AN INTEGRATED PERSONAL MOBILITY SERVICES ARCHITECTURE

By

Anirach Mingkhwan B.Sc., M.Sc.

A thesis submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy

Liverpool John Moores University
School of Computing and Mathematical Sciences

May, 2004
Acknowledgement

I am truly indebted to a number of people, each of whom has contributed to making this thesis possible.

First, I would like to express my deep gratitude to my supervisor, Prof. Madjid Merabti. Madjid has not only been a first class supervisor but also a friend to my family and I. He provided me with the opportunity to fulfil my dream, believed in my ability since we first met, and has always been ready with a constructive word during the course of my research.

I would also like to express my sincere thanks and deep appreciation to my second supervisor, Dr. Bob Askwith for his continuous precious support and assistance during the course of this research. I feel sure that this thesis would never have come into existence without his help, patience and dedication.

The emotional support, love and care are always important to keep one on his feet. Thanks to every single person for being nice and helpful, Omar Abuelma'atti, Peerasak Serikul, Tikamporn Thaweedech, Yuphin Ruangrit, Cherdchai Phosri, Nutthaporn Ruchikachorn, Kantavich Limsonwan, Paul Fergus, Gurleen Arora, Chris Bewick, John Haggerty, Tom Berry, David Llewellyn-Jones and Mei-Lan Chen. You are great colleagues and friends.

I am truly indebted to Wipha Sunthornmeesathian, Sompat Benjachaiyaporn, Sangob Kongka and Penkae Prajongjai for their kindness and incontrovertible belief in me. Finally, my thanks are due to my wife Chutarat and my own family. Their unfading interests in my efforts, and faith in my ability, have proved a continual source of inspiration to me.

I dedicate this thesis to my beloved parents.
Abstract

Next generation network computing systems are expected to provide a wide variety of Personal Mobility Services to users anytime, via a range of devices and anywhere in the world. Personal Mobility Services require an advanced architecture that integrates supported protocols, mechanisms and specialised functions to dynamically reconfigure applications. It should also integrate services that can accommodate environmental changes as the user roams from one location to another. Personal Mobility also has to deal with various Information Spaces where information is not just coming from the Internet but also from other nearby equipments and the environment around them. In a next generation network, Information and services should be available to mobile users in the same analogy as high street shopping.

Our research introduces an architecture that allows user nodes to perform specific tasks by operating as a centric-composer of the available services from both the local environment and the infrastructure network. The proposed Integrated Personal Mobility Services Architecture (IPMSA) incorporates a framework where the entire wireless nodes are in the same network and only one hop away from the public access point. For those one world wireless requirements, we implemented a Global Wireless Framework to complete the vision of IPMSA architecture. The IPMSA architecture will enable application on a network node in the Global Wireless Framework to use/provide services across the infrastructure network in a seamless manner.

In this global wireless network mobile devices need to communicate in an ad hoc manner and have the ability to perform Layer 2 multi-hop coordination. However, current protocol specifications including IEEE802.11 lack a definition of multi-hop support. This constitutes a challenge for the delivery and sharing of services in the integrated framework. We proposed a GWF Ad Hoc Bridge to allow clients in the Global Wireless Framework to provide multi-hop communication links to other nodes in the network. This extends the node communication range and provides more services accordingly.

We implemented our GWF Ad Hoc Bridge in the NS-2 network simulator to evaluate the performance of the Global Wireless Framework multi-hop communication. We also compare the performance of our bridge with the existing Ad Hoc routing protocols to get a better view of this novel approach. The services composition prototype that we also implemented proves that the concept of service integration provide better range of services in an interoperable heterogeneous domain. Joining these two concepts allows services to be offered from both the infrastructure network side and the local environment. In our future work, intelligent services composition and assistance mechanisms would be integrated to make applications built on this architecture more adaptable to the mobile user lifestyle.
# Table of Contents

Acknowledgement ....................................................................................... II  
Abstract ....................................................................................................... III  
Table of Contents........................................................................................ IV  
Table of Figures .......................................................................................... VI  
Table of Tables........................................................................................... VII  
List of Acronyms....................................................................................... VIII  

Chapter One: Introduction ......................................................................... 1  
1.1 The Future Network Environment.......................................................... 1  
1.1.1 Information Space ........................................................................... 2  
1.1.2 Network Structure.......................................................................... 4  
1.1.3 Services Classification: Structured and Unstructured Services ........ 8  
1.1.4 Services Composition....................................................................... 9  
1.2 Scope, Aims and Objectives of this Thesis............................................ 12  
1.3 Novel Aspects of this Work............................................................... 12  
1.4 Summary ............................................................................................. 14  
1.5 Thesis Structure................................................................................... 14  

Chapter Two: Related Works and Challenges.......................................... 16  
2.1 Architecture for Mobile Computing ...................................................... 16  
2.1.1 Service and Services Discovery Projects........................................... 17  
2.1.2 Service Integration Projects............................................................ 22  
2.1.3 Service Issues Summary.................................................................. 24  
2.2 Mobile Computing Networking .......................................................... 25  
2.2.1 Mobile Ad Hoc Network ................................................................ 26  
2.2.1.1 Address Autoconfiguration...................................................... 27  
2.2.1.2 Multi-Hop Communication.................................................... 29  
2.2.2 Mobile Ad Hoc Node and Internet Integration................................. 31  
2.2.3 Network Issues Summary............................................................... 34  
2.3 Challenges and the Way Forward ....................................................... 35  
2.4 Summary ............................................................................................. 37  

Chapter Three: Integrated Personal Mobility Services Architecture.... 38  
3.1 Architecture Requirements ............................................................... 39  
3.2 Architecture Overview......................................................................... 41  
3.3 Key Components of the Architecture................................................... 44  
3.3.1 Global Wireless Framework ............................................................ 44  
3.3.1.1 Automatic Address Configuration............................................ 46  
3.3.1.2 A GWF Ad Hoc Bridge............................................................ 46  
3.3.1.3 Global Wireless Network and Internet Integration..................... 48  
3.3.2 Services Integration.......................................................................... 49  
3.4 Summary ............................................................................................. 51
# Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Personal Mobility and devices</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Information Flow in Personal Mobility Information Space</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Node Situations in Information Spaces</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>Node-Centric Networking</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>Dynamic Services Composition</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>Device Services Utilisations</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>Virtual Appliances - Intercom</td>
<td>11</td>
</tr>
<tr>
<td>8</td>
<td>Wireless Networks Mapped to Ad Hoc Networking</td>
<td>26</td>
</tr>
<tr>
<td>9</td>
<td>The Integrated Personal Mobility Services Architecture</td>
<td>42</td>
</tr>
<tr>
<td>10</td>
<td>Overview of Global Wireless Framework</td>
<td>45</td>
</tr>
<tr>
<td>11</td>
<td>Multi-Hop Communication in Global Wireless Framework</td>
<td>48</td>
</tr>
<tr>
<td>12</td>
<td>Home Appliances Services Composition Concept</td>
<td>50</td>
</tr>
<tr>
<td>13</td>
<td>Flow Diagram of IP Configuration in GWF</td>
<td>55</td>
</tr>
<tr>
<td>14</td>
<td>MAXHOP Communication Range</td>
<td>59</td>
</tr>
<tr>
<td>15</td>
<td>IEEE802.11 MAC Wireless Distribution System Frame</td>
<td>59</td>
</tr>
<tr>
<td>16</td>
<td>Normal ARP Frame Format</td>
<td>60</td>
</tr>
<tr>
<td>17</td>
<td>Global Wireless Framework ARP Frame Format</td>
<td>61</td>
</tr>
<tr>
<td>18</td>
<td>Example GWF-ARP Table</td>
<td>63</td>
</tr>
<tr>
<td>19</td>
<td>A GWF Ad Hoc Bridge Algorithm Flowchart</td>
<td>65</td>
</tr>
<tr>
<td>20</td>
<td>Global Wireless Framework &amp; Mobile IP</td>
<td>67</td>
</tr>
<tr>
<td>21</td>
<td>Client Discovery and Services Mapping</td>
<td>70</td>
</tr>
<tr>
<td>22</td>
<td>Distributed Semantic Unstructured Services (DiSUS) Framework</td>
<td>72</td>
</tr>
<tr>
<td>23</td>
<td>DiSUS Framework Operation</td>
<td>73</td>
</tr>
<tr>
<td>24</td>
<td>HAIU Operation</td>
<td>75</td>
</tr>
<tr>
<td>25</td>
<td>NS-2 Wireless Header</td>
<td>80</td>
</tr>
<tr>
<td>26</td>
<td>NS-2 Link-Layer Header</td>
<td>80</td>
</tr>
<tr>
<td>27</td>
<td>NS-2 Schematic of a Wired Node</td>
<td>81</td>
</tr>
<tr>
<td>28</td>
<td>NS-2 Schematic of a Mobile Node</td>
<td>82</td>
</tr>
<tr>
<td>29</td>
<td>NS-2 Schematic of GWF Node</td>
<td>85</td>
</tr>
<tr>
<td>30</td>
<td>NS-2 DumbAgent Received</td>
<td>86</td>
</tr>
<tr>
<td>31</td>
<td>NS-2 GWF-ARP Header Detail</td>
<td>86</td>
</tr>
<tr>
<td>32</td>
<td>NS-2 GWF-ARP gwfnexthop() Function Detail</td>
<td>87</td>
</tr>
<tr>
<td>33</td>
<td>NS-2 GWF-Link Layer Packet Classification</td>
<td>88</td>
</tr>
<tr>
<td>34</td>
<td>GWF-MAC Multi-hop Handler</td>
<td>88</td>
</tr>
<tr>
<td>35</td>
<td>GWF-MAC Header File</td>
<td>89</td>
</tr>
<tr>
<td>36</td>
<td>Schematic Diagram of NS-2 MobileNode/MIPBS</td>
<td>91</td>
</tr>
<tr>
<td>37</td>
<td>Schematic Diagram of NS-2 GWF/MIPBS</td>
<td>92</td>
</tr>
<tr>
<td>38</td>
<td>Experimental Environment of IPMSA Prototype</td>
<td>94</td>
</tr>
<tr>
<td>39</td>
<td>IPMSA Client framework</td>
<td>95</td>
</tr>
<tr>
<td>40</td>
<td>DiSUS Prototype System Configuration</td>
<td>96</td>
</tr>
<tr>
<td>41</td>
<td>Services Composition Implemented Prototype</td>
<td>98</td>
</tr>
<tr>
<td>42</td>
<td>Profile Creation User Interface</td>
<td>98</td>
</tr>
<tr>
<td>43</td>
<td>Profile Selection and Invocation User Interface</td>
<td>99</td>
</tr>
<tr>
<td>44</td>
<td>Static Scenario Simulation</td>
<td>103</td>
</tr>
<tr>
<td>45</td>
<td>End2End Cumulative Comparisons of Static Scenario</td>
<td>105</td>
</tr>
<tr>
<td>46</td>
<td>Communication Path Throughput Comparisons in Static Scenario</td>
<td>106</td>
</tr>
<tr>
<td>47</td>
<td>Dynamic Scenario Simulation</td>
<td>108</td>
</tr>
<tr>
<td>48</td>
<td>End2End Protocols Dynamic Scenario Comparison</td>
<td>110</td>
</tr>
<tr>
<td>49</td>
<td>Communication Path Throughput Comparisons in Dynamic Scenario</td>
<td>111</td>
</tr>
<tr>
<td>50</td>
<td>GWF Communication Path</td>
<td>113</td>
</tr>
<tr>
<td>51</td>
<td>Contributions of our IPMSA work</td>
<td>122</td>
</tr>
</tbody>
</table>
# Table of Tables

