet-based extensions to cellular have become known as 2.5G technologies. One of these 2.5G technologies is the general-packetised radio service (GPRS), which uses the same time-division multiple-access (TDMA) as in GSM. GPRS uses shared time-slots between users that are allocated depending how much spare capacity is available. This gives GPRS potential data-rates of up to 170 kbps.

GPRS uses most of the existing GSM infrastructure, with additions to support packet data. Figure 3.4 shows an overview of a GSM/GPRS architecture. In GSM, the base station controller (BSC) provides handover and load controls for terminals, and the mobile switching centre (MSC) provides the routing into the PSTN and call-setups. Additions to BSC allow connections to the service GPRS support node (SGSN). The SGSN performs a similar function to the MSC by providing routing, encryption, authentication, access (Prasad & Ruggieri, 2003). To reach external packet networks such as the Internet, the gateway GPRS support node (GGSN) provides further mobility, routing, and location updating via the home location register (HLR).

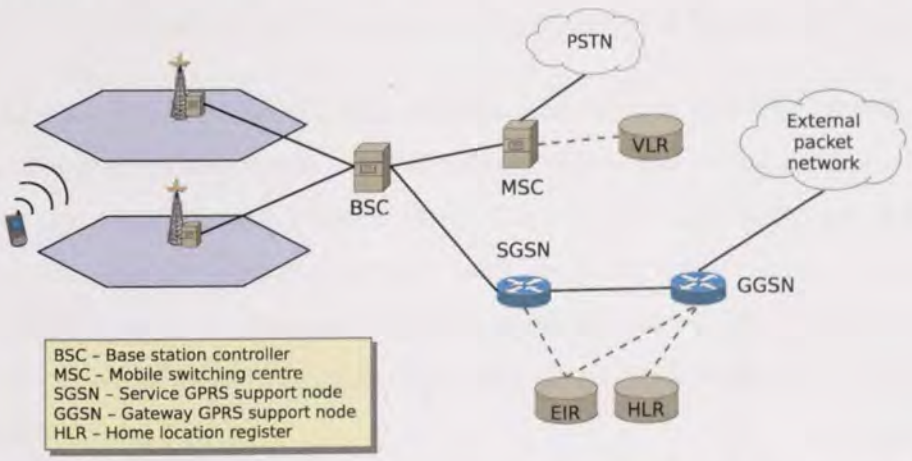


Figure 3.4: GSM-GPRS architecture overview (*adapted from*: Prasad & Ruggieri (2003, p.6))

Further developments for data services, such as Enhanced Data-rates for GPRS Evolution (EDGE), use different modulation schemes to provide up to 384-kbps with minimal changes to GPRS infrastructure (Prasad & Ruggieri, 2003). The benefits of GPRS/EDGE is 'always on' availability for quick access to data services, and pricing according to transmission volume rather than call duration as in HSCSD (Lloyd-Evans, 2002, p.3). GPRS supports the same QoS classes as UMTS, which are detailed in the following section.

3.4.2 Third-generation cellular

Third-generation (3G) cellular plans were proposed by the ITU in IMT-2000 to provide increased data-rates for multimedia services. More diverse services targeted for 3G include: video, voice, messaging, email, and web browsing. A wider range of services require more upgrades than those for 2.5G, and subsequently more expensive deployments.

The European variant of IMT-2000 is Universal Mobile Telecommunications Service (UMTS).

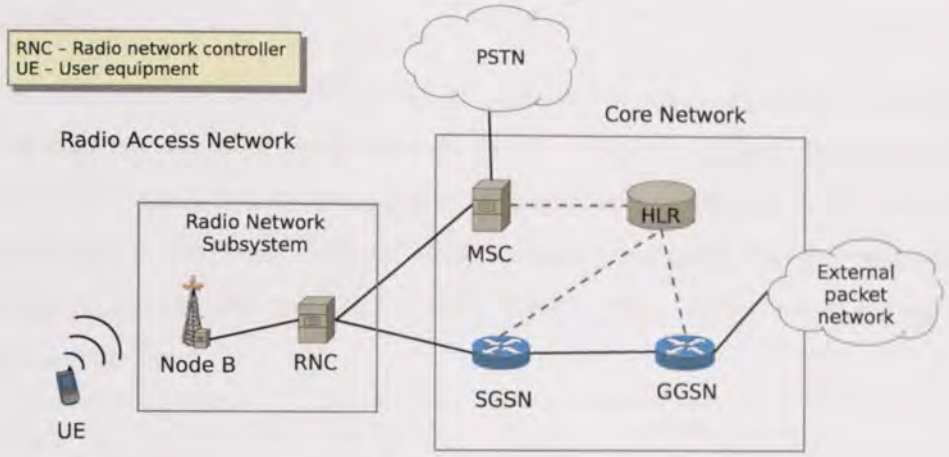


Figure 3.5: Overview of UMTS architecture (*Source*: Prasad & Ruggieri (2003, p.11))

Figure 3.5 presents an overview of UMTS. The radio access network (RAN) defines user equipment (UE) that are present in cells. Node B are base-stations that control multiple cells and are attached to a radio network controller (RNC). The RNC provides a similar role to BSC in GSM, providing circuit-switched voice to the MSC, and packet data to the SGSN in the core network (CN). Data in the CN is over asynchronous transfer mode (ATM) in Release 3 (also known as Release '99 in previous terminology), although all-IP is specified for Release 4 and 5 (Prasad & Munoz, 2003).

Development beyond existing 3G implementation of UMTS and cdma2000 have been defined by the 3rd Generation Partnership Project (3GPP) and 3GPP2 (North America). 3GPP Release 8 is the Evolved Packet System (EPS), which defines the next stage of UMTS to provide even higher data-rates, better QoS support, and an all-IP core network. The two main differences between previous versions of UMTS is a new core network, or Evolved Packet Core (EPC) using IP and the Evolved-UTRAN (E-UTRAN) for radio access. EPC is referred to as System Architecture Evolution (SAE) in the 3GPP standard documentation, and E-UTRAN as Long Term Evolution (LTE) (Lescuyer & Lucidarme, 2008). As well as being backward compatible with previous UTRAN, LTE provides low-latency radio access to support a range of services like VoIP, video, real-time gaming, and streaming. The equivalent to 3GPP2 for cdma2000 North American implementation to Evolved UMTS is 1xEV-DO Revision C[2]. Also known as Ultra Mobile Broadband (UMB), the Revision C of 1xEV-DO is based on an all-IP core network, and provides a peak theoretical data-rate of up to 288 Mbps.

[2]3GPP2 document C.S0084-000 available from `http://www.3gpp2.org/Public_html/specs/alltsgscfm.cfm`

QoS support

The fundamental difference in UMTS to 2G networks is data and voice packets are transmitted over the same medium. This presents the challenge to support end-to-end QoS perceived by the user as well as utilising the transmission medium efficiently (Baudet et al., 2001, p.41). To provide QoS in UMTS networks, the 3GPP has defined four traffic classes: conversational, streaming, interactive, and background (3GPP, 2004b). These traffic classes and are summarised in table 3.2.

QoS Class	*Transfer delay requirement*	*Transfer delay variation*	*Low bit error rate*	*Guaranteed bit error rate*	*Examples*
Conversational	Stringent	Stringent	No	Yes	VoIP, video-conferencing, audio-conferencing
Streaming	Constrained	Constrained	No	Yes	Broadcast services (audio, video), news, sport
Interactive	Looser	No	Yes	No	Web browsing, interactive chat, games, m-commerce
Background	No	No	Yes	No	E-mail, SMS, database downloads, transfer of measurements

Table 3.2: Traffic classes in UMTS (*Source*: Baudet et al. (2001, p.42))

The UMTS QoS architecture (figure 3.6 on the next page) is based on a number of bearer services that accept QoS attributes, as defined in 3GPP TS 23.107 (3GPP, 2004b, p.18). End-to-end bearer services provide QoS definition from one terminal equipment (TE) to another, and may traverse external non-UMTS networks. The UMTS bearer service provides QoS in UMTS by using services provided by the radio access bearer (RAB) and core network (CN) bearer. The RAB provides services and interfaces between the TE and UMTS terrestrial radio access network (UTRAN), while the CN bearer services provides interfaces for the UTRAN and external networks, such as the Internet and public switched telephone network (PSTN).

3GPP release 6 (3GPP, 2004b) specifies that the CN shall use DiffServ if IP is used, and DiffServ mappings for asynchronous transfer mode (ATM) to external networks. However, it is up to applications to supply QoS setup and resource specification through signalling, such as

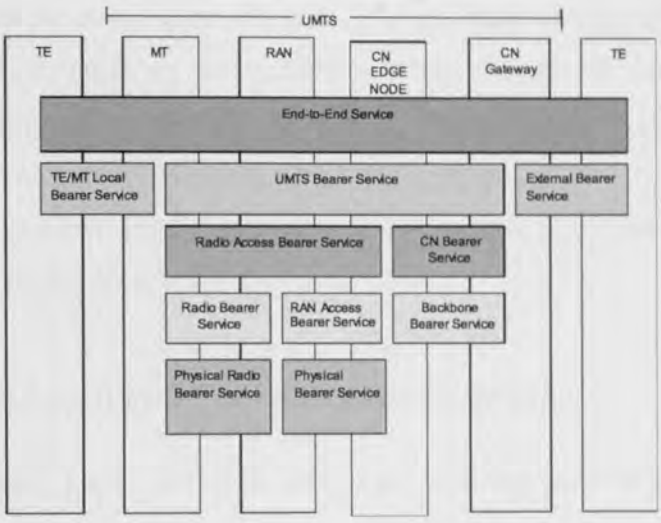


Figure 3.6: UMTS end-to-end QoS architecture (*Source*: (3GPP, 2004b, p.10))

SIP or SDP (Lloyd-Evans, 2002, p.230). The QoS settings used in the CN will thus determine the traffic treatment offered to subscribers.

3.4.3 Handoff

In cellular networks, terminal movement changes base-station association. Handoffs between base station cells must be fast enough to limit interruption to calls. This requires handoff procedures using variations of control:

- *Mobile-initiated.* Handoff is determined by the mobile node when received signal drops too low.
- *Mobile-assisted.* The mobile provides feedback to the network entities, that then make the handoff action.
- *Network-initiated.* This is performed by network entities to decide when a handoff between BS should occur. It may be based on metrics such as cell load, speed of node movement.
- *Network-assisted.* Entities within the network provides metrics and status to the mobile client, which then initiates handover procedures with the network. There are arguments that this is a useful approach when the mobile client has a choice of different air-interface types (Saito et al., 2005), which is relevant to next-generation networks, or those beyond 3G.

Handoff procedures can use a number of metrics, primarily based on signal strength indicators from local base stations (Stallings, 2002, p.293). In TDMA systems, cell associations are usually changed before setup on a new cell; a *hard handoff*. Alternatively, connections can be kept alive between cells for a short period before handoff; a *soft handoff*. Used in CDMA networks, soft handoffs are smoother and do not require the hysteresis in hard handoffs, reducing the 'ping-pong' effect (Chen, 2003, p.52).

3.5 Wireless Local and Personal Area Networks

WLANs in corporate, public, and home settings are replacing fixed-cable Ethernet for in-building connectivity. Wireless personal area networks (WPANs) include technologies for very short-range (within 10 meters) connectivity of devices. Common applications being used in WLAN and WPAN environments include: audio and video streaming; instant messaging; web and email; and voice. This section presents the common media-access methods and standards for WLANs and WPAN, and extensions to support QoS.

3.5.1 IEEE standards for WLAN

Common media-access technologies of WLANs are based on the IEEE standards 802.11 (a/b/g/n). These have become the *de facto* standards for WLAN-enabled devices. The most common WLAN-enabled devices are based on 802.11a/b/g. 802.11a was the first to be released, operating in the 5-GHz band and based on orthogonal frequency division multiplex (OFDM) in the physical layer. It is more resilient to interference and can provide up to 54-Mbps, whereas 802.11b can only offer data rates of 1, 2, 5.5, and 11 Mbps. 802.11b uses direct sequence spread spectrum (DSSS) in the physical layer, operating within a 30-MHz frequency-range of 2.4-GHz the industrial, scientific and medical (ISM) band (Bar-Shalom et al., 2003). The typical range of 802.11b devices can be up to 100 metres.

Enhanced data-rates for WLAN has been defined in 802.11g. It was developed to provide up to 54-Mbps in the 2.4-GHz band. It uses OFDM similar to 802.11a, but limiting the number of non-overlapping APs to three (Bar-Shalom et al., 2003). Further data-rate and range enhancements is provided by the 802.11n: multiple-input, multiple-output (MIMO). The 802.11n draft uses multiple receive and transmit antennas to overlay channels of the 2.4 and 5-GHz frequencies.

3.5.2 QoS support in WLANs

In standard 802.11, QoS features are minimal or rarely used. Most likely this is due to the standard being developed at a time when WLANs were expected to only cater for data (web and email), and not the mix of multimedia traffic that includes real-time services. The medium access control (MAC) layer provides two operations: distributed co-ordination function (DCF) and the point co-ordination function (PCF). DCF provides access based on the CSMA/CA (carrier-sense multiple-access with collision avoidance) mechanism for best effort delivery in ad-hoc mode and infrastructure modes. It is based on clients contending for airtime by random backoffs and RTS/CTS mechanisms. PCF is a polling mechanism that operates only in infrastructure mode with the access point (AP) operating as the Point Coordinator (PC). It is this PCF mechanism that aims to provide a fairer provision for time-constrained clients. However, stations polling cannot be guaranteed at regular intervals, due to unpredictable delays in beacon transmission (caused by periods of polling interspersed with DCF contention) and uncertain duration of client transmissions (Mangold et al., 2003). This equates to uncertain operation of fair provision, inadequate for real-time applications. Standard 802.11 specifications are therefore insufficient for providing QoS.

IEEE began working on the QoS extensions by forming the working group 802.11e. Early adoption as a subset of 802.11e was specified by the WiFi Alliance as WiFi Multimedia (WMM) (Alliance, 2004). The 802.11e standard (IEEE, 2005) was ratified in 2005 and extends MAC layer functionality by building upon existing 802.11 to provide the Enhanced Distribution Coordination Access (EDCA) and HCF (hybrid control function) Controlled Channel Access (HCCA).

- **EDCA** takes the mechanism of DCF and adds QoS support by introducing traffic categories (TCs) or service differentiation. This means that multiple traffic flows (up to eight) can exist within a particular station. Subsequently, each TC competes for channel access through a process of *virtual backoff*. It is this backoff procedure that enhances DCF, with priority flows requiring less time to wait for backoff and more chance of being transmitted.

- **HCCA** is similar to the PCF polling mechanism controlled by a hybrid co-coordinator function (HCF) at the AP.

Early simulation studies[3] have shown 802.11e is able to prioritise flows and improve QoS to high priority flows compared to legacy 802.11 schemes.

3.5.3 Personal-area technologies

WPANs are primarily short-range for allowing multiple different devices to interact, through dynamic forming of networks, such as pico-nets (Bluetooth) and sensor networks (ZigBee). WPANs are useful for generic data transmission for multimedia. Sending small files, and interactive audio is well suited to Bluetooth. For larger files and video streaming, developments in UWB technologies could potentially offer bandwidth increases of up to 4-Gbps (Razavi, 2008).

In sensor network deployments, with potentially thousands of nodes, low-power and reliable transmissions are important. The ZigBee platform provides short-range connectivity with data-rates up to 250-kbps; suitable for monitoring applications that send small, and infrequent messages (Adams & Heile, 2006). The physical layer is based on 802.15.4. The MAC portion uses CSMA/CA for transmission control. Operating at 2.4-GHz, the PHY sub-layer uses quadrature phase-shift keying (Q-PSK) as its modulation scheme for robust and reliable signal transmissions (Adams & Heile, 2006). Upper layers use the ZigBee stack for networking and transport of application messaging. The ZigBee stack provides devices with routing, and encryption, for applications.

Concluding Remarks

The wireless channel is a shared medium, thereby subject to complications not present in a wired channel. Wireless provides the ability to roam, but requires additional management by the network through mobility protocols. Wireless tends to be an access method for the last-hop of a network. The different types of access include wide area wireless networks, such as cellular (WWANs); wireless local area networks (WLANs); and personal area networks (WPANs). Each type of access method requires interface devices with different wireless channel and QoS capabilities.

Although data-rates are increasing in some wireless interface technologies together with better support for QoS, there are still limitations for truly ubiquitous QoS. A step forward would exploit the differences in access types for improved QoS; since WLANs provide higher

[3]Grilo & Nunes (2002); Gu & Zhang (2003); Mangold et al. (2003); Lindgren et al. (2003)

data-rates and are less-costly than cellular, but cellular provides wider coverage and controlled access. In the next chapter, the discussion moves to converged networks and heterogeneous access for the future wireless access networks.

Chapter 4

Heterogeneous Wireless Environments

Evolution of communications networks and services is driven by convergence towards IP core networks, while wireless becomes the access technology of choice. Improved data-rates for wireless enables wider range of services in more locations, such as third-generation cellular (3G). The vision beyond 3G, or forth-generation (4G), is an integration of networks using different radio access technologies and standards. However, this introduces additional issues for QoS and network complexity.

This chapter discusses heterogeneous wireless environments as a solution for providing QoS. An overlay of different network types creates opportunities for continuous connectivity. User and application QoS issues can be improved by exploiting multiple wireless interfaces and networks. Although, there are still unresolved issues with this concept that attracts research interest, such as mobility management and interface selection. Finally, a framework is presented that describes the solution context and interface selection as motivation for this thesis.

4.1 A Solution for Improved QoS

Wide-area cellular, WLANs, and personal area networks (PAN), have different capabilities and provide alternative options to access online services. One distinction is coverage area. The following discussion considers different wireless interface technologies to provide opportunities to improve QoS. Industry trends have shown a progression towards integration of previously separate network access methods, which is evident from large European Framework projects and standards organisations. From an *overlay pattern* of networks, the concept has progressed to being "*always best connected*" (Gustafsson & Jonsson, 2003), with multi-service architectures (Malyan & Lenaghan, 2003) integrating multiple access technologies and protocols.

4.1.1 The overlay pattern

wireless interface technologies that use different radio frequency bands and modulation schemes have led to an *overlay* of network coverage and geography (Stemm & Katz, 1998; Calvagna et al., 2005). The overlay pattern describes a model of data networks that decrease in coverage area: from satellite coverage to personal area networks in the order of a few square-meters (figure 4.1).

An overlay model illustrates the potential for better connectivity by using multiple access technologies. In homogeneous access networks, handoff[1] is a commonly used term for changing between base-station (BS) or access-point (AP). The overlay model distinguishes between *horizontal* and *vertical* handovers (Stemm & Katz, 1998). Horizontal handovers are between the same access technology (homogeneous), and vertical handover refers to changing between different link technologies (heterogeneous).

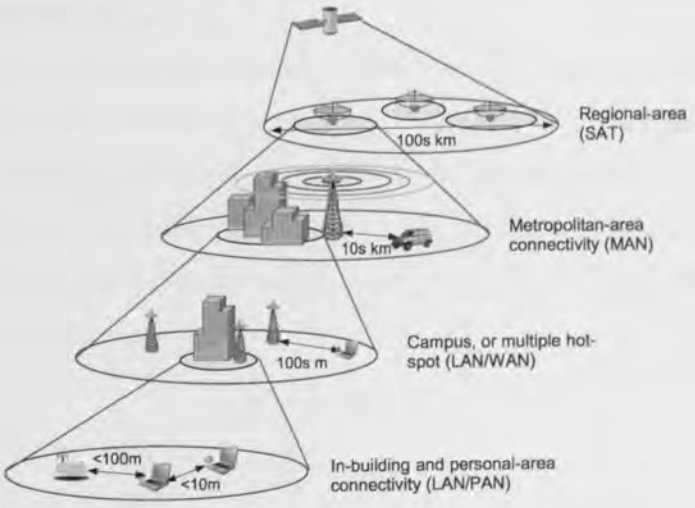


Figure 4.1: Overlay of networks (*adapted from*: Stemm & Katz (1998))

The common wide-area access type is cellular. Developments in 3G provide an improved platform for data services over cellular. In the residential, corporate, and public sectors, WLANs have become the main access choice. The benefits compared to wired options include: flexibility for local roaming, low installation cost, and scalability. With cellular providing wide area coverage, and WLANs providing high data-rates in hot-spots and indoors, these technologies are expected to play important roles in more integrated access networks (Lehr & McKnight, 2003).

[1] 'Handoff' or 'handover' are often used interchangeably in the literature. Though for subsequent discussions, handover is preferred when discussing changes between different access technologies.

4.1.2 Fixed-mobile convergence

Initiatives for integrated access networks are being pursued under terms such as, *fixed mobile convergence* (FMC). FMC aims to leverage services in mobile networks and other access types, such as voice over WLAN. Examples of FMC include unlicensed mobile access (UMA), also known as generic access network (GAN) (3GPP, 2007). The GAN defines an architecture for connectivity of cellular services to other access networks through dual-mode handsets, with additional protocol and mobility support. Other research projects have investigated generic access models in terms of security, roaming, handovers, and QoS. These issues were covered in European framework five projects, including: BRAIN (Broadband Radio Access for IP-based Networks, IST-1999-10050 (2001)); its successor, MIND (Mobile IP-based Network Developments, IST-2000-28584 (2002)); and MobyDick (IST-2000-25394, 2003). A framework six project aims to develop the ubiquitous heterogeneous access model, known as Ambient Networks (IST-507134, 2007).

3G cellular and WLANs interworking have become the initial focus for developing heterogeneous access networks. Standards bodies, such as the Third-Generation Partnership Project (3GPP), have followed the industry trend of FMC by defining frameworks for 3G-WLAN interworking (see, 3GPP 23.234 (3GPP, 2004a)). Integration options between WLAN and 3G (figure 4.2) are commonly cited as: open-coupled, loose-coupled, and tight-coupled (Prasad & Munoz, 2003; Ruggeri et al., 2005):

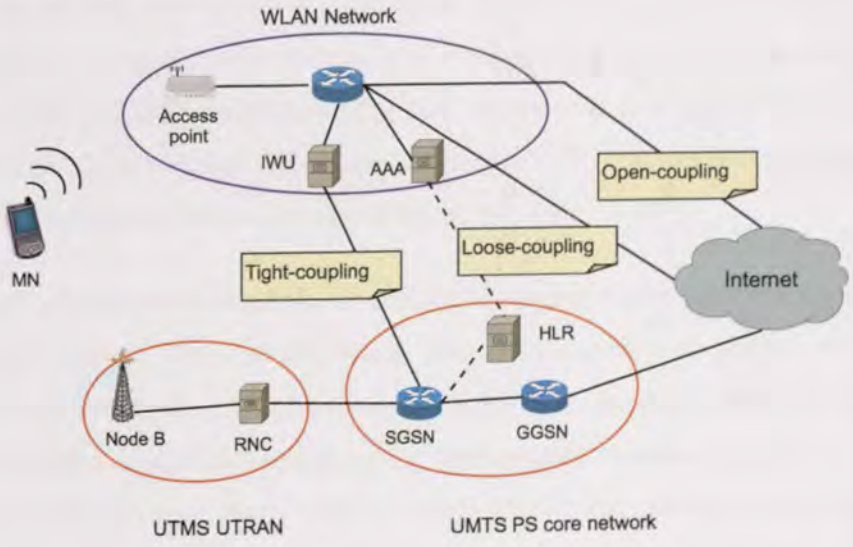


Figure 4.2: Coupling of UMTS 3G cellular and WLAN access networks

- **Open coupling.** There is no direct interface between the access network and cellular core. In figure 4.2, data traffic is otherwise routed via an external packet data network (PDN), such as the Internet. If the client mobile node (MN) is equipped with both interfaces, mobility protocols (mobile IP) can provide roaming between networks. Thus, the MN applications could be seamless, if the networks support it. Disadvantages of this type of connectivity include the potential for packet latency and duplication problems in mobile IPv4, which are detrimental to real-time services. However, optimisation schemes for fast handovers and mobile IPv6 could be employed.

- **Loose coupling** provides some connectivity with the WLAN access network and core UMTS network. The connection is defined as signalling between authentication, authorisation, and accounting (AAA) server and home-location register (HLR). The interface is used to provide billing and access-control only, with data sent across the Internet. A variation of this would use a mobile proxy on either the WLAN or UMTS side.

- **Tight coupling** provides a data and signalling connection between the WLAN access network and UMTS core network. A gateway proxy, or inter-working unit (IWU) in the WLAN network connects to the UMTS. Further variations of tight-coupling are given by Prasad & Munoz (2003, chp.2).

- **Integrated** describes a similar or variation of the tight-coupled approach. The WLAN access network may be seen as a cell of the UMTS network, or UMTS as an extension of WLAN. In the access network, an inter-working unit (IWU) is used between the serving GPRS support node (SGSN). The IWU can be used to emulate the radio network controller (RNC) of the UTRAN, and connect to SGSN. Or, the IWU can emulate the SGSN, and connect to the gateway GPRS support node (GGSN).

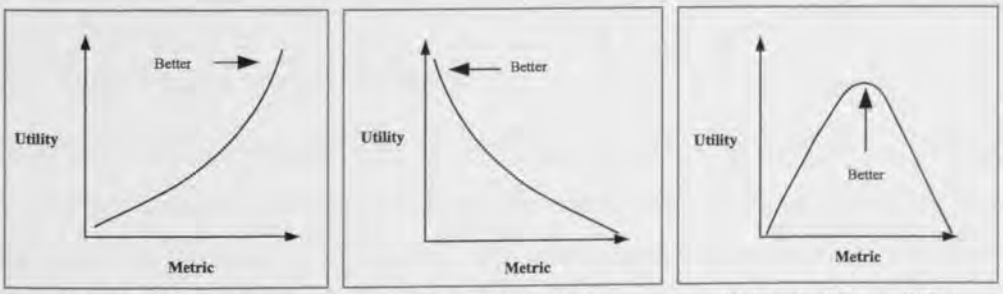
The types of connectivity between WLAN and UMTS are presented in industry standard efforts (3GPP, 2007, 2004a). These provide details of interworking recommendations for specific technologies for operators. A loose-coupled approach is more likely to be favoured initially by operators, as it provides a modular and readily implementable solution supported by existing infrastructure (Ruggeri et al., 2005). However, even a specific integration solution draws further complications and issues that are both technological and commercial.

4.2 Issues in Integrated Access Networks

Different architectures and integration methods have been proposed by industry and standards bodies. This provides a benefit for application QoS. As an access technology, WLANs benefit from higher data-rates and lower latency; though this depends on physical deployment and load. The frequency range of WLAN is in the shared Industrial, Scientific and Medical (ISM) unlicensed band, and therefore is susceptible to other device transmissions (interference). Deployment cost of infrastructure and spectrum licensing is much lower for WLAN than cellular, which determines service pricing and billing differences (Lehr & McKnight, 2003). The differences of access network also presents additional issues for QoS in handovers, session continuity, access control, and security.

4.2.1 QoS management

There are some commonly used measures of performance relating to speed, reliability, and availability. Technical measures of QoS, or metrics, include: throughput, delay, jitter (delay variation), latency, and packet loss. Metrics can be defined by their utility, as being: higher is better (HB); lower is better (LB); and nominal is best (NB) (Jain, 1991, p.40). Figure 4.3 depicts these three metric types. Where higher values are preferable (HB), utility increases. An example of this metric type is throughput in a file download. For metrics such as latency, jitter or packet loss, lower values (LB) have higher utility. The nominal type is probably least common though used to describe utilisation: lower values means unused capacity, but very high values are undesirable–since this begins to affect responsiveness or other performance metrics (Jain, 1991, p.40).



(a) Higher-better (HB) (b) Lower-better (LB) (c) Nominal-better (NB)

Figure 4.3: Metric types (*Source*: Jain (1991, p.41))

QoS metrics change with wireless and network conditions, and most measures are continuous. Figure 4.4 represents an abstract model of wireless QoS for three distinct links in a multi-

dimensional parameter space. Two example parameters represent the operating capabilities of the link, indicated by square regions. Within these parameters the current performance levels can change (dashed-line region). The oval shaded region represents an acceptable performance (determined by user perception or application requirements) of the parameters. Overlapping performance measures between link 1 and 2 (within the acceptable region), means that they both provide similar QoS. However, there may be instances when one link moves out of the acceptable QoS region. Comparing performance between different link types is further complicated by using more than two QoS metrics.

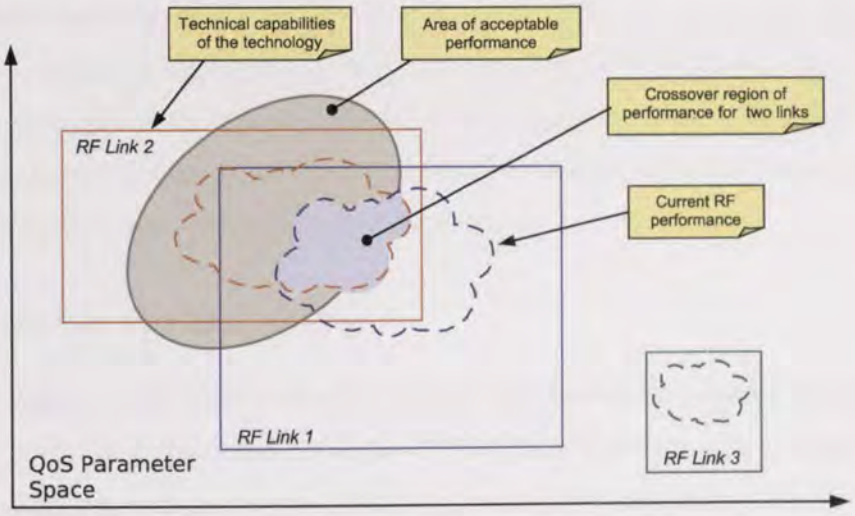


Figure 4.4: Parameter space possibilities and QoS performance crossover.

QoS performance can vary depending on different factors, for example network congestion and applications types. For interactive applications that require seamless connectivity, the QoS performance space can be affected by movement and roaming.

4.2.2 Roaming and session continuity

Roaming behaviour and QoS determine mobility requirements. Delay sensitive applications require seamless handovers within defined limits. Though in some situations, such as handovers, seamlessness cannot always be guaranteed. The mobility protocols can have a high handover latency–the time it takes to complete a handover[2]–which means services with strict latency requirements, such as voice, will incur interruptions or lost connections. Where seamlessness

[2]The measurement of this varies between protocol layers or occurs differently, i.e. handoff between base stations is fast compared to IP layer mobility. It is used here more towards IP layer mobility; changing between different access network.

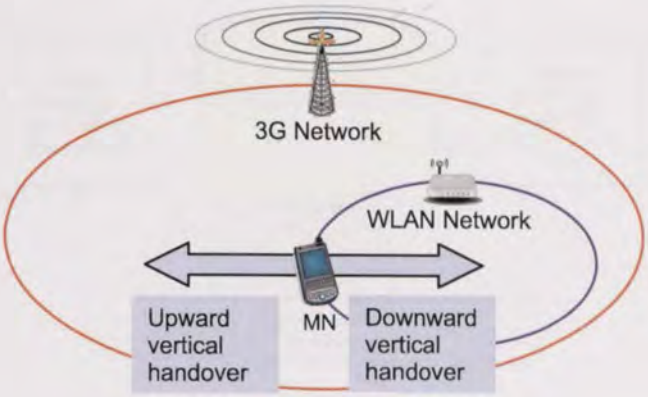
is less of a concern, handover latency can be flexible. Judgement of handover execution timing (handover latency) is difficult in an environment with continually changing conditions (Stemm & Katz, 1998). It is therefore important to minimise constant switching.

Handover direction also determines metrics that might be available. For example, an established connection to a base station or access point would provide a range of metrics, such as data rate, signal strength, jitter, and latency. This assumes that interface devices are powered simultaneously and connected to their respective networks. However, a wireless interface device could be powered-down at low battery levels using a sleep mode; a real possibility in mobile phone handsets. Also, situations may mean that metrics are incomplete; for instance, moving into range of a WLAN there is only physical and possibly link layer metrics available. Therefore QoS metrics are limited, until the terminal creates an active connection and some data is transferred. It is not known if a better performance exists end-to-end before a handover is made. Until then, only low-level metrics are available.

4.2.3 Interface selection problem

A device with multiple wireless interfaces creates opportunities for alternative connectivity and to be *always best connected.* Using the Cellular-WLAN context as an example, Cellular provides wide area coverage with lower data-rates and potentially greater cost than WLAN. This creates occasions when handovers can occur (figure 4.5). The handover direction could be towards WLAN (downward vertical handover) or towards the Cellular (upward vertical handover) (Stemm & Katz, 1998). Whenever a WLAN is available, it could be preferred over Cellular; depending on the application and movement. For a large file-transfer, the high-bandwidth, low-cost WLAN is preferred. However, for a voice call, reliability is more preferable than performing a handover (providing the current link is suitable). With a single QoS or policy metric this is a trivial decision; but judging multiple technical and QoS variables is problematic.

Human perception and decision-making has the intrinsic ability to evaluate and make judgements using small clues or in uncertain conditions. It is a capability that is difficult to replicate in engineering problems. Branches of psychology in cognition and decision theory research attempt to understand human processing of preferences and judgements. However, human judgements are not always rational in dealings with risk and benefit tradeoffs, instead engineering rational decision-making requires broad information, transparancy of the



(a) Roaming from cellular to WLAN

Figure 4.5: Roaming creates handover opportunities.

processes, and appreciation of consequences (Maes & Faber, 2003). The field of decision support systems uses decision theory in finance, cybernetics, and artificial intelligence (AI) to define formalisms in decision processes. The problem of wireless interface selection includes times where a choice or a comparison is required, such as the points of vertical handover or where coverage overlaps.

Intuitive decision-making processes are often difficult to synthesise in the digital domain, as computers require explicit description of variables, logic, rules, and context. One approach is to model the preference of a decision-making agent (a human), where the outcome is a measure of utility for actions or choices in accordance with preferences of an agent (Russell & Norvig, 2003).

4.3 Contextual Framework

Heterogeneous wireless architectures create more possibilities for QoS, but also more issues. The issues define 'factors' of the problem (figure 4.6). A solution space is defined from levels of factors that affect scope and complexity. The main factors relevant to the thesis are defined as:

- Applications, service types, and user preference.
- QoS metrics and measurement methods.
- Interface devices and dynamic events.
- Roaming and mobility effects.

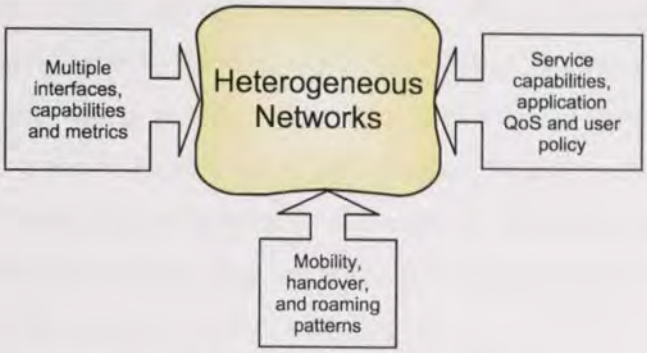


Figure 4.6: Factors affecting solution.

These factors are explored in the context of problem complexity to define a solution model that identifies relationships among factor levels. Problem scope and perspectives of QoS measurement are discussed further in this section to emphasise the depth of complexity. Finally, handover strategies are discussed as a solution for combining the issues in heterogeneous wireless networks.

4.3.1 Multiple factors

Complexity theory attempts to define representations of relationships between parts, variations, and quantity of information (Sporns, 2007). If a system is a representation of real object or objects, then "the degree of 'complexity' [is measured] by the quantity of information required to describe the vital system" (Ashby (1973), cited in Klir & Folger (1988, p.194)). Thus, complexity of a system is increased with the number of components; the interactions among components; and a level of emergence that stems from interacting components (Sporns, 2007).

Quantity of information and interactions of components are possible measures of complexity in systems. Other types of measure are those that exhibit randomness (Sporns, 2007), or are a type of *dynamical system* (Meiss, 2007) or *algorithmic information theory* (Hutter, 2007). Protocol interactions in networked environments exhibit complexity, through exchanging metadata, policies, and changing conditions. These interactions leads to emergent properties or perceptions elsewhere. When client devices request services from a network, additional traffic causes congestion at points in the network. Routers, servers, and switches use additional resources to process accumulated data traffic. Increased utilisation may incur other effects, such as increased latency and round-trip times, and possibly lost packets due to dropping algorithms.

Different wireless interfaces will have requirements for mobility and capabilities depending

on the network. Also, services and applications will have different requirements and capabilities for mobility, such as low handover latency in voice calls. Using mobility as an example, figure 4.7 defines complexity in roaming and mobility. The vertical handover problem requires additional mobility protocols handled above the link layer. Multiple individual applications and flows add to mobility factors complexity. Although, the OSI layered protocol stack is a separation of networking functions, there are protocols that require interaction among layers that complicate mobility management (see, Eddy (2004)).

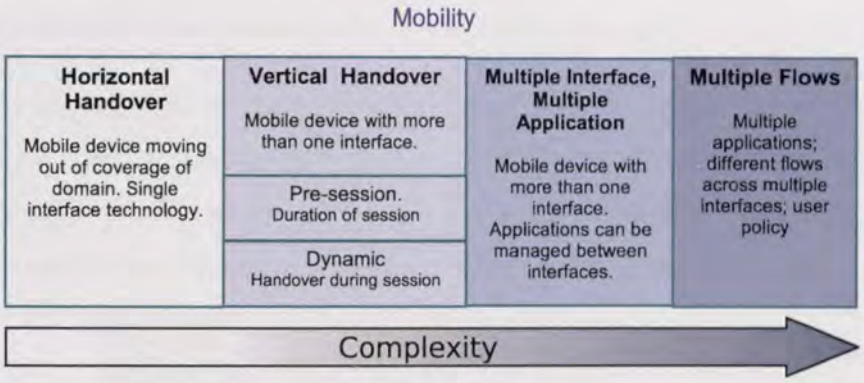


Figure 4.7: Mobility as a factor of increasing complexity.

The factors illustrated by figure 4.6 on the preceding page have varying degrees or levels. How these factors interact, or are made to interact, creates variations of a solution. A solution to vertical handover could be composed of how the problem factors are combined. The following discussion examines the details of these factors complexity.

- **QoS and performance factors**. This concerns the number of QoS metrics or other performance metrics. The choice and number of metrics is a problem. Also, how these metrics are measured, and where in the network stack they originate. Metrics can show different views of QoS depending on how they are calculated. Moreover, link or physical interface measures may not be the same or directly comparable between technologies or vendor types.

- **Application media flows.** At the simplest form, a single application with one flow, such as a file download or an audio stream. More complicated applications use more than one flow, i.e. a video conference involves audio, video, and synchronisation.

- **Mobility and roaming.** This introduces a complexity where the host device must maintain information of connections and procedures for changes in connections. Patterns

of usage and application type also affects mobility factors. For example, an interactive video that may require the users full attention. The device may not be very mobile, but used in a stationary location (the exception is the case of a passenger in a moving vehicle).

- **Multiple physical interfaces.** One interface device offers only a single choice of access type. Additional interfaces provide choice to access multiple networks, each with different capabilities and limitations. A more complex scenario could be to use multiple network connections simultaneously, rather than one at a time.

The factors of complexity can have varying levels. A solution matrix is shown in figure 4.8, with problem factors (columns) and the level of sophistication (rows); intersection of factors with levels define possible solution settings. Simple solutions would have less complex factor levels, whereas the most sophisticated solution would cover all expected levels.

	Factors			
	Application	QoS	Mobility	Interfaces
Solution complexity ↓	Single	Link metrics	Nomadic: initial connection	Single interface
	Two, similar traffic profiles	Link and networks conditions	Roaming: host moves, non-seamless	Dual interfaces, of different types
	Two, different traffic profiles	Link, network and application condition	Roaming: host moves, seamless	Multiple and different; multiple active connections

Figure 4.8: Matrix of solution settings.