Table 1 Private IP Address Space .............................................................................................................. 54
Table 2 The GWF-ARP Frame to Table Mapping .......................................................................................... 62
Table 3 Static Scenario Simulations Parameters ............................................................................................. 102
Table 4 Static Scenario Experiment Statistic .................................................................................................. 104
Table 5 Dynamic Scenario Simulations Parameters ......................................................................................... 107
Table 6 Dynamic Scenario Experiment Statistic ............................................................................................. 109
Table 7 Protocol Statistics Comparison ......................................................................................................... 114
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3G</td>
<td>Third Generation Mobile Communication Systems.</td>
</tr>
<tr>
<td>ACK</td>
<td>Acknowledgement</td>
</tr>
<tr>
<td>AOAC</td>
<td>Always-On Always-Connected</td>
</tr>
<tr>
<td>AODV</td>
<td>Ad hoc On Demand Distance Vector</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>APIPA</td>
<td>Automatic Private IP Addressing</td>
</tr>
<tr>
<td>ARP</td>
<td>Address Resolution Protocol</td>
</tr>
<tr>
<td>Bluetooth SDP</td>
<td>Bluetooth Service Discovery Protocol</td>
</tr>
<tr>
<td>BSD</td>
<td>Berkeley Software Distribution</td>
</tr>
<tr>
<td>BSS</td>
<td>Basic Service Set</td>
</tr>
<tr>
<td>CBR</td>
<td>Constant Bit Rate</td>
</tr>
<tr>
<td>CH</td>
<td>Correspondent Host</td>
</tr>
<tr>
<td>COA</td>
<td>Care-of-address</td>
</tr>
<tr>
<td>CORBA</td>
<td>Common Object Request Broker Architecture</td>
</tr>
<tr>
<td>CSMA/CA</td>
<td>Carrier Sense Multiple Access/Collision Avoidance</td>
</tr>
<tr>
<td>CTS</td>
<td>Clear To Send</td>
</tr>
<tr>
<td>DA</td>
<td>Directory Agents</td>
</tr>
<tr>
<td>DCE</td>
<td>Distributed Computing Environment</td>
</tr>
<tr>
<td>DCF</td>
<td>Distributed Coordination Function</td>
</tr>
<tr>
<td>DDR</td>
<td>Distributed Dynamic Routing protocol</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol</td>
</tr>
<tr>
<td>DiSUS</td>
<td>Distributed Semantic Unstructured Services</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name Server</td>
</tr>
<tr>
<td>DREAM</td>
<td>Distance Routing Effect Algorithm for Mobility</td>
</tr>
<tr>
<td>DSDV</td>
<td>Destination Sequenced Distance Vector</td>
</tr>
<tr>
<td>DSR</td>
<td>Dynamic Source Routing</td>
</tr>
<tr>
<td>DSSS</td>
<td>Direct Sequence Spread Spectrum</td>
</tr>
<tr>
<td>DVD</td>
<td>Digital Video Disc</td>
</tr>
<tr>
<td>ESS</td>
<td>Extend Service Set</td>
</tr>
<tr>
<td>FA</td>
<td>Foreign Agent</td>
</tr>
<tr>
<td>FIPA-ACL</td>
<td>Foundation for Intelligent Physical Agents - Agents Communication Language</td>
</tr>
<tr>
<td>FSR</td>
<td>Fisheye State Routing protocol</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GPSAL</td>
<td>GPS ant-Like Routing algorithm</td>
</tr>
<tr>
<td>GWF</td>
<td>Global Wireless Framework</td>
</tr>
<tr>
<td>HA</td>
<td>Home Agent</td>
</tr>
<tr>
<td>HAIU</td>
<td>Home Appliances Integration Unit</td>
</tr>
<tr>
<td>HARP</td>
<td>Hybrid Ad Hoc Routing Protocol</td>
</tr>
<tr>
<td>HAVi</td>
<td>Home Audio Video Interoperability</td>
</tr>
<tr>
<td>HP</td>
<td>Hewlit-Packart</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>IANA</td>
<td>Internet Assigned Numbers Authority</td>
</tr>
</tbody>
</table>
ICMP  Internet Control Message Protocol
IEEE  Institute of Electrical and Electronics Engineers
IETF  Internet Engineering Task Force
IP  Internet Protocol
IPMSA  Integrated Personal Mobility Services Architecture
IPv4  Internet Protocol version 4
IPv6  Internet Protocol version 6
Java  Java Remote Method Invocation
JDBC  Java Standard Database Connectivity
JMS  Java Messaging Service
JXTA  Juxtapose - SUN ® - Kernel which allows creating, enlarge, aggregate or erase peer to peer sessions.
LAN  Local Area Network
MAC  Media Access Control
MANET  Mobile Ad Hoc Network
MEWLANA  Mobile IP Enriched Wireless Local Area Network Architecture
MIPBS  Mobile IP Base Station
MIPMANET  Mobile IP for Mobile Ad Hoc Networks
MIPMH  Mobile IP Mobile Host
MN  Mobile Node
MPR  Multi Point Relays
NAT  Network Address Translation
NEMO  Network Mobility Working Group
NS-2  Network Simulator version 2
OGSA  Open Grid Services Architecture
OGSI  Open Grid Services Infrastructure
OLSR  Optimized Link State Routing Protocol
OSGi  Open Services Gateway initiative
P2P  Peer-to-Peer
PC  Personal Computer
PDA  Personal Digital Assistants
RARP  Reverse Address Resolution Protocol
RED  Random Early Detection
RFC  Request For Comments
RPC  Remote Procedure Call
RREP  Route Reply
RREQ  Route Request
RRERR  Route Error
RTS  Request To Send
SA  Server Agents
SIC  Service Integration Controller
SLP  Service Location Protocol
SOAP  Simple Object Access Protocol
TBBR  Tree Based Bidirectional Routing
TCP  Transmission Control Protocol
TORA  Temporary Ordered Routing Algorithm
TV  Television
UA  User Agents
UDA  UPnP Devices Architecture
UDDI  Universal Description Discovery and Integration
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UPnP</td>
<td>Universal Plug and Play</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locators</td>
</tr>
<tr>
<td>VN</td>
<td>Visiting Node</td>
</tr>
<tr>
<td>WAP</td>
<td>Wireless Application Protocol</td>
</tr>
<tr>
<td>WDS</td>
<td>Wireless Distributed System</td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>Wireless Fidelity</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
<tr>
<td>WLL</td>
<td>Wireless Local Loop</td>
</tr>
<tr>
<td>WPAN</td>
<td>Wireless Personal Area Networks</td>
</tr>
<tr>
<td>WRP</td>
<td>Wireless Routing Protocol</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Services Description Language</td>
</tr>
<tr>
<td>WSFL</td>
<td>Web Services Flow Language</td>
</tr>
<tr>
<td>WSIL</td>
<td>WS-Inspection Language</td>
</tr>
<tr>
<td>WWW</td>
<td>World Wide Web</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>ZEROCONF</td>
<td>Zero Configuration working group</td>
</tr>
<tr>
<td>ZRP</td>
<td>Zone Routing Protocol</td>
</tr>
</tbody>
</table>
Chapter One: Introduction

The convergence of mobile equipment and wireless technology bring us to a new era of network computing [Kleinrock 2000b]. Many kinds of small devices free the user from their desk enabling them to move around whilst still remaining connected to the network to continue their work. Today, mobile users already utilise a wide variety of mobile terminals ranging from mobile phones and personal digital assistants (PDA) to high-end multimedia notebooks. Personal mobility services require an advanced architecture that integrates supported protocol mechanisms, and special functionality to dynamically reconfigure applications whilst the user roams from one device to another. Personal mobility also has to deal with new information spaces where information and services are not just coming from the Internet but also from other equipment nearby and the environment around them. This also raises the challenges of how all the services from different networks can be discovered and managed ensuring efficient use. These services may be either from devices located in Ad Hoc networks or infrastructure networks accessible from an existing gateway. The services can reside in any networked appliance device from our home appliances to the multi-million computer at the company. The convergence of mobile equipment and wireless technology needs innovative ideas for the design and development of next generation wireless systems where mobile devices are able to interoperate in both Infrastructure and Ad Hoc Networks simultaneously. This thesis addresses these issues.

1.1 The Future Network Environment

In the scenario described above, Personal Mobility [Campadello et al. 1999] is another extension to mobile computing. In real life we do not just use one type of device – we use a plethora of computing devices to solve a number of technological requirements on a day to day basis such as desktop PCs, Notebooks, PDAs and Smart Phones. These, small, mobile and data processing devices will start to provide services and open up a new paradigm called “Smart Space”, whereby all devices become alive [Kleinrock 2000a]. Figure 1 shows the variety of personal mobility and computing devices.
Visualise a high street shopping area, which is a simple outdoor environment. The street is full of shops, restaurants, street vendors and other people. We pop in and out from one store to another (Internet/Web), buy a quick snack from a street vendor (Local Ad Hoc Services) and greet people we know (neighbour nodes). All of these activities happen within our focal view. Mobile devices within real-world environments have to work the same way as our shopping area analogy. This provides users of mobile devices, with the ability to interact and use services in the same way we interact with shops and people within our real environment.

1.1.1 Information Space

By considering the Mobile User as the centre of the surrounding information and services we find that, in reality, the environment the user moves into may have more than one network and each of them has its own services. The ability to select and use both of these types of services will offer the maximum flexibility for the user which is of paramount importance.

We define an Information Space as the environment that an end-user can get their information not just from their machine or Internet but also from their environment and equipment nearby [Mingkhwan et al. 2001]. In this information space devices can communicate with other devices and servers to exchange information, digital content and get the services they need the same way as the high street shopping analogy. The services in an Information Space are highly dynamic since some of them are provided by the surrounding ad hoc nodes that come and go over time.

The needs of personal mobility from information space go beyond the current WWW architecture. Mobile users do not just need information from WWW servers but also the services around them. To achieve this we need additional control and management
functionality whilst roaming. We need underlying service support architecture on the server and the client, which offers more than Web pages provided by the current WWW. This raises the major challenge of how to provide the services to users in a seamless fashion while they are moving.

The next scenario shows an example of how the information can flow in the personal mobility world. Imagine that a user with their mobile equipment is going to the museum to see a painting exhibition. The user device is still connected to the Internet and can reach the user’s home network to receive all the relevant information to do their work. On the other hand, the user can also enjoy the painting with the information that is sent to their terminal devices directly from the painting itself (not the museum server). In addition when combined with the information of the relevant painting in the room from the museum server, which is a local network, it will bring great benefits to users. This also allows flexibility to the museum staff to relocate any painting without updating the list of the painting. This kind of interoperable integrated service (Internet information, museum exhibition and painting information) needs an architecture that can provide the seamless interoperability of different types of networks.

From the example above, to achieve the concept of seamless integration of services in a Personal Mobility Information Space the concerned network should not only be the infrastructure network – the Ad Hoc network and the environment nearby have to be considered as one integrated network. It is the integration of these different types of networks, which brings all the available services together, that is the focus of our work.

![Figure 2 Information Flow in Personal Mobility Information Space](image-url)
Figure 2 illustrates the sources of information that mobile users can get from the Information Space. In the example scenario, we can see that the painting information is the nearby devices information, the groups display or room information is the local broadcasting that provided by the museum servers. We classify the information that flows through information spaces as follows:

- Information from the internet, it can be categorized into 3 types:
  
a. Pull information - information that users retrieve from the network by submitting their request to the network. Surfing the internet using a web browser is the good example for this kind of information.

b. Push information - a set of data relevant to the user needs, extracted from their profile, which is sent from the network to the user. These kinds of information come in many different types, the most popular form such as pop-up windows or e-mails. This is very useful if the criteria profile of the user is met. But mostly it is not and it becomes junk mail and annoying pop-ups.

c. Event monitoring information – a set of data reported to the user according to user specified rules (e.g. Tell me if my stock is lower than 300).

- Local Information Filtering (location based), by using a GPS system or the name of the place, mobile equipment can perform location filters including such things as traffic and weather reports; nearby restaurants and theatres; temporal filters sent “just in time” information such as the advertising of a restaurant's nightly specials menu and discounts. This can be from the network or from nearby services broadcasting information within the range of communication.

- Information about other services within the range of communication, e.g. in case the user wants to establish a connection with his group [Fleming et al. 2000; Kindberg et al. 2000; Kindberg et al. 2001], or find out any services needed.

The need for an integrated Information Space of mobile users will require the bringing together of the infrastructure with wireless ad hoc networks. In particular, the challenge is to bring together services that are provided within mobile ad hoc networks and services that are well defined in Infrastructure Networks like the Internet [Jonsson et al. 2000; Mingkhwan et al. 2002b; Kleinrock 2003]. A Service node has to be able to provide and discover its services throughout the Information Space by using Gateways that interconnect infrastructure networks and mobile ad hoc networks.

1.1.2 Network Structure

The original concept of Computer Network types are classified to Infrastructure and Ad Hoc as a function and objective of the network when it was setup.
- **Infrastructure Network**: The definition in information technology and on the Internet is the physical hardware used to interconnect computers and users. Infrastructure includes transmission media, telephone lines, cable television lines, and satellites and antennas, and also the routers, aggregators, repeaters, and other devices that control transmission paths. Infrastructure also includes the software used to send, receive, and manage the signals that is transmitted. Here we take the assumption of the infrastructure network as the Internet. In other definitions infrastructure network is a network that is set up to support specific needs within defined boundaries and usually is configured and controlled by network administrators.

- **Ad Hoc Network**: An ad-hoc (or "spontaneous") network is a local area network or other small network, especially one with wireless or temporary plug-in connections, in which some of the network devices are part of the network only for the duration of a communications session, or - in the case of mobile or portable devices - while in some close proximity to the rest of the network. In Latin, ad hoc literally means "for this," further meaning "for this purpose only," and thus usually temporary. The term has been applied to future office or home networks in which new devices can be quickly added. Here we take the assumption of the ad hoc network as the group of computers forming up a temporary network. Another feature that characterises ad hoc networks is the node itself is capable of self-configuration, which means no need for the help of network administrators.

In this classification the wired infrastructure network works well because all the nodes in the network are static. Wireless technology has added more value to the network other than just removing the need to physically wire devices; it allows the node to freely move around its environment while it is connected. The moveable capability of a device allows the device to move across other networks; therefore we need an infrastructure that enables seamless mobility.