In homogeneous networks, the problem space is limited by only one radio technology. With heterogeneous wireless environments, link metrics from two different link types could use different scales, or protocol characteristics that make direct comparison difficult. In WLANs, 802.11b uses modulation schemes with additional check-bits to reduce signal errors, thereby reducing the effective data-rate. Therefore, some QoS factors are not only dependent on signal conditions. This leads to a requirement of defining how measures of radio link quality are calculated.

4.3.2 QoS levels

QoS concepts and models, including those from ITU and IETF, offer perspective or logical differences in how QoS is specified and measured. Perspectives on QoS by user, services, and the network have led to conceptual frameworks. Solutions are discussed that present new modelling techniques and future developments of QoS descriptions.

Standards recommendations have been proposed for QoS models. ITU G.1000 (ITU-T, 2001a) defines requirements from customer or service provider, and performance or ratings for QoS assessment. Different levels from those of the customer and operator defines a QoS model based on achieved and required QoS (figure 4.9). Other ITU recommendations (G.1010: ITU-T, 2001b) define application, or service QoS constraints with an end-to-end view. It is not clear how network performance (bottom-up) and user services perspectives (top-down) are matched. This has led to proposals of QoS mapping solutions in the literature (Huard & Lazar, 1997; Calvagna et al., 2005).

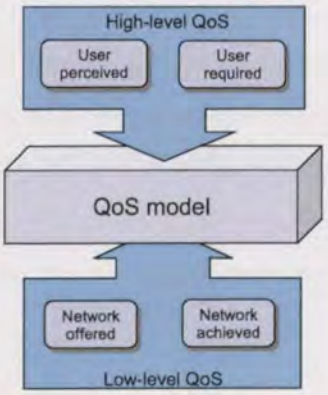


Figure 4.9: Perspectives of QoS levels

High-level QoS, or perceptive QoS is also presented in models from ITU (P.800: ITU-T, 1996). P.800 defines the mean opinion score (MOS) for subjective voice quality. However, there is less agreed approaches for other types of service. A study in the literature has focused on common application types for 3G network services (Papamilitiadis et al., 2004). The authors attempt to identify the effects of network characteristics and performance on different service types (instant-messaging and web-browsing). Although limited in scale (seventy-two respondents), the Papamilitiadis et al. (2004) study measures QoS aspects from different perspectives of protocols in the network stack.

A fine-grain, integrated view of network QoS attributes and quality of perception (QoP) is

given by Ghinea & Magoulas (2001). The aim is to obtain a mapping between a user perspective on performance and service provided by the underlying network. Perceptual QoS, or QoP, is a difficult concept to synthesise in relation to quantitative QoS measures (Burgstahler et al., 2003). Some of the problems described by Ghinea & Magoulas (2001) include:

- *Application-network QoS mapping.* There is less research attempting to integrate QoS measures across layers, in a usable framework. It is difficult to provide direct mappings between which attributes are seen by the application (frame-rate, audio synchronisation rate) and network level performance (packet/frame delay, packet loss-rates) (Ghinea & Magoulas, 2001).

- *Explicit measures of user satisfaction* with application or network performance. The levels of QoS at which a user become satisfied or dissatisfied.

Performance and parameter information structures, such as abstract syntax notation (ASN) used in management information base (MIB) schemes, have been primarily used in specific implementations of devices like 802.11. Such methods provide low-level measures of status and performance for localised protocol adaptation and optimisation. Network and application QoS provide high-level descriptions of performance, influenced by low-level metrics. Subtleties of such relationships are a challenge for a complete QoS representation model. Though, the benefits of formalising domain assumptions provide opportunities for cross-layer QoS information sharing between protocols.

4.3.3 Handover strategies

Strategies for vertical handovers often employ mechanisms for selection or decisions. The range of mechanisms have been reviewed in Kassar et al. (2008). Timer-based algorithms use signal metrics for handover control are detailed in Ylianttila et al. (2001, 2005). Others choose multiple input metrics based on signal or policies that are then processed by utility and cost functions (Wang et al., 1999; Zhu & McNair, 2006). More complex decision algorithms have been described that use policy and user preferences, including network and QoS conditions (Chan et al., 2001; Zhang et al., 2003; Song & Jamalipour, 2005). There is still no agreed framework for containing these strategies within the network protocol stack.

In 2003, a project was initiated at the IEEE, under task group 21 of 802 (LAN working group), to work on a standard for management of multi-access clients. The 802.21 version

2 draft (IEEE, 2006) defines functions for handover management between 802 and non-802 networks for media-access independence above interface technologies. The draft aims to:

- use link layer information to optimise the handover process;
- provide feedback to upper-layers;
- and support service continuity.

The logical architecture of 802.21 in a client stack is represented by an intermediary layer. This layer is defined as the Media-Independent Handover (MIH) Function (figure 4.10). According to the draft (IEEE, 2006), it provides services to the link layer and upper layers. **Event services** provides details on link status and quality. **Command services** are used to initiate changes, such as handovers. The **information service** provide bi-directional details of other access network characteristics, network-based settings, or user policy settings.

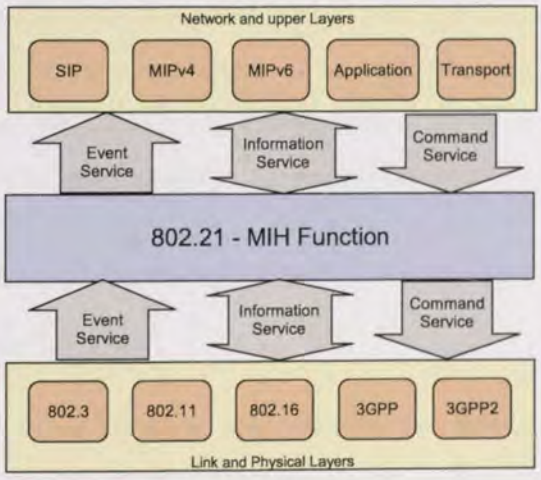


Figure 4.10: MIH function in 802.21 (*Source*: IEEE (2006, p.3))

Protocol layers above and below the MIH Function exchange messages through service access points (SAPs) (IEEE, 2006, p.28). The draft defines several types of SAP for interacting with the lower-layers and mobility management entities. Lower-layers are defined by media-dependent SAPs. For each interface device there are SAPs for data-link and physical layers. Media-independent SAPs define primitives for upper-layer entities.

As part of MIH, a mobility management entity is required to receive lower-layer data and make handover decisions. However, whilst the draft recognises the need, it does not specify the details of how selection might be performed. Handover selection as a concept is defined in

802.21 as a source of decision information; upper-layers are responsible for network selection (IEEE, 2006). Moreover, the draft suggest this as one of the 802.21 design principles:

> "MIH is a helper and facilitator function which helps in handover decision making. Upper layers make handover decisions and link selection based on inputs and context from MIH. Facilitating the recognition that a handover should take place is one of the key goals of MIH Function. Discovery of information on how to make effective handover decisions is also a key component." (IEEE, 2006, p.14).

With MIH as a facilitator of handover selection in the current draft, there is scope for definition of this selection entity. Other literature (Dutta et al., 2005; Cacace & Vollero, 2006) have developed early prototypes for testing 802.21 concepts. The work of Dutta et al. (2005) defines a testbed environment using SIP-based mobility management. The process of mobility is explained, but there is no specific detail of how the network is selected. Mobility management is also presented in Cacace & Vollero (2006). The authors define a *mobility manager* using mobile IPv6 in a testbed environment. As part of the MM, a handoff decision module is responsible for making network selection based on preferences and performance, that: "executes a simple priortized two-threshold algorithm for vertical handoffs" (Cacace & Vollero, 2006, p.4).

Concluding Remarks

Devices with multiple wireless interfaces introduce choice of network connectivity for overlapping coverage. The overlay pattern defines the concept of changing between wireless interface technologies (vertical handover) to take advantage of coverage. This premise can be extended to QoS, whereby different interfaces have different QoS capabilities and limitations. The fixed-mobile convergence concept is the industry led approach for heterogeneous wireless environments. Architectures have so far focused on 3G, or UMTS-WLAN integration. This narrows the problem scope, but there are still issues to be resolved.

Even with two different access methods, roaming and QoS issues add additional complexity. Movement creates points of handovers that can degrade QoS, but also has the potential for improving QoS. Using one or two metrics handover selection can be trivial. But for services that have different QoS requirements, one or two metrics are not sufficient to assess QoS. Frameworks such as 802.21 and handover strategies have been proposed that challenge

this problem. The algorithms employed in handover strategies have used decision theory and AI. These concepts are further explored in the following chapter as solutions for the wireless interface selection problem.

Chapter 5

Review of Decision-Making Techniques

In heterogeneous wireless environments more possibilities exist for applications to improve QoS and user experience through different access methods. With more more options for connectivity, an automated network selection system would require observing inputs from the environment and provide recommendations as outputs. Reasoning or decision-support systems are common in the literature of artificial intelligence (AI). Such systems provide mathematical foundations and algorithms for handling automated reasoning, which have been applied in areas of financial systems, diagnosis and operational-support systems.

The chapter introduces requirements of a system for link assessment and selection. A comparison of existing techniques is used to understand the issues in a decision-making process. Finally, the approach for link assessment is discussed.

5.1 System Definitions

Vertical handover in heterogeneous environments is complex. Wireless interface selection could include multiple sources of information or inputs, such as user expectations, conditions, or preferences. The system design should consider the following criteria:

1. Provide a common interface between other protocols or user-level applications.
2. Monitor multiple link statistics and events.
3. Implement actions or effects, such as handovers.

Figure 5.1 is a controller system for an intertface selection model. Inputs are processed by the system to find the optimal operating choice. These inputs are processed to generate outputs,

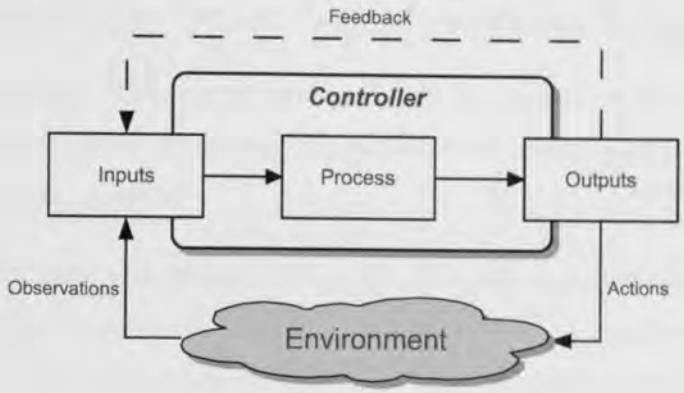


Figure 5.1: A controller component model for interface selection.

such as actions that affect the operating environment, or for feedback of some action or request. Solution options are generated for components of the system based on the following points:

1. Concepts for robots and intelligent behaviours can be applied as a framework for handover decision-making in heterogeneous wireless networks.

2. AI techniques can provide a link assessment process.

5.1.1 Context and inputs

The system context potentially involves many inputs, which may be unpredictable or unknown for every possible state of the system. Full information about link performance and QoS may be limited or unknown, without performing other actions, such as sending data or querying the network. Other factors that affect QoS are network characteristics or patterns; as in office hours where the number of users are at their highest. Consequently, the wireless network context introduces *uncertainty* into the decision-making. Types of information uncertainty have been described as "incomplete, imprecise, fragmentary, unreliable, vague, or contradictory" (Klir, 2006, p.6). System inputs may have uncertain characteristics that depend on whether they are measured or static, and how they are calculated. Types of input to the decision controller could include the following:

- *User preferences.* The QoS noticed during sessions, or *a priori* requirements fall under the category of perceived QoS. Multiple requirements or preference of services, constitute a policy. Policies have been commonly defined as generic SLAs that might be provided between an ISP and customer for network resources (Gurijala & Molina, 2004).

- *Application or services.* Common measures include: delay, throughput, jitter, and loss.
- *Link performance.* Performance measures below the application level are not a direct mapping to those above. Variation exists due to error correction, packet header overhead, and measurement methods.

Decisions are often made with incomplete or vague information, and complicated by expectations and outcomes. Making a good decision may not necessitate a good outcome, and not all bad decisions have a bad outcome; the decision process is different from the expectations of outcome (Ross, 2004, p.309). Moreover, Ross argues that decision-making in uncertainty should be done consistently and rationally, so that over the long-term, beneficial outcomes occur more often. Sometimes the correct decision may not be made or be the most optimal, but attempts to be 'good enough' most of the time (Ross, 2004, p.309).

5.1.2 Link selection process

Link selection could use multiple inputs that include: mobility, application performance, link capabilities, and link status. A selection procedure would need to interact with protocol instances to obtain performance data and perform control actions. It would also need to maintain a consistent structure of applications requirements and common measures of status metrics for different link types. The selection process is defined by the following criteria:

1. Combine measures of performance from multiple sources.
2. Use predefined requirements and capabilities as criteria to influence decision-making.
3. Adapt to use additional metrics; changes in link, application, and device conditions.

The system processing component is discussed as a *decision-making problem*. Though, in the literature this is distinct from *optimisation problems*. An optimisation problem is a refinement of parameters or inputs for improved output. In decision-making problems there is a choice between alternative options compared to criteria. However, some problems may require techniques of decision-making and optimisation; whereby a decision is made, then optimisation performed. For the purposes of simplicity, the methods discussed in this chapter refer to decision-making problems. The motivating question is therefore: what AI or decision-making methods can be used to develop a solution for access link assessment?

5.2 AI Methods

Link selection requires input processing to make control decisions. Related techniques for this type of problem are common in AI, where robotics and cybernetics research has led to techniques for performing tasks in challenging and dynamic environments. Although wireless networks do not have the same challenges as a Mars rover, there are some concepts of design that can be used: a robotic brain uses sensors to map its environment for a control process to decide on behaviour suitable for those conditions.

Existing AI techniques and frameworks that are relevant to the thesis problem are shown in figure 5.2. Probability is a commonly applied method for dealing with uncertainty based on statistical methods. Alternatives encode knowledge of experts or required behaviours. The third alternative is from decision theory and decision support for assessing options with multiple criteria. Subsequent discussions focus on techniques relevant to decision-making or refinement of options based on multiple, possibly conflicting, or uncertain inputs.

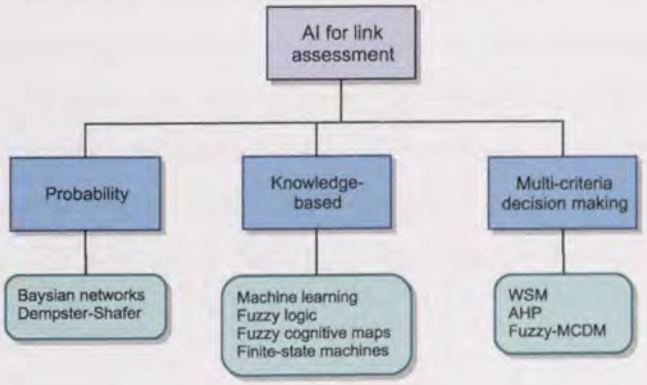


Figure 5.2: AI techniques relevant to link selection process

5.3 Probability-Based Reasoning

Probability originates from attempts to quantify unknown occurrence of stochastic events. This is defined as the *objectivist* viewpoint. Alternatively, a *subjective* probability is based on belief of an event occurring, though the importance of this distinction is debated (Russell & Norvig, 2003, p.430). Other distinctions are between probability and possibility; the latter often describing evidence theory (Klir & Bo, 1995). Alternative treatments of probability attempt to provide measures of the confidence of evidence itself. *Dempster-Shafer theory of evidence* provides a method of combining probabilities and confidences. Probability has been used in

methods for handling uncertainty in AI applications (Russell & Norvig, 2003). *Bayesian inference* is used in Bayesian networks[1] for conditional belief measures in a directed-graph model.

The wide use of probability theories and applications in the literature warrants consideration as an assessment and decision-making solution. Further discussion is given to the techniques of probability, such as Bayesian inference and Dempster-Shafer. These are explained and justified in the context of criteria in section 5.1.

5.3.1 Bayesian inference

Bayesian inference is based on the work of Thomas Bayes (1702-1761). Using *Bayes's rule*, new evidence can be used to change beliefs about some occurrence, or probability of some event. Since Bayes's rule is used in approaches for reasoning using probability, the following explains the core concepts and associated variations.

Probability measures can be unconditional or conditional. Unconditional probability is derived from statistical measures of prior occurrences, or those of belief of some events occurring, $P(A)$ and $P(B)$. Conditional probabilities are those based on other information in the domain, defined as $P(A|B)$: "the probability of A, given B" (Tozour, 2002, p.346). Using conditional probability $P(A|B)$, and the unconditional independent evidence of $P(A)$ and $P(B)$, the initial phrase can be reversed to become: $P(B|A)$, the probability of B given A (Tozour, 2002, p.346). Bayes's rule is given as equation (5.1) (further proof and examples found in Russell & Norvig (2003, p.480)).

$$P(B \mid A) = \frac{P(A \mid B)\,P(B)}{P(A)} \tag{5.1}$$

In AI systems there is often cause-and-effect relationships between concepts, or variables. The *Bayesian network*, or belief network allows for representation of event probabilities with causal relationships. A Bayesian network is usually represented visually as a directed acyclic graph; as in figure 5.3. Edges represent propositions that may be discrete or continuous random variables, and arcs represent direction of influence. These representations would be designed by a expert to adequately describe the problem context. Bayes's rule uses independent probabilities to derive new unconditional probabilities, which is a topology of causal effects. Figure 5.3 represents an example Bayesian network. Two independent probability variables are de-

[1] Also known as belief network, probabilistic network, causal map, and knowledge map (Russell & Norvig, 2003).

fined for events, like a burglary and an earthquake. In this example, the probability of these events is used to determine probability of a house alarm being triggered. The alarm node is conditional on the probabilities of an earthquake or burglary occurring. Using Bayesian inference, this network generates probabilities whether the alarm was triggered by a burglary, an earthquake, or both.

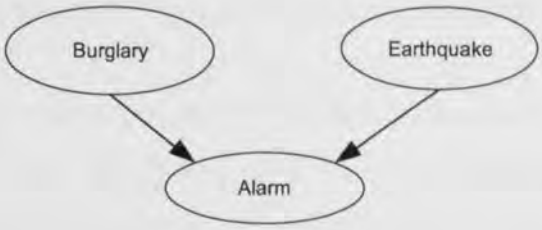


Figure 5.3: Example of a Bayesian network (*Source*: Tozour (2002, p.347))

The forms of Bayesian inference are useful in some contexts for reasoning with unknown variables. In particular, problems that involve randomness in variables. However, Bayesian inference is restrictive; it requires independent probability variables or proof of conditional independence (Laramee, 2002, p.358). Laramee continues to argue that appropriate estimation of conditional variables can be difficult if there is insufficient previous evidence or statistical significance. Where unknown variables or conditional variables are not independent, assumptions may be suitable for some problems, or further enhanced through statistical learning methods (see, Russell & Norvig, 2003, chp.20). However, it is argued that Bayesian updating of subjective values is not always applied correctly when representing changes in utility (Maes & Faber, 2003, p.99). These difficulties would likely rule-out this type of inference for most problems, except those that are simple, or sufficient domain knowledge is available and can be represented with statistical significance.

5.3.2 Theory of evidence

A computational approach to reasoning with evidence originated from the work of Dempster (1968). Further development by Shafer (1976) led to what is now known as Dempster-Shafer theory (DST). DST is a computational model that generalises the Bayesian theory of incorporating pieces of evidence. Like the Bayesian model, evidence is represented as probabilities in the interval [0, 1]. Probabilities about credibility of evidence provides a measure of confidence, reducing ignorance of an event. Evidence of ignorance is separate from the uncertainty

in occurrence of an event.

DST defines a belief function, $Bel(X)$, as a measure of the belief that the evidence supports a proposition, X. In the example of a coin toss (Russell & Norvig, 2003, p.525), classical probability would denote the chance of heads or tails as 0.5. Though, if no knowledge about the fairness of the coin is known, $Bel(heads) = 0$, as $Bel(\neg heads) = 0$. However, if some value of fairness is known to be 90% that $P(heads) = 0.5$, then DST will give $Bel(heads) = 0.9 \times 0.5 = 0.45$.

Individual pieces of evidence are a measure of belief corresponding to credibility or plausibility (Laramee, 2002). Those sources can confirm or negate a hypothesis or action proposition. Also, multiple probabilities can be used to update the probability of support for a concept. The combination of pieces of evidence is performed using Dempster's rule (Beynon et al., 2000).

5.4 Knowledge-Based Systems

Methods for reasoning have so far introduced probability-based methods, that are derived from statistical or estimated measures. Bayesian networks are based on contextual knowledge of relations, and therefore could be considered a form of knowledge-based system, or expert system. Knowledge-based systems contain information defined by experts in a specified context domain. Developing systems that have a constrained or limited purpose, can be beneficial to successful operating in complex environments. Methods that use limited scope computational models are evident in AI domains, such as robotics, pattern matching and classification. In these domains, reproducible behaviours are useful and expected.

Historically, such systems were pioneered in medical diagnosis, as attempts to replicate decisions similar to those of the expert. An expert's knowledge is captured (knowledge engineering) and encoded, usually as rules or a form of logic. Though rule-based systems and formalised logic are some methods of encoding domain knowledge, others include learning systems, fuzzy rule-based, cognitive mapping, and automata. The subsequent sections explain the relevant methods in these categories and as a potential solution of the system context in section 5.1.

5.4.1 Machine learning

Two commonly used techniques in machine learning include artificial neural networks (ANN) and evolutionary algorithms. Early use in neural networks were developed from attempts to

model the behaviour of the human brain. Modelling neurons was proposed by Warren McCulloch and Walter Pitts in 1943. Their findings showed that interconnections of artificial neurons could learn (Negnevitsky, 2002). It is this learning property that has given ANNs popularity in AI applications.

An ANN is essentially a knowledge-based system. Behaviour of a modelled process is encoded using neurons. The neuron processes input signals to produce a single output signal. Connections, or links between neurons are associated with weights. It is these weights that determine the strength of neuron output should 'fire' the input of the connected neuron (figure 5.4 shows these links in a multi-level ANN). Through learning procedures, these weights are adjusted and tuned to the required output of the neuron. Repeated learning sequences adjust the weights from training data, which brings the response of the ANN closer to the required behaviour. Further test sequences and statistical analysis are then used to determine the system is within acceptable error margins.

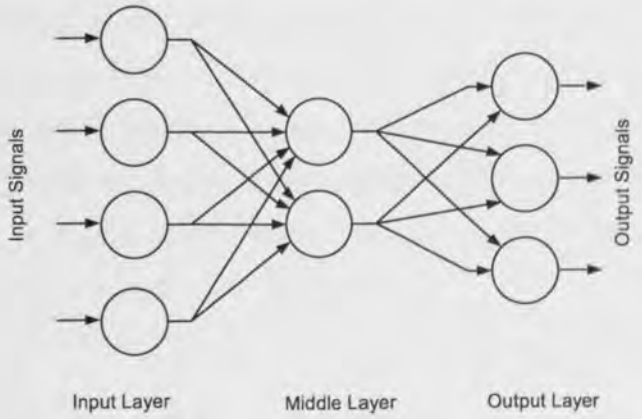


Figure 5.4: Example of multi-level ANN (*Source*: Negnevitsky (2002, p.165))

The neurons in ANNs are processors that combine one or more inputs to derive a single output. One or many neurons can be defined, with connections of neurons determining the architecture of the ANN. Also, neurons may be arranged in layers as in figure 5.4. This includes a middle, or hidden layer between input and output signals. The multi-layer model could use the input layer as a classifier of single inputs; the middle layer would then process, extracting features of multiple inputs (Negnevitsky, 2002). Finally, the output layer would classify the hidden layer for subsequent response signals.

Neural networks have been applied to robot navigation control. However, conventional ANN techniques are not applicable for all types of problems. To develop an ANN, suitable

training data is required. Suitable previous examples of behaviour or system conditions may not be available in all domains. Secondly, once a network is trained, significant changes are needed if the problem requirements change (Negnevitsky, 2002).

More recent research has combined neural network techniques with other AI or decision-making methods. The learning capabilities of ANN has been combined with fuzzy techniques to create hybrid intelligent systems (Cheng & Chang, 1999; Zheng & Kainz, 1999). Such systems are combined to exploit the benefits and reduce the limitations in each technique. The learning techniques of ANN, and the expressiveness of fuzzy terms and transparency of reasoning in fuzzy systems, are combined as Neuro-fuzzy systems. The following section provides more details of fuzzy systems.

5.4.2 Fuzzy systems

The concept of a fuzzy set was introduced by Lofti Zadeh in his seminal paper: "Fuzzy sets, Information and Control" (Zadeh, 1965). Zadeh defined a mathematical formalism for representing uncertainty in set theory. Classical set theory defines strict set membership: either part of the set or not; thus referred to as *crisp* sets. In the fuzzy set this boundary often has a smooth transition between membership and non-membership. Fuzzy sets have been proved to generalise the crisp set (Kosko, 1992), and so a crisp set is a special type of fuzzy set with a step function between the boundary of set inclusion.

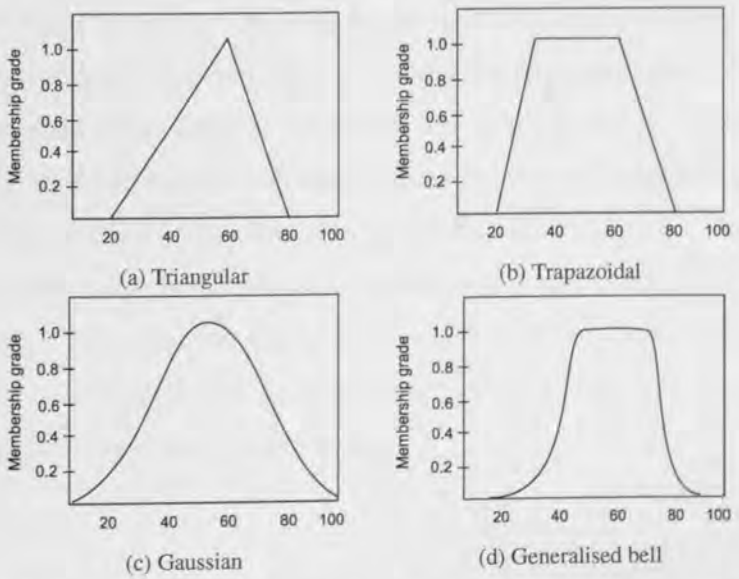
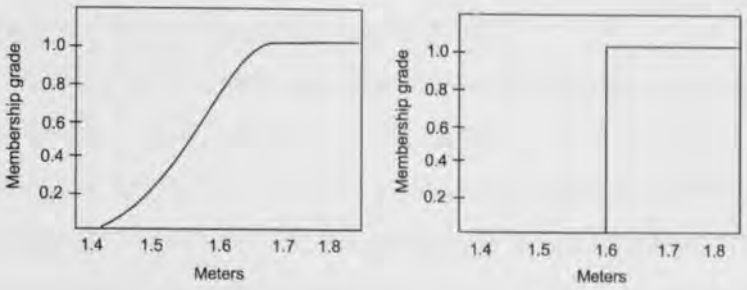


Figure 5.5: Examples of membership functions (*Source*: Jang et al. (1997, p.26))



(a) Fuzzy set (b) Crisp set

Figure 5.6: Tallness as a fuzzy set (a), and crisp set (b)

Fuzzy sets can have partial membership of objects, usually referred to as degree or grade of membership. Grade of membership is usually a value in the range 0 to 1, which is the degree that value x belongs to a fuzzy set (Klir & Folger, 1988, p.10). A representation of a fuzzy set is denoted by a membership function (MFs) (figure 5.5). Such MFs have a universe of discourse (the unit value) as the x-axis, and the degree of membership in the y-axis. The MF maps universe of discourse values to a grade of membership in the range [0, 1].

Fuzzy sets allow capture of uncertainty in concepts and terms. Often, probability is also used to represent uncertainty. Although both use the unit interval [0,1], fuzziness and probability are conceptually different. Probability measures are estimates of event occurrence in a stochastic domain. After the event has occurred, the probability is 1; it *has* occurred. Alternatively, fuzziness measures ambiguity in "the degree to which an event occurs, not whether it occurs." (Kosko, 1992, p.265). Fuzziness is more suited to human language reasoning. For instance, a fuzzy estimate of tallness will not be dependent on occurrence of some event. If an individual is selected whose height is 1.6 metres, they would be *tall* to a degree in the range [0, 1]. The MF shape determines a steady transition of grades of membership, from certain tallness to not so tall (figure 5.6a). In a crisp set, the set *tall* would have a crisp boundary, or step function (figure 5.6b). So, if someone is 1.59m they are not considered tall, but they are at 1.6m; this seems counter-intuitive. The term tall or tallness is considered vague, and so fuzzy sets allow a smooth transition from set membership. Fuzzy systems have been developed for reasoning using fuzzy sets, such as fuzzy logic.

Fuzzy logic

An application of fuzzy set theory is control systems. The principles of fuzzy set theory are applied to logic systems for the purpose of *approximate reasoning* (Klir & Folger, 1988). A

common application of this is fuzzy logic controllers (FLC).

A fuzzy system such as FLCs uses input variables as fuzzy sets and expert knowledge in the form of fuzzy logic rules. The logic rules provide control behaviour and are a class of expert system (Klir & Folger, 1988). Figure 5.7 is a schematic of a FLC. **Conditions**, or measured variables in a process to be controlled are defined as inputs for **fuzzification**. The fuzzification process maps input variables to membership functions to generate fuzzy values. Fuzzified values from the inputs are then processed by the **fuzzy inference engine**. This process uses fuzzy rules to derive fuzzy output signals. The knowledge is captured in a collection of linguistic rules (rule-base) in the form of IF-THEN statements for input and output variables. Following rule inference, the fuzzy values are **defuzzified** (converted to a continuous variable) to obtain output signals appropriate for the controlled process.

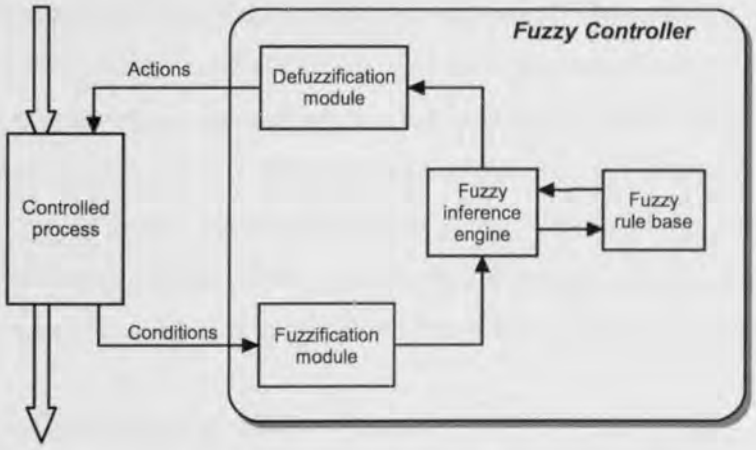


Figure 5.7: Schematic of general FLC (*Source*: Klir & Bo (1995, p.331))

Fuzzy sets and fuzzy rules provide an approximation of input and smooth transition between output signals. FLCs have found applications in control systems where proportional integral derivative (PID) controllers are commonly used. These are processes that measure, then provide a feedback loop to change control behaviour. A FLC requires clearly defined inputs and output signals that can be measured. Also, sufficient system knowledge can be represented in linguistic terms. Linguistic rules provide an intuitive method for describing the reasoning with inputs (Klir & Bo, 1995, p.330). The FLC method requires debugging and tuning of the rules for good, but not necessarily optimal output signals.

Systems using fuzzy logic have mainly been applied to control problems (Ibrahim, 2004), with variations used in other domains, such as medical diagnosis (Yuan et al., 2002). There are examples of fuzzy systems in wireless handoffs between base stations (Edwards & Sankar,

1998), and adapting protocol implementations (Fernandez et al., 2004). A fuzzy logic system for vertical handover in wireless networks has been published in Chan et al. (2001). The fuzzy system in Chan et al. (2001), defines rules for handling inputs on performance for initiating a handover (a control behaviour). It is combined with another fuzzy approach, multi-objective decision making, to perform a combination of performance and user preferences.

Simple models of FLCs are suited to problems with specific solutions. These are the problems covered by control problems with well defined inputs and output variables. However, in problems where the number of inputs or outputs could vary, FLC implementations can be inflexible. More adaptive fuzzy systems, such as Neuro-fuzzy systems, use learning in ANNs to train membership functions, whilst visible rules allow introspection in testing and verification.

The inference mechanism of FLCs uses linguistic rules to infer outputs from various inputs. Using FLCs in their basic form for the handover selection problem, showed the rule-based approach to be too restrictive and suited to only a small number of criteria (Wilson et al., 2005). Moreover, rules for decision-making require tuning to give suitable response behaviour, but this becomes less flexible with additional inputs or rules. Fuzzy systems that use linguistic rules for inference can become cumbersome, except where there is clear input parameters and defined scope (Sousa & Kaymak, 2002, p.56). Although some examples of fuzzy systems based on rules have been suggested, an extensible inference mechanism is required.

5.4.3 Sequential decision systems

Dynamic programming was coined by Richard Bellman (Bellman, 1957) as an approach to solve optimisation problems using a sequence of solved sub-problems. Derivative methods continue to be used in the field of AI, such as Markov decision processes (MDPs). In the domain of robotics, planning and control techniques are used in navigation systems: sensing inputs from the environment and changing directions.

In search and planning environments, agents use high-level factors to direct a strategy. High-level factors affecting planning and control include agents goals or policy, environment inputs (sensors), and actions (Russell & Norvig, 2003). This has led to behaviour-based AI and reactive planning algorithms, particularly in robotics research. In more simple planning problems, the environment of an agent may be fully observable. In this case, planning algorithms can look-ahead to environment conditions at a later point in space and time. The planning problem becomes more complex in the partially-observable domain (Bertoli & Pistore, 2004).

Certain conditions of the environment may be unknown until some other actions or variables change to reveal a new state. The state and planning paradigm provides the possibility to change behaviour based-on current state and history of states. Relevant models are discussed further that include cognitive models, and state automata.

Fuzzy cognitive maps

In previous sections, Bayesian networks defined causal relationships between concepts based on probabilistic variables. These models often require some form of expert knowledge about dependencies, and statistical or belief values. Bayesian networks models are derived from *influence diagrams*, as a model of causal relationships (Russell & Norvig, 2003). A very similar model to influence diagrams are *cognitive maps*. Cognitive mapping (Axelrod, 1976) is based on cognitive models as a representation of reasoning. They have been used to represent interactions for strategic planning in social and political systems evaluation (Ross, 2004). A cognitive map defines nodes, or edges to represent concept variables, with arcs as causal relations; as in directed graphs. However, relations or effects between concepts are limited; defined as positive or negative effects (or neutral).

Developments in cognitive mapping have introduced fuzzy concepts, known as *fuzzy cognitive maps* (see, Kosko, 1986). A fuzzy cognitive map (FCM) represents relations between concepts using fuzzy measures (Kosko (1986), cited in Ross (2004)). The concepts (nodes) are defined by a fuzzy set, with the membership of the set determining active concepts is in the system. Membership levels of concepts may change during the life of the model. A FCM with inputs and lifecycles is a fuzzy dynamical feedback system (Aguilar, 2005). Updating concept nodes is based on a summation of inputs until a stable state is obtained.

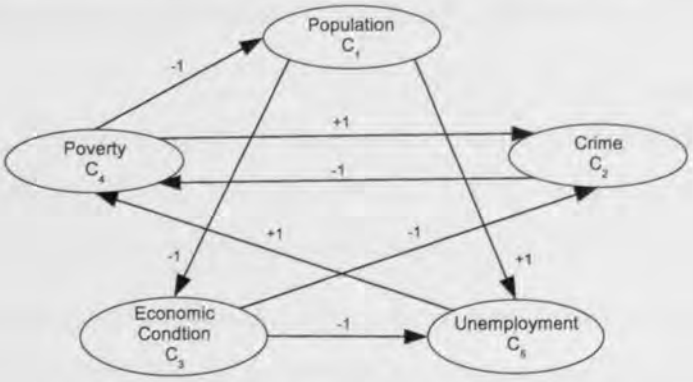


Figure 5.8: Socio-economic example FCM (*Source*: Kandasamy & Smarandache (2003))

The features of FCMs are similar to those of Bayesian networks, whereby relations between concepts can be used to infer effects on consequent nodes in the graph (figure 5.8). In effect, an FCM is an inference process based on convergence of input variables and knowledge of the decision maker. However, FCMs are models of a decision-maker's reasoning, and therefore subjective. By accepting subjectivity, FCMs models are encoded with biases and other prejudices (Kandasamy & Smarandache, 2003). Other complexities involve obtaining agreement between multiple designers; a decision-making problem itself.

Finite state machines

Finite state machines (FSMs) are derived from *automata theory*. They contain states and transitions, where changes between states defined by transition functions. Conventional formalisms include Mealy and Moore machines. A Mealy machine generates an output signal based on the input and the current state. The Moore machine output is based only on the current state. These differences are shown in figure 5.9.

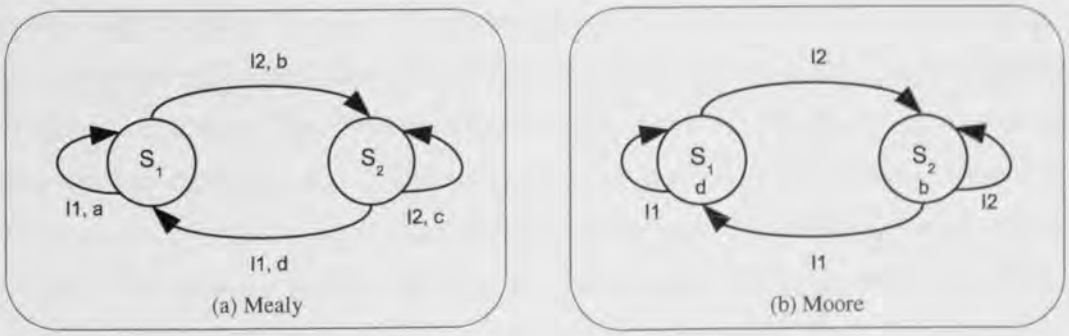


Figure 5.9: Moore and Mealy FSMs. S1 and S2 are the states, with I1 and I2 the inputs. Outputs are lower case letters. (*Source*: (Buchner & Funke, 1993))

FSM formalisms are well-established in computational methods. Variations in FSM concepts include:

- *PetriNets* use similar modelling of states and transitions, but support concurrent activation of states using tokens. Transitions generate tokens, which then accumulate in the states. A state activates transitions once tokens in the state meet a threshold value.
- *Fuzzy-FSMs*, or fuzzy-state automata (FSA), define states as fuzzy sets (Klir & Bo, 1995, p.369). Examples of FSAs can be found in the fuzzy journals[2], and medical diagnosis research (Steimann & Adlassnig, 1994).

[2]Fuzzy Sets and Systems, Elsevier scientific

- *Communicating extended FSMs* (CE-FSMs, (Byun et al., 2001)) have states and transitions, but with some additional features. CE-FSMs are intended to interact with other CE-FSMs, providing protocol-based activities. They also have conditional transitions, and can implement timing behaviours.

Applications of FSMs have been used for pseudo-intelligence in computer game AI (van Waveren, 2001), in software modelling (Wagner et al., 2004), and also for protocol design (Byun et al., 2001). Though variations such as CE-FSMs are suited to software design and modelling. A state-based approach is commonly used in software design patterns (Gamma et al., 1995). These tend to be event-based models and have open-source software tools for implementing FSMs (Rapp, 2007).

5.5 Multi-Criteria Decision Making

Decision-making is a large field of research including a wide range of theories in psychology, operational management, and control systems. Approaches to decision-making can be *descriptive* or *normative* (Sousa & Kaymak, 2002). In descriptive decision-making, human cognitive processes are studied. These tend to be psychological questions with the aim of discovering how and why humans reason a certain way, or arrive at solutions to problems. Normative decision-making is defined as a rational choice amongst alternatives based on available information. The normative view is often adopted by mathematical and engineered approaches, requiring formalisms and discovering optimal solutions. Such approaches tend to be used in the context of management operations, control theory and optimisation. There is an extensive body of work in normative decision-making methods. Common problems have emerged, such as comparing multiple options, or optimising actions.