Most of the networks that we have today are Infrastructure. Every node that exists has to register to use an IP address, for example in Internet applications users may use DHCP servers for automatic assignment. Network computing was designed based on the concept of the network belonging to some kind of administrative authority, responsible for the management of that domain. Ad Hoc Networks are more flexible, being formed by a group of machines when they are required. They do not form a permanent structure. All the nodes that appear in an Ad Hoc Network can communicate with each other and use any kind of services offered. We expand the idea of Ad Hoc to allow a subset of nodes within the network to simultaneously access services in the Infrastructure Network [Mingkhwan et al. 2002b].
To illustrate the states of nodes in the network, we classify the network conditions as shown in Figure 3.

There are three main states for network service nodes in an Information Space as shown in Figure 3. Nodes A, B and C represent different states of network capability. All can be wired or wireless. What we consider here is what structure the node belongs to and consequently what are its capabilities.

Node A is representing a node that only operates in the Infrastructure Network, a normal node in a wired or wireless network that has an address assigned to it. The node is always connected to the infrastructure. An example for this kind of node is a normal office environment where the network is managed by a Network Administrator. Node A can access only services provided within the Infrastructure Network.

Node B operates in an Ad Hoc Network environment, where every node has equal opportunities to share and use the services offered by the network. Every node like B can join with the existing ad hoc networks. Since the network is not maintained, many challenges are brought up such as security, availability and reliability of services, and the services range [Chlamtac et al. 2003].

Node C represents the most complicated situation, compared to A and B. Node C has the capability to access and operate in both the Ad Hoc and the Infrastructure Network. The challenge for Node C, in order to get the benefit from both Ad Hoc and Infrastructure Networks is to access the services of both at the same time. To do this the node must join...
seamlessly with the Infrastructure Network through the gateway without being part of it. This can be achieved if the wireless link of the access point also behaves in an ad hoc manner and use the same network address with the mobile node. Our research considers nodes operating in this kind of environment.

What is interesting about the research in Ad Hoc Networks is that the vision is still very much about a node connecting to a network; the trick is in how the nodes cooperate together without an infrastructure to support them. We think that for Personal Mobile Computing this vision is not ideal. What we think would be more appropriate would be for the network to be formed around the user, so that the devices the user directly interfaces with are central and the ones they indirectly communicate with are at the network edges [Askwith et al. 2004]. We call this model **Node-Centric Networking**.

![Figure 4 Node-Centric Networking](image)

In our work we introduce a new concept of a network called, Node-Centric network, an example of such a network is illustrated in Figure 4. From the different point of view with Mobile Ad Hoc Network (MANET), our Node-Centric network does not form the network that can interact within the group of nodes but communicates with every nearby node and infrastructure network in its range of communication.

Node-Centric network is the idea that a node is an individual not a group of nodes. Every node has its own communication range which is its communication range ability. In a Node-Centric Network, many of the tasks that the user might want to perform will take place in a node communication area, i.e. between the user’s personal devices. On a slightly larger scale some of the tasks the user performs might be between one or more devices belonging to the personal mobile computing system of another user, or to other more general devices in the adjacent environment, such as the fixed Networked Appliances in the home.
1.1.3 Services Classification: Structured and Unstructured Services

There are many technologies to provide services for users while they are moving. For example, Internet access can be provided to users in many different ways like WWW, WAP and I-Mode technologies [Pilioura et al. 2003]. Many people are talking about m-commerce (the use of mobile technologies to conduct e-commerce). In the Information Space, devices can communicate with other devices and servers to exchange information and get the services they need. During personal mobility the available information within the user Information Space is changing, so we also have to deal with services that may be unstable.

Apart from joining all the services available from both Infrastructure and Ad Hoc networks together - we also consider that Radio or Television like appliances do not have to connect to Infrastructure network, the user should simply just unpack it, and the Plug-and-Work Service function should introduce itself to the home appliances networked, automatically configure itself and start to provide services directly to the user’s device on their request. In this case it was introduced to a home environment that has a services structured controller, and is willing to join new devices with the existing services.

In our work we define services within the Information Space as structured and unstructured:

- **Structured Services**: Structured Services are provided with the help of third party services, e.g., Directory Service, Name Server and Proxy. These kinds of services typically have complex structures, such as network connectivity, database access and multimedia functions. Examples of projects providing these kinds of services approaches are CORBA [Siegel 1998], Java RMI [Gisolfi 2003] and GRID Computing [Foster et al. 2002].

- **Unstructured Services**: Unstructured Services provide services independent of any third party intervention. This concept is based on a simple service definition, such as a kiosk that provides quick information, a TV remote control that simply changes the channel or a file-sharing application that exchanges digital content. The main example of this type of service is the Web Service [Englander 2002; Newcomer 2002] which is focused on simple, Internet-based standards to address heterogeneous distributed computing.

There are a great number of different types of structured services provided to users over the Internet and mobile ad hoc networks, but unstructured services are left far behind and our research provides a foundation on which to build. Service nodes in our system can provide either Structured or Unstructured Service types (or both). The main idea here is how we can provide both types of services to Information Spaces anytime we want. The user devices are responsible for discovery of what kind of services, and how many are available in the Information Space that they migrate to.
Internet-based structured services like JINI [Jini 2002] and UDDI [Walsh 2002] are already well defined, however they are incapable of providing a service within a dynamically changing network environment. In our work, we overcome this limitation by situating services within the Information Space, using decentralised P2P concepts [Verbeke 2002].

1.1.4 Services Composition

To demonstrate the framework and show how to utilise the integrated services, we use the home appliances network as a case study to show how the services provided should look and how we can benefit from them. This section will discuss how the services of network appliances can be best organised.

The proliferation of networked appliances and their increasing complexity and varied functionality make it ever harder for a specialist let alone an ordinary home user to configure these appliances to provide a given service. Furthermore many of the common applications and their sophisticated functions would allow for a better exploitation of these features both for the benefit of performance, simplicity and reduction in the implications of many of their capabilities. For example a TV, DVD player and radio have all got output speakers and they are capable of producing sound, but there is no common framework to harness these functionalities. For this we proposed the use of Implicit Functions. Implicit Functionality is defined as that which is configured on demand without any expressly stated characteristics. We propose a framework that enables Implicit Functionalities in a home appliances network. In this network, home appliances are structured around a central service integration controller, which integrates their offered services and efficiently coordinates the communications between all service-enabled appliances independent of the protocol type being used. Implicit Functionalities can be formed based on the controller's ability to discover and dynamically compose services to form specified configurations.

Imagine while one watches a DVD using the TV's visual and audio services, the TV's Teletext function is displaying a recipe on the intelligent fridge screen, which is being used by the person who is preparing dinner in the kitchen. In the mean time the TV's free-to-air receiver unit can be used to stream the signal of a documentary on the Discovery Channel to the upper floor where it is displayed on a small window on the children's personal computer screen. The phone rings and the person who is reading a book in the bedroom picks it up, and although a physical intercom system does not exist, is capable of broadcasting a message to the household using the available audio appliances found within the home environment, to inform the relevant person that they have a phone call. The possibilities of deploying such systems and creating these on demand virtual appliances are only limited by the user's imagination.
In another scenario, imagine a situation whereby all the appliances within the home are connected. Each appliance exposes its capabilities by advertising its services to all the appliances within its immediate environment. Using a 3G mobile phone, a video call conversation is made. In the process of having the conversation, the user arrives at home. While entering the house the phone registers its services including 3G receiver, video, microphone and speaker with the services registry. The user now has the capability to create or invoke an Implicit Function in order to determine better services to support the video conversation. They could integrate the visual service of the TV with the audio of the HiFi and still use the microphone and communication channel of the mobile phone in one single implicitly created profile and activate it seamlessly without any disruption to the conversation as illustrated in Figure 5.

![Figure 5 Dynamic Services Composition](image)

From the example scenarios described above, our novelty resides in the ability to capitalise on the redundancy and offer all the services provided by these complex appliances. The high independence of the services enables users to compose numerous Implicit Functions from the available services such as the intercom system described in the first scenario, or to typically provide better services as described in the second scenario. Additionally one can also make use of the redundant services when only one function of the appliance is used. For example, as shown in Figure 6, if the TV is used to play a DVD all its other services are normally off. In our system the free-to-air receiver is still able to be used to broadcast the signal to other visual displays, whether it is a computer or another TV. In this case, the user utilises an unused service instead of buying new TV receiver card for the PC.
Another interesting feature provided by our system is the creation of a virtual appliance such as an intercom Implicit Function. As we described in the previous section, this appliance does not physically exist but is able to be built by combining the available services of other registered appliances. In this case the microphone from the telephone and the speaker from the audio appliances throughout the house, such as TV, Hi-Fi system, Computer Speaker, can be combined to produce a virtual intercom appliance as shown in Figure 7.

To be able to do this, the system has to provide the ability to interrupt services, allowing other appliances to take over the control of the appliances being used according to the priority classification. Another example of a virtual appliance is a virtual remote control capable of invoking all the functionalities provided by each appliance in a given profile configuration.
1.2 Scope, Aims and Objectives of this Thesis.

The aims of our research are to integrate the services that are offered to users from both Infrastructure Networks and Ad Hoc Networks that a user migrates to and utilise the services that are available. In order to achieve these objectives we first have to address two issues.

- What are the requirements to provide services for mobile users in the Information Space? Both of the requirements from the mobile user, network system and services should be taken into account to make sure that all of them will be seamlessly integrated.

- What existing techniques are appropriate to developing a solution for these requirements? The literature surveys shall examine the building blocks of the Integrated Personal Mobility Services as well as providing the knowledge to the Mobility Services.

The proposed Integrated Personal Mobility Services Architecture allows clients in Personal Mobility environments to use concurrent services from different types of networks, either Infrastructure or Ad Hoc Networks. This will provide the flexibility to use any kind of service in our Information Space by means of open services architecture. The aims of this thesis are:

- To give the mobile devices the ability to communicate in both Ad-Hoc and Infrastructure Networks simultaneously.

- To propose a system that can provide the network and ensure that each mobile device is the centre of its own network which moves along with the mobile device.

- To provide a concept of how the services will be organised to efficiently use the architecture.

- To demonstrate / evaluate the proposed solution that shows how the network can overcome the requirements and the visions.

1.3 Novel Aspects of this Work

In this section, we discuss three of our main novel contributions to achieve our goal to provide and utilise the integration of services in the personal mobility environment:

- **Integrated Personal Mobility Services Architecture (IPMSA):** In our published papers [Mingkhwan et al. 2001, 2002a, 2002c], we described how we provide Unstructured and Structured services to mobile users. The IPMSA client framework was developed in our experiment based on Jini Client with the addition of the functionality to handle Unstructured Services in conjunction with unstructured Jini Services. The IPMSA client combines the integrated services
and client properties to implement the core features of the IPMSA. The novelty of this is in the flexibility of services that can be discovered and utilized seamlessly from both ad hoc and infrastructure networks.

- **Global Wireless Framework:** Our framework uses one large, auto-assigned, private, non-Internet-routable IP address space to provide anytime, anywhere services to any nomadic end user devices in the Information Space [Mingkhwan et al. 2002b]. Every mobile node maintains their network just within their communication range (if a repeater/bridge is not applied). Outside the nodes communication cover area it can use the duplicated IP since it was not involved or related. Instead of using a router for Multi-Hop communication, we are using a Link-Layer repeater/bridge scheme [Mingkhwan et al. 2003a]. In such a network, mobile devices need to communicate in an ad hoc manner and have the ability to access the Infrastructure network simultaneously [Mingkhwan et al. 2003b]. The proposed framework has the following advantages,

  - The Node-Centric network concept in our Global Wireless Framework joins a number of networks within the environment into one single domain.
  - A GWF Ad Hoc Bridge providing the multi-hop communication in the Link-Layer offers more flexible choices to the implementation of ad hoc systems.
  - Global Wireless Framework provides an easier integration of ad hoc and the infrastructure network (Internet), with less modification of the gateway and most importantly the client.

- **Dynamic Services Composition:** Our work [Mingkhwan et al. 2004] proposes a dynamic service composition framework to dynamically incorporate the appliances services and controls into a Services Integration Controller operating in a central command, which we call the Home Appliances Integration Unit (HAIU). Our framework makes three novel contributions:

  - **Functionality Utilisation** - Complex appliances like TVs and HiFis can expose their resources and services for simultaneous use.
  - **Dynamic Composition** - The ability to use the services of better performing appliances when they are readily available.
  - **Virtual Appliance** - The ability to create an appliance that does not physically exist using the available services.
1.4 Summary

At present a user’s awareness of network services are guided or controlled by what the network offers. However in the future we will see many different networks in any environment. The challenge is to move from the connection with the central network to one with multiple networks with a very small area, say 100 metres. How do we allow users to travel with their small personal area network and interact with many locally offered services, analogous to high street shopping?

Our research scenarios [Mingkhwan et al. 2002a] are based on the notion that a user will gather information and services from their surroundings and require interoperability between multiple mobile network standards [Abuelma'atti et al. 2002a]. These may be either communicating with surrounded devices and/or connecting to existing infrastructure networks.

The problems of present mobile Ad Hoc network devices are that they are only temporary networks. The only way to get information from infrastructure networks like the Internet is to connect to the services access point. The weak points of these getting connected are they cannot discover other services in their ad hoc environment if those services are not joined to the Infrastructure Network.

Our research is based on the assumption that all mobile devices can have an IP address, but to discover what kind of IP paradigm provides the most benefit to a nomadic user, easy to configure and seamlessly works with the Infrastructure Networks. We present the Global Wireless Framework [Mingkhwan et al. 2003b] within the Integrated Personal Mobility Services Architecture to be solutions to our research requirements.

1.5 Thesis Structure

The remainder of the thesis is structured as follow:

- In Chapter Two, “Related Works and Challenges”, we discuss the Personal Mobility foundation we require in order to look at the solutions for this paradigm. We look at the related research and what Personal Mobility services are, Mobile IP & IPv6, Wireless Technologies, Ad Hoc networks, MIPMANET, Location Services and Services Discovery techniques. At the end we raise the challenges for the current system that are left to be explored.

- In Chapter Three, “Integrated Personal Mobility Services Architecture”, we explain the Integrated Personal Mobility Services Architecture and how each component is put together to deliver the goal of the project. By introducing our novel contribution to the Integrated Personal Mobility services solution we first outline the general concept of our architecture and framework. The Global Wireless Framework [Mingkhwan et al. 2003b] is our framework that we
proposed for integrating Mobile Ad Hoc Networks and an Infrastructure Network(s) together which allows nomadic users to move around and get services from both Structured and Unstructured service providers. Integrated Personal Mobility Services Architecture (IPMSA) is our proposed architecture to provide Structured and Unstructured Services to nomadic end-users that also provides the managing functionality that is needed to keep the service functions available while users are moving.

- **In Chapter Four**, “Architecture Design”, we analyse and propose the functional and protocol needs for each part of Integrated Personal Mobility Services Architecture in detail. This is including the address and node configuration, multi-hop communication and the integration of the node with the Internet while moving in the freely ad hoc manner.

- **In Chapter Five**, “Simulation and Prototype Implementation”, we show how we set up the test bed and evaluation plan for both the Global Wireless Network framework and Integrated Personal Mobility Services Architecture (IPMSA) including the work plan for the rest of our research. We take the design and implement it to prove that our proposed system is working. We use NS-2 simulator for the Global Wireless Framework implementation model and use TCP, XML and JXTA to implement the prototype for the services integration part.

- **In Chapter Six**, “Global Wireless Framework Simulation Study”, we run the simulation in the different situation and analyse the results to show the performance of our work compared against the existing technology.

- **Finally In chapter Seven**, “Conclusions and Future Work”, we conclude this thesis, summarising the findings and the problems that we had so far, discuss the outstanding issues and our idea for future work.
Chapter Two: Related Works and Challenges

There is a great deal of work going on in the area of mobile computing and mobility services. Many new technologies are enabling mobile communication services networks. Such networks enable users to seamlessly move between different networks to perfectly use needed services. Many research efforts have been focusing on personal mobility services and the system architecture that can support their movement in the environment. In our investigation, we have identified two related issues that have to be considered in this chapter of the thesis. They are mobile computing services and their supporting networks.

2.1 Architecture for Mobile Computing

One of the efforts in personal mobility is the Universal Inbox of the ICEBERG project from U.C. Berkley [Wang et al. 2000]. It achieves the goal of integrating telephony and data services by enabling the possibility to transparently communicate between end-to-end points that use different kinds of devices. However it does not address the integration of structured and unstructured services.

While ICEBERG tries to provide a transparent infrastructure of cross system services, other projects like CoolTown [Kindberg et al. 2001], Monads [Raatikainen et al. 1999] and Salutation [Salutation 1999] are aiming at providing services. CoolTown implements the WEB presence paradigm. By enabling WEB access to devices, many kinds of services can become available. The limitation of this work is its lack of automatic discovery of services. In this work, users have to get services by browsing the device’s web portal.