The remaining discussion identifies some common techniques in formalising problem descriptions and compares some common algorithms. Techniques based on multi-criteria decision making (MCDM) support choice comparison using multiple decision criteria. The following sections describe MCDM terminology in relation to the context of network selection, and variations for introducing preference in decisions. Further variations are discussed that combine fuzzy concepts.

5.5.1 Terminology

A decision process or situation is required when some agent has choices to make, or a course of action to follow. This implies that certain information about the state of the world is available to be collected, and this collection of information is part of the process (Sousa & Kaymak, 2002). The decision act or commitment is another part of the process. The following are some characteristics of decision methods relating to the problem context:

- **Multi-attribute**. One or many criteria can be used, that requires aggregation.
- **Multistage**. Some decision processes may require different levels of criteria aggregation, or dependencies between decision actions based on temporal events (Sousa & Kaymak, 2002, p.18).
- **Conflict** or trade-off in criteria. In a MCDM problem there are often conflicts or trade-offs from multiple sets of criteria.
- **Incommensurable criteria**. Multiple criteria may use different units and therefore difficult to compare directly.

A generic decision situation is defined by the a tuple, $DS = (A, W, C)$, where W is a set of states of a world or domain; C is a set of consequences; and A is the set of acts that an agent can perform on the world (Halpern, 2003). A MCDM solution is commonly defined by a set of criteria, some alternatives, and preference weights. Weights associated with criteria can be used to change the importance of the criteria when it comes to applying the decision-making method. These can be arranged in a *decision matrix* (figure 5.10).

	C_1	C_2	$\dots$	C_n
Alts.	$(w_1$	w_2	$\dots$	$w_n)$
A_1	a_{11}	a_{12}	$\dots$	a_{1n}
A_2	a_{21}	a_{22}	$\dots$	a_{2n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
A_m	a_{m1}	a_{m2}	$\dots$	a_{mn}

Figure 5.10: A decision matrix

5.5.2 Conventional methods

Methods for normative decision-making support, including their variation, are extensive in the literature (Triantaphyllou, 2000). The following discussion explains a selection of these techniques (commonly cited and most simple) in terms of their characteristics and effects for a decision-making procedure.

WSM

Weighted sum model (WSM[3]) is a simple criteria aggregation approach. Values of criteria for each alternative are multiplied by corresponding value in a weight vector. The sum of criteria values scores is used to rank the order of preference. For n criteria and m alternatives the score for alternative A_i is given by (Triantaphyllou, 2000, p.6):

$$A_i = \sum_{j=1}^{n} a_{ij} w_j, \qquad for\, i = 1, 2, \ldots, m. \tag{5.2}$$

WSM is simple method of combining criteria. When the criteria are the same units of measurement, they can be summed directly using WSM. Though, if criteria are incommensurable, some transformation needs to be applied, such as normalisation or scaling. Transformation of criteria values is also required to represent negative and positive criteria preference (Burgess, 2003). Basic WSM (without scaling) algorithm only allows one type, i.e. higher-better or lower-better, but not both. This is based on the underlying assumption of WSM; that of "additive utility" (Triantaphyllou, 2000, p.6). Criteria values are summed to give the position rank of alternatives, so those criteria with higher individual scores will give a higher rank; for lower values, a lower rank is the result. This has the effect of trading poor criteria for good performance of other criteria; though strength depends on criteria weightings.

AHP

The analytic hierarchy process (AHP) uses pair-wise comparisons of criteria to obtain a preference score. Criteria are entered into a pair-wise decision matrix. Table 5.1 shows the preference values used for pair-wise comparison. The result is a decision matrix of criteria preferences. AHP uses the same maximum aggregation function of WSM (5.2).

[3]Sometimes referred to as "simple additive weighting" (SAW).

Intensity	Definition
1	Equal importance
3	Moderate importance
5	Strong importance
7	Demonstrated importance
9	Extreme importance
2, 4, 6, 8	Intermediate values

Table 5.1: AHP importance scales (Source: (Song & Jamalipour, 2005, p.3)

AHP has found many applications in the literature of decision-making research (Triantaphyllou, 2000). It has also been applied in the context of this thesis. Specific to the problem of heterogeneous handover, a study by Song & Jamalipour (2005) has used AHP to map subjective user preferences on services to a set of weights for criteria.

5.5.3 Fuzzy MCDM

Common approaches to MCDM attempt to structure the normative, one-time decision-making involving multiple criteria and alternatives. Chapter 4 introduced some of the complexities in wireless heterogeneous environments of user perception, uncertainty, and risk. A class of solutions have been developed that uses fuzzy concepts in MCDM. The remainder of this section presents a decision-making procedure developed by Richard Bellman and Lofti Zadeh (Bellman & Zadeh, 1970), that combines fuzzy techniques with MCDM conventions.

Bellman-Zadeh algorithm
1. Creating the decision matrix requires a set of alternatives: $A = \{A_1, A_2, \ldots, A_i, \ldots, A_n\}$; and a set of criteria (goals and constraints): $Z = \{C_1, \ C_2, \ \ldots, \ C_m\}$. Using these to form a decision matrix, a_{nm} is the values for alternatives, A of criteria C.
2. Create fuzzy membership functions, F_j that represent the criteria. Following the Bellman-Zadeh approach, goals and constraints are treated the same by the decision function.
3. Weight factors are vector elements in the interval, $w_j \in [0,1]$, $j = 1, \ldots, n$ and normalised as: $\sum_{j=1}^{n} w_j = 1$
4. With a decision matrix of fuzzy values and a corresponding weight vector for criteria, a decision function is defined as: $D^w(\mu_{i1} \ldots, \mu_{in}) = 1, \ldots, m$
5. The result of the decision function is a fuzzy decision vector of F: $F = D^w(F_1 \ldots, F_n)$

Figure 5.11: Bellman-Zadeh algorithm (*adapted from*: (Sousa & Kaymak, 2002, p.33))

The MCDM technique from Bellman & Zadeh (1970) introduced fuzzy sets as representations of criteria. The algorithm is summarised in figure 5.11 as five steps (Sousa & Kaymak, 2002, p.33). The Bellman-Zadeh approach begins with defining goals and constraints. **Goals** are an approximate value representing a value to be achieved. A **constraint** is an approximate value that represents a minimum for operation. These are conceptually different, but the mapping of criteria measures using **membership functions** means that they are treated the same by the algorithm. It is possible that the importance of criteria upon the decision are not equal. Certain criteria can have less or more influence on the decision, which can be altered using a **weight** vector. Values in the decision matrix for alternatives, A_m, are aggregated and combined with weights by a **decision function**, D^w. The type of decision function depends on the aims of the decision maker and problem. Some commonly used functions provide different treatment or compensation among criteria (Sousa & Kaymak, 2002). The result of the aggregation procedure yields a vector of values, F, in the interval [0, 1], corresponding to alternatives (rows) in the decision matrix.

The Bellman-Zadeh procedure can be modified for different results, depending on the preference of the decision maker:

- The **fuzzy membership functions** of the criteria can be varied in shape and universe of discourse (range of crisp values from minimum to maximum). This has the effect of changing the boundaries of transition from support to non-support of the criteria.
- The **Decision function** can be replaced to change the aggregation of fuzzy values for inputs. The type of decision function changes the treatment of how criteria weights affect the final calculation. The final aggregation is numerical, therefore a ranking of alternatives is required; the one with highest support being the most preferable (Ribeiro, 1996).

Extensions and variations to the fuzzy MCDM approach have been developed and detailed in meta-studies (Ribeiro, 1996; Carlsson & Fuller, 1996). A fuzzy variation on classical MCDM methods has been applied to ranking of handover criteria (Zhang, 2004). The approach uses membership functions to map linguistic variables into crisp inputs to a MCDM solution (Chen et al. (1992), cited in Zhang (2004)). An alternative approach described in Triantaphyllou (2000, chapter 13), uses fuzzy membership functions to define preferences and input criteria to a decision matrix, and compares fuzzified versions of AHP, WSM, and TOPSIS. These approaches allow fuzzy terms to be used with conventional MCDM methods.

Choosing between MCDM solutions is a decision-making problem itself. Comparisons among conventional MCDM or fuzzy MCDM algorithms in their treatment of criteria preferences, weights, and rankings is a problem for decision theory research. The problem relevant to this thesis, is: which solution provides flexibility in preference elicitation, treatment, and trade-off among criteria? The variations of AHP that introduce fuzziness (Triantaphyllou, 2000; Zhang, 2004) could be used. Criteria in Bellman-Zadeh can be described through membership functions and weights, and Sousa & Kaymak (2002) have shown that different decision functions can be used to change the behaviour criteria of aggregation. Therefore, **fuzzy decision making is used to combine multiple types of criteria for wireless interface assessment.**

Concluding Remarks

This chapter firstly introduced some of the requirements of a system for link-layer assessment and selection. In heterogeneous wireless environments, the operating context can have extensive inputs and interactions with other protocols. An assessment system would assess link performance based on user policy, QoS, and link capabilities. The problem raises the issue of flexibility for multiple sources of inputs, while maintaining a consistent and structured decision process. Similar problems and applications in the literature of AI techniques have warranted the discussion provided in this chapter.

The suitability of probability for handling risk could be a basis for studying handovers from this perspective; that changing between networks establishes a risk of not meeting expectation. However, most of the time there will be only a small amount of randomness or risk; but mostly the type of network will give clues as to operating conditions. Also, the reliance on statistical representations requires history in input data patterns, or at least subjective guesses for probabilities. The requirement of previous data for training and testing restricts learning systems, such as ANNs.

Knowledge-based and fuzzy systems are a compelling technique for the solution, especially for handling human-based reasoning and preferences. Using rules, these systems can be expanded by the decision-maker. In fuzzy systems such as FLCs, these rules can be augmented by linguistic terms ('small', 'some', 'very'). From these system requirements, a fuzzy variation of the MCDM paradigm is considered for further investigation. Although discrete decision processes, they allow for any number of criteria and alternatives, with flexibility in choice of aggregation mechanism.

There are also other subtleties in the decision-making process: that of obtaining inputs and performing actions. Control methods, such as FSMs have been used successfully in providing a formalised language for protocols, and could further augment the decision-making process. A protocol for heterogeneous handover could be conceived that captures environment events and actions, and also utilises fuzzy MCDM for discrete decision-making. A framework is therefore required to integrate fuzzy decision-making and FSM approaches.

Chapter 6

A Simplified Agent Framework for Optimising Handover Decisions

Handover decision-making is made complex by the different capabilities and requirements of each wireless technology and multimedia applications in a heterogeneous wireless environment (as described in Chapter 4). This implies there are further challenges in selecting the "best" wireless interface to use. Issues that need to be addressed are:

1. Interactions with network stack and operating environment, including performance monitoring, link events, and performing mobility actions.

2. Combining requirements with performance conditions for interface suitability.

This chapter provides a framework that combines decision-making and control techniques for interface selection. Using fuzzy decision-making techniques, ratings of link suitability reflect multiple and sometimes incommensurable QoS and user policies. A simplified agent framework is presented that integrates performance monitoring, decision process logic, and interface assessment.

6.1 Scope and Context

Problems of decision-making and handover in a heterogeneous network context include many issues: QoS management and monitoring; user requirements modelling; cross-layer design; handover and mobility management in roaming scenarios; and protocol integration and service mapping for QoS. All of these issues pose questions investigated by large-scale European framework projects, such as MobyDick (IST-2000-25394, 2003) and Ambient Networks (IST-507134, 2007). Therefore, it is necessary to define a manageable boundary to questions and

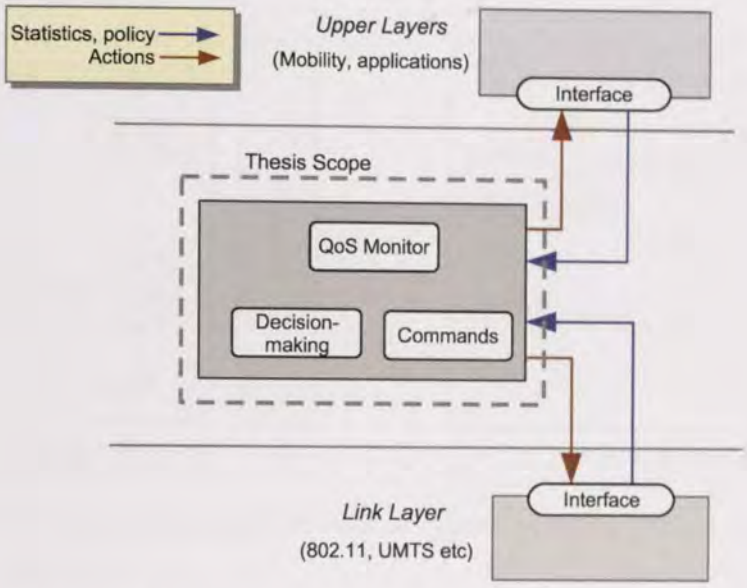


Figure 6.1: Information and processing element for heterogeneous handover selection.

issues presented by this problem.

The scope for the thesis is shown in figure 6.1. Wireless interface protocols and upper layers provide statistics and policy information to the **QoS monitor** which stores them in a data model. This data is used by a **decision-making** process to assess multiple pieces of information about candidate interfaces. External actions are processed by the **commands** component applications programming interface (API) to the client system. The main components within the scope are examined in detail in this chapter.

6.2 Agent Model for Decision-Making

Decision-making in uncertain and dynamic environments is not a new problem in AI and robotics. For example, decision models and algorithms have been used to create robots that can follow paths and avoid objects, or control avatar behaviour in computer game design (van Waveren, 2001).

In the thesis context, a solution is required to select the most preferable link according to support information, such as user criteria, application requirements, and QoS capabilities. To manage wireless interface handovers, the program requires up-to-date statistical information from the TCP/IP stack. Other information may also be important, such as link events, network discovery, and connectivity. These possibilities require a framework that responds to dynamic events and supports future changes in interface technologies. AI related research has developed

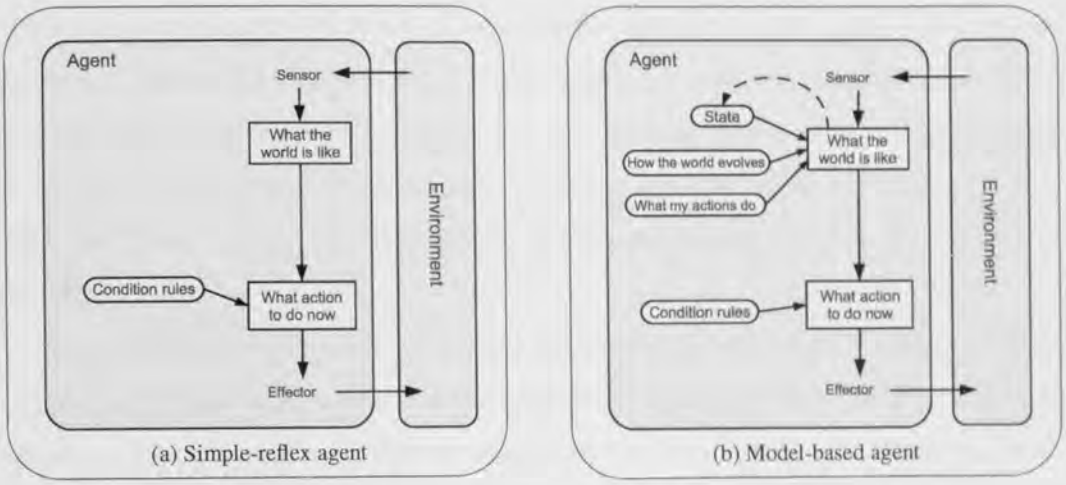


Figure 6.2: Reactive agent models (*Source*: Russell & Norvig (2003, p.47, 48))

techniques for conceptualising intelligent behaviours in computing domains (Bryson, 2001; Laukkanen et al., 2002). The following sections discuss these concepts and how they can be applied to interface selection in a wireless context.

6.2.1 The agent metaphor

Explorations of metaphysical aspects of intelligence have been explored in AI literature Minsky's *Society of Mind* (Minsky, 1988). Others have developed usable frameworks, such as the *Subsumption Architecture* (Brooks & Connell, 1986) for robotics, and the cognitive Soar[1] architecture (Laird et al., 1987; Newell, 1991). These are all efforts to pursue generalised architectures for AI and have been the basis of many subsequent software architectures and research[2].

A common starting position in creating AI is *software agents*. Simple agents have been defined as programs that 'sense' the environment and perform 'actions' to achieve some goal. This is a simplistic description of an agent program that responds to events, or conditions as they occur: a *simple-reflex agent* (Russell & Norvig, 2003). Figure 6.2a is a schematic of a simple-reflex agent that senses and perform actions. Though initially limited, it can be extended to retain some state of the environment, with knowledge of how the environment changes (Russell & Norvig, 2003).

A *model-based agent* (figure 6.2b) supplements the simple-reflex agent with additional domain knowledge to take actions based on rules to reach some goal. These are examples

[1] Previously known as SOAR (State, Operator And Result); now commonly referred to as Soar.
[2] Further exploration of intelligence and the many facets of the domain, can be found in Brooks (1991).

of simple reactive agents which use rules to change behaviour depending upon environment conditions. Other characteristics which can be applied to agents are autonomy or mobility (moving from one execution environment to another). Also, agents can include communication protocols for interacting with other agents. Learning is another characteristic that is often desirable. Using learning algorithms and AI techniques allows the agent to be adaptable in certain conditions.

Research in agent approaches has yielded behaviour-orientated design (BOD): a methodology for creating reactive agents that use hierarchical behaviours (Bryson, 2001). Another reactive approach is used in the Pyrobot framework for robotic design and simulation (Blank et al., 2006), which also contains behaviour-based and finite-state techniques for developing reactive agents.

6.2.2 Simplified agent prototype

The agent metaphor has been proposed in the context of network management (Bieszczad et al., 1999). The approach from Bieszczad et al. (1999) aims to manage network entities and address limitations in conventional protocols, by using autonomously communicating agents. Such agent frameworks are based on standards and designed to be autonomous, interact with environment through messages, and respond to changing conditions. Following this line of reasoning, an agent for network selection with decision-making and control logic is proposed. The agent is defined with the following aims:

- Utilise sources of data sensed from the environment which includes setup parameters, performance data, and QoS data.
- Monitor changes in the environment and adapt to changes in performance or events.
- Propose and implement changes to the wireless interface options that reflect QoS and user policy.

Agent descriptions and components are derived from simplifications of concepts from BOD (Bryson, 2001) and implementations of robotic control systems (Blank et al., 2006). The following concepts are described in relation to the scope (6.1 on page 79).

Events. A monitor module collects real-time performance statistics and events from the operating environment. An event can be internal or external to the agent. External events

could occur from a change in link metrics or status, and discovery of new links. Internal events can be generated by agent procedures.

Brain. The brain process is the main control loop in the agent. Types of brain are defined from AI techniques, such as Subsumption architectures, machine learning, or finite-state machines (FSMs). An FSM-based brain has been the basis of control for other AI projects[3]. FSMs can encode states of the environment, and execute some required behaviour: environment events can result in a change of state. This effect allows the agent to respond to environment conditions using different behaviours, thereby creating a plan for the agent.

Behaviour. Defined as a series of procedures that can later be called by the brain when in certain states. These behaviours are tactical effects the brain can use to support control. In Pyrobot, these are modular Brain components. For the prototype, a behaviour can access **QoS Policy** or metrics model, such as assessing link preference. The brain can support multiple behaviours to activate rules or actions when needed, which are used by the brain logic to make some change to the environment (an output).

Commands. Outputs are actions that require an effect in the agent environment (outside the agent). These outputs are kept separate from behaviours that generate them for modularity. Also, outputs can be written for different operating environments. These define APIs for access to operating system functions, such as cross-layer signalling, initiating handovers, or configuring links. Commands are loaded specific to the client platform; allowing the same brain and behaviours to run in a simulated or real system by loading different command modules.

[3]For controlling game-bots in Quake III Arena (van Waveren, 2001).

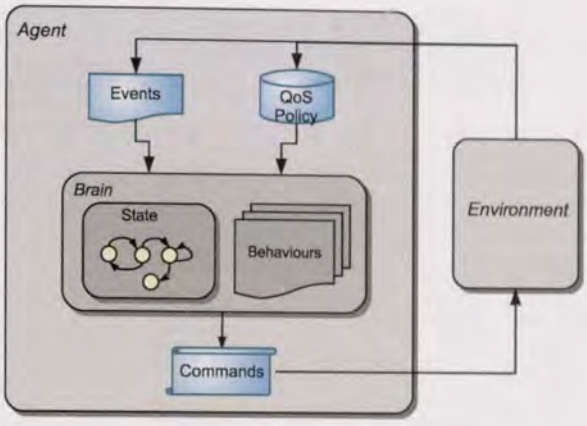


Figure 6.3: Block diagram of agent components.

These concepts are shown graphically in figure 6.3 and relate to the behaviour-orientated design of a solution agent. QoS policies, commands, and events are represented as separate entities that support the functions of the brain and behaviours. Such concepts are still abstract, needing further descriptions on how they apply to a solution for wireless interface selection. The following section begins to address this with a design for agent functions as a controller in the protocol stack.

6.3 Agent Functional Components

The previous section outlined the model for agent components. A handover agent layer (HAL) defines an agent-based container for the brain object (decision logic and behaviours). The agent also provides support functions: update functions to the QoS data model, event lists, and command procedures (figure 6.4). The **QoS Monitor** component provides performance statistics, user policy, and events collection from data sources in the environment. These data sources can be link drivers, applications, or operating system protocols that are outside the scope of HAL. A **decision-making** component is responsible for assessing collected data and events, which uses agent-based concepts described in the previous section, such as brain and behaviours (these will be discussed in later sections of this chapter). Also, HAL must issue changes to wireless interface and other system protocols through the **commands** interface.

Functions that support cross-layer signalling is one approach for providing statistics, events, and commands. However, the thesis scope limits further development beyond that of abstract mappings, and will be focused on agent functions. The HAL framework should support mod-

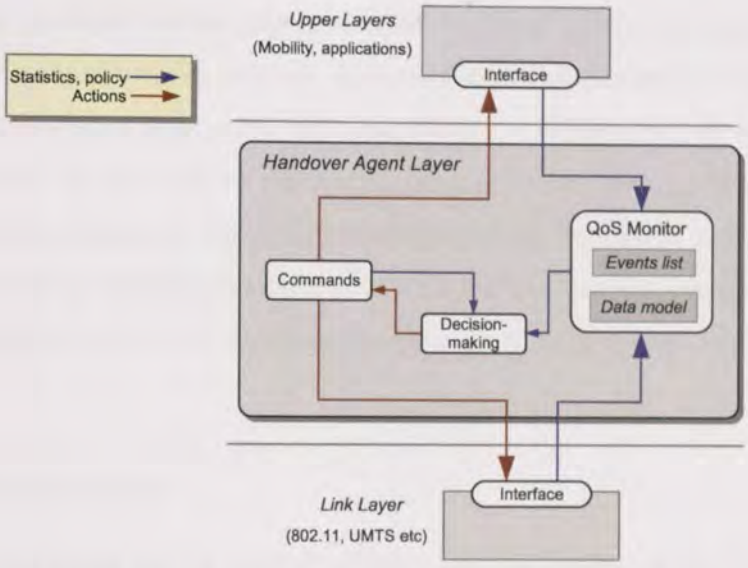


Figure 6.4: HAL controller interaction between main objects within the protocol stack.

ularity; so that statistics and command functions for simulation and testing can be replaced by other modules for real-system testing. The premise is that HAL functions provide performance and event data processing. In the following sections, these components are explained further.

6.3.1 Performance monitoring

The agent performs assessment using performance statistics of protocol stack objects. Links and applications are the primary sources of dynamic performance statistics, but other types could be static or change infrequently, such as user policy and network capabilities. The monitor function stores information on data sources and their attributes during a session in a data model (figure 6.5). The model is updated with raw inputs from sources by the monitor function, and decouples data updates from the decision logic.

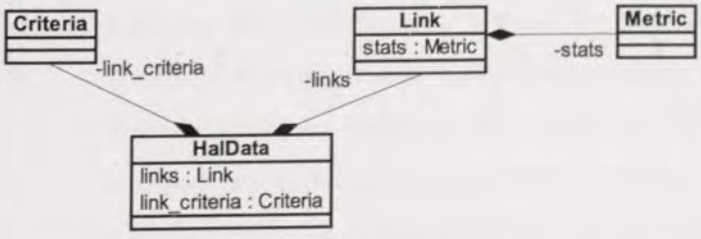


Figure 6.5: Data model for QoS Monitor component.

The logical data schema (figure 6.5) defines **HalData** as the root class. An instance of this class contains objects and operations relating to system data. Objects based on the **Link**

class represent interface interfaces supported by the operating system, containing data on link characteristics and performance statistics. Each type of statistic captured is represented by a **Metric** object. For example, physical layer attributes such as, received signal strength, signal noise ratio (SNR), and signal quality describe the dynamic performance of a wireless interface. Others may be determined by polling or signalling protocols for metrics that could include delay, delay variation, and packets loss. For decision-making, each metric type must have an associated **Criteria**. The criteria maybe defined for a specific service class, for instance voice or web-data.

6.3.2 Event monitoring

As with data monitoring there is a choice of which direction updates occur. The HAL could regularly poll data source objects, or data sources can call an agent function with an event type for changing link conditions. A service point in the logic would need to listen for changes in interface status and metrics changes.

Potentially useful events have been listed in a draft of 802.21 (IEEE, 2006) for media-independent handover (MIH). As part of the MIH Function, the MIH Event Service defines categories of events that include: predictive; state change; link parameters; administrative; link transmission; and link synchronisation. Further details of these are provided in the standard draft (IEEE, 2006, p.32-38), but some are considered for use in the prototype design. Table 6.1 are some possible events used by the HAL.

Event	*Description*
Link.dataChanged	Caused when a change in link statistic(s) is changed.
Link.newLink	If a new link has been discovered.
Link.connectSuccess	A success callback after a connect command has been issued.
Link.connectFail	A fail callback after a connect command has been issued.
HAL.assess	Maybe called on a regular basis by the brain step function.
HAL.prepare	A callback after the brain behaviour has assessed alternative links.
HAL.unstable	A callback from regular assessment behaviour; if the current link is not favourable.

Table 6.1: Sample events and triggers.

Events can be either internal or external to HAL. Internal events may be called by other HAL components as the result of functions or commands. External events could arrive by cross-layer signalling from link objects and services registered by HAL. The agent handles new events (and any parameters) by the event function. Events are processed at each increment of the brain.

Some events represent changes in link status. Other measures of status could be derived from metrics collected by interfaces to determine an abstract measure of status. For example, if only radio metrics are used, such as a signal strength and signal quality, these could be used to derive a calculation of link stability or other predictive metrics. Another use of these metrics could be availability: a device is unavailable if signal strength and signal quality is below some threshold, or there is no reception of beacon frames.

6.3.3 Commands

The 802.21 draft describes a MIH Command Service (IEEE, 2006, p.38-41). MIH Commands originate from upper-layers of the stack to change link layer protocols, that include: requesting status of interfaces; performing handover stages; and scanning links. Some of these actions are similarly used by HAL (table 6.2).

Command	*Description*	*Parameters*
Handover	Finalise handover procedures	target
Prepare	Prepares a link/network for handover	target
Connect	Used to create an initial connection	target

Table 6.2: Agent commands

Metrics evaluation and events can result in an action being taken, or caused by changes in the process model or from changes in state. External actions (commands) are outside the operating mode of HAL, such as changing of the routing table, or interaction with mobility protocols. Therefore, HAL must interact with a platform specific API to affects system settings.

6.4 Control Using Finite States Machines

The HAL uses a fuzzy decision-making (FDM) module to support one-time assessment of inputs given at a single point in time; suitable for well-defined inputs. With other types of

information that are temporal or have a sequence history, it is difficult to represent them in the same format without losing some of the information. A mobile device is subject to dynamic and unpredictable events, and changes in protocol operations; roaming could change connectivity, or network discovery. Knowledge about how the world evolves would be useful to adapt agent behaviour, which has been suggested in approaches to intelligent agent architectures (Russell & Norvig, 2003). In the following sections, finite state machines (FSMs) describe a decision process, combined with FDM.

6.4.1 Brain model for assessment process

Wireless interface selection is based on application QoS requirements and performance. Factors under consideration are constantly changing during the lifetime of a session. Performance of QoS and links may change and new events occur. A decision process must adapt actions depending on events and state of the environment. Interfaces may be in different states, networks may be discovered or lost, QoS policy and service may change. Sequential decision processes inherit a history or progression of states based on temporal events. Methods exhibiting such characteristics have been developed using FSMs to provide a reactive control system for changes in events and state variables.

The principle concepts of behaviour-based design and those from Pyrobot simulator (Blank et al., 2006), are used to define a brain model for controlling HAL processing. A brain is an abstract concept, in that there could be many specialised brains for different purposes. Examples of brain types defined in the Pyrobot toolkit include, Brooks' Subsumption architecture, and FSM-based brains. The latter of these inspired the brain model for HAL. Figure 6.6 is an overview of the types of functions the FSM-brain uses. At each cycle, any events (if none, a default event performed) are processed in sequence by the brain using the FSM model. The FSM may use stored behaviours that corresponds to current state within the FSM. Any actions or commands generated by the behaviours or FSM are then processed by specific HAL functions (callbacks).

Conceptually, states in the FSM are used to represent modes of processing available to HAL based on conditions in the environment. Transitions from states occur through events that may contain parameters or conditional predicates. The FSM brain can use different behaviours depending on the conditions or state of the environment. Using transition parameters and predicates allows changes in behaviour actions depending on states of other variables.

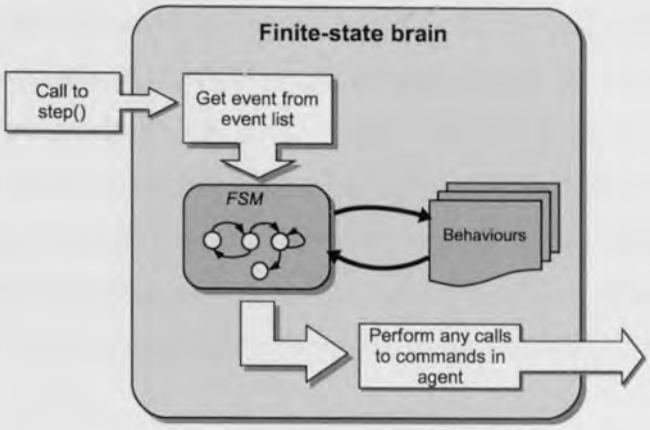


Figure 6.6: Main functionality of FSM brain module.

6.4.2 FSM notation

Finite state machines have found application as AI formalisms in control behaviours for gamebots (van Waveren, 2001) and descriptions of protocols (Byun et al., 2001). The following points are common properties of FSMs:

- FSMs contain *states*, and connections between certain states as *transitions*.
- Transitions to the next state could incur *actions*.
- A FSM is always in one state at a time.

Providing a structured definition and visual representation (figure 6.7), FSMs have properties of deterministic behaviour (Bertoli & Pistore, 2004). A sequence of possible states based on transitions, can be followed to define all possible states based on *inputs* or *events*. Taking any state as the current state, there is a history (previous state) and a future (possible next states). Only when a certain input is provided that triggers a transition from a state, movement to new state can occur.

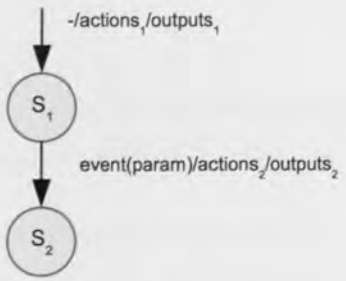


Figure 6.7: Notation for a simple FSM (*Source*: Byun et al. (2001, p.4))

Descriptions of FSMs and automata can be extended to state-charts of UML (Unified Modelling Language) and patterns in software engineering (Gamma et al., 1995). Protocol design in telecommunication systems can use features such as transition parameters, timers, and predicates (Byun et al., 2001). Using predicates and transition parameters for state transitions introduces the possibility of non-deterministic behaviour. Figure 6.8 defines notation describing an Extended FSM (E-FSM) with predicates. E-FSMs use similar concepts from formal adaptable planning concepts, such as those described in Bertoli & Pistore (2004).

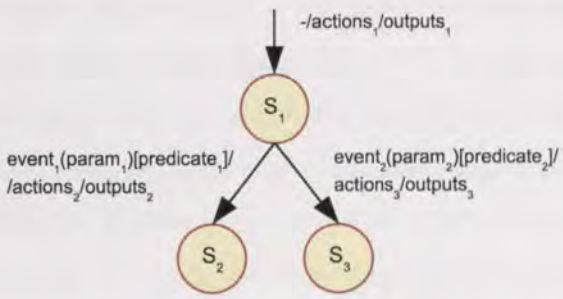


Figure 6.8: Notation for FSM with predicates (*Source*: Byun et al. (2001, p.5))

Concepts from E-FSMs can be implemented[4] for the brain model in HAL. Transition predicates and parameters are used for certain transitions which are explained in the following sections.

6.4.3 FSM prototype

A sequence of decision logic is described as an FSM by interconnections of states, transitions, and actions. Three versions of an FSM prototype are described, each with a different goal and capabilities.

- **FSM-1.** Select an initial network that best matches the service and preferences of the user. It continues to assess all available networks, but there is no subsequent handover control capability.
- **FSM-2.** Once a connection has been established, continue to assess performance of the current link, but only assess other links when poor (fail-over).
- **FSM-3.** Active assessment of alternative link selection opportunities, providing optimal selection from available interfaces and newly discovered links.

[4]Similar concepts with code implementation (see Rapp, 2007).

The FSMs for handover and decision control prototypes are described using the notation of section 6.4.2 (diagrams in this section are shown with less detail; full details are shown in Appendix A.2).

FSM-1

This FSM establishes an initial interface at the start of a session. The design goal is to select and then constantly assess, but only for monitoring purposes; no handover actions are performed.

The model uses two states (figure 6.9). Firstly, there is no interface selected (`NotSelected`), which is the start state after initialising the HAL brain module. If there are at least two available interfaces (since one link gives only one choice), the monitor function updates the data model with current statistics. At every *step* of the Brain, transition *assess* is called and the associated behaviour to assess all links. If there is a suitable link, a *useLink* transition is invoked and the state is changed to `Selected`. At the same time as the *useLink* transition, output from the behaviour is a command to *connect* to target link, which is executed by the handler function, *command*.

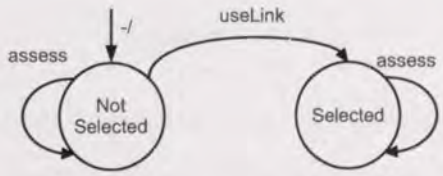


Figure 6.9: *FSM-1*

This FSM aims to setup an initial connection, continuing to assess all networks, but has no handover behaviour. Using this approach, the simplest FSM can be conceivably used with human decision-making. For example, this FSM could alert the user of the assessment, leaving the subsequent decision on the user. The next step in a more sophisticated FSM, with states and transitions for handovers.

FSM-2

FSM-2 extends the initial selection approach of FSM-1 to include assessment once a interface is selected. The aim is to monitor current conditions of the interface and QoS conditions from upper layers. If the current levels become unsuitable, other links are assessed for handover.

When a link is ready, control is moved to the `Selected` state (figure 6.10). In this state, additional behaviour is added that monitors the current selected link (more QoS metrics may

be collected from the upper layers). If the assessment behaviour determines that current link has become less preferable or unstable, a transition is made to `CompareLinks` state. If there is a more desirable link available, a transition is issued to a `PrepareHandover` state. This state initiates a command to prepare for handover to the target link. The command is mapped to the handler function for execution. If this is successful, a *connectSuccess* transition will be made to `Selected` for continued monitoring.

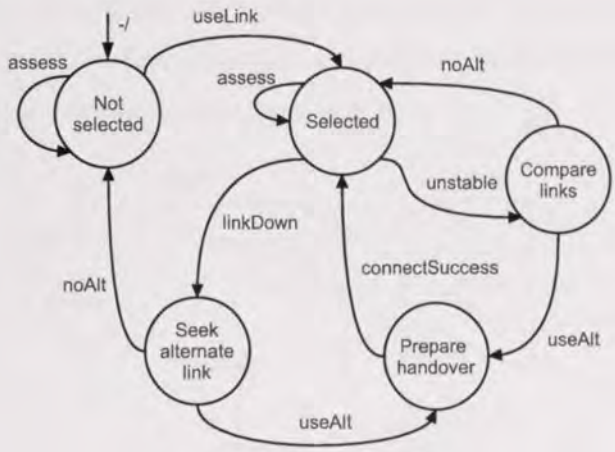


Figure 6.10: *FSM-2*

Some states (`SeekAlternateLink` and `CompareLinks`) use the same behaviours, but have different transitions. For example, `SeekAlternateLink` state can only be entered by a loss of connectivity in the current interface (*linkDown* event when in `Selected` state), and if no alternative, then continue to assess from the `NotSelected` state.

Although FSM-2 has the additional behaviour for monitoring and link handovers over FSM-1, it is still limited. The design only checks the current link performance using combined assessment of metrics. Only if there is a deviation from the constraints are alternatives assessed. This misses opportunities of handovers to more desirable alternative links; FSM-2 does not assess others if the current link is still suitable.

FSM-3

Using additional events and behaviours, FSM-3 aims to provide active monitoring for changes in interfaces status, such as discovery of new links or loss of connectivity. As in FSM-2, the selected state checks the currently selected interface (AssessCurrent behaviour), but also uses AssessCurrentAll logic. This new behaviour provides a logic that returns an *unstable* event if

there is a more preferable interface (behaviours are discussed in section 6.5 on the following page).

The monitoring process in the agent will provide *linkUp* event if a new link is discovered (figure 6.11). If the model is in the `Selected` state, the *linkUp* event causes a transition to the state `CompareLinks`. This state has behaviour that compares the current active link with the new detected link using comparable criteria. If the new link has a higher rank than the current link, a transition to `PrepareHandover` is made. Alternatively, if there is no increase in rank above the current link, the new link is rejected (but added to the known list of links) and control is returned to the `Selected` state without any link change.

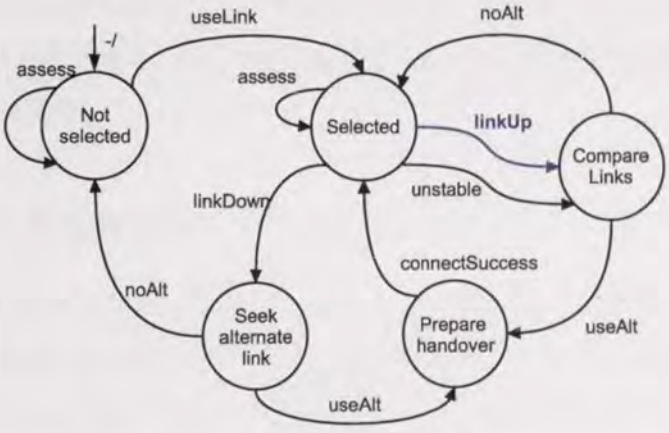


Figure 6.11: *FSM-3*

FSM-3 extends the FSM-2 design to support handling of detected links and active assessment. The previous goal is achieved by continually assessing the performance of the current link then changing the states if further actions are required. FSM-3 modifies this goal to monitor events, such as discovered interfaces, and assess whether they would be better than the current link. Not only are new interfaces checked, but known networks performance may change and become more desirable.

6.4.4 Changing state to change behaviour

The FSM is event driven and activated by the parent agent. In the agent model (figure 6.3 on page 84), events are passed at regular intervals; for example every 0.5 seconds. An event waiting to be processed is mapped to FSM functions that issue transitions. The transition to a next state can cause actions–methods (or functions) in the code–and perform outputs. Any outputs, or commands are invoked by the command function. This allows the brain to make

abstract commands, which the HAL module maps to real commands via an API for interacting with the operating environment.

The rationale for using sequential states and event-based transitions, is that FSM models provide a structure with deterministic and logical progression of state changes. States and transitions provide different actions depending on the current state and events (inputs). Some actions are modelled as modular behaviours (as presented in the agent architecture, see 6.2.2 on page 82). Behaviours are a set of procedures attached to the brain. In the FSM prototype, behaviours are called from state transitions. Additional logic such as conditional checking and entry/exit functions could be added to the FSM implementation. However, to compare alternative brains without FSMs, behaviours are used to encapsulate some of the logic and support reuse. The following section describes the behaviours and the low-level logic used to support high-level FSMs.

6.5 Modular Behaviours

The Pyrobot framework groups low-level logic into modular units called behaviours (Blank et al., 2006). These are designed to be called at regular intervals or when in certain states (as in the FSM brain). Behaviours allow a modular approach to supplementing functionality of the brain module. For the HAL prototype, the following behaviours are proposed.

- Assessing all candidate interfaces for initial selection (initial).
- Assessing the currently in use interface (preservation).
- Assessing all links for changes in other link conditions and opportunities for handover (optimisation).

Behaviours use the status of interfaces in some of the logic. It is necessary for HAL to record interface status for such processes. Each physical link device belongs to the set *all links* (figure 6.12). Those that are powered-up and capable of connectivity are in the subset of *active links* (these are added to the HAL data model). At points in time, interfaces from the active set can be chosen *(link choice)* and/or in-use *(link used)*. These categories of links are used to support behaviour logic and stored in variables in the HAL data model.

The purpose of the behaviour is to allow different processing depending on the brain module and at different times of a session. Types of behaviours are defined in the following subsections.

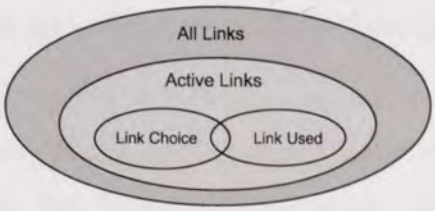


Figure 6.12: Link status terms

6.5.1 All link assessment (AssessAll)

Behaviour logic for assessing all links is shown in figure 6.13. Statistics and other data about the interface capabilities are used by the FDM module to obtain a ranked list. At the top of this list will be the most preferable link according to the decision criteria; this will be the link with largest FDM score. With a target selected, the behaviour adds a *connect* rule to be executed by the brain.

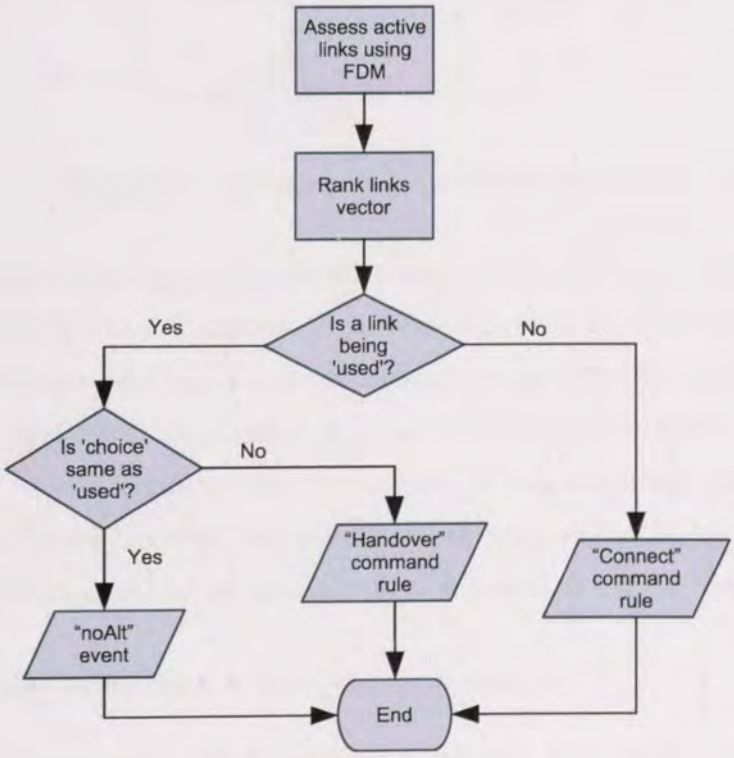


Figure 6.13: Algorithm for AssessAll behaviour

For situations where there is already an interface being used, if the choice is different (a new top rank) a *handover* command rule with target interface as parameter is issued. If `link_choice` is the same as the `link_used` interface, then no actions are required and `noAlt` event is issued; `link_used` is already the most suitable.

6.5.2 Currently selected link assessment (AssessCurrent)

When a link is 'used', there may be more detailed statistics available, such as application or network layer QoS metrics. In this case, the AssessCurrent behaviour uses detailed QoS statistics to assess the current link for handover assessment. The procedure in figure 6.14 assesses the current link performance only.

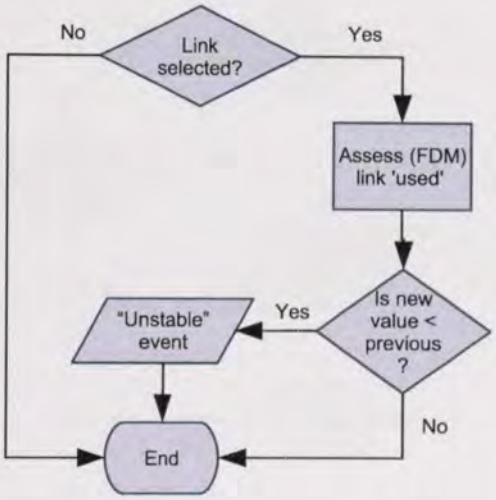


Figure 6.14: Algorithm for AssessCurrent behaviour

Performance statistics are passed to the FDM module and a single rank-score is generated. With the AssessAll behaviour the comparison was relative to other alternative interface options. A single FDM output value has no point-of-reference to determine the suitability. This is a problem of mapping the fuzzy number to a relevant control action; in this case the value that makes the assessed link as unstable or unsuitable. A simple approach uses the previous assessment value, and if it is lower, then generate an `unstable` event to change the FSM state. More advanced approaches may be used, such as moving averages and hysteresis functions.

6.5.3 Assessing all for current link (AssessCurrentAll)

This behaviour is used with a different intention to previous descriptions. *AssessCurrentAll* would assess all available interfaces as in *AssessAll*, but instead of issuing a handover or connect commands, an `unstable` event is issued if the highest rank changes (figure 6.15). This behaviour is intended to be used in a continuous assessment whilst there is an already selected interface. Only if the rankings change, should further actions be taken.

Depending on the HAL brain used, the `unstable` event could lead to further assessment

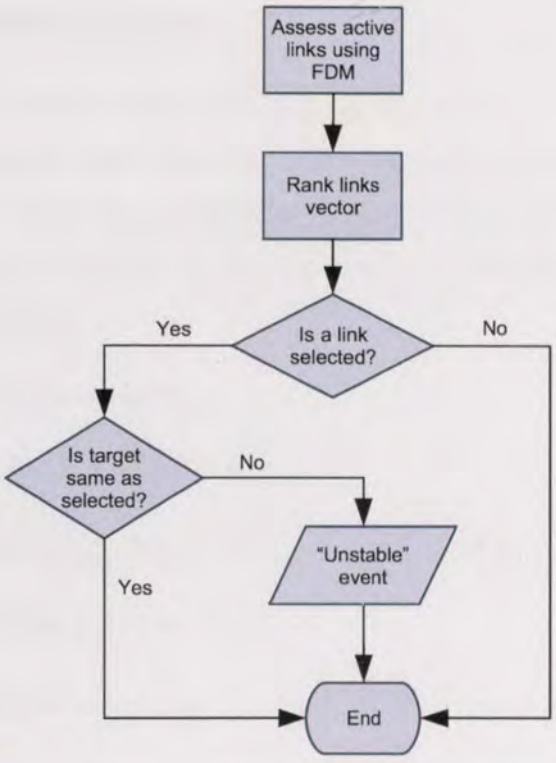


Figure 6.15: Algorithm for AssessCurrentAll behaviour

logic. In FSM- 2 and 3, the `unstable` event re-evaluates the interfaces again in the next state. Overall, the aim of the behaviour is to constantly check the suitability of other interfaces that may match preference policies.

6.5.4 The role of FDM in behaviours

HAL reacts to events and data in the environment by FSMs. States in the FSM use behaviours that solve a small problem (similar to the Pyrobot toolkit (Blank et al., 2006)). Functions of FDM module for assessing interface statistics are used by behaviours described previously. Behaviours are used to combine techniques for a hierarchical division of processing (or problem sub-division); from low-level measures of performance in the FDM module, to high-level sequential plans for events and actions using the FSMs. The next section explains the details of FDM processing.

6.6 Fuzzy Decision-Making

Elements of decision-making include defining goals and constraints. These goals and constraints are used to calculate option suitability values. According to the definitions from Bellman & Zadeh (1970), goals and constraints are treated the same, and represented by fuzzy membership functions over the set of alternatives. A decision-making problem can be defined by (Sousa & Kaymak, 2002):

- A finite set of alternative choices.
- A set of goals and/or constraints - the criteria.
- Mappings of criterion per alternative - the membership values.
- Importance or weight factors of criteria.
- A decision function which defines the ranking of alternatives to the satisfaction of criteria.

The behaviours defined procedures or logic for assessment. FDM is used for the low-level assessment of multiple criteria and alternatives, shown by figure 6.16. Inputs include preferences collected from policies and performance indicators. Mappings of input values from **criteria** membership functions yield fuzzy values. These are combined with weights for criteria by the **decision function**. The output is a numerical vector of scores for representing support for each alternative option.

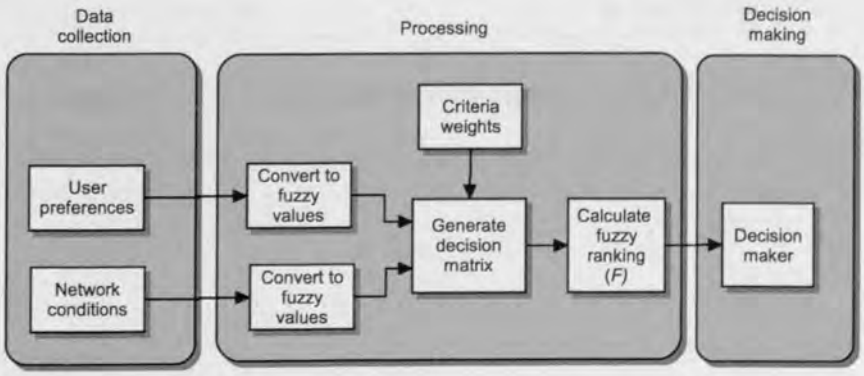
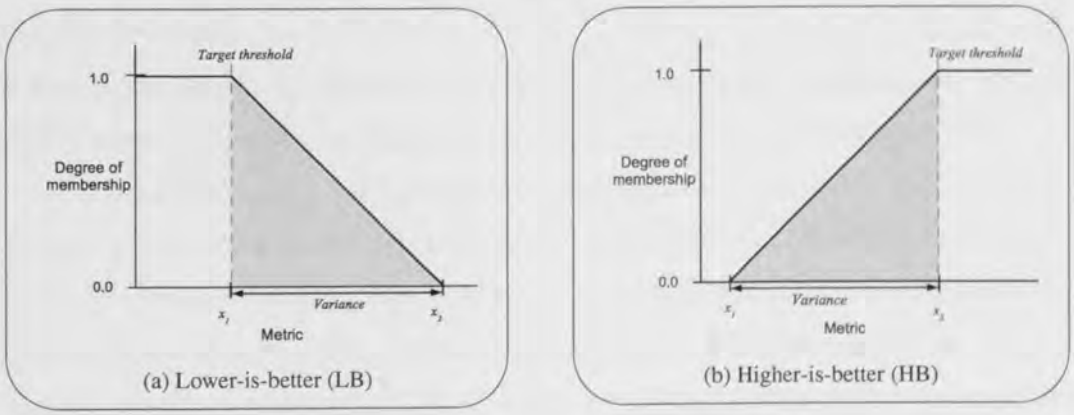


Figure 6.16: Schematic system model of FDM operation

6.6.1 Setting criteria

The FDM uses inputs values mapped to membership functions. Each function defines a degree of membership for criteria values. The line between inclusion and exclusion is fuzzy, where parameters of the function define a geometric shape. Parameters can take numerical ranges and thresholds, or linguistic terms. Figure 6.17 shows functions types[5] used in the module. For a linear and triangular shape, two parameters are used. A **target threshold** value is used to define the minimum (HB) and maximum (LB) acceptable values for each type. A **variance** value defines the level of slope, or support between degrees of membership.

An example criterion is suggested for a service type that requires QoS for low latency: less than 400ms. Translating this to membership function parameters is shown by a lower-is-better function in figure 6.18. The criteria is given a target threshold value of 100ms; anything less is given a membership of 1. A variance of 300ms is added for a smooth transition between thresholds. For values 100ms to 400ms, there will be a falling degree of membership and therefore less support for the criteria as the variable value increases.



(a) Lower-is-better (LB)

(b) Higher-is-better (HB)

Figure 6.17: Fuzzy membership functions used for defining criteria.

[5] Although membership functions can be any shape, such as sinusoidal, triangular MFs are a simple representation for initial prototypes.

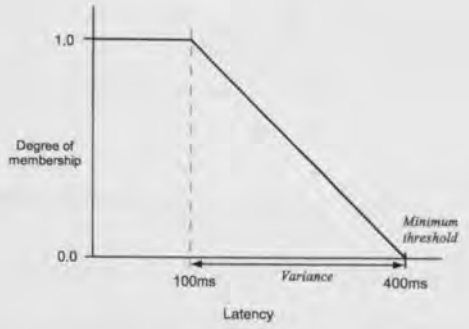


Figure 6.18: Linear (LB) membership function for criteria: low latency.

6.6.2 Decision functions

Fuzzy values from membership functions are used to populate a decision matrix of fuzzy values for criteria and alternative options. The next step is to aggregate the fuzzy values using a decision function to obtain a single value. Three types of aggregation are can be used: *conjunctive, disjunctive*, and *compensatory* (Sousa & Kaymak, 2002). Conjunctive aggregation aims to satisfy all criteria. It uses intersection (minimum) operators. Disjunctive functions satisfy at least one of the criteria, described by union (maximum) operators. Compensatory methods apply a mixture of conjunctive and disjunctive characteristics for a trade-off effect.

For conjunctive aggregation, operators are triangular norms, or *t-norms*. In fuzzy terminology, t-norms are a form of intersection of sets. The t-norms have the effect of aggregating towards the worst criteria (the minimum value). Contrary to this, triangular co-norms, or *t-conorms* provide a fuzzy union, or maximisation aggregation. Such functions are used when at least one criteria is good enough, and likely to have a higher aggregation result the more criteria are used (Sousa & Kaymak, 2002). Further axiomatic descriptions of t-norms and t-conorms can be found in Klir & Bo (1995).

Compensatory or averaging functions allow partial trade-off amongst criteria, as opposed to extreme compensation of conjunctive and disjunctive operators. This tends to be the case in human decision-type problems, where there is often a trade-off between best and worst criteria (Zimmermann & Zysno, 1980). The geometric mean operator is an example of averaging aggregation that compensates between criteria (6.1). Output D is the rating for a particular option for n criteria, where $\mu_j$is the fuzzy membership value.

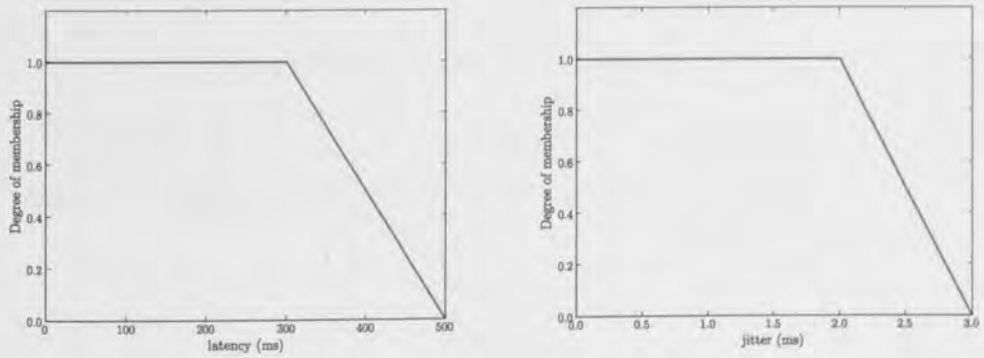
$$D(\mu_1, \ldots, \mu_n) = \prod_{j=1}^{n} \mu_j{}^{1/n} \tag{6.1}$$

Example of geometric mean operator

Given a set of alternatives, A_1 and A_2, a voice application has two key criteria: latency and jitter. Interactive voice has strict requirements on latency. According to ITU-T (2001b), latency or one-way-delay should be under 400ms, and jitter should be less than 2ms (not taking into account jitter buffering). Using these values as application requirements, decision criteria are defined as C_1: 'latency lower than 400ms'; and C_2 'jitter lower than 2ms'.

Defining criteria would usually be a judgement of the decision maker and those relevant to the problem. In this example, acceptable values for the criterion are defined by application constraints. Criteria C_1 and C_2 are defined by membership functions in figure 6.19. Input values for each alternative and criteria are mapped to membership values. Obtained fuzzy values represents the degree of satisfaction of criteria for each alternative. Using some values of a for each alternative and criteria in a decision matrix:

	C_1	C_2
A_1	350	1.5
A_2	450	2.5



(a) Latency

(b) Jitter

Figure 6.19: Possible membership functions for QoS.

Mapping of raw values to fuzzy values from the membership functions in figure 6.19 is performed. The decision matrix now contains the fuzzy values of raw inputs to membership mappings:

	C_1	C_2
A_1	0.75	1
A_2	0.25	0.5

The decision function performs the combination of membership values on the matrix. A geometric decision function is applied to the fuzzy decision matrix in this example (defined by equation 6.1). The result is a fuzzy rank-scores of alternatives:

$$F = \{(A_1, 0.87), (A_2, 0.35)\}$$

The output ranking, F gives A_1 the higher rank and therefore the preferred choice in this example. The algorithm assumes equal compensation among criteria (equivalent weights), and does not account for importance between criteria. All class of operators can be extended to use weightings of criteria importance (Sousa & Kaymak, 2002).

Introducing weights

In the previous example all criteria are treated equally. There may be some problems where criteria are not equal in importance. In this case, weights can be used in decision functions to change the effect of criteria on the rank-score. The method introduces weights of criteria to calculations of the decision function, known as "transforming the decision function" (Sousa & Kaymak, 2002, p.58). For the geometric mean function (6.1), the following weighted form is defined as:

$$D^w(\mu_1, \ldots, \mu_n) = \prod_{j=1}^{n} w_j \mu_j{}^{1/n} \tag{6.2}$$

Usually, the weights are in the range [0,1], and normalised according to:

$$\sum_{j=1}^{n} w_j = 1. \tag{6.3}$$

6.6.3 Ranking methods

Output from the decision function is a vector of rank-scores of each alternative. The next procedure is to rank these results for a preferred alternative. A simple method is to rank in descending rank-scores. Therefore a result, $D(0.4, 0.6, 0.7)$ would give ranking of $D(2, 1, 0)$; where 0.7 is the preferred option.

Methods for discerning between options with the same rank-scores, or ties, are: random, average, and first-occurrence. Random option is as the name implies, and so could change rankings for consecutive data; and likely to cause inconsistent decisions. Averaging ranks still gives no distinction between the options. The first-occurrence method ranks ties according position in the vector. So, vector of rank-scores D(0.4, 0.4, 0.7), gives D(1, 2, 0); the first occurrence of 0.4 is ranked higher than the second. Although this may be biased to positions in the case of ties, it will provide consistency for a simple ranking method.

6.6.4 FDM algorithms

This section defines algorithms for FDM methods described in the previous section. Algorithm 6.1 assumes a criteria vector, and a matrix for alternative options. The output is a vector of values in the interval [0,1] for each alternative in A .

Algorithm 6.1 Fuzzy Decision Making Main

fdmEvaluator:
Input: Criteria vector C, and Alternative matrix, A
Output: Ranked alternatives vector, R
$C \leftarrow$list of criteria functions
$A \leftarrow$performance measures for each criteria
fuzzyMatrix $\leftarrow$*generateFuzzyMatrix(C, A)*
$R \leftarrow$*generateRanking(C, fuzzyMatrix)*

Algorithm 6.1 passes vector and matrix to the function *generateFuzzyMatrix* (algorithm 6.2), which matches the values from vectors in matrix A to the criteria in C. Items in C represent membership functions mappings. For each alternative m, there are n data values, which correspond to criteria, C. Values in A_{mn}, are passed to the function in C_n that returns the fuzzy value. The fuzzy matrix, F, is then returned.

Algorithm 6.2 Fuzzy Matrix Generator

generateFuzzyMatrix(C, A):
Input: Criteria *C*, and Alternative values, *A*
Output: Fuzzy matrix, *F*
F ←[]
for *n*←0 *to A.length* **do**
  alt ←[]
  for *n*←0 *to A[n].length* **do**
    alt[m] ←*C[m].getFuzzyValue(A[n][m])*
  end for
  F[n]←*alt*
end for

Algorithm 6.3 describes the **weighted geometric mean decision function** (Sousa & Kaymak, 2002, p.45), using criteria vector and fuzzy matrix of values to obtain a vector of ranking values per alternative. Each fuzzy value in the matrix is multiplied by the corresponding criteria weight. The n^{th}root of the product for A_{nm} values gives the ranking of that alternative. Vector *R* is returned, corresponding to *A*. Further processing of the result vector may be required, such as the degree of difference between ranks, or whether the results are much different from previously generated rankings.

Algorithm 6.3 Fuzzy Decision Function: weighted geometric mean

generateRankScores(C, F):
Input: Fuzzy matrix, *F* and criteria, *C*
Output: Fuzzy ranking vector, *R*
R ←[]
weightsVector ←[]
for *m*←0 *to C.length* **do**
  weightsVector[m] ←*C[m].weight*
end for
weightsVector ←*normalise(weightsVector)*
for *n*←0 *to F.length* **do**
  altVector←[]
  for *m*←0 *to F[n].length* **do**
    altVector[m] ←*weightsVector[m]*×*F[n][m]*
  end for
  R[n] ←*product(altVector)*$^{\wedge(1/F[n].length)}$
end for

Algorithm 6.3 is suggested for initial testing as part of the decision logic to provide compensatory behaviour among criteria. Fuzzy decision algorithms are interchangeable and others can be used to provide different trade-off effect between criteria. The following section outlines

an extension to the previous algorithms to add more flexibility in the compensation effect.

6.6.5 FDM enhancements

Many variations of decision function have been suggested in the literature (Carlsson & Fuller, 1996; Ribeiro, 1996; Sousa & Kaymak, 2002, chap. 3). Averaging and compensatory functions have been shown to apply trade-offs among criteria, with criteria importance controlled by weights. The following is a variation of the decision function a modifier parameter for aggregation tuning.

A *generalised weighted averaging operator* (GWAO) introduces a new element: γ, or *grade of compensation* (Zimmermann & Zysno, 1980) or *index of optimism* (Kaymak & van Nauta Lemke, 1998). The parameter changes the value of rank-scores. For large values of γ (towards ∞) good values of alternatives have more of an effect; when γ tends towards $-\infty$, poor values for alternatives are emphasised (Sousa & Kaymak, 2002, p.46). The effect of γ in generalised averaging functions can be shown through *sensitivity analysis* (Kaymak & van Nauta Lemke, 1998). These studies suggest that γ could be used to reflect the level of risk (optimistic or pessimistic) of the decision maker. GWAO with index of optimism parameter, is defined as (Kaymak & van Nauta Lemke, 1998, p.7):

$$D_i(\gamma) = \left\{ \sum_{j=1}^{n} w_j \mu_{ij}^{\gamma} \right\}^{1/\gamma}, \quad \gamma \in \mathbb{R} \setminus \{0\} \tag{6.4}$$

Algorithm 6.4 uses the definition of the **generalised mean operator** defined in equation 6.4 with the new optimism parameter, γ.

Algorithm 6.4 Fuzzy Decision Function: generalised mean operator

Algorithm generateRankScores(C, F, γ):
Input: Fuzzy matrix, *F;* and criteria, *C;* and parameter, γ.
Output: Fuzzy ranking vector, *R*
$R \leftarrow []$
$weightsVector \leftarrow []$
for $n \leftarrow 0$ *to C.length* **do**
  $weightsVector[n] \leftarrow C[n].weight$
end for
$weightsVector \leftarrow normalise(weightsVector)$
for $n \leftarrow 0$ *to F.length* **do**
  $altVector \leftarrow []$
  for $m \leftarrow 0$ *to F[n].length* **do**
    $altVector[m] \leftarrow weightsVector[m] \times F[n][m]^{\wedge\gamma}$
  end for
  $R[n] \leftarrow sum(altVector)^{\wedge(1/\gamma)}$
end for

Concluding Remarks

Managing handovers and QoS assessment has many factors and implications in heterogeneous environments. This chapter defined an initial scope and solution framework to address a subset of questions posed by Chapter 4. An agent framework from AI and robotics was used as a design metaphor for managing complexity of cross-layer interactions and aggregating information from different sources. This defined functions of: performance monitoring and policies; a control logic for interpreting input data; and modularity for different configurations and behaviour requirements. This aims to support handover control logic for different link types, QoS, and user policies.

At the core of the framework is control logic to handle event changes and assessment of link and QoS statistics. This uses finite-state machines to provide a sequential logic and perform actions when in different states. Modular behaviours are defined as separate procedures, such as assessing link performance. Assessment behaviour described in this chapter, uses fuzzy decision-making techniques to give suitability rankings of interface options.

The proposed framework and its components provide a behaviour-based agent for wireless interface assessment and handover control. Outputs from the handover agent layer (HAL) decision-making reflect QoS requirements and conditions to provide timely handovers and appropriate interface selection. The following chapter will explain implementation details and an approach to evaluating the HAL prototype.

Chapter 7

Evaluation Method for Interface Selection Agent

Testing approaches to handover selection requires tools to model more complex wireless architectures. Wireless simulation models already exist, but those for heterogeneous environments are still emerging. As research poses further questions, such tools commonly need to be modified. Though the more capable tools allow such development, there is no standard or 'best practice' approach for wireless heterogeneous environments.

The handover agent layer (HAL), described in the previous chapter, is middleware for optimising vertical handovers. What follows is an evaluation approach using analytical techniques to decision-making components and network simulation for HAL. Experimental design provides controlled treatments for inputs and simulation scenarios.

7.1 Evaluation Approach

Three evaluation approaches are often used in computer systems performance studies: analytical, simulation, and measurement (Jain, 1991). Fuzzy decision-making (FDM) elements, such as those used in the HAL prototype, have been studied using analytical techniques (Kaymak & van Nauta Lemke, 1998; Triantaphyllou, 2000; Zhang, 2004). Since HAL is also a component of a greater system (wireless networked environment), simulation or empirical studies of network protocols are useful. The research questions that need to be answered defines the evaluation approach for a controllable environment; key questions that motivate the study:

1. What methods can be used to model the factors of a wireless environment?

    (a) What level of detail should be modelled?

(b) What tools can be used?

2. What are the relevant factors of interest; or those to be modelled?

The three approaches to performance study (analytical modelling, simulation, and empirical measurement) provide varying benefits and complexity in development. Selection of an appropriate approach is based on criteria such as, duration of study, expertise in modelling tools, accuracy, cost, and interactions of variables (Jain, 1991, p.24).

Analytical approaches can be used to explore the effects and interactions of variables. Approaches for evaluating FDM algorithms have used *sensitivity analysis* (Kaymak & van Nauta Lemke, 1998; Zhang, 2004). Sensitivity analysis compares outputs results from interactions of input parameter levels, similar to full-factorial experiments that compare output variables between types of decision algorithm (Triantaphyllou, 2000, chap.13). Analytical approaches have the benefit of greater control of factors and metrics. However, more control over modelling of the input space can mean that other system interaction are not shown.

Discrete event simulation (DES) is often used in network testing as it offers richer modelling of the operating environment than analytical techniques, though with less control of parameters. Modelling expensive hardware and distributed systems in simulation is often less costly and provides better repeatability than empirical measurements. Network simulation tools provide models of different wireless technology, mobility, topologies, and traffic models. However, a simulation model is still an abstraction constrained by problem scope. Therefore, selection of evaluation technique is a trade-off in controllable parameters, accuracy, and detail of abstraction (Jain, 1991, chap.3).

7.1.1 Modelling wireless environments

Wireless environments are notoriously complex to model fully, whereas wired physical properties are more predictable and simpler to model (Heidemann et al., 2001b). Specific simulation tools can provide models of the wireless channel, from simple free-space to more sophisticated multipath models. The level of detail in the model is determined by the problem scope. A study concentrating on the effects of signal attenuation in urban environments would require detailed propagation and geographic models. Therefore, there must be a trade-off in model detail depending on the significance of parameters and system boundaries (Heidemann et al., 2001a).

A model of the environment can be built using high-level programming language or dedicated simulation tools. Systems can be modelled in C++ or Java, but these tend to be costly in expertise, time, and debugging for more detailed models (Jain, 1991). Specific simulation languages, such as SIMULA and SIMSCRIPT provide well defined functions for simulation, but limited by scope of the language syntax (Jain, 1991). High-level programming languages provide a flexible syntax that can be adapted to support simulation features such as, event scheduling, data collection and timing (Jain, 1991). SimPy is an open-source package for DES in the Python language; providing object-orientation, portability and the necessary features of DES.

Alternatives to general-purpose tools are dedicated domain tools, such as network simulation. Network simulation tools provide packet, physical channel models, and workload generators. Commonly used tools include the commercial Opnet and open-source NS-2 (NS-2, 2008). An interesting simulator, NCTUns[1], routes simulated packets through the TCP/IP stack of a modified Linux kernel to provide realistic packet treatment (Wang et al., 2003). NS-2 was chosen for generating network simulation data, with Python and SimPy used to model components of HAL. Further detail of simulation tools is provided in Appendix B.1.

7.1.2 Subject prototypes

Experimental design literature suggest that a mixture of approaches is usually required (Jain, 1991). This section describes different variations of prototypes (test subjects) for evaluation approaches. Analytical techniques can be used for refining parameters used in simulation, or simulation to verify the analytical model (Jain, 1991, chap.3). An evaluation approach (figure 7.1) uses analytical techniques for decision-making component, and simulation to evaluate HAL in a wireless network environment. For decision-making analysis, several variations are compared with the most appropriate being considered for use in simulation tests. The subsequent simulation studies use modified versions of HAL for comparisons.

Analytical techniques are used for assessing behaviour of decision-making algorithms described in Chapter 6 with other commonly used algorithms. Table 7.1 shows other methods of decision-making using different parameters or decision functions for criteria aggregation. A weighted-sum model (F-WSM) is used as a control subject; as an example of simple aggregation. The other functions also perform aggregation, but using different parameters.

[1]Freely available for non-commercial use, at: `http://nsl.csie.nctu.edu.tw/nctuns.html`

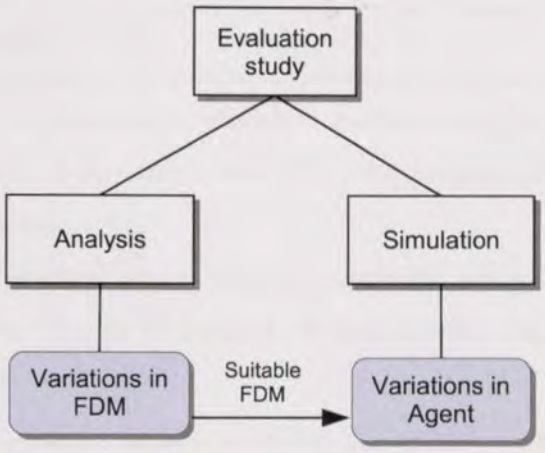


Figure 7.1: Evaluation approach using different decision-making methods for analysis and HAL configurations for simulation.

Prototype	*Description*
F-WSM	Based on the weighted-sum model (WSM): fuzzy variation.
F-Geo	Geometric mean fuzzy decision making (FDM) function.
F-WGeo	Weighted geometric mean fuzzy decision making (FDM) function.
F-GenO	Optimistic version of generic-FDM.
F-GenP	Pessimistic version of generic-FDM.

Table 7.1: Decision-making component prototypes for analytical study.

Simulation-based data is used for experiments that use different settings for HAL components. The state-based HAL uses three different state models. Table 7.2 summarises the prototype implementations using the decision-making method selected from the analytical experiments. The FSM prototypes are compared to a non-FSM model of the FDM-Brain. Using FDM functionality, a subsumption architecture (Brooks & Connell, 1986) is used to control triggering behaviours; the implementation uses Pyrobot (Blank et al., 2006) examples and libraries. Details of FDM-Brain and other agents are described in Appendix A.1.

Prototype	*Description*
FDM-Brain	The FDM algorithm selected from analysis stage. Uses a stack to implement behaviours instead of an FSM (based on Brooks' subsumption architecture).
FSM-1	Select an initial network that best matches the service and preferences of the user. Continue to assess all available networks, but there is no handover control.
FSM-2	Once a connection has been established, optimise the performance of the current network.
FSM-3	Assess opportunities of alternative networks that benefit the level of service and user preferences.

Table 7.2: HAL prototypes for simulation study.

Each prototype and support classes are implemented in Python (version 2.5.1) programming language. Python has a concise syntax and no compiling stage, which is a benefit for rapid prototyping. Other benefits include dynamic types, full object-orientation, and support of third-party tools and libraries.

In comparison, Python performs well among other dynamically-typed and interpreted languages, and (in some cases) with compiled languages (Prechelt, 2000). The study by Prechelt (2000) ranks Python favourably for program length (concise syntax) and implementation time, though C and C++ provide lower processing time and less memory consumption. Therefore, the preference depends on the importance of run-time performance versus the program design. However, Beazley (2003) describes programs that combine compiled language components for processor and memory intensive functions, and scripting for high-level configure functions.

For initial prototyping, HAL does not require the performance aspects of compiled languages, since the agent will perform high-level processing in simulation time, and in the interval of seconds. The benefits of Python are that it provides a rapid prototyping and testing platform, that can be migrated to a real-time environment with nominal changes (possibly performance and device interactions APIs).

7.2 QoS and Decision Criteria

Measured criteria values are converted to a common scale for comparison, using fuzzy membership functions (including fuzzy linguistic terms). The QoS criteria could include those from figure 7.2. First-order criteria are usually hard to quantify without further descriptions, such as timeliness. Second-order criteria can be used to give indications to first-order criteria: latency and jitter for timeliness. For example, signal strength (RSS) and coverage metrics help to describe the availability of a wireless link. This section explains some of the more common QoS requirements for decision criteria definitions used in the experimental designs.

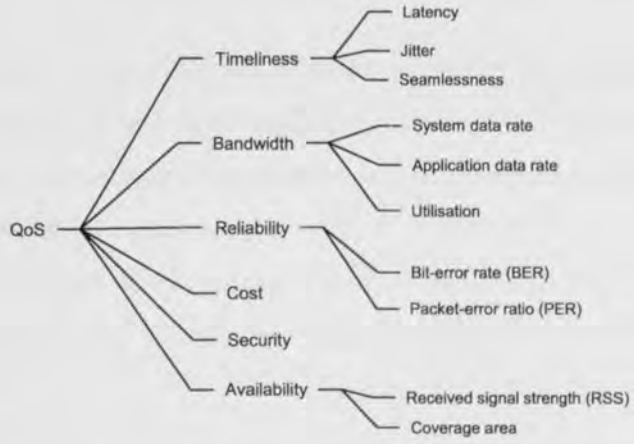


Figure 7.2: QoS criteria in wireless networks (*Adapted from*: Burgess (2003, chap.7))

Each criteria must be defined by parameters to create the membership function. The functions translate measured or static parameters into fuzzy values, used to define the required level of service. Decision criteria are defined for categories of service types. Table 7.3 are some common services types and associated parameters composed from QoS recommendations available in the literature (ITU-T, 2001b).

Category	*Label*	*Unit*	*Target*	*Variance*	*Type*	*Weight*
Conversational voice	Throughput	kbps	64	10	HB	medium
	Latency	secs	0.3	0.2	LB	high
	Jitter	secs	20	30	LB	high
	Packet loss	%	1	2	LB	high
File transfer	Throughput	kbps	1000	8000	HB	medium
	Latency	secs	1	4	LB	low

Table 7.3: QoS categories requirements as criteria parameters

Membership functions in the FDM module define a transition region of set membership; where the set is the criteria (see, 6.6.1 on page 99). Therefore, the variance value in table 7.3 is used to define the slope and function shape. Membership functions for initial testing use triangular and trapezoidal shapes, though other types can be used, such as Gaussian curves. For simplicity trapezoidal functions are defined for types of preference: lower-better (LB), and higher-better (HB). Each of these criteria are grouped into specifications for QoS classes, such as conversational voice and background data.

7.3 Decision-Making Analysis

As part of the proposed solution, the decision-making module determines the suitability of alternative options from multiple input measures (criteria) when combined using a decision function. The type of decision function and criteria weights changes the trade-offs when criteria are combined (aggregated) to generate rank-scores of alternatives. It is the rank-score which is used to select the suitable alternative based on the criteria: high scores represent a closer match. The following sections describe the evaluation approach for decision functions using sensitivity analysis.

7.3.1 Sensitivity analysis of decision functions

The sensitivity analysis method show interaction of multiple independent variables and effect on the dependent variable. In this problem context, the decision function is the subject with independent variables the criteria measures or weights and the dependent variable rank-score. The rank-score is the decision function output in the range [0, 1]. Tests involve different decision function and parameters to answer the question: how does the criteria and weights affect the rank-score?

Decision functions were selected that provide a compensatory or trade-off effect. Table 7.4 shows the functions which are used in the test subjects. All functions use fuzzified inputs in the interval [0, 1]. The generic functions F-GenO and F-GenP use the parameter γ to calculate outputs. To evaluate the effect of this parameter, the sensitivity analysis method by Kaymak & van Nauta Lemke (1998) is used. A vector of γ values is compared against weights and alternative inputs values. The tests for decision-function analysis uses input data (performance metrics) and normalised weights to calculate rank-scores for alternatives (interface options).

Label	*Name*	*Decision function*
F-WSM	Weighted sums	$D^w(\mu_1, \ldots, \mu_n) = \sum_{j=1}^{n} w_j \mu_j$
F-Geo	Geometric	$D(\mu_1, \ldots, \mu_n) = \prod_{j=1}^{n} {\mu_j}^{1/n}$
F-WGeo	Weighted geometric	$D^w(\mu_1, \ldots, \mu_n) = \prod_{j=1}^{n} {w_j \mu_j}^{1/n}$
F-GenO, F-GenP	Weighted generic	$D^w_\gamma(\mu_1, \ldots, \mu_n) = \left\{ \sum_{j=1}^{n} w_j \mu_j^\gamma \right\}^{1/\gamma}$

Table 7.4: Decision-functions of test subjects for aggregating fuzzy inputs.

7.3.2 Experiments

A series of tests were used to analyse characteristics of the decision module, with the test subjects (see section 7.1.2) as fuzzy decision-making functions. The following descriptions define each test case and their objectives. Overall, these experiments aim to compare parameters in criteria aggregation algorithms and select a reasonable option for use in the HAL model.

Test case A1

This case aims to assess the generic decision-function parameter for values of γ. Using techniques of Kaymak & van Nauta Lemke (1998), negative and positive values of γ are used to show the effect on output variable given an input matrix and criteria weights. Two experiments are defined:

- A1.1: one alternative is used with two criteria; weights are varied for each criteria.
- A1.2: two alternatives - one better, one worse - for two criteria; weights are varied for each criteria.