The Monads project [Raatikainen et al. 1999] combines nomadic computing and software agent technologies to create a solid basis for future nomadic applications. Partitioning applications have been introduced in this project to support personal mobility. This partitioning of an application services approach is also only from the network side and aims to provide continuous service of one application while users are moving from one device to another. Partitioning applications [Koskimies et al. 2000] is concerned with improving
network support for applications. In our work, we wish to provide a broader availability of services as required by the notion of Information Spaces.

2.1.1 Service and Services Discovery Projects

A service is a network-enabled entity that provides a specific capability. It is defined in terms of the protocol one uses to interact with it and the behaviour expected in response to various protocol message exchanges (i.e., service = protocol + behaviour). Therefore, a service definition may permit a variety of implementations. A service may or may not be persistent (i.e., always available), be able to detect and/or recover from certain errors; run with privileges, and/or have a distributed implementation for enhanced scalability.

Location services are defined as applications that deliver location-based information where and when it is needed. Users can access these services via a desktop Web browser, a mobile phone, a personal digital assistant (PDA), a pager etc. Diverse applications include fleet tracking, emergency dispatch, sensor monitoring, roadside assistance, stolen vehicle recovery, navigation, directory services, customer location services for targeted marketing and advertising, and many more.

Recent work in Distributed Computing has focused on the problems of mobility both of software components of systems and of the devices on which they operate [Rosenberry et al. 1992]. One of the primary difficulties is that of location, both of distributed services and mobile devices. Common approaches to device location, for example IP Routing [Baker 1995] and DNS [Mockapetris 1987], require that the name of a machine reflects its physical location to some degree.

Similarly, distributed architectures built on IP and DNS usually require that programs exporting services not migrate between devices, if clients are to be able to access those services. Mobile services will therefore be considered as equivalent and transparent in terms of location to mobile devices.

The issue of transparent location is tackled by a number of different schemes, which can be divided, for the most part, into three general categories;

- Schemes such as the Globe Object Model’s [Hauck et al. 1997] location service, that use pure names which convey no information to the client as to how the corresponding entity should be located. These systems are seldom as scalable as we might like, since their implementations require that some central node be aware of every object in existence.

- Schemes such as Mobile IP [Perkins 1998a] and the CORBA LifeCycle [Object Management Group 1998; Siegel 1998] service, which use impure or composite names, most often comprising a home agent location and a key. These systems...
can fail when the home agent is unavailable for some reason and can suffer from performance degradation when the mobile entity is far away from its home.

- Hybrid schemes such as ALICE [Cunningham 1998], in which the mobile entity’s home is effectively mobile itself and the composite name of the entity is transparently merged to reflect its current home, older names of the entity being forwarded to its current location transparently. This approach is more efficient than the first one and more manageable than the second but suffers similar robustness problems to the latter. It is also unclear as to how long these surrogate homes should hold forwarding information for.

**Service Discovery Protocols** obtain the description of a device in one of two ways. For example, with the Bluetooth protocol, discovery information is exchanged directly between two devices. After a connection is established, the devices exchange information about their capabilities using the Bluetooth service discovery protocol (SDP) [Avancha et al. 2002; Huang et al. 2002]. Universal Plug and Play operates in essentially the same way.

Jini [Li et al. 2000] handles the discovery process differently. When a client first connects to a network, it looks for a server that has been set up as a registry, a so-called look-up server. The client registers with the server, giving its capabilities. When it needs the services of another device, such as a printer or fax, it goes to the server to see what printers and faxes are registered.

In the case of both Jini and Bluetooth, the information a device needs to know; what devices are available on the network and what those devices are capable of, is the same. But in the case of Bluetooth, the device gets the information directly from the equipment it wants to tie up with, while in the case of Jini a device must go to a special server to get it.

Looking again at Jini, this protocol provides a method for locating and loading special Java code called a proxy object. The proxy object is provided by the discovered device. Because it is written in Java, it is designed to be written once and run on any Java engine, allowing any Java application to access and use the discovered device. However, the value is limited to a closed Java environment. In reality, the pervasive networks will not be solely Java.

The term **Web Services** [Fisher 2002] describes an important emerging distributed computing paradigm that differs from other approaches such as DCE (Distributed Computing Environment) [Rosenberry et al. 1992], CORBA, and Java RMI [Bodoff 2002; Englander 2002] in its focus on simple, Internet-based standards, e.g. XML (eXtensible Markup Language) [Green 2002; W3C 2003] to address heterogeneous distributed computing. Web services define a technique for describing software components to be accessed, methods for accessing these components, and discovery methods that enable the identification of
relevant service providers. Web services are a programming language, programming model, and system software-neutral. Web services standards are being defined within the W3C [W3C 2003] and other standards bodies and form the basis for major new industry initiatives such as Microsoft .NET Framework [Platt 2003], IBM Dynamic e-Business [Gisolfi 2003], and Sun ONE [Sun 2003]. We are particularly concerned with three of these standards: SOAP, WSDL, and WS-Inspection.

- The Simple Object Access Protocol (SOAP) [Englander 2002; W3C 2003] provides a means of messaging between a service provider and a service requestor. SOAP is a simple enveloping mechanism for XML payloads that defines a Remote Procedure Call (RPC) convention and a messaging convention. SOAP is independent of the underlying transport protocol; SOAP payloads can be carried on HTTP, FTP, Java Messaging Service (JMS), and the like. We emphasize that Web services can describe multiple access mechanisms to the underlying software component. SOAP is just one method of formatting a Web service invocation.

- The Web Services Description Language (WSDL) [W3C 2001] is an XML document for describing Web services as a set of endpoints operating on messages containing either document-oriented (messaging) or RPC payloads. Service interfaces are defined abstractly in terms of message structures and sequences of simple message exchanges (or operations, in WSDL terminology) and then bound to a concrete network protocol and data-encoding format to define an endpoint. Related concrete endpoints are bundled to define abstract endpoints (services). WSDL is extensible to allow description of endpoints and the concrete representation of their messages for a variety of different message formats and network protocols.

- WS-Inspection [Ballinger et al. 2001] comprises a simple XML language and related conventions for locating service descriptions published by a service provider. A WS-Inspection language (WSIL) document can contain a collection of service descriptions and links to other sources of service descriptions. A service description is usually a URL to a WSDL document; occasionally, a service description can be a reference to an entry within a Universal Description, Discovery, and Integration (UDDI) [OASIS 2002] registry. A link is usually a URL to another WS-Inspection document; occasionally, a link is a reference to a UDDI entry. With WS-Inspection, a service provider creates a WSIL document and makes the document network accessible. Service requestors use standard Web-based access mechanisms (e.g., HTTP GET) to retrieve this document and discover what services the service provider advertises. WSIL documents can also be organized in different forms of index.

Various other Web services standards have been or are being defined. For example, Web Services Flow Language (WSFL) [Leymann 2001] addresses Web services orchestration, that is, the building of sophisticated Web services by composing simpler Web services.
The Web services framework has two advantages for our requirements. First, our need to support the dynamic discovery and composition of services in heterogeneous environments necessitates mechanisms for registering and discovering interface definitions and endpoint implementation descriptions and for dynamically generating proxies based on (potentially multiple) bindings for specific interfaces. WSDL supports this requirement by providing a standard mechanism for defining interface definitions separately from their embodiment within a particular binding (transport protocol and data encoding format). Second, the widespread adoption of Web services mechanisms means that a framework based on Web services can exploit numerous tools and extant services, such as WSDL processors that can generate language bindings for a variety of languages, for example, Web Services Invocation Framework (WSIF) [Apache 2003], workflow systems that sit on top of WSDL, and hosting environments for Web services (e.g., Microsoft .NET and Apache Axis).

The Service Location Protocol (SLP) [Guttman 1999] is an IETF standard for service discovery and automatic configuration of clients. It provides for fully decentralized operation and scales from a small, un-administrated network to an enterprise network where policy may dictate who should discover which resources.

SLP establishes a framework for resource discovery that includes three “agents” that operate on behalf of the network-based software:

- User Agents (UA) perform service discovery on behalf of client software.
- Service Agents (SA) advertise the location and attributes on behalf of services.
- Directory Agents (DA) aggregate service information into what is initially a stateless repository.

SLP has two modes of operation:

- When a DA is present, it collects all service information advertised by SAs, and UAs unicast their requests to the DA.
- In the absence of a DA, UAs repeatedly multicast the same request they would have unicast to a DA. SAs listen for these multicast requests and unicast responses to the UA if it has advertised the requested service.

The formal specification has not yet been standardized for SLP Operation over IPv6, but SLP is designed to provide service discovery facilities that will work for networks using IPv6 [Guttman 2001a]. Once the database is settled regarding which string representation to use in URLs for numerical IPv6 address, some minor changes will be needed. Service URLs [Guttman et al. 1999] containing numerical addresses will require a different format from what IPv4 uses, and link-local address will require some special handling in IPv6. For example, DAs that obtain service registrations with link-local numerical address must not
forward them using the link on which they were registered. Also, the address to use for site-local scoped multicast operations differs in IPv4 from what it is in IPv6 [Hinden et al. 1998].