Test case A2

Surface mapping used two input criteria against a output variable: *rank-score*. This is a measure of how well the alternative matches the criteria, in the range [0, 1]; where one being fully met (suitable), and zero is unsuitable. An initial test of decision component uses two input vectors

to generate a surface map of response values. Each input is a vector of values, which are then input to each decision-function. The result is a combined matrix of values that correspond to each value in the input vector.

Surface mapping plots value ranges of two inputs against the output variable. Throughput (kbps) and latency (ms) were used as criteria for a voice application. Input values were set using arrays of values, creating the decision matrix for a single alternative option (value ranges in brackets):

Alt.	Throughput (C_1)	Latency (C_2)
A_1	[200, 500]	[100, 500]

Experiments were generated by using different weights (table 7.5). In the first experiment, weights are set as equal. Experiment 2 used unequal weights defined by linguistic terms. Finally, weight pairs were varied at regular intervals, producing results for combinations of weights pairs for each criteria. This results in a total of 35 data sets (5 subjects $\times$ 7 weight pairs).

Experiment	C_1	C_2
1	w = 1	w = 1
2	w = 'medium'	w = 'high'
3	w = (0.1, 0.3, 0.5, 0.7, 0.9)	w = (0.9, 0.7, 0.5, 0.3, 0.1)

Table 7.5: Experiment weight settings for test case A2

Test case A3

For more than two input variables, sensitivity analysis shows variations in output variable. Using a similar approach to Zhang (2004), multiple input parameters were set for different alternatives, then input weights are varied to produce a change in the output variable. Plotting the independent variable (weights) against output variable (suitability), illustrates the effects for each decision-function for alternatives. Sensitivity analysis was used to assess the effect of changes in a particular variable on the ranking value. Inputs were defined using two criteria for four alternatives:

Alt.	Throughput (C_1)	Latency (C_2)
A_1	64	50
A_2	50	400
A_3	64	400
A_4	60	50

Alternatives option A_1 and A_2 of the decision matrix are best and worst criteria, respectively. Option A_3 is poor for C_1, but good for C_2. Whereas A_4 is good for C_1, but poor for C_2. Using these settings, experiments are defined as:

- A3.1: decision outputs using equal criteria weight.
- A3.2: decision outputs using variation in criteria weight.
- A3.3: Randomised inputs for equal weights

7.4 Agent Program for Simulation

This section describes a SimPy program to model HAL component interactions with a simulated environment. A pilot model is used initially to create a rudimentary environment based on matrices of QoS variables, followed by a trace-based model to use variables input from external simulation trace-files.

A collection of processes are defined for an initial model (figure 7.3). The **environment** is modelled as a SimPy Process object that stores performance statistics from wireless interfaces, modelled as a **link** object. Link objects are initialised and stored in a **client** process object. The client contains the initialised HAL (or other agent prototype). Each link object queries the environment object for current statistics, and updates the client agent. At regular intervals the client increments the agent object by calling the step method.

The client also contains methods that are passed to the agent for command processing. Initialising the client, these methods are passed to the agent, which can be called if the logic invokes a matching command. Methods include changing environment variables, such as link selection or other link command.

Wireless scenarios from NS-2 generate trace files for each interface statistic (figure 7.4). Variables are indexed by time; so a trace file of events, such as changes in parameters are updated by a SimPy process from the time index. A SimPy method reads each line and adds

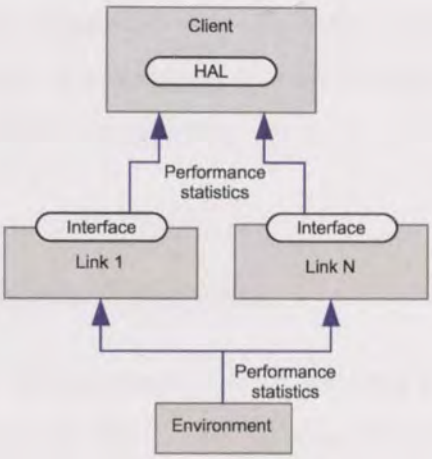


Figure 7.3: Block diagram of SimPy simulation program

an event to the scheduler that should be run at that time in the simulation; in this case updating some variables. The link processes in the SimPy program calls a function (at a regular time interval) to update the client process on link statistics.

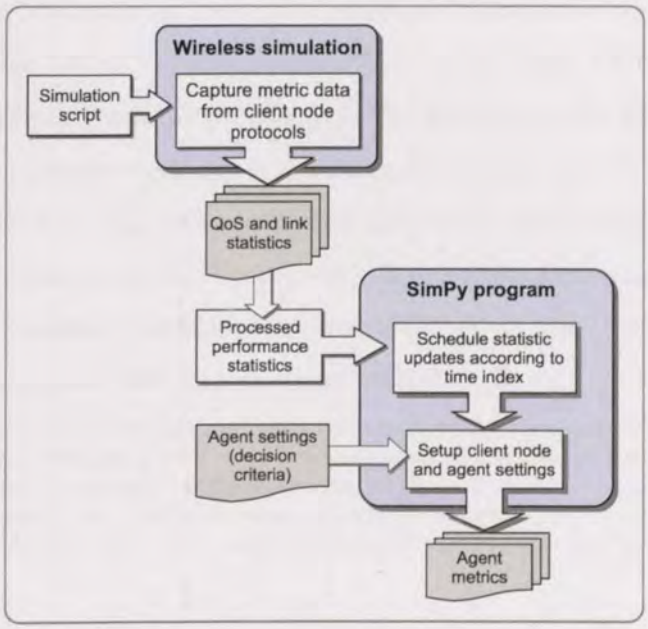


Figure 7.4: Trace-based data and simulation output procedure.

Using this approach any program or simulator can be used to generate the traces, whilst the SimPy program manages the interaction of the agent components. Data from empirical testing could also be processed by the SimPy program. This could be used for verifying the simulation data, by providing realistic changes in input statistics. The limitations in this approach are that client actions, such as causing handovers, may affect other statistics (handover delay).

The preferred solution would integrate the agent code into the network simulator, and provide simulation-time control actions. Though the aim of initial testing is to optimise the behaviour of the agent configuration, further integration and production issues are beyond the scope of this thesis.

7.5 Models for Wireless Simulation

Detailed representations of wireless properties are provided by purpose-built network simulation. Third-party extensions for NS-2 include wireless and mobility protocols, and large community base for support. A third-party tool relevant to this study, *Multi InteRfAce Cross Layer Extension* (MIRACLE) for NS-2 (Baldo et al., 2007), provides cross-layer messaging and enhanced modularity. The package supports dynamic libraries[2], modular network stacks, and enhanced models for propagation.

Wireless simulations for UMTS and WLAN use modified sample scripts from MIRACLE (version 1.2.1). The NS-2 (version 2.31) is built on Ubuntu Linux (server version 7.10), running as a virtualised guest OS on VMware server (version 1.0.6). Table 7.6 details the network parameters used (further details in Appendix C.2). The physical models used are based on the MIRACLE extension described in Baldo et al. (2007). For propagation, the MIRACLE physical layer calculates path loss using Hata model along with Jakes' model (Jakes, 1974) for fading calculations. The Baldo et al. (2007) paper also details interference method calculated from SINR curves, and a multirate modulation scheme for 802.11b/g. UMTS models are modified Eurane[3] code and based-on 3GPP Release 4 architecture[4].

[2]Dynamic libraries in NS-2 allow other third-party extensions to be loaded in simulation scripts in a modular fashion, without the need to recompile NS-2 base code. Though they must be patched to support dynamic loading according to `http://www.dei.unipd.it/~baldo/ns_dl_patch/`.

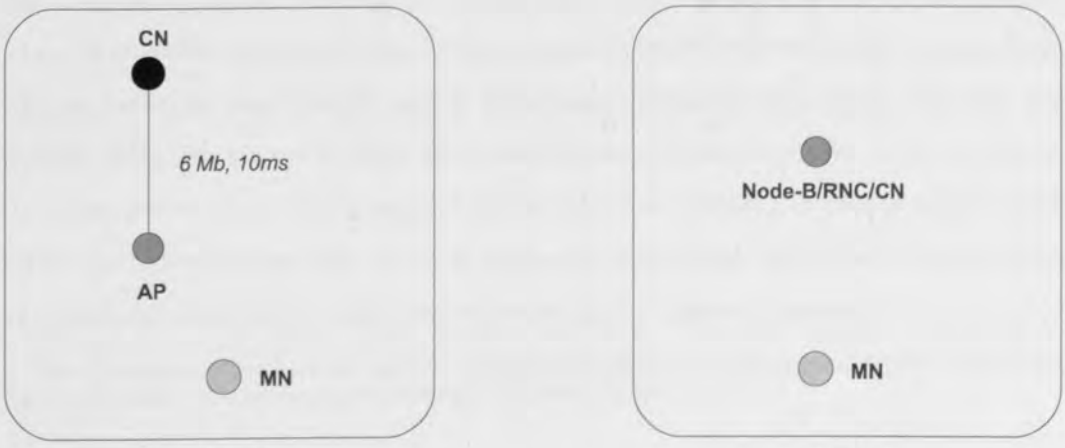
[3]Eurane UMTS models for NS-2: `http://eurane.ti-wmc.nl/eurane/`

[4]According to: `https://mail.dei.unipd.it/pipermail/nsmiracle-users/2008-July/000187.html`

Parameter	*WLAN sim 1*	*UMTS sim 1*
Terrain	1km × 1km	1km × 1km
Number of nodes	1	1
Node placement	MN (100, 100)	MN (100, 100)
Frequency	2.437-GHz	
Transmit power	15 dBm	21 dBm
Propagation model	MIRACLE two-ray ground	FreeSpace
MAC	802_11g (multirate)	UMTS/MAC/ME
MAC bit rate	6 Mbps	2 Mbps
Mobility model	Deterministic	Deterministic

Table 7.6: Network parameters for NS-2 simulations

Each simulation defines access points (AP) for 802.11g WLAN and base station for UMTS, application workloads, and a roaming scenario. QoS performance and signal metrics is recorded by the mobile host (MH). Figure 7.5 shows the node topologies for WLAN and UMTS simulations. The WLAN AP is connected to a correspondent node (CN) by a fixed-line link. For simplicity, the UMTS Node-B, core network, and CN objects are combined (though these could be expanded to separate nodes for more complex topologies, such as competing nodes and background traffic).



(a) WLAN (b) UMTS

Figure 7.5: Node topologies for WLAN and UMTS.

7.5.1 Traffic parameters

Simulations require traffic load specifications for client devices, defined as workloads. Service types for workload activity include: voice, video and non-real-time data, such as file transfer or web-browsing. Workloads define a pattern of activity. For instance, user requests a voice service, which then begins an application sending packets across the network.

Voice traffic in simulations are based on the G.711 protocol using UDP and constant bit-rate (CBR) of 50 packets per second. Based on packets of 180 bytes (includes application headers), this gives an application data-rate of 64 kbps. For simplicity, there is no silence-suppression (available in other voice protocols) or header compression; giving a constant stream of packets, hence CBR is appropriate. Further details of application settings is given in Appendix C.2.

Metrics

During and after simulations, procedures are scheduled to record performance statistics of the MH. QoS metrics are collected by calling procedures in the simulation script. The procedure takes the mean of samples from received packets at one second time intervals. Simulation trace-files are post-processed where required metric data is not recorded during simulation time. QoS metrics are recorded and processed to provide separate files with observations indexed by the second.

Where measurements are calculated on a packet-by-packet basis, an average or smoothing function is used. This is required to reduce the effect of signal fluctuations in data calculated from individual packets or frames; for example the corner effect of WLAN signal changes (Zhang et al., 2003). Post-processing scripts are used to find the received signal strength (RSS) using an *averaging filter*[5], which outputs the average of samples in a second. For jitter and one-way delay, the average is taken from instantaneous measurements for received packets. The instantaneous jitter, or interarrival delay is calculated according to method used by RTP in RFC 3550 (Schulzrinne et al., 2003) which uses an exponential filter. Further details on the collection procedures and trace-file parsing scripts can be found in Appendix C.2.

[5]Though other smoothing functions could be used, such as exponential moving-average or Savitzky-Golay filters (Savitzky & Golay, 1964) implementations provided in Press et al. (1992, p.650).

7.5.2 Roaming scenarios

In a handover assessment study, changes in network conditions and points of handover are of particular interest. This study uses roaming scenarios (figure 7.6) to initiate changes in network conditions, such that some links or networks can become better or worse over the course of the simulation.

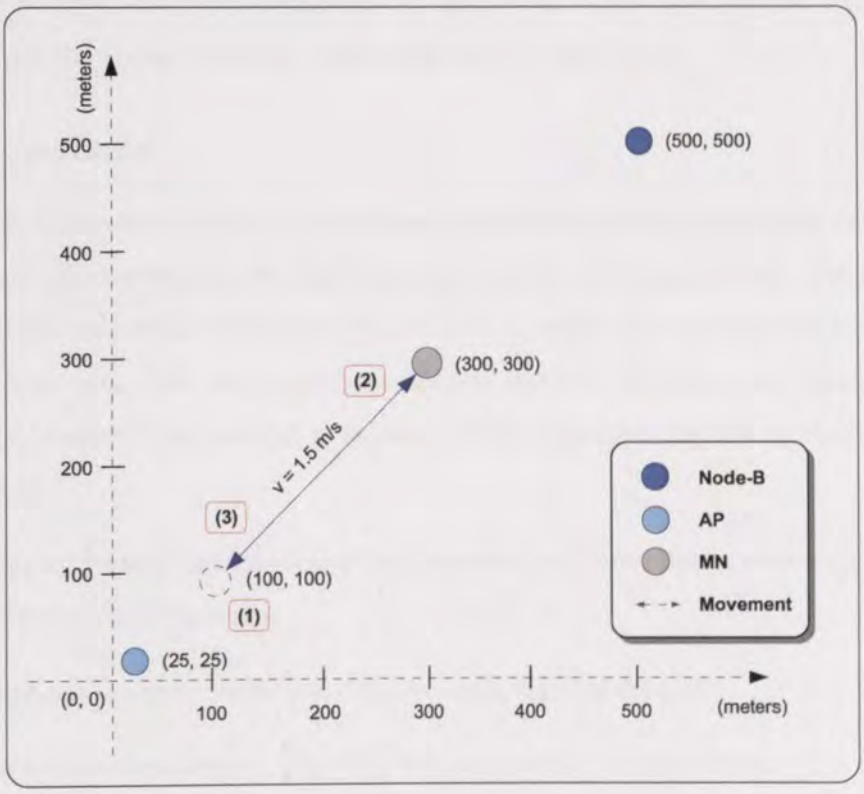


Figure 7.6: Roaming scenarios.

1. **Stationary**. The MH is at a fixed point within range of at least two access networks (WLAN and UMTS). Other factors can be varied such as the number of competing nodes or application traffic.

2. **Moving-out**. The MH starts within range of a WLAN hotspot, then proceeds to move away from the AP at velocity (v) slightly faster than average walking speed of 1.5 m/s, or 5.4 kph.

3. **Moving-in**. The MH starts within range of a cellular BS only. The MH then moves towards the WLAN hotspot provided by AP at the same velocity in 2.

Of the three scenarios, the stationary scenario provides a benchmark for other roaming scenarios. This gives six sets of results: three for WLAN and three for UMTS simulations. The overall aim of these simulations provides QoS and interface performance metrics that HAL can use to make handover decisions. The moving-out and moving-in creates a transition region where the WLAN becomes less preferable and more preferable, respectively. Although there are many possible scenarios for roaming and application usage, these provide a reasonable starting point for making selections when conditions of a link change.

7.5.3 Experiments

After running simulations, traces for performance metrics are processed and read into the agent mobile host client defined in the SimPy program (section 7.4 on page 116). This program defines factors for experiments using: decision criteria, scenes, and prototype models. Experiment settings using these factors provides different inputs to the prototypes, shown in table 7.7. The experiments are arranged to evaluate wireless interface selection by the prototypes according to:

- Roaming pattern: the movement and transition between different network types, and dynamic QoS performance.
- Number of decision criteria and different application requirements.

Test cases use two experiments: one with only one criteria; and two criteria. The traffic requirements are defined in Appendix C.2.

Test case	*Scene*	*Experiment*	*Traffic criteria (target values)*	
			Throughput	Latency
S2.1	1	1	64 kbps	-
		2	64 kbps	300ms
S2.2	2	1	64 kbps	-
		2	64 kbps	300ms
S2.3	3	1	64 kbps	-
		2	64 kbps	300ms

Table 7.7: Summary of test cases for trace-based simulation: S2.

Concluding Remarks

Using the HAL prototype from Chapter 6, an approach for performance evaluation was proposed. Test subjects are divided between analytical and simulation evaluation. Decision-making components are used as "subjects" for sensitivity analysis of output variables. The simulation approach uses inputs metrics generated by NS-2 simulation data. Subjects for simulation are configurations of HAL: decision-making only, and decision-making plus finite-state machines.

Experimental results for these tests were analysed and compared in the following chapter. Test cases and experiments provide data for indicators based on varying factor levels. The analytical tests aim to give insight into decision parameters, and simulation to how the framework behaves within a wireless environment.

Chapter 8

Performance of Handover Agent Layer

The agent framework in Chapter 6 defines wireless interface selection for QoS criteria and metrics using decision-making techniques and finite state machine logic. Performance metrics are used to compare interfaces and generate suitability rank-scores. Rank-scores generated by the decision module were evaluated using an analytical approach. The chosen decision settings were used in the simulation test cases to compare different prototypes of agent logic.

8.1 Analytical Tests

Analysis of the decision-making module was necessary to evaluate aggregation output. Results from these tests were used to determine decision module settings for simulation experiments. Analytical tests were performed on the decision module described in Chapter 6 (section 6.6 on page 98) using different criteria aggregation functions (table 8.1). These functions were compared using sensitivity analysis techniques to compare rank-scores of alternative options (response variable) for input criteria.

Prototype	*Description*
F-WSM	Based on the weighted-sum model (WSM): fuzzy variation.
F-Geo	Geometric mean fuzzy decision making (FDM) function.
F-WGeo	Weighted geometric mean fuzzy decision making (FDM) function.
F-GenO	Optimistic version of generic-FDM with optimism parameter, γ.
F-GenP	Pessimistic version of generic-FDM with optimism parameter, γ.

Table 8.1: Selected decision-making prototypes for analytical study.

8.1.1 Generic function with optimism parameters (A1)

The generic operator uses a variable, γ as a parameter that changes trade-off direction between criteria; also known as the "index of optimism" (Sousa & Kaymak, 2002, p.46). Using sensitivity techniques of Kaymak & van Nauta Lemke (1998), the effect of γ values on rank-scores was shown. Figure 8.1 shows data from test A1.1. For a set of inputs (2 criteria, 1 alternative), lines *a* to *e* are combinations of weights; alternating between each criteria. The overall pattern is for rank-scores to increase towards 1.0 when γ is set towards ∞.

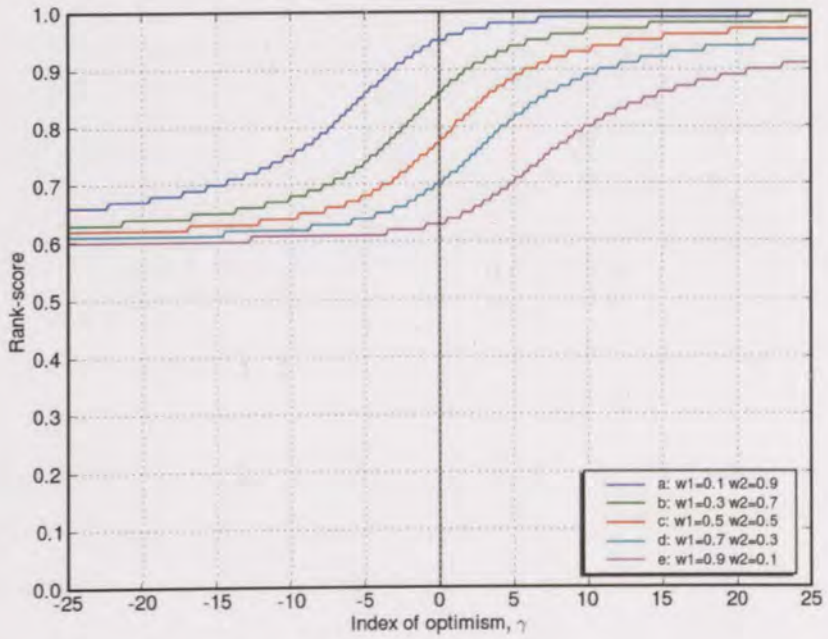


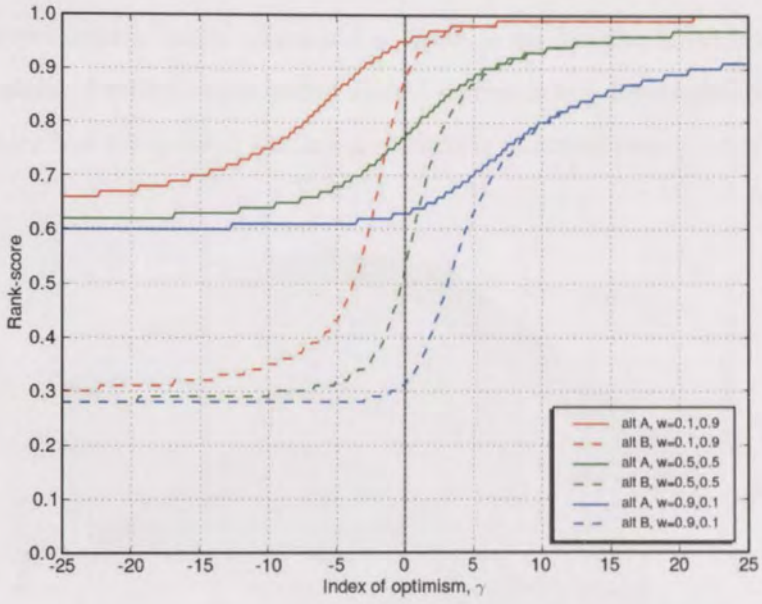
Figure 8.1: Test A1.1; rank-scores for index of optimism on criteria with different weights.

A second test (A1.2) was performed by adding a second alternative (see figure 8.2 on the following page). Alternative A has inputs that are more preferable than B. For values of γ in the region $[-\infty, -5]$, option B rank-scores are lower than those for A. Conversely, when weights are set unequally with less importance on the poorer criteria, rank-scores of options A and B become the same as γ tends towards ∞.

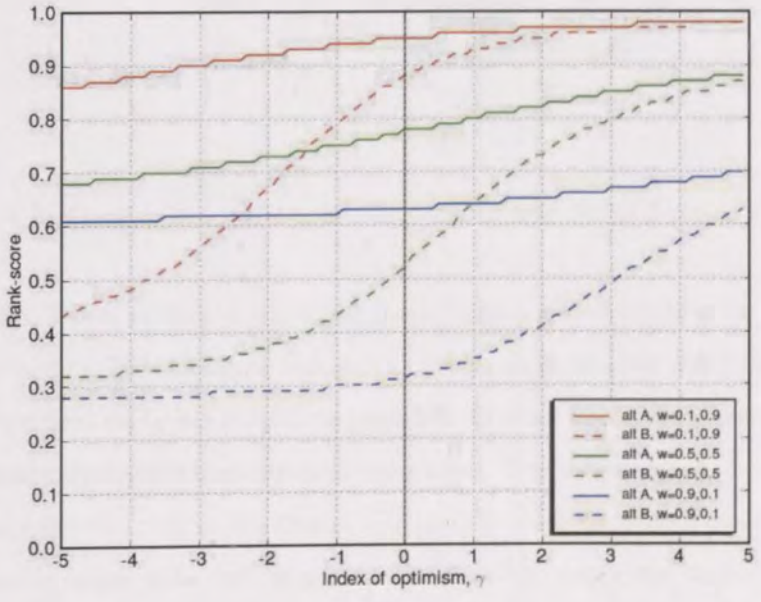
Selecting parameters

For the tests subjects that use the generic decision function, two variations are pessimistic (F-GenP) and optimistic (F-GenO). The γ values determine criteria trade-off and are reflected in the rank-score. Generally, for higher values of γ, the higher the rank-score; conversely, lower γ values reduce the rank-score.

Parameters were selected for F-GenO decision function $\gamma = 2$, and for F-GenP $\gamma = -2$. These values were chosen based on the curves from figure 8.2b, at points were there is separation of rank-scores from good (alt A) and poorer (alt B) options.



(a)

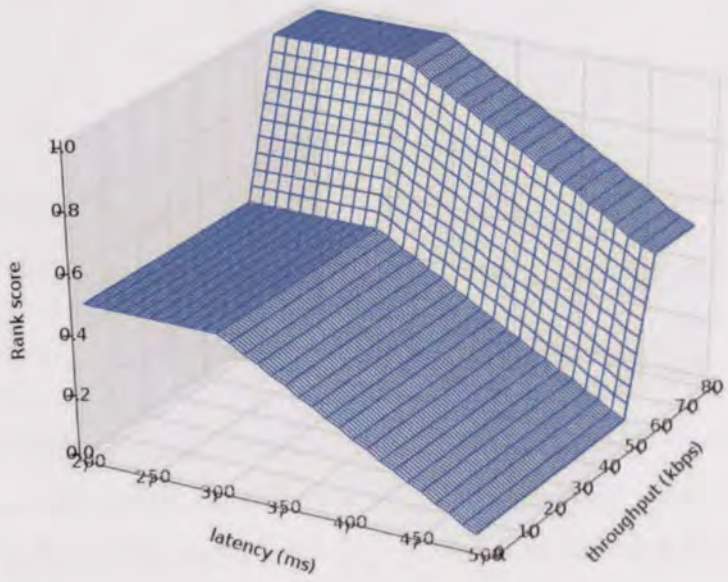


(b)

Figure 8.2: Rank-scores of index of optimism values, using two alternatives: A is best, B is worse. Plot (b) is a close-up of the region [-5, 5] in plot (a).

8.1.2 Surface mapping (A2)

The tests use two criteria (vector sequences) as inputs to the decision function that calculates the output values. Plotting output values against corresponding inputs generates a surface, which visualises how the decision function is combining measured criteria values (figure 8.3).



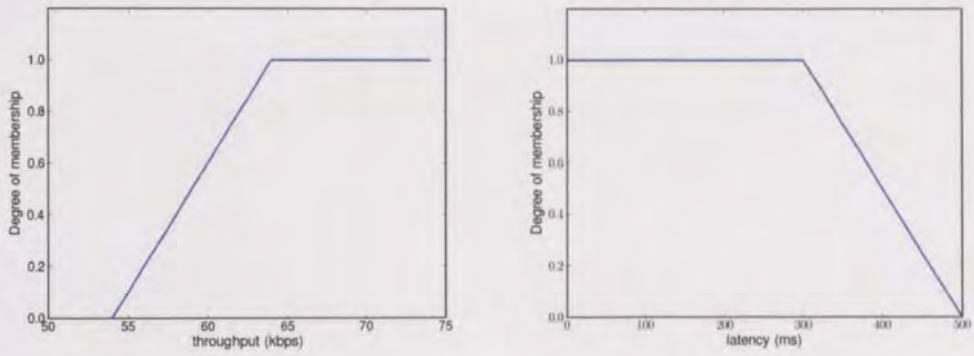
(a) Equal weights

Figure 8.3: Example of a surface plot of outputs using F-WSM decision function.

The intersection of criteria in figure 8.3 (here, latency and throughput) at points on the surface correspond to the calculated rank-score. Scores are calculated using implementaions of decision functions explained in 6.6.2 on page 100. In other words, each point on the z-axis is the calculated output value from inputs (x and y axes). This shows visually how the decision function aggregates two criteria; the change in z value or slope for certain inputs. From these plots, changes in output value can be observed and used to verify that the behaviour of the decision function is appropriate. It is also noted that the shape of the surface is determined by criteria membership functions and weights, which is explained in subsequent tests.

The first test, A2.1 (fig. 8.5 on page 131) compares two variables for each subject and are equal weighted for the two variables: *throughput* and *latency*. These were chosen to illustrate decision functions context of the problem: comparing multiple QoS criteria. Other criteria could be used, such as jitter, packet loss rate, cost, or any available QoS performance metrics

determines requirements of the decision-maker for matching input measures. In A2 tests, the criteria were defined by membership functions for throughput and latency (figure 8.4).



(a) Throughput MF

(b) Latency MF

Figure 8.4: Criteria and options settings

The rank-scores for criteria values can be affected using weights. Unequal weights for criteria has the effect of changing how the criteria value contributes to the final rank-score. In test A2.2 the same variables are used, this time the following variation in weights are used:

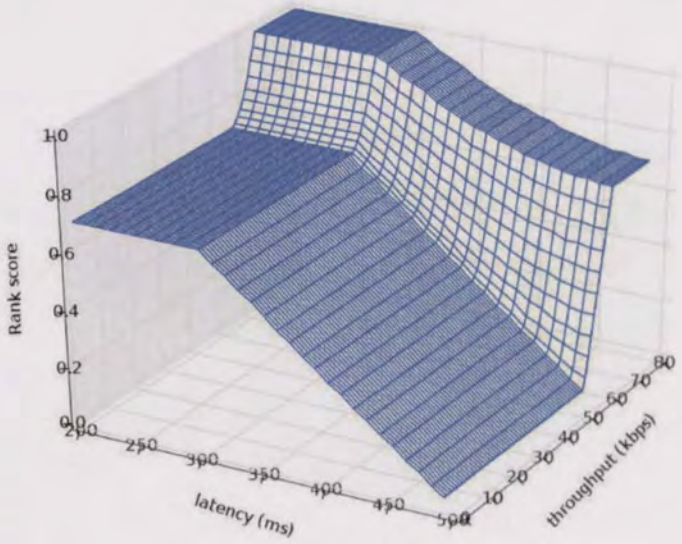
Criteria	*A2.1*	*A2.2*
Latency	1.0	high (0.725)
Throughput	1.0	low (0.225)

Table 8.2: Weights for A2 tests.

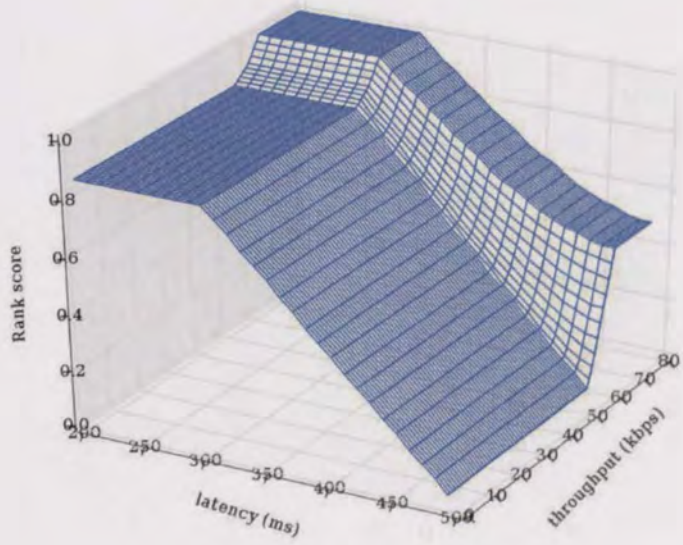
Running the test with weights from table 8.2, produces surface plots to compare against A2.1. For F-GenO (fig. 8.5 on page 131) and F-WSM (fig. 8.6 on page 132), the surface is raised (higher rank-scores) in test A2.2 at points for lower throughput values: between 0-55 kbps. This is because there is less weight attached to the throughput criteria. Also, as the latency criteria is also less than test A2.1, the rank-score is raised when throughput is good and latency is poor.

In tests A2.1 and A2.2 for F-GenP (fig. 8.8 on page 134), the difference is less striking (except for a slight curve in the surface for latency 300-500ms), whereas F-WGeo shows a larger difference (fig. 8.7 on page 133). For test A2.1 (equal weights), the surface pattern is similar to F-GenP: zero rank-score below 55 kbps, and a slope from 1.0 to 0.0 between 300

and 500 ms latency. In the weighted test A2.2 this shape is compressed; giving a maximum rank-score of 0.2 in the satisfactory region.

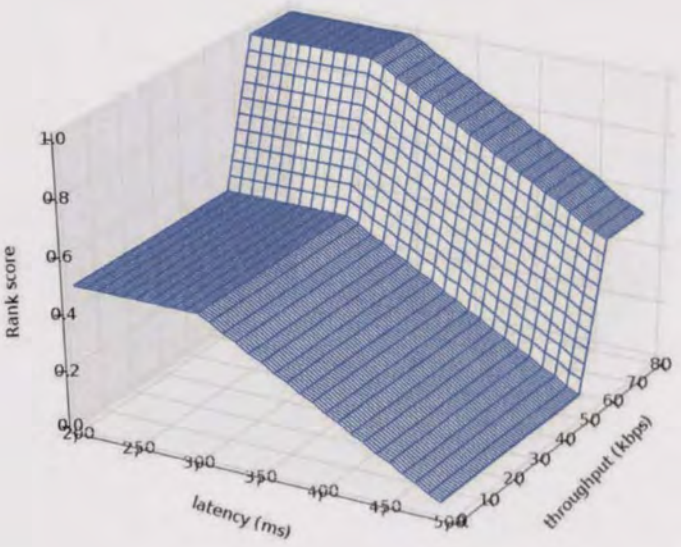


(a) Equal weights

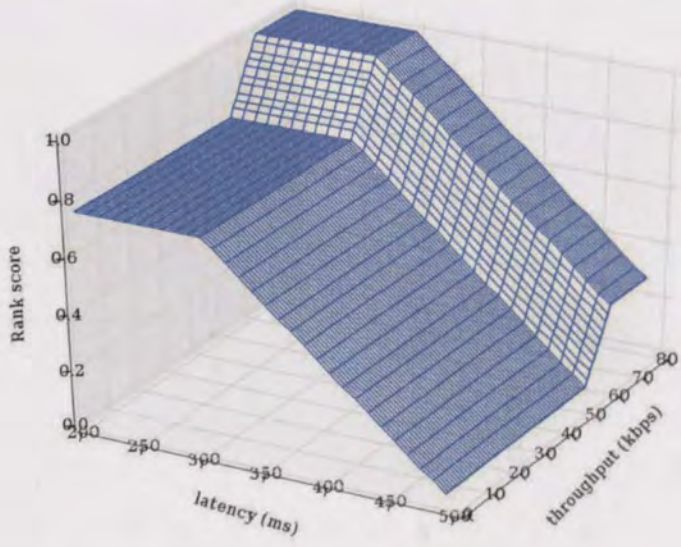


(b) Varied weights

Figure 8.5: F-GenO decision-function output (z-axis) as a surface from vectors of two input criteria (x and y axes).

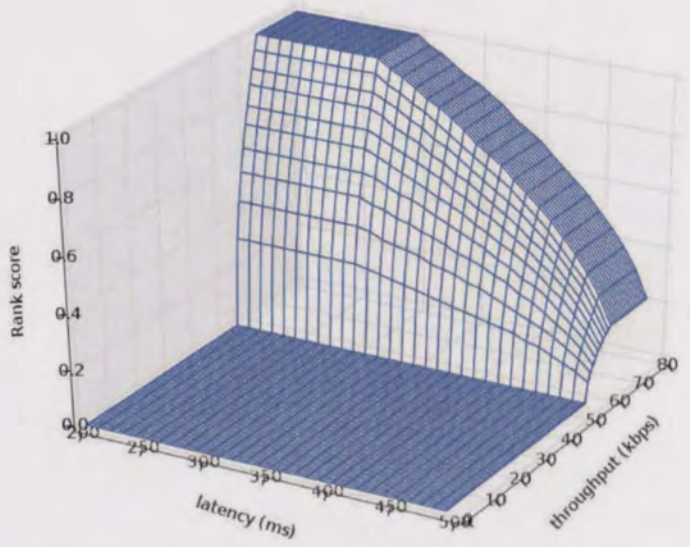


(a) Equal weights

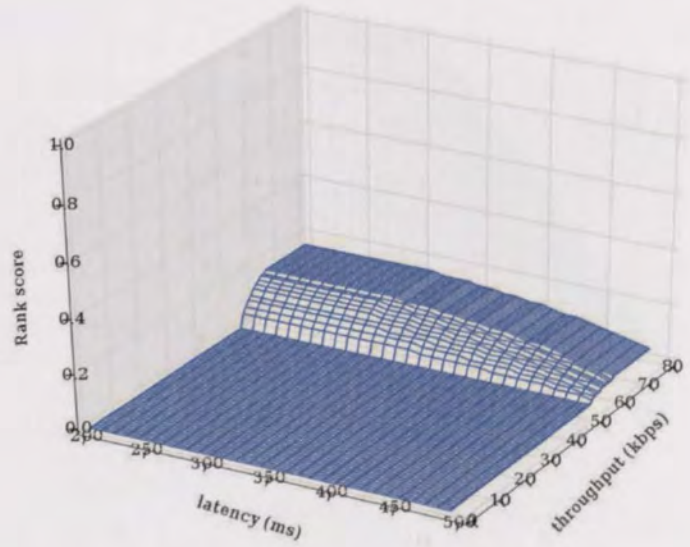


(b) Varied weights

Figure 8.6: F-WSM decision-function output (z-axis) as a surface from vectors of two input criteria (x and y axes).

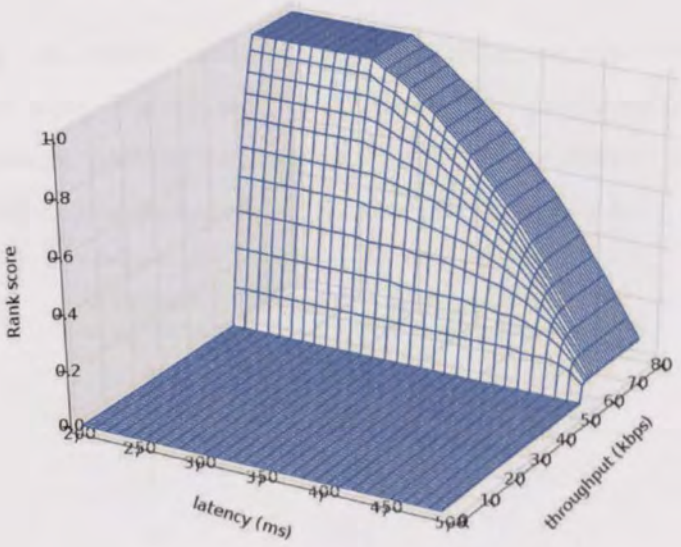


(a) Equal weights

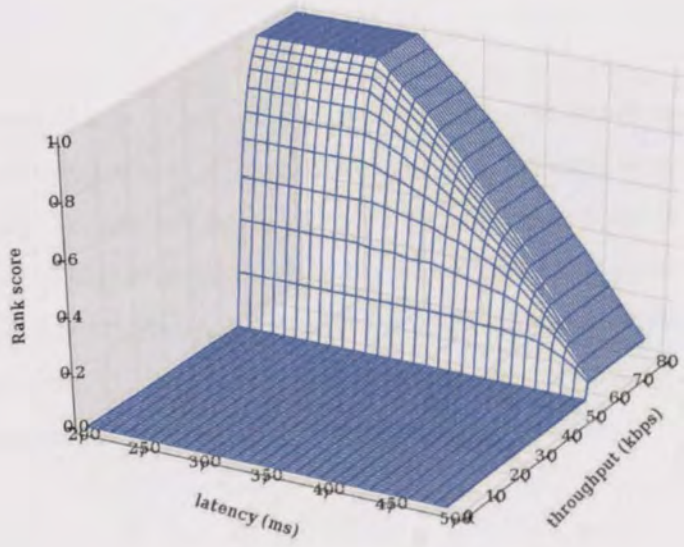


(b) Varied weights

Figure 8.7: F-WGeo decision-function output (z-axis) as a surface from vectors of two input criteria (x and y axes).



(a) Equal weights



(b) Varied weights

Figure 8.8: F-GenP decision-function output (z-axis) as a surface from vectors of two input criteria (x and y axes).

8.1.3 Sensitivity analysis (A3)

In the following tests, criteria values are fixed and weightings are varied to show the weights affect the output score. The first test (A3.1) compares four alternatives (A) for two criteria (throughput, latency). Alternative options are set with data that make A_1 the best option, A_3 and A_4 that satisfies one of the criteria, and A_2 the worst option (table 8.3).

Alternative	*Throughput (kbps)*	*Latency (ms)*
A_1	64 ↑	50 ↑
A_2	50 ↓	400 ↓
A_3	64 ↑	400 ↓
A_4	60 ↓	50 ↑

Table 8.3: Settings for alternatives of A3

The second set of tests (A3.2) uses the same setup, but with variations in criteria weights to show changes in rank-scores. The weight for one criteria (latency) is varied from 0.1 to 1.0 in intervals of 0.1. The example figure 8.9 shows the different rank-scores for alternatives A_n when the latency criteria weight is increased. In this example, A_1 is given higher rank-scores than other for all weights, and A_4 is better than A_3. The differences in rank-score diverge as the weight approaches 1.0, but the rank-score is lower when the weight is less (the geometric mean and weight normalisation could be causing this effect).

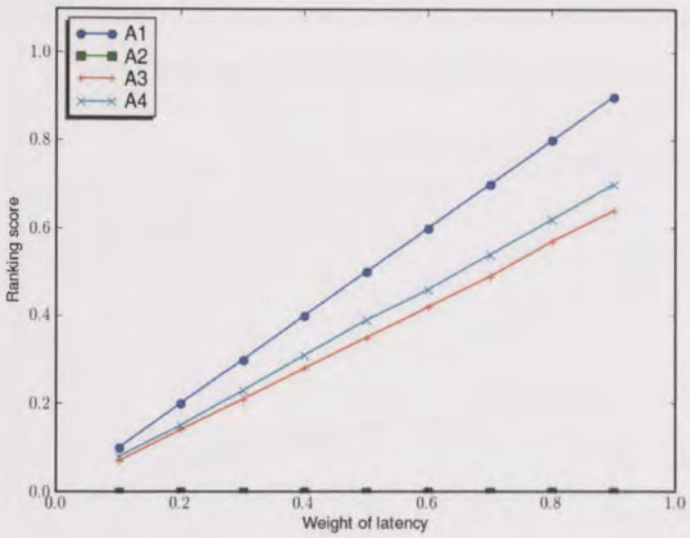
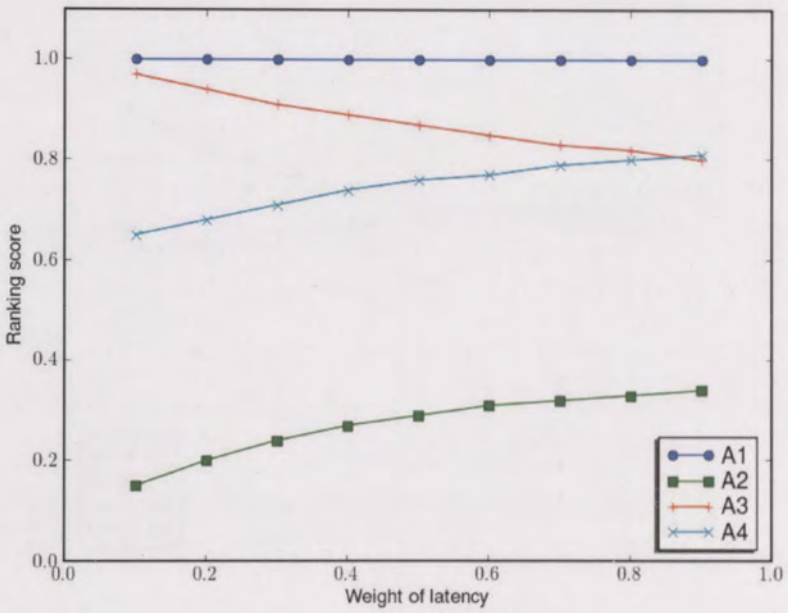


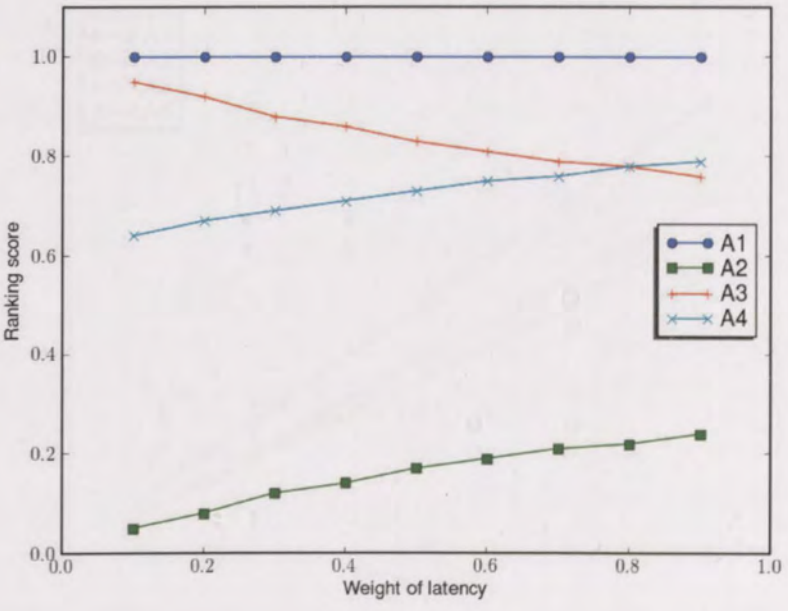
Figure 8.9: Example (F-WGeo) of rank-score sensitivity to criteria weights

The F-GenO and F-WSM functions rank-scores change in a similar pattern (figure 8.10 on the next page). For both functions the worst alternative (A_2) produces a low rank-score, around 0.2. Alternative A_3 has a poor latency value but good throughput, so when latency has a weight of 0.1 the rank-score is close to 1.0. When the importance weight is increased, the rank-score for A_3 decreases to 0.8. Conversely, the rank-score for A_4 increases because it has a better latency value.

Function F-GenP shows a similar pattern, but with lower rank-scores (plot (a), figure 8.11 on page 138). For alternative A_2, rank-score is zero (as is F-WGeo), and A_3 rank-score decreases lower with increasing weight. A different pattern is shown by F-WGeo (plot (b), figure 8.11 on page 138). Instead of decreasing with weight, A_3 increases in rank-score. The rank-score for other alternatives also starts low and increases when weight of latency increases.

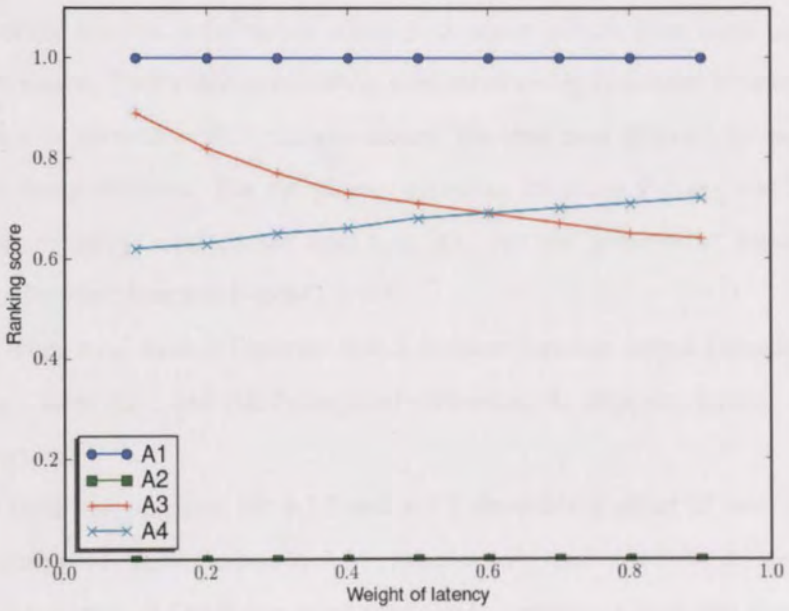


(a) F-GenO

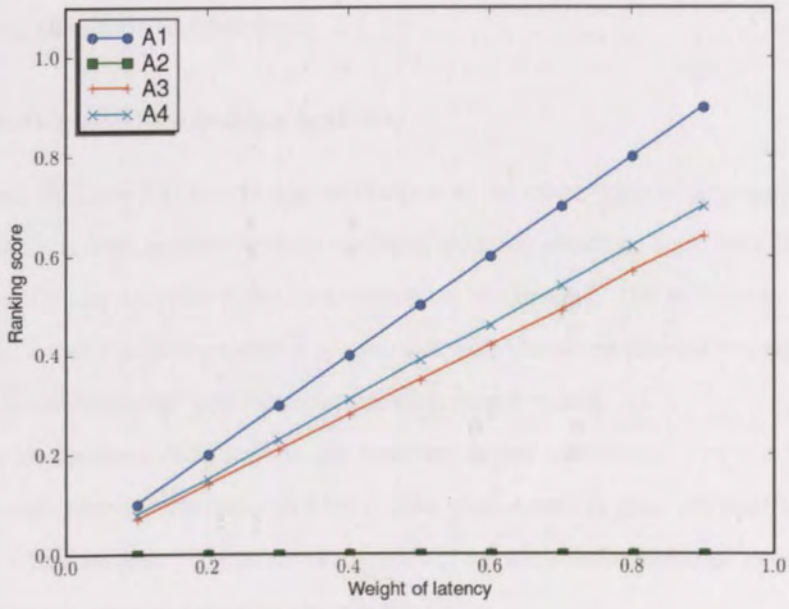


(b) F-WSM

Figure 8.10: Variations in weights for criteria C1 of test A3.2



(a) F-GenP



(b) F-WGeo

Figure 8.11: Tests A3.2 for variations in weights in (a) F-WGeo and (b) F-GenP

8.1.4 Discussion

Decision-making analysis experiments aimed to compare effects from input parameters on options rank-scores. Fuzzy decision-making uses membership functions of criteria to assess suitability against alternative performances values. The tests used different decision-functions to aggregate fuzzy numbers. For the generic averaging functions F-GenP and F-GenO, the optimism parameter (γ) were chosen from tests A1. For the 'pessimistic' function F-GenP, $\gamma = -2$; for optimistic function F-GenO, $\gamma = 2$.

Surface maps were used to illustrate how a decision-function output changes with inputs and weights. Tests A2.1 and A2.2 compared differences in response variable between the decision-functions.

Further sensitivity analysis for A3.2 and A3.3 showed the effect of increasing criteria weight on alternatives rank-scores. In A3.2, functions F-GenO and F-WSM showed similar pattern in rank-scores. F-GenP was more sensitive to weights changes for alternatives with poor criteria. This corresponds with Sousa & Kaymak (2002, chp.3) discussion on averaging functions and γ parameter; they change the trade-off between at least one criteria (disjunctive), and satisfying all criteria (conjunctive).

Generic function provides decision flexibility

As the generic decision function is a generalisation of the other types of aggregation (Sousa & Kaymak, 2002), it adds another level of tuning to decision-making. Tests have shown that by varying the optimism operator γ, the rank values can be changed. This equates to changing the level of risk: lower γ values provide a greater difference between alternatives ranks; higher γ values reduce the difference and increases the rank output values.

Weights also make a difference in the function output rank-score. For γ values above 0, alternatives with poor criteria data can have similar rank-scores to good alternatives, if the poor criteria has a low weight. The effect is a trade-off between importance of good performing criteria, and worse criteria, varied by the γ value.

As these tests only used two criteria, there is scope for further studies using more criteria. However, the current tests are enough to provide an indication of the aggregation functions behaviour in criteria trade-off. The generic function with optimism operator provides finer control over the aggregation process, and thus flexibility for decision-makers. It is possible to use different γ values depending on the situation context, user policy, or service types to change

criteria trade-off effects.

8.2 Wireless Simulations

The wireless simulations used data from NS-2 to drive the responses from prototype models described in the previous chapter. Simulations used three roaming scenarios for WLAN coverage: stationary (scene-1), moving-out (scene-2), and moving-in (scene-3). Metrics were captured based on a regular sampling interval of 1 second. QoS metrics are those measured at the application layer. Signal metrics were extracted from NS-2 trace-files for MAC layer events. The simulation data must be arranged in a format for experiment inputs.

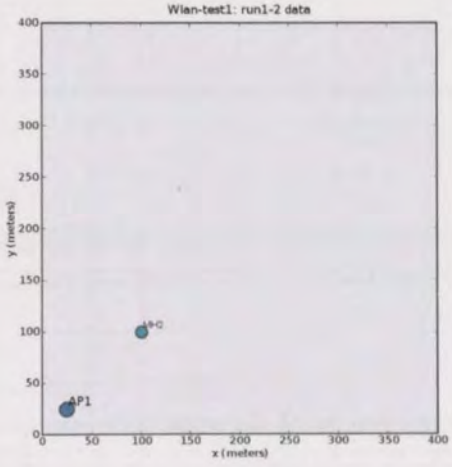
Separate simulations for WLAN and UMTS were run for each scene. Data from the simulation run were used as inputs by the SimPy program containing the agent prototypes (section 7.4 on page 116). The output metrics of the prototypes were captured for each test case experiment. This section reports the results from test specifications using trace-based inputs: set 2 (S2). A test-case defines a combination of the roaming scenario, parameter settings, and experiments. For each experiment, a set of results were generated for each prototype.

Test case	*Experiment*	*Scene*	*Mobile Nodes*	*Metrics used*
S2.1	1	Stationary	1	signal, throughput
	2		1	latency, throughput, signal
S2.2	1	Moving out	1	signal, throughput
	2		1	latency, throughput, signal
S2.3	1	Moving in	1	signal, throughput
	2		1	latency, throughput, signal

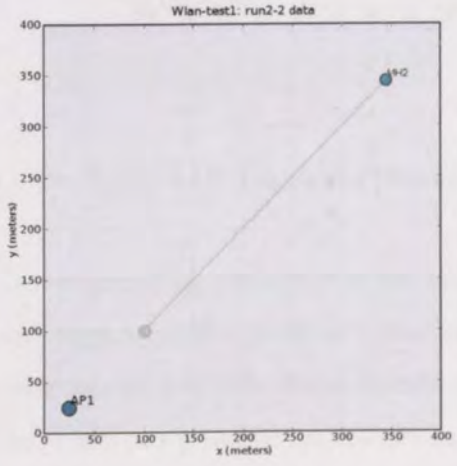
Table 8.4: Summary of experiments

The first test case used a simulated WLAN and UMTS wireless nodes in a non-roaming scene. Figure 8.12a shows the environment topology generated from WLAN and UMTS simulations (scene-1). Horizontal and vertical axis are the position in meters. In this figure, the WLAN access point (labelled AP1) is placed at position (25, 25), and the mobile host (labelled MH2) is at (100, 100). The correspondent host for the MH traffic is present in the simulation but not shown on the diagram as position is irrelevant. Also, the UMTS base station is placed at

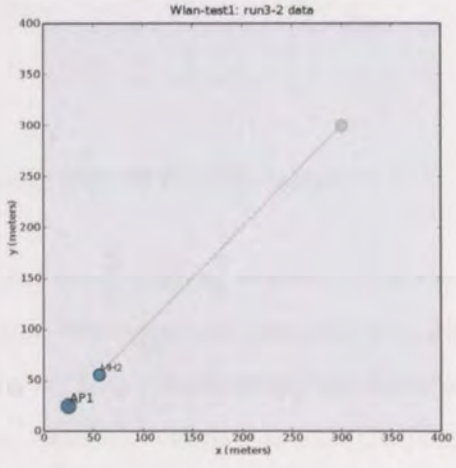
the centre at (500, 500) to cover the simulation region. The scenes for moving-out and moving-in are shown in figures 8.12b and 8.12c respectively. In these figures, the MH start position is indicated by the grey circle and moves according to the roaming pattern until the end of the simulation.



(a) S2.1 (stationary)



(b) S2.2 (moving-out)

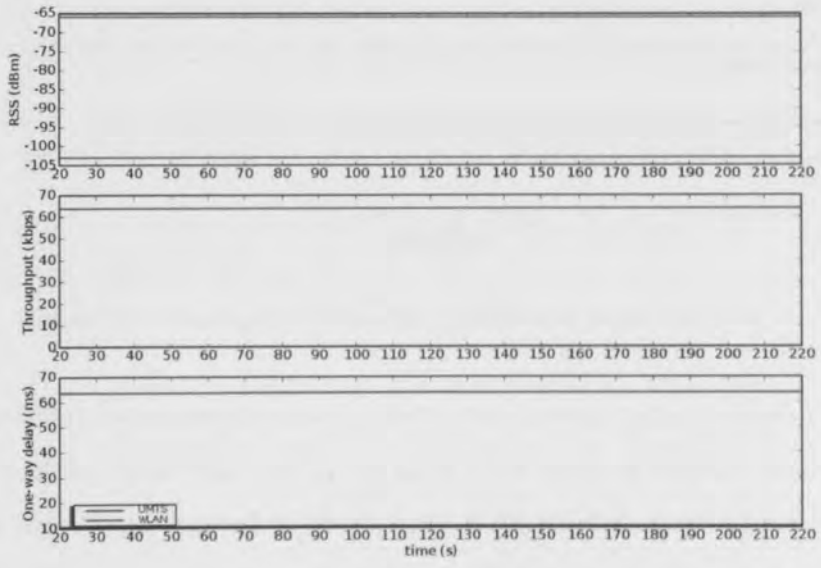


(c) S2.3 (moving-in)

Figure 8.12: Scenes for test S2

In experiment 1, only signal data is used for connectivity and throughput is used for decision criteria. The second experiment uses throughput and latency data for handover judgement. Figure 8.13 shows the time-series trace plots of these inputs. The horizontal axis is the duration of the simulation in seconds, which starts after a 20 seconds 'warm-up' period (data before

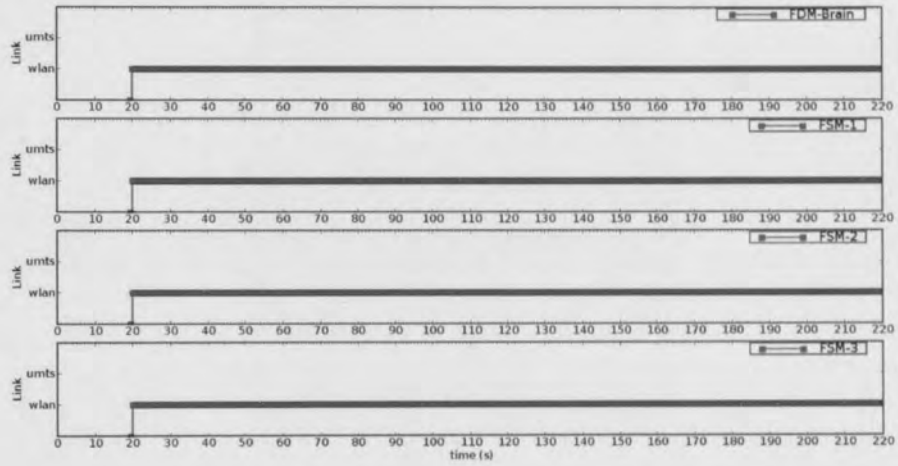
this point are not used in experiments). Respective vertical axis are the metric from received packets at the mobile host; in this case, signal in dBm, throughput at the application (more accurately, goodput), and one-way delay (milliseconds) of packets from the correspondent host. There is little change in the values during the simulation since the MH stays in range of AP at all times and there is no movement.



(a) WLAN

Figure 8.13: Time-series plots of S2.1 metric data for WLAN simulation.

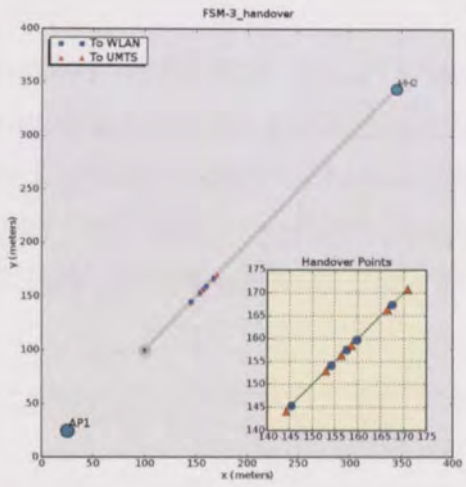
An output of the prototypes is link choice during the simulation. Figure 8.14 is a time-series comparison of a prototype's interface selection. These plots are generated from traces by sampling the prototype choice variable every second. This value indicates the current state of the prototype's preferred link and not an an actual handover initiation, as other components (FSM or behaviour logic) may determine handover timing.



(a) UMTS

Figure 8.14: Example of time-series plot for link choice (case S2.1).

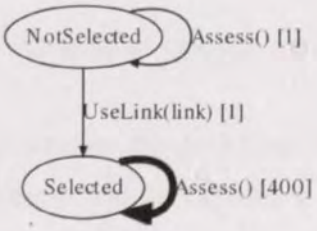
The actual handover execution is simplified by the prototype calling a function during the SimPy simulation. These were recorded to show actual points of interface handover points initiated by a prototype. Figure 8.15 shows a plot of the MH trajectory with handover points indicated along the path with an inset plot of the transition region. From the start position (100, 100) an initial connection is made to WLAN (blue circle). As MH2 moves away from AP1, there are handover points indicated in the inset plot, firstly to UMTS after reaching (145, 145). This example plot shows subsequent handover indicators along the trajectory of the MH.



(a) Experiment 1

Figure 8.15: Example of handover plot with inset of transition region.

For prototypes that use FSMs, a further level of detail shows how many times events cause transitions to new states. Figure 8.16 shows an example of the FSMs for S2.1 generated from trace data. Each state is connected by transitions which have a label and how many times it was called in brackets. The thickness of the line also indicates the frequency of the transitions. In subsequent test case observations, these are used to explain the more significant differences between FSM prototypes.



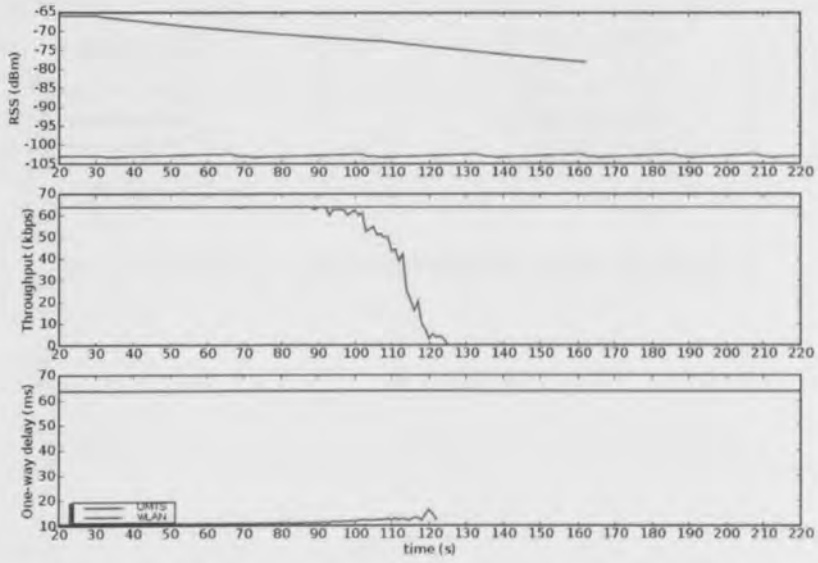
(a) Experiement 1

Figure 8.16: FSM-1 trace with transition counts for S2.1 test (experiment 1).

In the following sections, further results from the roaming scenarios are explained in the format of these examples. Finally, a discussion is provided on some of the key observations.

8.2.1 Test case S2.2

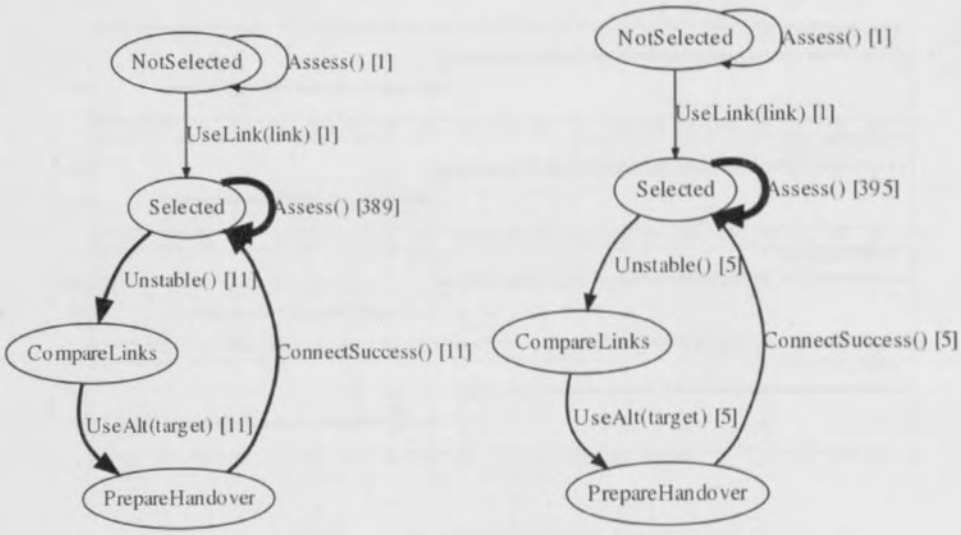
In test case S2.2, the MH with HAL moves out of a WLAN coverage. Moving at an average walking speed, MH signal degrades as shown by the RSS plot in figure 8.17. After ~100 secs the signal becomes too low to sustain the QoS for the voice traffic as shown by decreasing throughput and one-way delay plots. After 125 seconds the throughput drops to zero after a period of errors at the MAC layer. UMTS metrics do not change during roaming.



(a) UMTS

Figure 8.17: Time-series plots of inputs for S2.2.

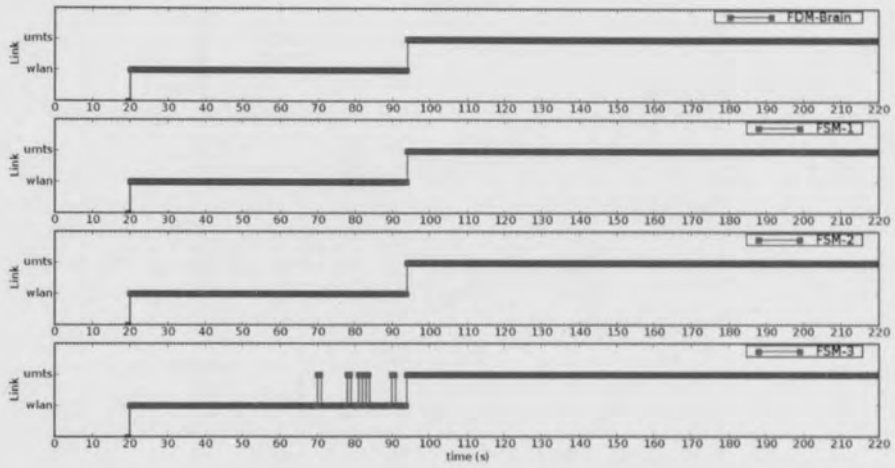
The choice plot for S2.2 experiments are shown in figure 8.19 on page 147. All prototypes except for FSM-3 select UMTS after 69 secs for experiment 1 and 75 secs for experiment 2. Selection for FSM-3 selects UMTS in both experiments and performs handover actions the same number of times. For experiment 1 with just one QoS criteria (throughput) there were 11 handovers, though for experiment two there were 5. This is shown by `ConnectSuccess` transition from `PrepareHandover` state in FSM-3 traces (figure 8.18). Handover points for FSM-3 are indicted in figure 8.20 on page 148.



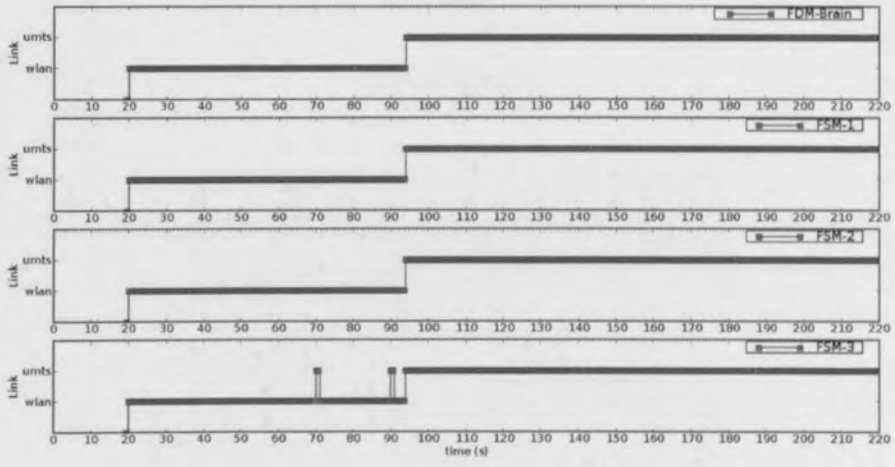
(a) Experiement 1

(b) Experiement 2

Figure 8.18: FSM-3 trace with transition counts for S2.2 test.

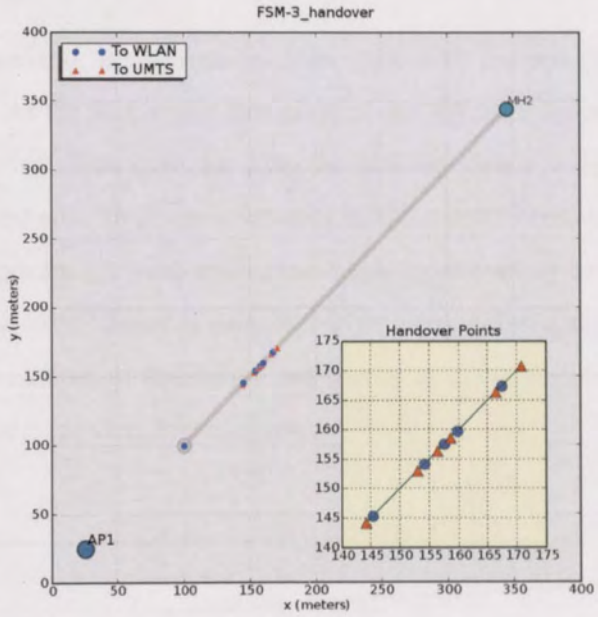


(a) Experiment 1

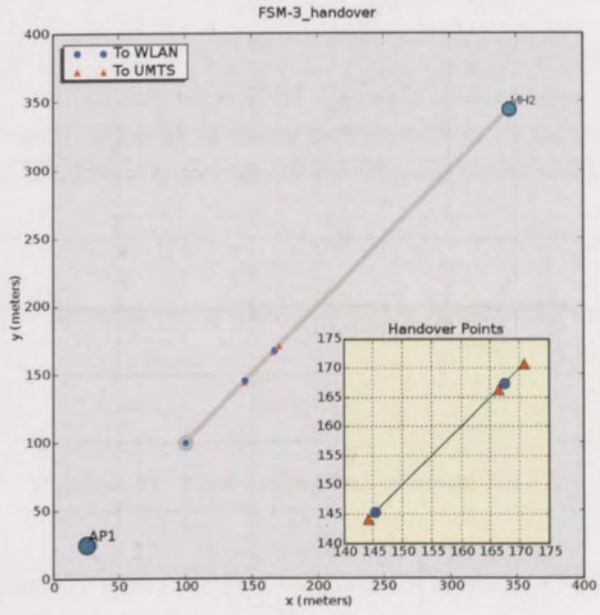


(b) Experiment 2

Figure 8.19: Time-series plots of prototypes link selection for S2.2



(a) Experiment 1

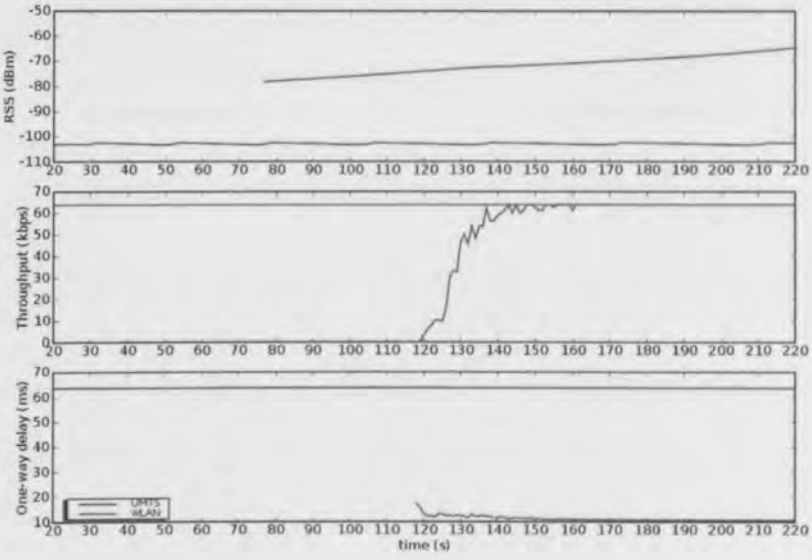


(b) Experiment 2

Figure 8.20: Handover plots of FSM-3 handover for S2.2

8.2.2 Test case S2.3

In this moving-in scenario, the MH moves from (300, 300) towards (100, 100) at a speed of ~1.5 meters/sec. As the MH moves into range of the AP, there becomes a decision point as whether to handover. Data collected from the MH are shown in figure 8.21. The RSS for WLAN is detected after 75 seconds, whereas RSS for UMTS remains steady. After 120 seconds the throughput for the voice application begins to increase up to the required 64 kbps at 160 seconds. Similarly, latency is measured at the same point, and becomes stable after 160 seconds. The variations of throughput and latency at in this region could be due to the multi-rate MAC layer responding to packet errors.

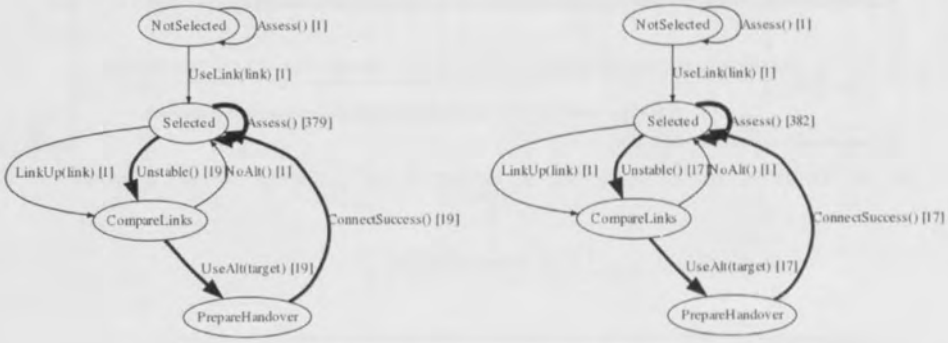


(a) WLAN

Figure 8.21: Time-series plots of inputs for S2.3

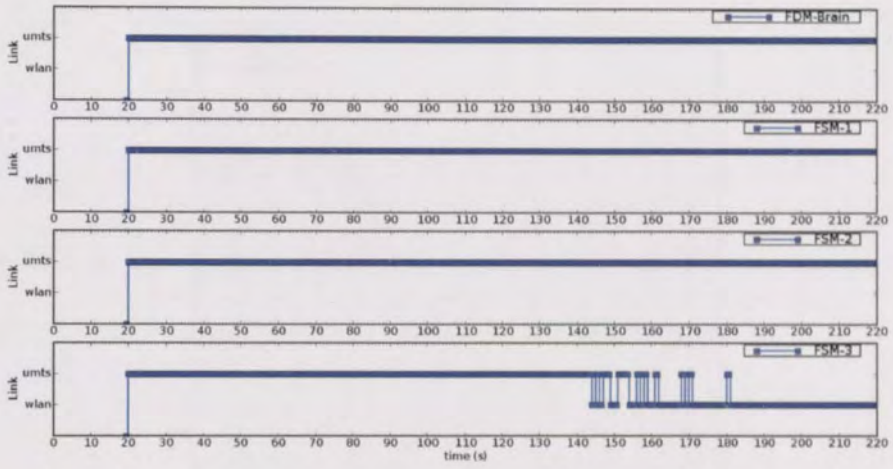
Figure 8.23 on page 151 shows interface selections of prototypes for the experiment 1 that only uses throughput, and experiment 2 that uses two metrics (throughput and one-way delay). The plots for both experiments are similar. All prototypes connect to UMTS, initially as the only link available. QoS metrics improve for WLAN when the MH moves into range of the AP. Only FSM-3 has the logic to take advantage of this, the other prototypes FSM-1, FSM-2, and FDM-Brain do not assess other links unless the current link becomes unstable. Although FSM-3 performs this assessment, the result is not ideal. In the transition region where there is

a potential handover the selection flips between UMTS and WLAN, finally setting on WLAN once the metrics have settled. Figure 8.24 on page 152 shows the points of handover with regards to the position. The MH performs 19 handovers in experiment one and 17 in experiment two, as indicated by the FSM traces of figure 8.22. In both experiments the FSM gets caught in a loop `Selected->CompareLinks->PrepareHandover` in the period 143-181 seconds.

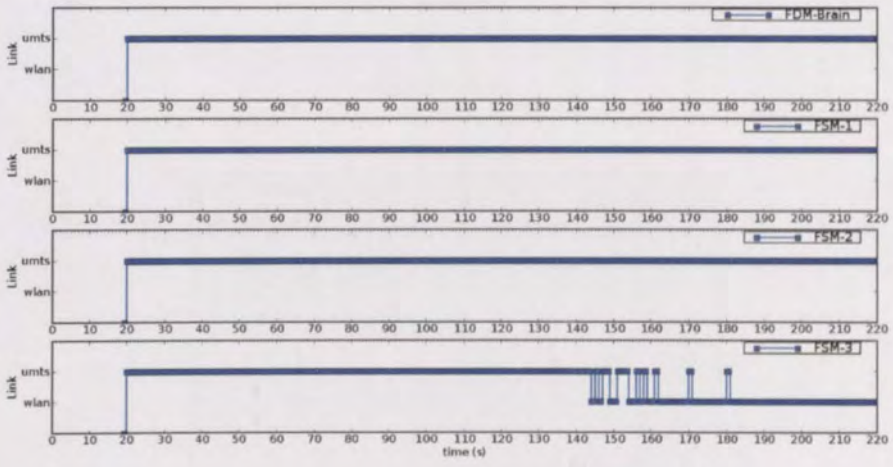


(a) Experiement 1 (b) Experiement 2

Figure 8.22: FSM-3 trace with transition counts for S2.3 test.