**Jini** [Jini 2002], provides a Java-oriented set of mechanisms and programming interfaces for automatic configuration. Jini leverages Java's uniformity across platforms, providing powerful semantics for service discovery operations. The Jini discovery architecture is similar to that of SLP. Jini agents discover the existence of a Jini Lookup Server which collects service advertisements in a manner analogous to DAs in SLP [Guttman et al. 1999]. Jini agents then request services on behalf of client software by communicating with the Lookup Server. Jini requires the presence of one or more Lookup Servers.

Jini’s discovery mechanism offers some advantages to Java-based clients. The Lookup Server uses object-oriented matching to determine which services support the client’s requested Java interface. Both Jini and SLP use attributes to find services that match the client’s requirements, but where SLP uses string-based attributes and weak typing, Jini employs Java objects throughout.

Service discovery with SLP returns a URL denoting a service’s location. Jini, on the other hand, returns an object that offers direct access to the service, using an interface known to the client.

Wireless networks and portable client devices provide new design opportunities for computer/communications systems. The **HP Labs “Cooltown” Project** [Kindberg et al. 2000; Kindberg et al. 2001], explores these opportunities through an infrastructure to support “web presence” for people, places and things. It embed web servers into devices like printers and put information into web servers about things like artwork; this groups physically related things into places embodied in web servers. Using URLs for addressing, physical URL beacons and sensing of URLs for discovery, and localized web servers for directories, it can create location-aware but ubiquitous systems to support nomadic users. On top of this infrastructure, Internet connectivity can be leveraged to support communications services. Web presence bridges the World Wide Web and the inhabited physical world, providing a model for supporting nomadic users without a central control point.

The **Open Services Gateway initiative** Service Platform (OSGi) [OSGi 2000; Marples et al. 2001] specification is a Java based application layer framework that gives service providers, network operators, device makers, and appliance manufacturers vendor, neutral application and device layer APIs and functions. This strategy enables virtually all emerging home networking platforms, protocols and services to seamlessly interoperate with back end services, using existing residential telephone, cable TV, or electrical wiring.
The OSGi Service Platform specification focuses exclusively on providing an open application layer and gateway interface for service gateways. Because of that it complements and enhances virtually all current residential networking standards and initiatives. Some of these include JINI, Bluetooth [Bhagwat 2001], CAL, CEBus [Markwalter et al. 1988], HAVi [HAVi 2000; Lea et al. 2000], HomeRF [Negus et al. 2000], Home API, HomePNA, and HomePnP [Zahariadis et al. 2002].

The OSGi Service Platform is a collection of APIs that define standards that define a service gateway. These APIs define a set of Core and Optional APIs that together define an OSGi-compliant gateway. Where possible the OSGi Service Platform leverages existing Java standards, such as JINI and JDBC [Ellis et al. 2001]. The core APIs addresses service delivery, dependency and life cycle management, resource management, remote service administration, and device management. All of the core APIs are either contributed by a member or developed by the OSGi Alliance technical working groups.

The optional set of APIs defines mechanisms for client interaction with the gateway and data management. In addition, several existing Java APIs are included in the optional services. This includes JINI and several other Java standards.

The OSGi Alliance has delivered specifications that are an open standard based on Java technology. The specification process is also open to members and based on those developed by other, similar organizations that enjoy a wide range of industry support.

**Universal Description, Discovery and Integration (UDDI) [UDDI.ORG 2002]** is a major advance in Web services. It is the first cross-industry effort driven by platform providers, software developers, marketplace operators, and business leaders that comprehensively addresses the problems limiting the growth of service-centric computing, and that will benefit businesses of all sizes by creating this global, platform-independent, open framework to: discover each other; define how they interact over the Internet; and share information in a global registry that will more rapidly accelerate the global adoption of service-centric computing.

The UDDI specifications take advantage of the World Wide Web Consortium (W3C) and the Internet Engineering Task Force (IETF) standards such as Extensible Markup Language (XML) and Domain Name System (DNS) protocols. Additionally, cross-platform programming features are addressed by adopting early versions of the proposed Simple Object Access Protocol (SOAP) messaging specifications found at the W3C Web site.

**2.1.2 Service Integration Projects**

There are also significant projects that are dealing with harvesting the availability of services and integrating them to complete a given task. Here we discuss as an example OGSA and UPNP.
Grid concepts and technologies were first developed to enable resource sharing within far-flung scientific collaborations. Applications include collaborative visualization of large scientific datasets (pooling of expertise), distributed computing for computationally demanding data analysis (pooling of compute power and storage), and coupling of scientific instruments with remote computers and archives.

Building on both Grid and Web services technologies, the **Open Grid Services Architecture (OGSA)** [Foster et al. 2002] uses its Open Grid Services Infrastructure (OGSI) [Tuecke et al. 2003] to define mechanisms for creating, managing, and exchanging information among entities called *Grid services*. A Grid service is a Web service that conforms to a set of conventions (interfaces and behaviours) that define how a client interacts with a Grid service. These conventions, and other OGSI mechanisms associated with Grid service creation and discovery, provide for the controlled, fault-resilient, and secure management of the distributed and often long-lived state that is commonly required in advanced distributed applications.

Microsoft’s **Universal Plug and Play (UPnP)** [Microsoft 2000; Miller et al. 2001], defines an architecture for pervasive peer-to-peer network connectivity of intelligent appliances, wireless devices, and PCs of all form factors. It is designed to bring easy-to-use, flexible, standards-based connectivity to ad-hoc or unmanaged networks whether in the home, in a small business, public spaces, or attached to the Internet. UPnP provides a distributed, open networking architecture that leverages TCP/IP and the Web technologies to enable seamless proximity networking in addition to control and data transfer among networked devices.

The UPnP Device Architecture (UDA) [UPnP.ORG 2003] is more than just a simple extension of the plug and play peripheral model. It is designed to support zero-configuration, “invisible” networking, and automatic discovery for a breadth of device categories from a wide range of vendors. This means a device can dynamically join a network, obtain an IP address, convey its capabilities, and learn about the presence and capabilities of other devices. A device can leave a network smoothly and automatically without leaving any unwanted state behind.

Technologies leveraging the UPnP architecture include Internet protocols such as IP, TCP, UDP, HTTP, and XML. Like the Internet, contracts are based on wire protocols that are declarative, expressed in XML, and communicated via HTTP. Using Internet protocols is a strong choice for UDA because of its proven ability to span different physical media to enable real world multiple-vendor interoperation and to achieve synergy with the Internet and many home and office intranets. The UPnP architecture has been explicitly designed to accommodate these environments. Further, via bridging, UDA accommodates media
running non-IP protocols when cost, technology, or legacy prevents the media or devices attached to it from running IP.

2.1.3 Service Issues Summary

Much recent research has addressed the wide range of issues in mobile computing and services discovery. Some of which have focused on reliable connections [Kim et al. 1999], some, for example MAP [Duda et al. 1997], provides a middleware layer that works with a mobile host to collect results and manage the data transfer path. AOAC [Fleming et al. 2000; Helal et al. 2001], HP Cool Town project [Kindberg et al. 2001] and JINI [Jini 2002] work on an automatic discovery of services and other devices from the physical surroundings using mobile equipment. Furthermore partitioning application services have been proposed as a method to enable personal mobility [Koskimies et al. 2000]. However, none of these proposals have addressed a solution to seamlessly carry on using services from an Ad Hoc network to an Infrastructure Network and vice versa. This thesis focuses on the capability of the dynamic changing of user terminals and how to continue working on different networks and user interfaces. In summary, the available research presents many very interesting and useful ideas; however, this work does not meet our requirements for a Personal Mobility Information Space, introduced in Chapter One, individually or collectively.

Existing service discovery architectures such as Universal Plug and Play Device Architecture (UDA), the Jini Specification, Service Location Protocol (SLP) and the Bluetooth Services Discovery Protocol (SDP) do not present appropriate characteristics for mobile ad hoc networks. For example, the services discovery protocols do not concern the reliability of the services. They are satisfied with only finding out a service. They do not consider whether the service would be able to serve the requester. For example the Jini Lookup and Discovery Protocols allow a service client to find a printing service (hardware oriented service) or a personal banking request (software oriented service), but the protocols are not powerful enough to find the geographically closest printing service that has the shortest print queue, or the banking service with the least delay time while the requester is moving around. Current architectures maintain a directory server containing the necessary information to identify the services. For example the core component of Jini is the Lookup Server which acts as a service directory. Devices looking for services, register with this directory and obtain appropriate services information. However, this mechanism is not suitable for dynamic ad hoc networks because it is difficult to maintain an up-to-date directory. UPnP, SLP, Bluetooth SDP and other approaches operate similarly. Although Bluetooth SDP has been designed for mobile Wireless Personal Area Networks (WPAN), it addresses primarily service discovery problems. It provides neither access to services nor service advertisement, and there is no event notification when services become unavailable. Due to these limitations, we believe that a Personal Mobility Services Architecture that considers
specifically the characteristics of mobile ad hoc networks is important especially for the Unstructured Services type.