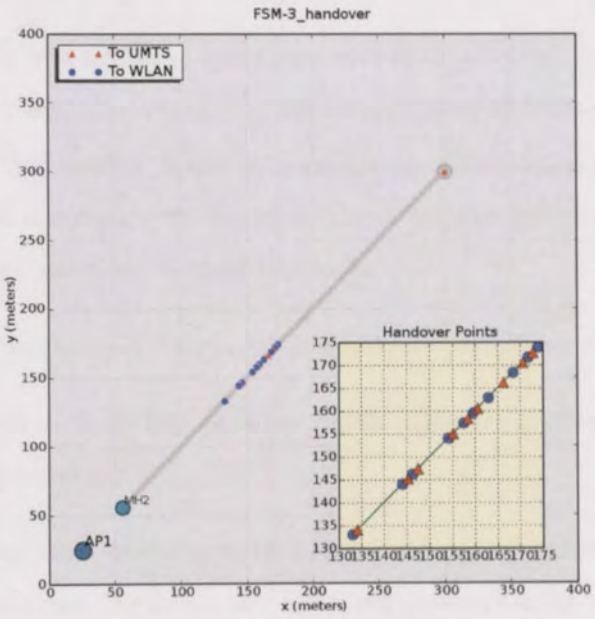


(a) Experiment 1

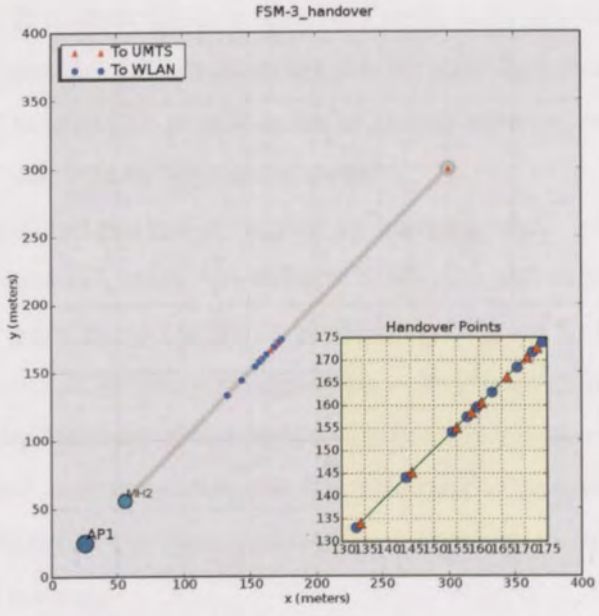


(b) Experiment 2

Figure 8.23: Time-series plots of prototypes link selection for S2.3



(a) Experiment 1



(b) Experiment 2

Figure 8.24: Handover plots of FSM-3 handover for S2.3

8.2.3 Discussion

The prototype models with additional agent logic were evaluated with wireless simulation data provided by the NS-2 simulator. A roaming pattern moved the MH between available WLAN to provide dynamic QoS metrics. It was these changes in metrics that required the prototypes to weigh options and make decisions (based on preset decision criteria of application type). The following research questions motivated the study:

1. *How does the number of decision criteria affect interface selection?*
2. *Does the interface choice and handover points reflect the performance of application and policy requirements?*

The first question was based on the hypothesis that prototypes should select interfaces based on application requirements. Although, this was partly answered in the analytical tests by the parameters of decision module. In the simulation tests, the same decision module settings were used; only the subsequent decision logic was varied between prototypes. This logic used the decision module to combine inputs to generate a single value of interface suitability. Although this value is used to judge link suitability, it is the agent logic that determines when to handover. Points of handover are periods in the simulation when the performance measures degrade sufficiently, such as at the edge of link visibility.

Question two continued this line of enquiry, by exploring where selection decisions and handovers occur in time and space. As different application and policy settings affect the decision-making, changes in QoS performance affect the response of the prototypes logic. These *transition regions* are periods in the simulation when a handover decision can be made, such as when two interfaces have similar performance or either becomes degraded. The roaming scenarios provided an artificial change in link conditions to generate a transition region. Agent response in this region was thus a combination of agent logic and the application criteria to QoS performance mapping.

Relevance of decision criteria

As demonstrated by the decision analysis results (section 8.1.4 on page 139), options in the decision module are determined by multiple criteria and corresponding input. Although these results do not provide an optimum number or type of criteria, the resulting value is a function of criteria and aggregation parameters. In the simulation test cases, the judgement of link suit-

ability is only as good as the inputs. More precisely, the measured (dynamic) metrics should be calibrated to the corresponding criteria. This means that criteria defined for application metrics must use metrics measured at the application layer, and MAC layer criteria must correspond to MAC metrics, and so on. A judgement on the interface suitability is therefore only as good as the inputs of metric and criteria mapping.

Criteria and metrics in the simulations were specified for one and two dynamic metrics. These were throughput and one-way delay of received packets measured at the application layer. A voice application was used to generate traffic and criteria was set for a G.711 type call. The simulation models for WLAN and UMTS links were run separately, and metrics were then combined; an unlikely scenario. In a more realistic setting application traffic would be sent over one interface and then handed-over to a new interface, so the same type of metrics would not be available for the other link until the handover was made. Although the simulation and experimental study was simplified by using the same criteria and metric types on both links and no handovers, the decision module is capable of using different criteria. It would need to be extended to measure, for instance MAC layer throughput and latency, then criteria to judge those metrics. Handover mobility would also need to be included in the simulation model as the procedure for handover may, depending on mobility protocols used, affect conditions through increased delays and packet losses.

In the tests, FSM-2 assessed the current selected link, but only assessing others if it becomes unsuitable with regards to requirements criteria. The effect of this, is that it would only change when necessary. The FSM-3 prototype actively assesses all interfaces (those with connectivity) with available criteria data, and could potentially find gains for the user in some criteria, such as lower cost per byte, or higher throughput. For FSM-2 the logic behaved as expected, but FSM-3 there were unacceptable number of undesirable handover requests in transition regions (false positives). In both moving-out (S2.2) and moving-in (S2.3) tests, the experiments with less criteria generated more handover requests using FSM-3. However, there is not enough data to suggest a correlation.

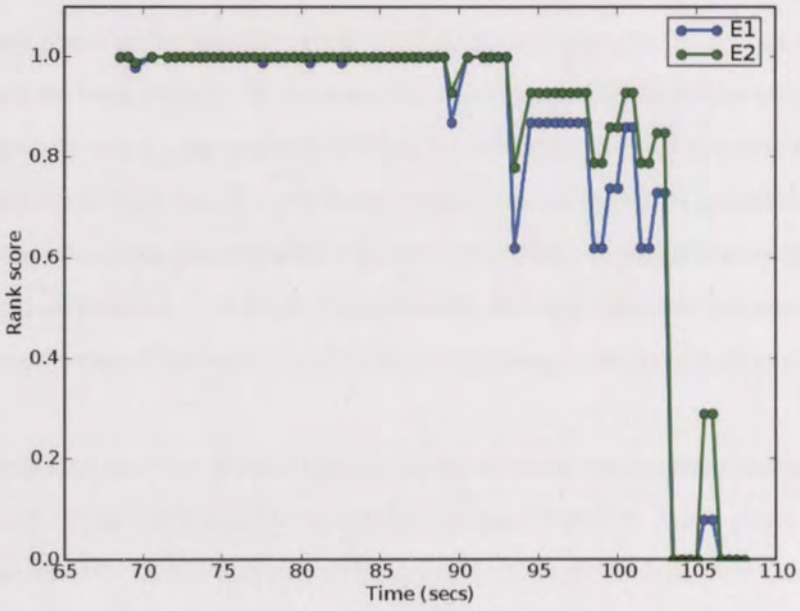
Prototypes at points of handover

Since the purpose of the the agent framework is to provide interface selection decisions, the simulations have roaming scenarios to change the conditions. Changing conditions of QoS on different interfaces creates a transition region where a selection change can occur to maintain

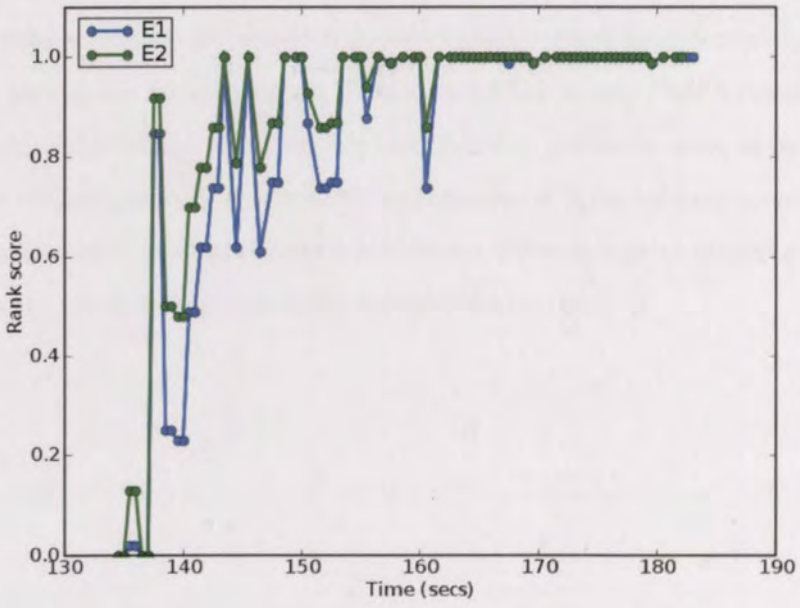
or improve QoS. Link selection or choice plots showed the output from prototypes during the simulation.

The FSM-2 prototype results were consistent with the designed logic: it only assesses other links when the current link drops in rank-score. FSM-3 logic continually assesses all interfaces for opportunities. FSM-3 needs to assess all other interfaces most of the time, to find improvements on the current selected link. For FSM-3 in S2.2, excessive selections and handovers were triggered during the transition region. A behaviour was used that triggers a handover if the current assessed rank-score for the link is below the previous value. At 68.5 seconds the prototypes agent rank score for WLAN changed from 1.0 to 0.99 for experiment 1, and 74.5 for experiment 2. Both are caused by the same behaviour: a lower rank-score for consecutive link assessments causes a change to selection. This is shown by figure 8.25a) for the period 68-110 seconds where FSM-3 WLAN rank-scores are assessed using the behaviour AssessAllCurrent. The scores for experiment 2 (E2) are less variable than E1 and the rank-scores begin to worsen later.

As the test environment did not model handovers in sufficient detail to assess delays and mobility management, these effects on prototypes are unknown. Real-time applications will be more susceptible to mobility management, and thus have a smaller margin of error in delay; requiring additional timing control. Further tests would be required for assessing mobility protocols and handover delay effects.



(a) S2.2



(b) S2.3

Figure 8.25: Comparison of FSM-3 rank-scores for WLAN link.

Concluding Remarks

The analytical phase of the research aimed to discover the trade-off effects from different settings in fuzzy decision-making. Subject models were aggregation functions of the mean and a generic function using a parameter to change the effect of trade-off between criteria. The generic decision function used this parameter to represent the degree of optimism: whether to push down the score when poor criteria is present (pessimistic), or push the score up when there is good criteria (optimistic). A generic decision function and optimism parameter ($\gamma = -2$) provides a good trade-off between criteria while the score can change enough to reflect criteria differences.

Simulation tests used the generic function in the decision module and combined this with different setups of the agent framework developed from Chapter 6. Agent prototypes using a Subsumption model (FDM-Brain) and variations of finite-state machine logic (FSM-1, -2, -3) were compared using data from wireless simulations. NS-2 was used to generate metric traces of a voice application over WLAN and UMTS according to roaming scenarios. Three scenarios defined a mobile host that is: stationary, moves away from WLAN, and moves into range of WLAN. In these scenarios, the transition region of changing link conditions caused handover requests in moving-out for FDM-Brain, FSM-2, and FSM-3; only FSM-3 caused handovers for moving-in. Logic defined by FSM-3 is insufficient to determine when to handover. The tests caused multiple unnecessary handover requests due to a limited rank-score comparison method. An alternative ranking method is needed or a different logic to suggest the difference between interface rank-score is significant enough for a handover.

Chapter 9

Conclusions

This thesis proposed a new vertical handover strategy using a combination of artificial intelligence (AI) methods. A prototype hybrid framework design managed inputs, data representation, and control actions. It was also shown that handover logic can be encoded as finite-state machines, with user QoS policies used to determine wireless interface suitability.

A trend towards heterogeneous wireless environments and varied types of media services requires that QoS and user satisfaction are prominent in next-generation networks. The problems in next-generation heterogeneous wireless environments include many levels of complexity; from link coexistence to user-centric policies and contexts. The research focused on an approach to improve QoS by leveraging the differences in wireless networks; developing the overlay pattern as proposed in Stemm & Katz (1998). Moreover, wireless access networks vary in areas of coverage, capabilities, and dynamic conditions. A strategy and methods to manage these issues would be beneficial to the user experience and possibly to the network–as intelligent devices could adapt to changing network conditions.

Chapter 2 provided some insight into the challenges of providing QoS for networks, users, and applications. Architectures proposed for QoS (Aurrecoechea et al., 1998; Mustill & Willis, 2005) are mainly afterthoughts, that introduce another layer of complexity into the network (Schormans & Pitts, 2004; Vin, 2005). However, in noisy mediums, such as wireless, QoS awareness is inherently more important. Chapter 3 presented wireless networks and their support for QoS.

Heterogeneous wireless environments are a concept of multiple types of interface with a common, or inter-operable core network; as described in Chapter 4. More inter-operable and cross-layer approaches to next-generation wireless networks have been shown in the literature (Kawadia & Kumar, 2005; Liu et al., 2006; Baldo & Zorzi, 2007). There is also pressure

from operators and effects of convergence within the industry (Economist, 2006). This has implications of further complexity in the network and terminal devices, through more protocols and configuration. Potential solution devices are user-centric and context-aware, and more adaptable to changes in the network.

Chapter 5 reviewed AI and decision-making techniques. Selected techniques of fuzzy decision-making (FDM) (Bellman & Zadeh, 1970; Sousa & Kaymak, 2002), finite-state machines (FSM) (Byun et al., 2001), and an agent framework from behaviour-based design (Bryson, 2001), were explained in Chapter 6. An evaluation approach described in Chapter 7 used analytical techniques for decision-making components and wireless simulation. Wireless roaming scenarios for cellular UMTS and WLAN networks were simulated using NS-2 to generate QoS and performance metrics data. The simulation data was used as inputs for handover and selection by the agent framework, and results were reported in Chapter 8.

The thesis has presented issues and defined a model of complexity within heterogeneous wireless environments. Agent prototypes were developed for user policy and QoS requirements to make optimised decisions. The decision component reflects pre-defined criteria when judging interface suitability, and thus the final point-of-handover. Logic for handover control was described in two FSM models: one that would only handover if the current link QoS was poor; while the other assessed alternative links for improved QoS. The outcome was prototypes that used a combination of FSMs for formalising the logic of discontinuity in the handover process, and QoS optimised decision-making for comparing criteria in wireless interface assessments. Experiments involving simulation were limited to three criteria and two wireless interfaces, thus further experiments and probably modifcations are required to strengthen the case for this agent-based solution.

9.1 Discussion

The scope of heterogeneous environments covers many concepts that include: session continuity, mobility, security, authentication, and inter-working. The thesis scope covered vertical handover strategies for optimising user and QoS policies. This still has many open issues, such as psychological factors of risk and subjective assessments, together with techniques for planning and decision-making.

Handover strategies in the literature have used multiple criteria to make assessments. Decision-theoretic and AI solutions have employed user-centric and soft decision methods, such as mul-

tiple attribute decision-making (MADM), and fuzzy logic. MADM and fuzzy variations allow criteria to be defined in vague, subjective terms within a region of uncertainty. The fuzzy decision-making (FDM) component used in the thesis allows this combination of multiple factors and trade-offs by importance, but also enables fine-tuning assessments in terms of subjective risk. As shown in the analytical experiments (section 8.1 on page 124), an *optimism operator,* γ, changes decision-functions (aggregation of decision criteria) between optimistic and pessimistic decisions. Optimistic decision parameter $\gamma = 2.0$, gives higher rank-scores for good criteria, trading-off poor performing criteria. Whereas the pessimistic decision parameter $\gamma = -2.0$ produces lower rank-scores when poor criteria are present. The combination of importance weights was shown to provide a flexible, but largely predictable pattern of rank-scoring of interface assessments.

The FDM component was used as part of finite-state logic for decision processing. FSM models were used to perform actions that depend on events or status updates from the device. These could be interface changes and metric updates. According to the architecture in Chapter 6, the FSM control would use behaviours for specific states. These behaviours use the FDM to assess interface suitability.

For the FSMs with modest functionality (FSM-1 and FSM-2), selection results were as expected. However, the logic for FSM-3 caused unexpected results. FSM-3 has the aim of comparing all available links, while the other models compare only in a fail-over capacity. The logic used in FSM compares the rank-score for each link, if the current selected link is no longer the highest rank-score it issues an *unstable* event. An unstable event causes a change to a new state where the assessment is performed again using the AssessAll behaviour. If there is a worse rank-score, then a handover is initiated by changing to another state in the FSM. As shown by the rank-score for the transition periods (figure 8.25 on page 156), FSM-3 rank-scores for WLAN change constantly during this period. Since the rank-score determines the suitability of the link, the logic was too simplistic in using this value.

Based on these results, the behaviour logic used in FSM-3 could benefit from an additional procedure to calculate when the difference between rank-scores is significant enough to warrant a handover. A possible solution is to use hysteresis on the rank-scores; only performing handover if there is a improvement by some margin. The modular approach means making this change is not too cumbersome, and can be achieved by replacing the behaviour module. Where the behaviour is called is then a matter of modifying the state or transition in FSM. Al-

though the logic in FSM-3 is not quite desirable, the agent approach helps reduce the problem complexity.

Agent concept for reducing complexity

Using the agent paradigm attempted to create a hierarchy of logic, or bottom-up reasoning. The behaviour-based design approach from robotics, uses many lower-level behaviours performing relatively simple tasks. Higher-level logic combines behaviours, or more accurately, uses different behaviours depending on other factors like internal or external status of the agent. With regard to the agent design in this thesis, figure 9.1 shows this hierarchy as a separation of logic. The Agent container provides interaction to the outside world, collecting status data and sending messages with external objects. The FSM is a representation of the agent's environment and the scope of logic for interacting with other agent objects. Other objects are data structures and events monitored by the agent. Such events causes changes in the FSM, which in turn *may* trigger other actions or *behaviours*. The behaviours are short procedures or rules that perform some task within a limited scope. In the prototype, behaviour scopes are defined for assessing interfaces. The assessment process uses fuzzy decision making constructs to aggregate multiple metric data according to decision criteria. Combining these 'layers of control' creates a modular decision-making and control approach.

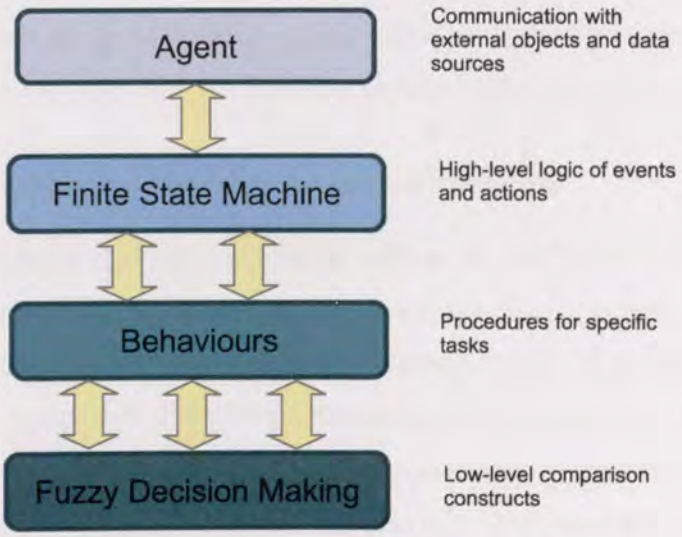


Figure 9.1: Logic and task hierarchy in the hybrid agent.

For a user-centric perspective in next-generation networks, handover strategies described in the literature are designed to allow additional decision-making information. Information about the network, interfaces, QoS, and users needs to be collected and formalised. Methods in the literature have been described for cross-layer signalling (Shakkottai & Rappaport, 2003), and network-based information databases. Emerging standards, such as 802.21 (IEEE, 2006) uses cross-layer signalling and adaptable data structures to gather and store data from different sources. However, 802.21 is a framework that describes interaction protocols and types of data, but not specific algorithms for making handover decisions. The HAL framework proposed in this thesis would compliment 802.21 and related frameworks for user and QoS optimised handovers.

9.2 Further Research

This thesis highlights a number of directions for further research in heterogeneous wireless environments. It is a domain complicated by technical issues of QoS, mobility, security, session-continuity; and non-technical user issues of operator policy and market trends. This research focused on handover strategies and multi-factor decision-making. A possible extension could develop QoS representation schemes, collecting performance and status information from the network and wireless interfaces. Another extension could develop the simulation framework to use cross-layer messaging and network architectures that support interaction among roaming protocols and session continuity. The following sections expand these themes further.

9.2.1 QoS classification for richer network information

A simple description of QoS and performance metrics was used in this thesis, but richer representation schemes have been proposed in the literature. These include examples of QoS taxonomies (Sabata et al., 1997) and QoS implementations using semantic schema (Dobson et al., 2005). A recent draft of 802.21 for media-independent handovers (IEEE, 2006), uses the resource description framework (RDF) and web ontology language (OWL); based on the eXtensible Markup Language (XML). The proposed RDF/OWL approach in 802.21, provides an "information service schema" (IEEE, 2006, p.58) for hierarchical and distributed data structures with flexible querying. As new link technologies are added the schema can be updated with new information. Such a schema for QoS provides more flexibility describing QoS and wireless interface metrics.

QoS descriptions could benefit from ontological implementations, but this requires further research. There are also questions of interactions with other protocols, but with work ongoing in the IEEE 802.21 working group, a standards-based QoS description is not yet finalised. Also, the QoS mapping aspect could use lessons learnt from other research projects that use QoS mechanisms like IntServ and DiffServ: the *wireless adaptation layer* (WAL). WAL has a component for end-to-end QoS using application requirements and IP type of service (ToS) fields (Prasad & Munoz, 2003, p.216).

9.2.2 Simulating vertical handovers for session continuity

Vertical handover concepts are still evolving in terms of modelling and simulation, with those in the public domain limited to customised models. This thesis used wireless simulation of NS-2, but not mobility protocols such as mobile IP. However, these tend to be third party tools that do not easily integrate with other components, or are designed for different versions of the simulator code (in the case of NS-2). This makes new modules, such as a mobile host with multiple wireless interfaces, require modifications to simulation code. This means open-source modifications are non-standard, and vary between research projects. Researchers developing new protocols for heterogeneous wireless environments would benefit from common models and data-sets.

An extension to NS-2, *ns-miracle* (Baldo et al., 2007), provides a modular framework for dynamic libraries and cross-layer messaging. Dynamic libraries allow different protocols and third-party extensions to be loaded in simulation models. Cross-layer messaging functionality is relevant to the thesis, as the prototype uses metrics and event data from the protocol stack. This approach could be extended to test how the prototype performs in session continuity scenarios. The current thesis models do not currently simulate this close integration of handover strategy and mobility protocols.

The ns-miracle libraries could be used to provide metrics and control messaging between protocol objects and the HAL agent. Integrating the current code would require additional programming, as HAL is written in Python, and NS-2 is written in C++ and Tcl, although there are open-source tools[1] that allow programs and libraries to interact between languages with less modifications to existing code.

[1] Elmer (`http://elmer.sourceforge.net`) provides a bridge for Python code in C, C++ and Tcl applications.

9.2.3 Other extensions

The research used FSMs for sequential event and action processing, and decisions are based on assessments from FDM. Richer context information could improve the decision and selection process. Other AI techniques, such as dynamical planning could be used for adaptation. Goal planning methods have been used in full and partially observable domains (Bertoli & Pistore, 2004). A realisation of active goals that can generate plans depending on environment changes could provide a high-level reasoning approach. The HAL agent described in Chapter 6 could be considered a simpler version of a planning agent (deterministic plans defined as FSM models), but more detailed than conditional programming.

In this thesis, the problem was shown (section 4.3 on page 49) to be a multi-dimensional decision space. Rather than provide a solution for the extreme of all factors, the HAL vertical handover strategy optimises for one application or service class at a time. Though in more realistic settings, users are likely to use different types of application, or switch several times within a session. Extending the solution to more than one service type, a further round of assessments could be used to prioritise between applications or hierarchical separation of criteria assessment (Sousa & Kaymak, 2002, chap.3). A similar approach was used by Yeh et al. (2000), which defines sub-criteria. Other AI or decision-making methods could be introduced at other points in the decision process, such as AHP for determining preference between applications.

The research has implications for economic use of electrical resources with more computing devices in homes, offices and industry. Adaptable and 'intelligent' devices with wireless connectivity for WPANs, WLANs, and WWANs are finding more applications in automated control and monitoring. The environmental impact of computing requires smarter, low-powered devices that adapt to changes in their environment. Reports from computing activity account for 2% of global CO_2 (830m tonnes) emissions from human activity in 1997, rising to 1.4 billion tonnes by 2020 (Economist, 2008). The report also describes potential computing CO_2 savings from other industries of 7.8 billion tonnes; with 1.7 billion from smart buildings. Wireless and low-powered devices could become the enabler for smart devices, backed by standards and Internet technologies.

As a consequence of more wireless availability, devices need to become 'intelligent' or adaptable. The concepts used in this thesis and implemented in an agent prototype provides a framework design for interface selection in heterogeneous wireless environments. Future devices will contain more protocols and potential for wireless connectivity in order to improve QoS for services with an adaptability that is foremost user-centric.

Bibliography

(2005). *Fast Handovers for Mobile IPv6.* IETF, RFC 4068.

3GPP (2004a). *3GPP system to Wireless Local Area Network (WLAN) interworking ; System description (Release 6).* 3GPP, TS 23.234.

3GPP (2004b). *Quality of Service (QoS) concept and architecture (Release 6).* 3GPP, TS 23.107.

3GPP (2007). *Technical Specification Group GSM/EDGE Radio Access Network; Generic access to the A/Gb interface; Stage 2 (Release 7).* 3GPP, TS 43.318 V7.3.0.

Adams, J. & Heile, B. (2006). 'Busy as a ZigBee' [Online]. Available: `http://www.spectrum.ieee.org/oct06/4666`. (accessed: June 2008).

Aguilar, J. (2005). 'A Survey about Fuzzy Cognitive Maps Papers (Invited Paper)'. *International Journal of Computational Cognition* **3**(2):27–33.

Ahmavaara, K., Haverinen, H. & Pichna, R. (2003). 'Interworking Architecture Between 3GPP and WLAN Systems,'. *IEEE Communications Magazine* **41**(11):74–81.

Alliance, W.-F. (2004). *Wi-Fi CERTIFIED for WMM - Support for Multimedia Applications with Quality of Service in Wi-Fi Networks.* Wi-Fi Alliance.

Almes, G., Kalidindi, S. & Zekauskas, M. (1999a). *A One-way Delay Metric for IPPM.* IETF, RFC 2679.

Almes, G., Kalidindi, S. & Zekauskas, M. (1999b). *A One-way Packet Loss Metric for IPPM.* IETF, RFC 2680.

Almes, G., Kalidindi, S. & Zekauskas, M. (1999c). *A Round-trip Delay Metric for IPPM.* IETF, RFC 2680.

Ashby, W. R. (1973). 'Some peculiarities of complex systems'. *Cybernetic Medicine* **9**(2):1–7.

Aurrecoechea, C., Campbell, A. T. & Hauw, L. (1998). 'A survey of QoS architectures'. *Multimedia Systems* **6**(3):138–151.

Axelrod, R. (1976). *The Structure of Decision: Cognitive Maps of Political Elites*. Princeton University Press, USA.

Baldo, N., Maguolo, F., Miozzo, M., Rossi, M. & Zorzi, M. (2007). 'ns2-MIRACLE: a modular framework for multi-technology and cross-layer support in network simulator 2'. In *ValueTools '07: Proceedings of the 2nd international conference on Performance evaluation methodologies and tools*, pp. 1–8, Brussels, Belgium. Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering; ICST, ICST.

Baldo, N. & Zorzi, M. (2007). 'Fuzzy Logic for Cross-layer Optimization in Cognitive Radio Networks'. In *4th IEEE Consumer Communications and Networking Conference (CCNC 2007)*, pp. 1128–1133, Las Vegas, NV, USA. IEEE.

Bar-Shalom, O., Chinn, G., Fleming, K. & Gadamsetty, U. (2003). 'On the Union of WPAN and WLAN in Mobile Computers and Hand-Held Devices'. *Intel Technology Review* **7**(3):20–36.

Baudet, S., Besset-Bathias, C., Frene, P. & Giroux, N. (2001). 'QoS Implementation in UMTS Networks'. *Alcatel Telecommunications Review* **1st Quarter 2001**.

Beazley, D. M. (2003). 'Automated scientific software scripting with SWIG'. *Future Generation Computer Systems* **19**(5):599–609.

Bellman, R. E. (1957). *Dynamic Programming*. Princeton University Press, Princeton, New Jersey. USA.

Bellman, R. E. & Zadeh, L. A. (1970). 'Decision Making in a Fuzzy Environment'. *Management Science* **17**(4):141–164.

Bertoli, P. & Pistore, M. (2004). 'Planning with extended goals and partial observability'. In *14th International Conference on Automated Planning and Scheduling (ICAPS'04)*, pp. 270–278, British Columbia, Canada.

Beynon, M., Curry, B. & Morgan, P. (2000). 'The Dempster-Shafer theory of evidence: an alternative approach to multicriteria decision modelling'. *Omega* **28**(1):37–50.

Bieszczad, B., Biswas, P. K., Buga, W., Malek, M. & Tan, H. (1999). 'Management of heterogeneous networks with intelligent agents'. *Bell Labs Technical Journal* **4**(4):109–135.

Black, U. (2000). *QoS in wide area networks.* Prentice Hall PTR, Upper Saddle River, NJ, USA.

Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. & Weiss, W. (1998). *An Architecture for Differentiated Services.* IETF, RFC 2475.

Blank, D., Kumar, D., Meeden, L. & Yanco, H. (2006). 'The Pyro Toolkit for AI and Robotics'. *AI Magazine* **27**(1):39.

Braden, R., Clark, D. & Shenker, S. (1994). *Integrated Services in the Internet Architecture: An Overview.* IETF, RFC 1633.

Braden, R. E. ., Zhang, L., Berson, S., Herzog, S. & Jamin, S. (1997). *Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification.* IETF, RFC 2205.

Brooks, R. A. (1991). 'Intelligence Without Reason'. In *12th International Joint Conference on Artificial Intelligence*, pp. 569–595, Sydney, Australia.

Brooks, R. A. & Connell, J. (1986). 'Asynchronous Distributed Control System for a Mobile Robot'. In *SPIE's Cambridge Symposium on Optical and Optoelectronic Engineering*, pp. 77–84, Cambridge, MA, USA.

Bryson, J. J. (2001). *Intelligence by Design: Principles of Modulatiry and Coordination for Engineering Complex Adaptive Agents.* Ph.D. thesis, Massachusetts Institute of Technology, USA.

Buchner, A. & Funke, J. (1993). 'Finite-state Automata: Dynamic Task Environments in Problem-solving Research'. *The Quarterly Journal of Experimental Physcology* **46A**(1):83–118.

Burgess, M. S. E. (2003). *Using multiple quality criteria to focus information search results.* Ph.D. thesis, Cardiff University, UK.

Burgstahler, L., Dolzer, K., Hauser, C., Jahnert, J., Junghans, S., Macian, C. & Payer, W. (2003). 'Beyond technology: the missing pieces for QoS success'. In *RIPQoS '03: Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS*, pp. 121–130, Karlsruhe, Germany. ACM Press.

Byun, Y. J., Sanders, B. A. & Keum, C.-S. (2001). 'Design Patterns of Communicating Extended Finite State Machines in SDL'. In *8th Conference on Pattern Languages of Programs*, Monticello, Illinois, USA.

Cacace, F. & Vollero, L. (2006). 'Managing mobility and adaptation in upcoming 802.21 enabled devices'. In *WMASH '06: Proceedings of the 4th international workshop on Wireless mobile applications and services on WLAN hotspots*, pp. 1–10, New York, NY, USA. ACM Press.

Calvagna, A., Corte, A. L. & Sicari, S. (2005). 'Mobility and quality of service across heterogeneous wireless networks'. *Computer Networks* **47**(2):203–217.

Calvagna, A. & Modica, G. D. (2005). 'A cost-based approach to vertical handover policies between WiFi and GPRS'. *Wireless Communications and Mobile Computing* **5**(6):603–617.

Campbell, A. T., Gomez, J., Sanghyo, K., Chieh-Yih, W. & Turanyi, Z. R. (2002). 'Comparison of IP micromobility protocols'. *IEEE Wireless Communications* **9**(1):72–82.

Carlsson, C. & Fuller, R. (1996). 'Fuzzy multiple criteria decision making: Recent developments'. *Fuzzy Sets and Systems* **78**(2):139–153.

Chakravorty, R., Vidales, P., Subramanian, K., Pratt, I. & Crowcroft, J. (2004). 'Performance Issues with Vertical Handovers - Experiences from GPRS Cellular and WLAN hot-spots Integration'. In *Second IEEE International Conference on Pervasive Computing and Communications*, Washington, DC, USA. IEEE Computer Society.

Chalmers, D. & Sloman, M. (1999). 'A Survey of Quality of Service In Mobile Computing Environments'. *IEEE Communications Surveys* **2nd Quarter 1999**.

Chan, P. M. L., Hu, Y. F. & Sheriff, R. E. (2002). 'Implementation of fuzzy multiple objective decision making algorithm in a heterogeneous mobile environment'. In *IEEE Wireless Communications and Networking Conference*, pp. 332 – 336.

Chan, P. M. L., Sheriff, R. E., Hu, Y. F., Conforto, P. & Tocci, C. (2001). 'Mobility management incorporating fuzzy logic for a heterogeneous IP environment'. *IEEE Communications Magazine* **39**(12):42–51.

Chen, S. J., Hwang, C. L. & Hwang, F. P. (1992). *Fuzzy Multiple Attribute Decision Making*. Springer-Verlag.

Chen, Y. (2003). *Soft Handover Issues in Radio Resource Management for 3G WCDMA Networks*. Ph.D. thesis, Department of Electronic Engineering, Queen Mary University, UK.

Cheng, R.-G. & Chang, C.-J. (1999). 'A QoS-Provisioning Neural Fuzzy Connection Admission Controller fo Multimedia High-Speed Networks'. *IEEE/ACM Transactions on Networking* **7**(1):111–121.

Chin, A., Gupta, A., Narjala, R. & Vallabhu, V. (2003). 'Seamless Connectivity to Wireless Local Area Networks'. *Intel Technology Review* **7**(3):58–67.

Cisco (2005). *DiffServ – The Scalable End-to-End QoS Model*. Cisco Systems.

Cisco (2006). *Understanding Delay in Packet Voice Networks*. Cisco Systems, 5125.

Demichelis, C. & Chimento, P. (2002). *IP Packet Delay Variation for IP Performance Metrics (IPPM)*. IETF, RFC 3393.

Dempster, A. P. (1968). 'A generalization of Bayesian inference'. *Journal of the Royal Statistical Society* **Series B**(30):205–247.

Dobson, G., Lock, R. & Sommerville, I. (2005). 'Quality of Service Requirement Specification using an Ontology'. In *13th IEEE International Requirements Engineering Conference, SOCCER 05 Workshop*, Paris, France.

Dunmore, M. & Pagtzis, T. (2005). 'Mobile IPv6 Handovers: Performance Analysis and Evaluation'. Tech. Rep. 32603/ULANC/DS/4.1.1/A1, 6net.

Dutta, A., Das, S., Famolari, D., Ohba, Y., Taniuchi, K., Kodama, T. & Shulzrinne, H. (2005). 'Seamless Handover across Heterogeneous Networks - An IEEE 802.21 Centric Approach'. In *Wireless Personal Multimedia Communications (WPMC) 2005*, pp. 31–35, Aalborg, Denmark.

Economist, T. (2006). 'Your television is ringing'. The Economist, 12 October 2006.

Economist, T. (2008). 'Computing sustainability'. The Economist, June 21st 2008.

Eddy, W. M. (2004). 'At What Layer Does Mobility Belong?'. *IEEE Communications Magazine* **42**(10).

Edwards, G. & Sankar, R. (1998). 'Microcellular handoff using fuzzy techniques'. *Wireless Networks* **4**(5):401–409.

Erickson, T. (2002). 'Some problems with the notion of context-aware computing'. *Communications of the ACM* **45**(2):102–104.

Fernandez, M. P., de Castro P. Pedroza, A. & de Rezende, F. (2004). 'Dynamic QoS Provisioning in DiffServ Domains Using Fuzzy Logic Controllers'. *Telecommunication Systems* **26**(1):9–32.

Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, USA.

Ghinea, G. & Magoulas, G. (2001). 'Quality of Service for Perceptual Considerations: An Integrated Perspective'. In *IEEE International Conference on Multimedia and Expo*, pp. 752–755, Tokyo, Japan.

Gozdecki, J., Jajszczyk, A. & Stankiewicz, R. (2003). 'Quality of service terminology in IP networks'. *IEEE Communications Magazine* **41**(3):153–159.

Grilo, A. & Nunes, M. (2002). 'Performance Evaluation Of IEEE 802.11e'. In *13th IEEE Symposium on Personal, Indoor and Mobile Radio Communications: PIMRC*, Lisbon, Portugal.

Gu, D. & Zhang, J. (2003). 'QoS Enhancement in IEEE802.11 Wireless Local Area Networks'. *IEEE Communications Magazine* **41**(6):120– 124.

Gunasekaran, V. & Harmantzis, F. C. (2008). 'Towards a Wi-Fi ecosystem: Technology integration and emerging service models'. *Telecommunications Policy* **32**(3-4):163–181.

Gurijala, A. & Molina, C. (2004). 'Defining and Monitoring QoS Metrics in the Next Generation Wireless Networks'. In *IEE Telecommunications Quality of Service (QoS 2004)*, pp. 37–42, Savoy Place, London, UK. IEE.

Gustafsson, E. & Jonsson, A. (2003). 'Always Best Connected'. *IEEE Wireless Communications* **10**(1):49–55.

Halpern, J. Y. (2003). *Reasoning About Uncertainty*. MIT Press, Massachusetts, USA.

Hardy, W. C. (2001). *QoS: Measurement and evaluation of telecommunications quality of service*. Wiley and Sons Ltd.

Hawick, K. A. & James, H. A. (2003). 'Middleware for Context Sensitive Mobile Applications'. *Workshop on Wearable, Invisible, Context-Aware,Ambient, Pervasive and Ubiquitous Computing* **21**.

Heidemann, J., Bulusu, N., Elson, J., Intanagonwiwat, C., Lan, K.-c., Xu, Y., Ye, W., Estrin, D. & Govindan, R. (2001a). 'Effects of Detail in Wireless Network Simulation'. In *Proceedings of the SCS Multiconference on Distributed Simulation*, pp. 3–11, Phoenix, Arizona, USA. Society for Computer Simulation.

Heidemann, J., Mills, K. & Kumar, S. (2001b). 'Expanding confidence in network simulations'. *IEEE Network* **15**(5):58–63.

Hersent, O., Gurle, D. & Petit, J.-P. (2000). *IP Telephony: Packet Based Multimedia Communications Systems*. Pearson Education Limited, UK.

Huard, J.-F. & Lazar, A. A. (1997). 'On QOS Mapping in Multimedia Networks'. In *Computer Software and Applications Conference, 1997. COMPSAC '97. Proceedings., The Twenty-First Annual International*, pp. 312–317, Washington, DC, USA.

Hutter, M. (2007). 'Algorithmic information theory'. *Scholarpedia* **2**(3):2519.

Ibrahim, H. A. (2004). *Fuzzy logic for embedded systems applications*. Elsevier Science, USA.

IEEE (2005). *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*. ANSI/IEEE, Std 802.11e.

IEEE (2006). *Draft IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services*. ANSI/IEEE, IEEE P802.21/D00.05.

IST-1999-10050 (2001). 'Broadband Radio Access for IP based Networks (BRAIN)' [Online]. Available: `http://web.archive.org/web/20031023194821/http://www.ist-brain.org/`. (accessed: October 23, 2003).

IST-2000-25394 (2003). 'Moby Dick - Mobility and Differentiated Services in a Future IP Network' [Online]. Available: `http://www.ist-mobydick.org/`. (accessed: October 30, 2007).

IST-2000-28584 (2002). 'Mobile IP based Network Developments (MIND)' [Online]. Available: `http://web.archive.org/web/20050304071121/http://www.ist-mind.org/`. (accessed: March 4, 2005).

IST-507134 (2007). 'Ambient Networks: Mobile and wireless systems beyond 3G' [Online]. Available: `http://www.ambient-networks.org`. (accessed: November 1, 2007).

ITU (2008). 'Key Global Telecom Indicators for the World Telecommunication Service Sector' [Online]. Available: `http://www.itu.int/ITU-D/ict/statistics/at_glance/KeyTelecom99.html`. (accessed: August 11, 2008).

ITU-T (1994). *Terms and definitions related to quality of service and network performance including dependability.* ITU-T, E.800.

ITU-T (1996). *P.800: Methods for subjective determination of transmission quality.* International Telecommunication Union, P.800.

ITU-T (2001a). *Communications quality of service: A framework and definitions.* ITU-T, G.1000.

ITU-T (2001b). *G.1010: End-user Multimedia QoS Categories.* International Telecommunication Union, G.1010.

Jain, R. (1991). *The Art of Computer Systems Performance Analysis: Techniques for experimental design, measurement, simulation, and modelling.* John Wiley and Sons Inc, USA.

Jakes, W. (1974). *Microwave Mobile Communications.* John Wiley and Sons, New York, NY, USA.

Jang, J. S. R., Sun, C. T. & Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing.* Prentice-Hall Inc., Upper Saddle River, NJ, USA.

Johnson, D., Perkins, C. & Arkko, J. (2004). *Mobility Support in IPv6.* IETF, RFC 3775.

Kandasamy, W. B. V. & Smarandache, F. (2003). *Fuzzy Cognitive Maps and Neutrosophic Cognitive Maps.* Xiquan, Pheonix, Arizona, USA.

Kanter, T. G. (2003). 'Going Wireless, Enabling an Adaptive and Extensible Environment'. *Mobile Networks and Applications* **8**(37):37–50.

Kassar, M., Kervella, B. & Pujolle, G. (2008). 'An overview of vertical handover decision strategies in heterogeneous wireless networks'. *Computer Communications* **31**(10):2607–2620.

Kawadia, V. & Kumar, P. R. (2005). 'A cautionary perspective on cross-layer design'. *IEEE Wireless Communications* **12**(1):3–11.

Kaymak, U. & van Nauta Lemke, H. (1998). 'A sensitivity analysis approach to introducing weight factors into decision functions in fuzzy multicriteria decision making'. *Fuzzy Sets and Systems* **97**(2):169–182.

Klir, G. J. (2006). *Uncertainty and information: foundations of generalized information theory.* John Wiley and Sons Inc, New Jersey, USA.

Klir, G. J. & Bo, Y. (1995). *Fuzzy Sets and Systems: theory and applications.* Prentice-Hal, Inc., NJ, USA.

Klir, G. J. & Folger, T. A. (1988). *Fuzzy Sets, Uncertainty, and Information.* Prentice-Hal, Inc., USA.

Kosko, B. (1986). 'Fuzzy cognitive maps'. *International Journal of Man Machine Studies* **24**:65–75.

Kosko, B. (1992). *Neural Networks and Fuzzy Systems : a dynamical systems approach to machine intelligence.* Prentice Hall, Englewood Ciffs, NJ, USA.

Laird, J. E., Newell, A. & Rosenbloom, P. S. (1987). 'SOAR: an architecture for general intelligence'. *Artificial Intelligence* **33**(1):1–64.

Laramee, F. D. (2002). *A rule-based architecture using the Dempster-Shafer Theory.* In Rabin , S. (ed.), *AI Game Programming Wisdom*, pp. 358–366. Charles River Media, Hingham, Massachusetts: USA, 1st edn.

Laukkanen, M., Helin, H. & Laamanen, H. (2002). 'Supporting nomadic agent-based applications in the FIPA agent architecture'. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pp. 1348–1355, New York, NY, USA. ACM Press.

Lehr, W. & McKnight, L. W. (2003). 'Wireless Internet access: 3G vs. WiFi?'. *Telecommunications Policy* **27**(5-6):351–370.

Lescuyer, P. & Lucidarme, T. (2008). *Evolved Packet System (EPS): The LTE and SAE Evolution of 3G UMTS*. John Wiley and Sons Ltd, West Sussex, UK.

Lindgren, A., Almquist, A. & Schelen, O. (2003). 'Quality of service schemes for IEEE 802.11 wireless LANs: an evaluation'. *Mobile Network and Applications* **8**(3):223–235.

Liu, Q., Zhou, S. & Giannakis, G. B. (2006). 'Cross-layer modeling of adaptive wireless links for QoS support in heterogeneous wired-wireless networks'. *Wireless Networks* **12**(4):427–437.

Lloyd-Evans, R. (2002). *QoS in Integrated 3G Networks*. Artech House Inc., Norwoord, MA, USA.

Ma, L., Yu, F. & Randhawa, T. (2004). 'A New Method to Support UMTS/WLAN Vertical Handover Using SCTP'. *IEEE Wireless Communications* **11**(4).

Maes, M. A. & Faber, M. H. (2003). 'Issues in utility modelling and rational decision making'. In Maes , M. A. & Huyse , L. (eds.), *11th IFIP WG7.5 Working Conference on Reliability and Optimization of Structural Systems*, pp. 95–104. Balkema Publishers, pp. 95-104.

Mahonen, P., Riihijarvi, J., Petrova, M. & Shelby, Z. (2004). 'Hop-by-hop toward future mobile broadband IP'. *IEEE Communications Magazine* **42**(3):138–146.

Malyan, R. & Lenaghan, A. (2003). 'A Multi-service Architecture to Support Mobile IP Applications over Heterogeneous Wireless Networks'. In *EUROCON 2003*, vol. 1, pp. 313–315, Ljubijana, Slovenia.

Mangold, S., Choi, S., May, P., Klein, O., Hiertz, G. & Stibor, L. (2003). 'Analysis of IEEE 802.11e for QoS support in wireless LANs'. *IEEE Wireless Communications* **10**(6):40–50.

Manner, J., Toledo, A. L., Mihailovic, A., Munoz, H. L. V., Hepworth, E. & Khouaja, Y. (2002). 'Evaluation of mobility and quality of service interaction'. *Computer Networks* **38**(2):137–163.

Meiss, J. (2007). 'Dynamical systems'. *Scholarpedia* **2**(2):1629.

Minsky, M. L. (1988). *The Society of Mind*. Simon and Schuster, New New York, USA.

Mishra, A., Shin, M. & Arbaugh, W. A. (2004). 'Context Caching using Neighbor Graphs for Fast Handoffs in a Wireless Network'. In *IEEE Infocomm*, Hong Kong.

Morrison, G. (2005). 'QoS for Applications'. *BT Technology Journal* **23**(2):28–36.

Mustill, D. & Willis, P. J. (2005). 'Delivering QoS in the next generation network - a standards perspective'. *BT Technology Journal* **23**(2):48–60.

Negnevitsky, M. (2002). *Artificial Intelligence: A Guide to Intelligent Systems*. Pearson Education Ltd., Essex, UK.

Newell, A. (1991). *Unified theories of cognition*. Harvard University Press, Cambridge, MA, USA.

NS-2 (2008). 'Nsnam' [Online]. Available: `http://nsnam.isi.edu/nsnam/index.php/Main_Page`. (accessed: May 15 2008).

Papamilitiadis, K., Zisimopoulos, H., Gasparroni, M. & Liotta, A. (2004). 'User quality of service perception in 3G mobile networks'. In *IEE Telecommunications Quality of Service (QoS 2004)*, pp. 64–69, Savoy Place, London, UK. IEE.

Parziale, L., Liu, W., Matthews, C., Rosselot, N., Davis, C., Forrester, J. & Britt, D. T. (2006). *TCP/IP Tutorial and Technical Overview*. IBM, Poughkeepsie, NY, USA.

Paxson, V., Almes, G., Mahdavi, J. & Mathis, M. (1998). *Framework for IP Performance Metrics*. IETF, RFC 2330.

Perkins, C. (2002). *IP Mobility Support for IPv4*. IETF, RFC 3344.

Prasad, R. & Munoz, L. (2003). *WLANs and WPANs Towards 4G Wireless*. Artech House, UK.

Prasad, R. & Ruggieri, M. (2003). *Technology trends in wireless communications*. Artech House, UK.

Prechelt, L. (2000). 'An Empirical Comparison of Seven Programming Languages'. *Computer* **33**(10):23–29.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. (1992). *Numericl Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 2 edn.

Rapp, C. W. (2007). 'SMC: The state machine compiler' [Online]. Available: `http://smc.sourceforge.net/`. (accessed: May 7, 2007).

Razavi, B. (2008). 'Gadgets Gab at 60 Ghz'. *IEEE Spectrum* **45**(2):40–45.

Ribeiro, R. A. (1996). 'Fuzzy multiple attribute decision making: a review and new preference elicitation techniques'. *Fuzzy Sets and Systems* **78**(2):155–181.

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. & Schooler, E. (2002). *SIP: Session Initiation Protocol.* IETF, RFC 3261.

Ross, T. J. (2004). *Fuzzy Logic with Engineering Applications.* John Wiley and Sons Inc.

Ruggeri, G., Iera, A. & Polito, S. (2005). '802.11-Based Wireless-LAN and UMTS interworking: requirements, proposed solutions and open issues'. *Computer Networks* **47**(2):151–166.

Russell, S. J. & Norvig, P. (2003). *Artificial Intelligence: a modern approach.* Prentice-Hal, Inc., USA.

Sabata, B., Chatterjee, S., Davis, M., Sydir, J. J. & Lawrence, T. F. (1997). 'Taxomomy of QoS Specifications'. In *Workshop on Object-Oriented Real-Time Dependable Systems - (WORDS '97)*, p. 100, Newport Beach, California, USA. IEEE.

Saito, Y., Kuroda, M. & Ishizu, K. (2005). 'Design of Media Independent Handover Interface for Beyond 3G Terminal'. In *Wireless Personal Multimedia Communications (WPMC)*, pp. 1404–1408, Aalborg, Denmark.

Savitzky, A. & Golay, M. J. E. (1964). 'Smoothing and Differentiation of Data by Simplified Least Squares Procedures'. *Analytical Chemistry* **36**(8):1627–1639.

Schormans, J. A. & Pitts, J. M. (2004). 'So You Think You Can Measure IP QoS?'. In *IEE Telecommunications Quality of Service (QoS 2004)*, pp. 151–155, Savoy Place, London, UK. IEE.

Schulzrinne, H., Casner, S., Frederick, R. & Jacobson, V. (2003). *RTP: A Transport Protocol for Real-Time Applications.* IETF, RFC 3550.

Shafer, G. (1976). *A Mathematical Theory of Evidence.* Princeton University Press.

Shakkottai, S. & Rappaport, T. (2003). 'Cross-Layer Design for Wireless Networks'. *IEEE Communications Magazine* **41**(10):74.

Song, Q. & Jamalipour, A. (2005). 'An adaptive quality-of-service network selection mechanism for heterogeneous mobile networks'. *Wireless Communications and Mobile Computing* **5**(6):697–708.

Sousa, J. M. C. & Kaymak, U. (2002). *Fuzzy Decision Making in Modelling and Control.* World Scientific, London, UK.

Sporns, O. (2007). 'Complexity'. *Scholarpedia* **2**(10):1623.

Stallings, W. (2002). *Wireless communications and networks.* Prentice-Hal, Inc., New Jersey, USA.

Steimann, F. & Adlassnig, K. P. (1994). 'Clinical monitoring with fuzzy automata'. *Fuzzy Sets and Systems* **61**(1):37–42.

Stemm, M. & Katz, R. H. (1998). 'Vertical handoffs in wireless overlay networks'. *Mobile Network and Applications* **3**(4):335–350.

Stewart, R. (2007). *Stream Control Transmission Protocol (SCTP).* IETF, RFC 4960.

Stewart, R., Xie, Q., Tuexen, M., Maruyama, S. & Kozuka, M. (2007). *Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration.* IETF, RFC 5061.

Tozour, P. (2002). *Introduction to Bayesian networks and reasoning under uncertainty.* In Rabin , S. (ed.), *AI Game Programming Wisdom*, pp. 345–357. Charles River Media, Hingham, Massachusetts: USA, 1st edn.

Triantaphyllou, E. (2000). *Multi-Criteria Decision Making Methods: A Comparative Study.* Kluwer Academic Publishers, Netherlands.

van Waveren, J. P. (2001). 'The Quake III Arena Bot'. Master's thesis.

Vin, H. M. (2005). 'Multimedia systems research: a retrospective OR whatever happened to all that QoS research?'. In *NOSSDAV '05: Proceedings of the international workshop on Network and operating systems support for digital audio and video*, pp. 25–26, New York, NY, USA. ACM Press.

Wagner, F., Wagner, T. & Wolstenholme, P. (2004). 'Closing the Gap Between Software Modelling and Code'. In *11th IEEE International Conference and Workshop on the Engineering*

of Computer-Based Systems (ECBS'04), p. 52, Washington, DC, USA. IEEE Computer Society.

Wang, H. J., Giese, J. & Katz, R. H. (1999). 'Policy-Enabled Handoffs Across Heterogeneous Wireless Network'. In *2nd IEEE Workshops on Mobile Computing and Applications (WMCSA 1999)*, New Orleans, LA, USA.

Wang, S. Y., Chou, C. L., Huang, C. H., Hwang, C. C., Yang, Z. M., Chiou, C. C. & Lin, C. C. (2003). 'The design and implementation of the NCTUns 1.0 network simulator'. *Computer Networks* **42**(2):175–197.

Wei, Q., Farkas, K., Prehofer, C., Mendes, P. & Plattner, B. (2006). 'Context-aware handover using active network technology'. *Computer Networks: The International Journal of Computer and Telecommunications Networking* **50**(15):2855–2872.

Wilson, A., Lenaghan, A. & Malyan, R. (2005). 'Optimising Wireless Access Network Selection to Maintain QoS in Heterogeneous Wireless Environments'. In *Wireless Personal Multimedia Communications (WPMC) 2005*, Aalborg, Denmark.

Yeh, C.-H., Deng, H. & Chang, Y.-H. (2000). 'Fuzzy multicriteria analysis for performance evaluation of bus companies'. *European Journal of Operational Research* **126**(3):459–473.

Ylianttila, M., Makela, J. & Pahlavan, K. (2005). 'Analysis of handoff in a location-aware vertical multi-access network'. *Computer Networks* **47**(2):185–201.

Ylianttila, M., Pande, M., Makela, J. & Mahonen, P. (2001). 'Optimization scheme for mobile users performing vertical handoffsbetween IEEE 802.11 and GPRS/EDGE networks'. In *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, vol. 6, pp. 3439–3443, San Antonio, TX, USA.

Yuan, Y., Feldhamer, S., Gafni, A., Fyfe, F. & Ludwin, D. (2002). 'The development and evaluation of a fuzzy logic expert system for renal transplantation assignment: Is this a useful tool?'. *European Journal of Operational Research* **142**(1):152–173.

Zadeh, L. A. (1965). 'Fuzzy sets'. *Information and Control* **8**:338–353.

Zhang, W. (2004). 'Handover Decision Using Fuzzy MADM in Heterogeneous Networks'. In *IEEE Wireless Communications and Networking Conference (WCNC 2004)*, Atlanta, USA.

Zhang, W., Jahnert, J. & Dolzer, K. (2003). 'Design and evaluation of a handover decision strategy for 4th generation mobile networks'. In *57th IEEE Vehicular Technology Conference (VTC 2003 Spring)*, Jeju, Korea.

Zheng, D. & Kainz, W. (1999). 'Fuzzy rule extraction from GIS data with a neural fuzzy system for decision making'. In *GIS '99: Proceedings of the 7th ACM international symposium on Advances in geographic information systems*, pp. 79–84, Kansas City, Missouri, United States. ACM Press.

Zhu, F. & McNair, J. (2006). 'Multiservice Vertical Handoff Decision Algorithms'. *EURASIP Journal on Wireless Communications and Networking* **2006**(Article ID 25861):1–13.

Zimmermann, H. J. & Zysno, P. (1980). 'Latent Connectives in Human Decision Making'. *Fuzzy Sets and Systems* **4**(1):37–51.

Appendix A

Prototype Components

The following sections are details of the prototype implemenations of agent code, behaviours, and fuzzy decision-making (FDM) components. Full program code is made available at: `http://intstack.wikidot.com/research`.

A.1 Agent Models

The following are models of the Agent prototypes for the Subsumption agent design (used for the FDMbrain), and finite-state machine based agent (used in FSM prototype).

Subsumption agent

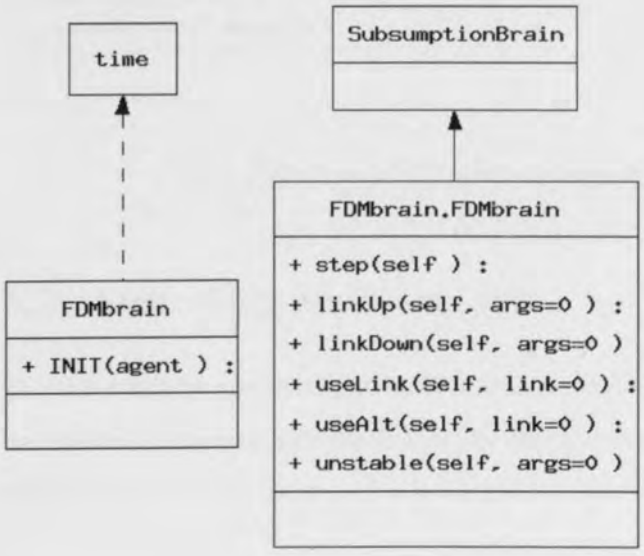


Figure A.1: UML class diagram of FDMbrain.py

Finite-state agent

The FSM agent is a subclass of Brain and defined in the module **FSMcontrol.py**. It exposes methods to call agents in the FSM code. The FSM model is generated from State Machine Compiler scripts (.sm) into Python. Upon initialising the FSM model, FSMcontrol class provides calls to the FSM transition from its own methods.

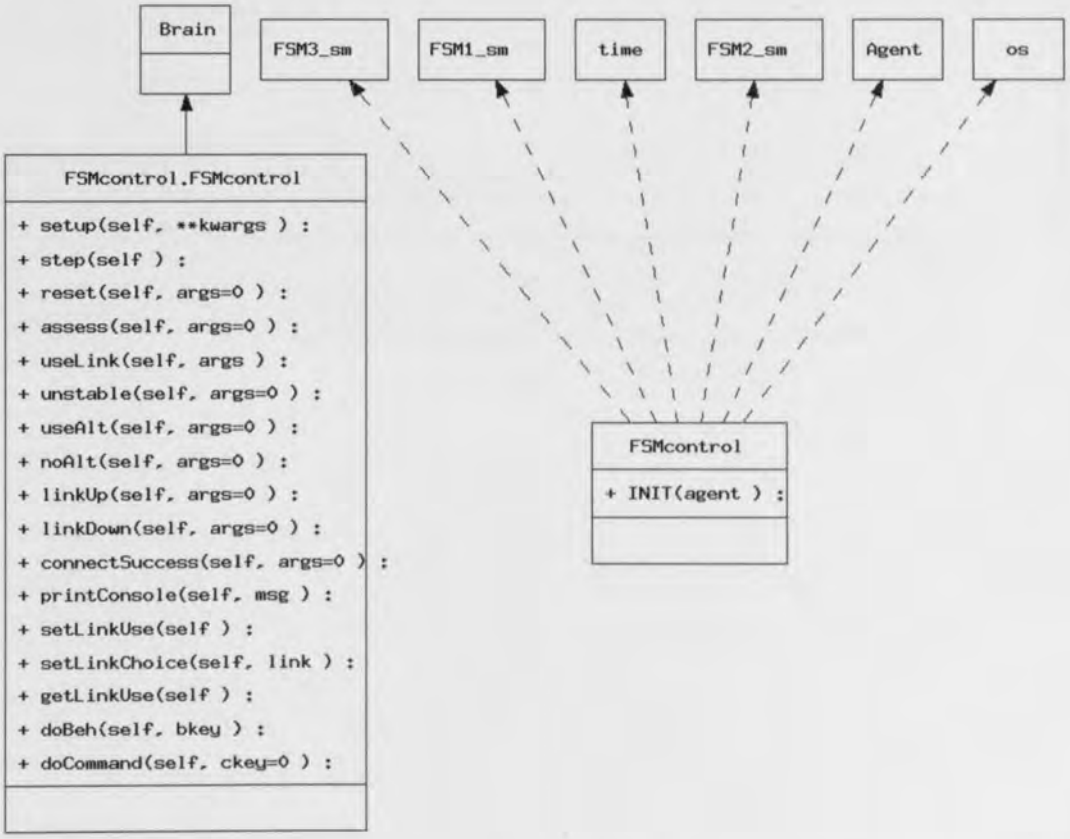


Figure A.2: UML class diagram of FSMcontrol.py

A.2 Finite-State Machines Logic

The following are generated graphs of finite-state machine logic used in the prototype implementations. They were generated using the debug option from State Machine Compiler Rapp (2007).

FSM-1:

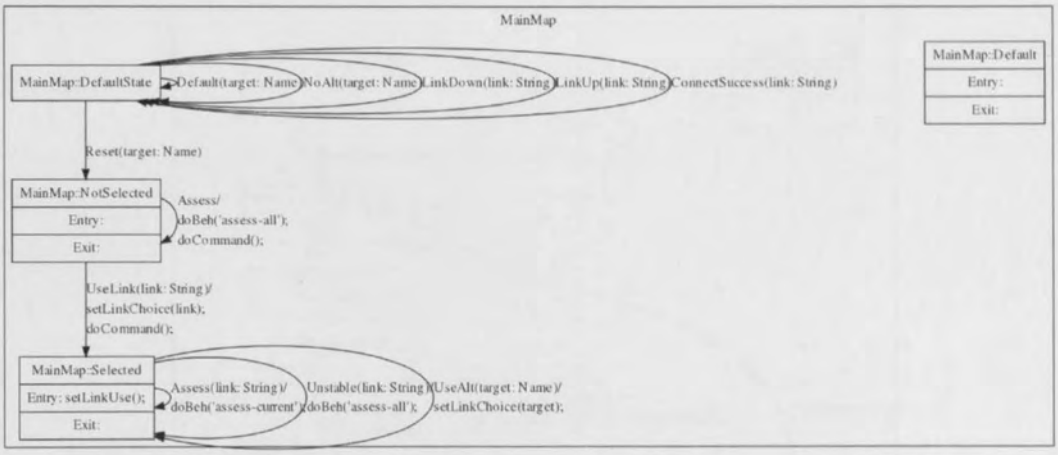


Figure A.3: Detailed graph of FSM1 states and transitions

FSM-2

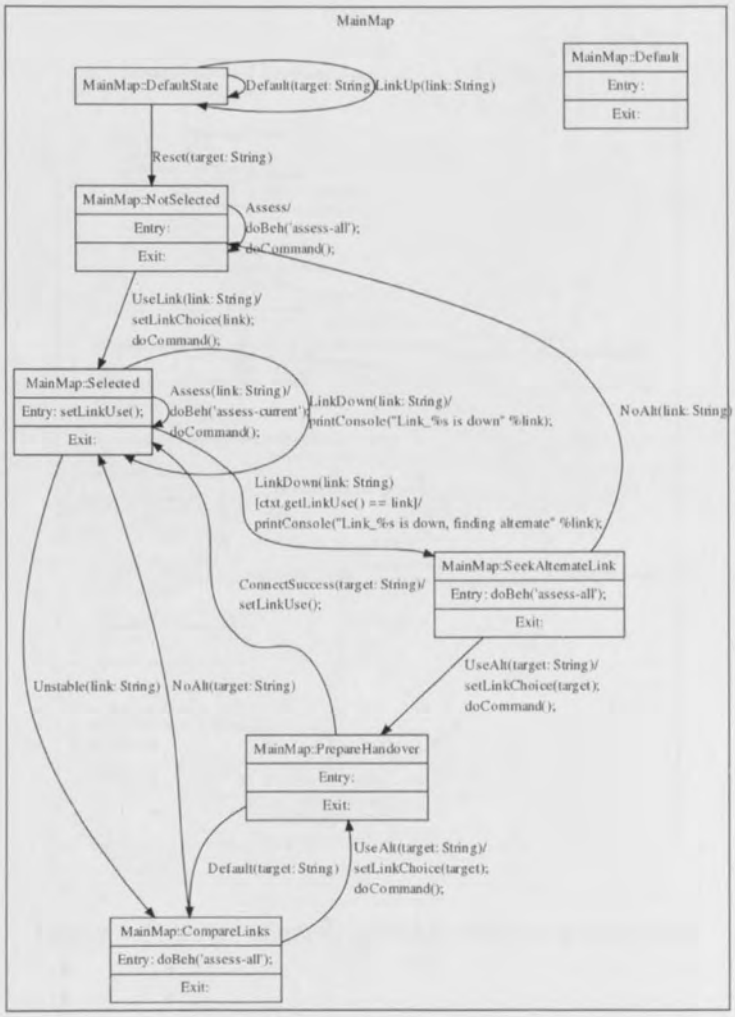


Figure A.4: Detailed graph of FSM2 states and transitions

FSM-3

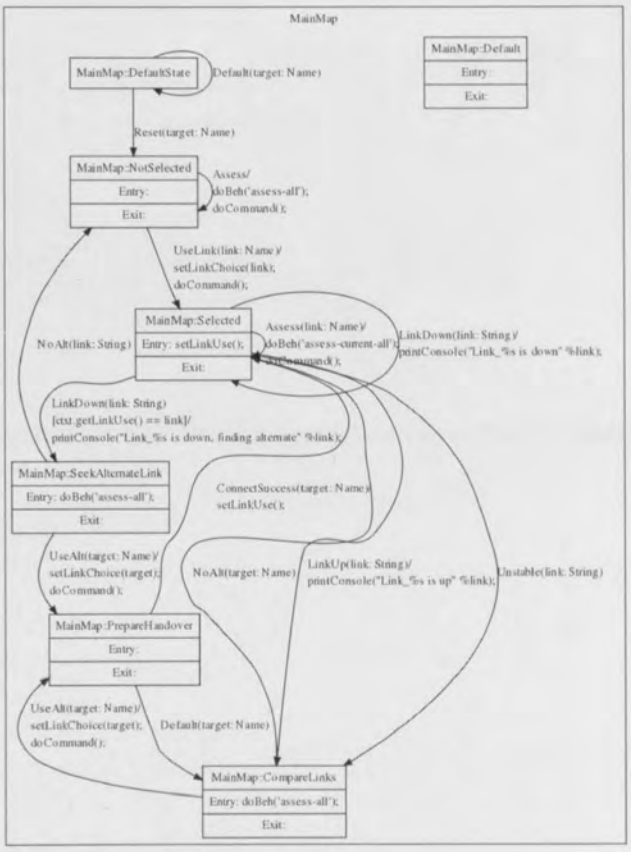


Figure A.5: Detailed graph of FSM3 states and transitions

A.3 Fuzzy Decision Making and Behaviours

The following is a UML diagram of the classes in the module **fdm.py**. It contains the classes for FDM (DecisionMatrix, Criteria) and behaviours that uses FDM (AssessAll, AssessCurrent, AssessCurrentAll).

Figure A.6: UML class diagram of Python module fdm.py

Appendix B

Research Methods

B.1 Simulation Tools

Tools for wireless simulation were considered in relation to criteria (figure B.1). All except SimPy provide models of specific physical wireless and networking protocols, such as TCP/IP stacks, wireless local-area networks (WLAN), and cellular (UMTS). NS-2 has more contributed models than NCTUns, as it more widely used in networking research. Opnet also has a community providing contributed protocol models (though requires a valid support license).

	Option			
Criteria	NCTUns	NS-2	Opnet	SimPy
Models detailed wireless protocols	✔	✔	✔	✘
Shows detailed interaction between inputs	✘	✘	✘	✘
Can include external programs (Python)	✔	✔	✘	✔
Can run customised protocols	✔	✔	✔	✔

Figure B.1: Comparative matrix for wireless simulation tools and criteria.

Inclusion of non-protocol programs into the simulation environment is possible in all options, but Opnet would require a rewrite in to its modelling method and Proto-C (a programming language similar to C). SimPy allows Python programs to be used in simulation process, and other programs source could be included by interface coding libraries. NS-2 can provide this similar inclusion of programs source (although not running pre-compiled programs). NC-

TUns can run other pre-compiled binaries in simulation time due the kernel modifications and timing mechanisms, and limited interaction through an inter-process communication (IPC) API Wang et al. (2003).

All options provide capabilities for designing custom protocol models, some more favourable than others. Being a generic simulation tool, SimPy would require too much effort from first-principles and simply repeating existing work of other network simulation tools. Though for higher-level abstraction or problems with restricted scope, SimPy could be beneficial. Although Opnet allows customised protocols, they must follow specific modelling approach. Both NCTUns and NS-2 use C++ and the basis for protocols implementations, though NS-2 uses a split programming model of C++ and OTcl/Tcl.

B.2 Experimental Design

Experiments are defined for analytical study of decision components–the FDM algorithms. Simulations are first performed using initial pilot study of SimPy (set S1). More realistic modelling of wireless channel, mobility, and application traffic is performed using NS-2 (set S2). The details of parameters and metrics for each set are defined in this section; results and observations of test-runs are reported in the subsequent chapter.

The purpose of experimental design is to identify parameters, metrics, and variables involved in performing evaluative testing. Factors are configured as sets of experiments to assess performance of the prototypes. Parameters and factors are inputs, and metrics (outputs) are the dependent variables. Analytical and simulation setups are defined as the basis for test cases in the following sections.

The test descriptions uses various configuration of parameters, metrics and factors for later analysis. Test specifications are based on factorial principles Jain (1991). Commonly used in *design of experiments*, factorial design uses combinations of factors (a variable that has discrete levels) to create experiment sets. A common design is two-level full-factorial: 2^k, where 2 is the number of levels and k is the factors.

A test set defines a suite of test cases (figure B.2), such as the analytical and simulations sets. Each test case defines parameters and factors to be varied, and metrics of interest. The variations in factors creates a number of experiments. An experiment is the lowest level, representing an isolated configuration or treatment. Therefore, a test case defines the point of interest for parameters, factors, levels, and metrics. Subsequent test cases can be defined to try

different factors or levels.

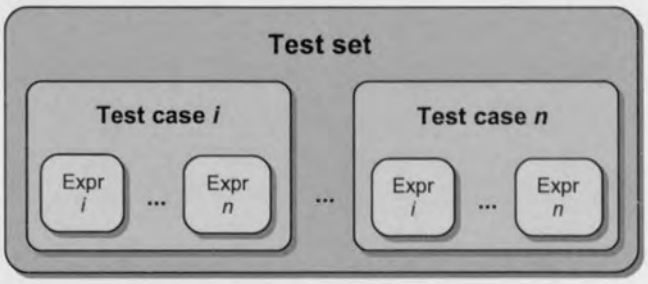


Figure B.2: Hierarchy of test sets.

Simulations are run to answer a point of interest, configured by input parameters that result in output metrics. Table B.1 shows input and output options for simulation configurations. Metrics calculations or observations performed for each simulation run. These include observations of performance values used as inputs for HAL assessment, to outputs of the decision module. Simulation parameters are the settings used for each simulation run. Certain parameters of interest are varied to compare differences or effects on output metrics. These are factors that set at different levels, such as having two levels of workload: voice and data traffic.

Type	*Description*
Metric	*Application performance:* observations of application statistics over duration of the simulation. *Link rankings:* observations from decision-making that give the rank and value of link during the simulation. *Selection choice* over time.
Parameter	*Roaming pattern* and host settings. *Link settings* determining network availability, positioning. *Number of wireless nodes*: this includes the MH used as the subject of the assessment. *Workloads*, or requests for service: different applications have parameters for QoS requirements and flow characteristics. *Decision criteria* or policy: QoS criteria used for assessment.
Factor	*Workloads*: the levels of workload could change by having a single application type, and varying the type among experiments. More experiments can be run by including simultaneous applications. This is the more complex scenario. *Decision criteria*: changing the policy using different settings of importance for workload and user preferences. Some policy such as flow types, have some common requirements or importance levels. *Roaming*: a static position, moving-in and moving-out patterns will be used. Static positioning means that the client device will stay in one position throughout the session. Moving-in defines the client (subject) moving into a WiFi area of coverage. Moving-out is the reverse path from moving-in pattern.

Table B.1: Simulation output metrics, input parameters, and factors for test sets.

Appendix C

Experimental Setup

The following sections describe the detail of the settings used for simulations and experiments. Where full code listings are not displayed, full simulation driver programs and utility scripts are available at: `http://intstack.wikidot.com/research`.

C.1 Analysis Settings

C.2 Simulations Settings

The following settings were defined for NS-2 version 2.31 with *ns-miracle*[1] 1.2.1 installed. The following table is the common parameters used by WLAN and UMTS simulations.

Parameter name	Parameter value
Terrain	1000 m x 1000 m
Duration	250 seconds
Number of mobile nodes	1
CBR sources	2*
Data payload	160 bytes
Packet rate	50 packets/sec

Table C.1: Shared parameters for NS-2 simulations (*MH and CH)

The following table is link parameter settings for WLAN and UMTS simulations.

[1] Available from: `http://www.dei.unipd.it/wdyn/?IDsezione=3966`

Parameter	wlan-test-1	umts-test-1
Frequency	2.437-GHz	
Transmit power	15 dBm	21 dBm
Propagation model	MIRACLE two-ray ground	FreeSpace
MAC	802_11g (multirate)	UMTS/MAC/ME
MAC bit rate	6 Mbps	2 Mbps

Table C.2: Link parameters for NS-2 simulations

The following table is the roaming settings used for the three scenarios. It refers to the MH in the scene.

Parameter	scene-1	scene-2	scene-3
Movement model	-	Fixed waypoint	Fixed waypoint
Speed	-	1.564 m/sec	1.564 m/sec
Start position	(100, 100)	(100, 100)	(300, 300)
Direction	-	(300, 300)	(100, 100)

Table C.3: Roaming parameters for NS-2 simulations

C.3 Simulation Metric Calculations

The metrics used for simulation cases are calculated from NS-2 (with ns-miracle) tracefiles. An entry in the tracefile has a time index for receive or send event from a layer in the node stack. For example, the following extract is a transmission from an application of a wireless node:

```
448 s 2.200000000 2 CBR PRT   1Mbps  0.0.0.0 --> 255.255.255.254  SRC
      0.0.0.0:0 DST 2.0.1.0:0 SN=60 TS=2.200000 SZ=160 RFTT
     =0.010658
449 s 2.200000000 2 PRT IPR   1Mbps  0.0.0.0 --> 255.255.255.254  SRC
      0.0.0.0:0 DST 2.0.1.0:0 SN=60 TS=2.200000 SZ=160 RFTT
     =0.010658
450 s 2.200000000 2 IPR IP1   1Mbps  0.0.0.0 --> 1.0.1.1  SRC
     0.0.0.0:0 DST 2.0.1.0:0 SN=60 TS=2.200000 SZ=180 RFTT
     =0.010658
```

```
s 2.200000000 2 IP1 LL1   1Mbps  1.0.1.2 --> 1.0.1.1  SRC
    1.0.1.2:0 DST 2.0.1.0:0 SN=60 TS=2.200000 SZ=180 RFTT
    =0.010658
s 2.200075000 2 LL1 PH1   6Mbps  1.0.1.2 --> 1.0.1.1  SRC
    1.0.1.2:0 DST 2.0.1.0:0 SN=60 TS=2.200000 SZ=214 RFTT
    =0.010658
s 2.200075000 2 PH1 CH   6Mbps  1.0.1.2 --> 1.0.1.1  SRC
    1.0.1.2:0 DST 2.0.1.0:0 SN=60 TS=2.200000 SZ=214 RFTT
    =0.010658
```

Where fields are defined as[2]: `flag time nodeid fromlayer tolayer <layer data...>`

- Flag is 's' for send, 'r' for receive, and 'D' for drop.
- Time in seconds of the event.
- Nodeid is the id of the node.
- Fromlayer and tolayer indicate the protocols layers direction of the packet.

From this data, the python script **tracefile_parser.py** is run to extract QoS related metrics for the selected host. In these methods, an average is taken for packets received by the application layer (CBR) in the interval period (one second). The following method (C.1) extracts the signal level from received packets at the link layer.

Listing C.1: Signal calculation method

```
27 def sigSample(ts, sig):
28     '''just take the mean from sample period - use smoothing?'''
29     ts_str = re.split('\.', str(ts))
30     ts_sec = eval(ts_str[0])
31     if G.delta_time['sig'] == 0:
32         G.delta_time['sig'] = ts_sec
33     if (ts > G.ctime):# and (ts <= ts+G.interval):
34         if (ts >= G.interval+G.delta_time['sig']) and (size(G.
               avg_metrics['sig']) > 0):
35
36             av_sig = mean(G.avg_metrics['sig'])
37             ofiles['av_sig'].write('%s %s\n' %(G.delta_time['sig'],
                   av_sig))
```

[2]As defined by https://mail.dei.unipd.it/pipermail/nsmiracle-users/2007-August/000023.html

```
38              G.delta_time['sig'] += G.interval
39              G.avg_metrics['sig'] = array(sig)
40          else:
41              G.avg_metrics['sig'] = append(G.avg_metrics['sig'], sig)
```

To calculate delay and jitter, the method from RTP Schulzrinne et al. (2003) is used in C.2. It uses two consecutive received packets to calculate the time differences. A smoothing parameter is used for jitter as per RTP, but this is not verified and remains unused

Listing C.2: Delay calculation method

```
43 def delayCalc(rj, sj):
44     '''calc jitter (using eg from http://wiki.wireshark.org/
           RTP_statistics)'''
45     #D(i,j) = (Rj - Ri) - (Sj - Si) = (Rj - Sj) - (Ri - Si)
46     dif = (rj-G.ri)-(sj-G.si)
47     lat = rj-sj
48     #J(i) = J(i-1) + (|D(i-1,i)| - J(i-1))/16
49     jit = G.jit_prev + ((dif - G.jit_prev)/16)
50     G.si = sj    # current times to prev times
51     G.ri = rj
52     G.jit_prev = jit
53     return (lat, jit)
```

From each packet received by the application layer, the instantaneous packet delays calculated from the previous method collected. The average of these values for a sampling duration is written to the output file. Listing C.3 shows the code for calculating average delay per second.

Listing C.3: Delay average method

```
55 def commonParse(line):
56     '''pass in a line a parse for stats'''
57     data = re.split('\s+', line)
58     ts = eval(data[1])    # the receive time (current packet)
59     #---- sequence numbers
60     s_seq = re.search('[Ss][Nn]=\d+', line) # match the sequence no.
61     if s_seq:
62         d = re.split('=', s_seq.group())
63         seq = eval(d[1])
64         #print '%f %f' %(ts,seq)
65         ofiles['seq'].write('%s %s\n' %(ts, seq))
```

```
66      #--------now parse line for a packet timestamp value
67      s_ts = re.search('[Tt][Ss]=\d+.\d+', line)
68      if s_ts:
69          d = re.split('=', s_ts.group())
70          p_ts = eval(d[1])     # packet time (current packet)
71          i_lat, i_jit = delayCalc(ts, p_ts)     # instantaneos values
72          ts_str = re.split('\.', d[1])
73          ts_sec = eval(ts_str[0])
74          if G.delta_time['delay'] == 0:
75              G.delta_time['delay'] = ts_sec
76          if (ts > G.ctime):
77              #------ update samples array with instant. vals, save to
                    file, and move on
78              ofiles['i_jit'].write('%s %s\n' %(ts, i_jit))
79              ofiles['i_lat'].write('%s %s\n' %(ts, i_lat))
80              if (ts >= G.delta_time['delay']+G.interval) and (size(G.
                    avg_metrics['lat']) > 0):
81                  #------ calc and save average for sample period
82                  av_lat = mean(G.avg_metrics['lat'])
83                  av_jit = mean(G.avg_metrics['jit'])
84                  ofiles['av_lat'].write('%s %s\n' %(G.delta_time['
                        delay'], av_lat))
85                  ofiles['av_jit'].write('%s %s\n' %(G.delta_time['
                        delay'], av_jit))
86                  G.delta_time['delay'] += G.interval
87                  G.avg_metrics['lat'] = array(i_lat)
88                  G.avg_metrics['jit'] = array(i_jit)
89              else:
90                  G.avg_metrics['lat'] = append(G.avg_metrics['lat'],
                        i_lat)
91                  G.avg_metrics['jit'] = append(G.avg_metrics['jit'],
                        i_jit)
```

Appendix D

Additional Results

These are the supplemental results data and plots accompanying the test cases. Experiment setup scripts are made available at: `http://intstack.wikidot.com/research`.