2.2 Mobile Computing Networking

A network is a set of nodes that are connected together using wires or wirelessly to share resources and exchange information. We simply classify networks around us according to management as, Infrastructure or Ad Hoc networks. The network type that we are considering in this thesis is a wireless network that is related to mobile nodes. This includes both Infrastructure wireless networks and mobile ad hoc networks.

An “Infrastructure wireless network” is a wireless networking system that is provided by a network provider e.g. a WLAN, a mobile phone network, etc. Nodes in this type of network are connected and configured by the provider/administrator.

A "mobile ad hoc network" is an autonomous system of mobile routers and allied hosts joined by wireless links, the union of which form an illogical graph. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may work in a standalone fashion, or may be joined to the larger Internet.

To put ad hoc networking into its prospective, let's take a look at the whole wireless communication in the present-day. Beginning with cellular systems, which rely heavily on infrastructure – coverage is provided by base stations, radio resources are managed from a central location, and services are integrated into the system – they lead to the good and predictable services of today's cellular systems. As shown in Figure 8, Magnus Frodigh et al [Frodigh et al. 2000] depicts this two-dimensional aspect as it relates to ad hoc networking. We can note that as we move away from central management, we find that systems are moving in the direction of pure ad hoc operation. This can also be classified in terms of single or multiple hops.

Without having fully significant control, but given the direct mode of communication in HiperLAN/2 [Johnsson 1999], neighbouring nodes can communicate directly with one another. Thus, the transport of traffic is not necessarily provided by access points. The dependency on centrally administered coverage is further reduced when end-user terminals relay traffic in a multi-hop fashion between other terminals and the base station (cellular multi-hop). A similar approach applies to commercial or residential Wireless Local Loop (WLL) [Kibati et al. 1999] multi-hop access systems, primarily conceived for Internet access.
Fully decentralized radio, access, and routing technologies enabled by Bluetooth [Haartsen 2000], IEEE 802.11 ad hoc mode [IEEE-SA 1999], PRnet (Packet Radio Network) [Shoch et al. 1979] station less mode, mobile ad hoc network (MANET) [MANET 2003], and concepts such as the Personal Area Network (PAN) or PAN-to-PAN communication fit more or less entirely into the ad hoc domain. The MANET initiative by the Internet Engineering Task Force (IETF) also aims to provide services via a fixed infrastructure connection to the Internet.

Short-range ad hoc networks can simplify intercommunication between various mobile devices such as notebooks, cellular phones and PDAs by forming a PAN, and thereby eliminate the tedious need for cables. This could also extend the mobility provided by the fixed network to nodes further out in an ad hoc network domain. The Bluetooth system is perhaps the most promising technology in the context of personal area networking but IEEE 802.11 is also very active and makes its way by the varying support of technologies and the wide use of Ethernet born users.

In the following sections we take into consideration the work that is related to the basic requirements that we have defined in Chapter One for a mobile ad hoc networking system. This includes; address autoconfiguration, multi-hop communication and Internet Integration.

### 2.2.1 Mobile Ad Hoc Network

A Mobile Ad Hoc Network (MANET) [MANET 2003], an active work by IETF, is a collection of nodes forming a temporary network without the aid of any centralized administration. In this network, each node communicates over wireless channels, moves
freely and may join and leave the network at anytime. The evolution of wearable and pervasive computing meant that, not only laptops, palmtops and mobile phones can become nodes of an ad hoc network, but also any conceivable device such as home appliances. Interesting research has been going on to provide wireless enabling techniques and efficient routing protocols in these domains. Such networks have a potential values in areas such as military communication, post-disaster rescue efforts as well as commercial appliances.

Today, the vision of ad hoc networking includes scenarios such as, people carrying devices that can connect on an ad hoc basis to both other users' devices and local information points. Such connections can be used for example, in airports to; retrieve or update on flight departures, gate changes, and so on.

In order to enable such dynamic scenarios, devices that are out of the range of access points should be able to receive traffic relayed form other devices in the rout. Therefore a mixture of single and multiple radio hop communications should be allowed. Furthermore nodes should not even be administered by central DHCP systems to provide them with addresses to participate in communications. Therefore address autoconfiguration methods should be deployed. In the following sections we give details of the available literature in the areas of address autoconfiguration, multihop communication and integrating ad hoc networks with the Internet.

2.2.1.1 Address Autoconfiguration

Zeroconf [Guttman 2001b] networking is an emerging field of research from the Internet Engineering Task Force. It works on providing IP level networking without needing a network administrator. It provides methods to assign an IP address to a host, discover network services by names or characteristics and to translate between names and addresses.

IPv4 Address Autoconfiguration [Cheshire et al. 2001; Perkins et al. 2001], provides stateless address configuration by choosing from a pool of 65024 possible IP addresses in the range of 169.254.1.0 to 169.254.254.255 (that is 169.254/16, with the top 256 and bottom 256 addresses reserved for future purposes).

The algorithm to choose an address is relatively simple. The host seeds a random number generator with the hardware address (MAC address) of the interface, and randomly chooses an address in the allowed range. It should then do an ARP probe for the address, and if there are any responses, i.e. a conflict is detected, it chooses another address at random and try the ARP probe again. If there are no answers, i.e. no conflicts in the local range, the host is free to assign and use the selected address. It should then do a couple of gratuitous ARPs to flush any ARP caches which may have old data, and can then use the
address for further networking. The host must continue to monitor the network traffic in order to respond to any ARP probes advertised for the address that the host is currently using.

**IPv6 Address Autoconfiguration** [Thomson et al. 1996; Perkins et al. 2001] also provides stateless address configuration that takes advantage of the size of the IPv6 address space and the uniqueness of link layer addresses such as MAC addresses.

Each interface generates a link-local address, which is never routed off the local link. This is done by taking the hexadecimal value FE80 and appending the link layer address - typically, the MAC address - with zeros filling the remaining bytes between the link local identifier, and the link layer address. In IPv6 notation, this is FE80::(mac address), where the :: notation indicates that enough zeros are inserted to pad the resulting address out to the required 128 bits.

That address is then configured onto the network interface. At this stage, the address is considered tentative, and the host joins appropriate multicast groups. One of the multicast groups (solicited-node) is then used to check that the tentative address is unique by using a Neighbour Solicitation message, which is part of ICMPv6. If a node on the link is already using the desired address, i.e. duplication is detected, then the host that is already using the address replies with a Neighbour Advertisement, and the tentative address is abandoned, and autoconfiguration stops on that host.

If there is no answer to the Neighbour Solicitation message, then the address is unique, and the interface transitions from tentative to preferred. The host then sends a Router Solicitation to the all-routers multicast address. Each router will respond with a Router Advertisement message and for each response with the autonomous bit set, an address is generated consisting of the prefix provided in the Router Advertisement, and the interface hardware address. These addresses are assigned to the appropriate interface, and are then available for use.

**IPv6 Global addresses**, which are identified by an FP of 001, and which are also called aggregate-able global unicast addresses, are equivalent to public IPv4 addresses. Global addresses are globally routable and reachable on the IPv6 Internet.

As the name implies, global addresses can be aggregated or summarized to produce an efficient routing infrastructure. Unlike the current IPv4-based Internet, which has a mixture of both flat and hierarchical routing, the IPv6-based Internet is designed to support efficient hierarchical addressing and routing. The scope of a global address, which is the region of the IPv6 inter-network over which the address is unique, is the entire IPv6 Internet.
Although IPv6 has a solution for global addresses available to cover with a huge amount of nodes that will be connecting to the site, but it only supports for the nodes that are directly connected to the site infrastructure.

2.2.1.2 Multi-Hop Communication

In MANET, a considerable amount of routing protocols has been proposed. These protocols have reactive strategies (AODV [Perkins et al. 2002], DSR [Johnson et al. 2001a] and TORA [Park et al. 1998, 2001]), pro-active strategies (DSDV [Perkins et al. 1994], WRP [Murthy et al. 1996]), flat topologies (ZRP [Haas et al. 2001], OLSR [Clausen et al. 2003], and FSR [Pei et al. 2000]), hierarchical topologies (DDR [Nikaein et al. 2000], HARP [Nikaein et al. 2001]), geographical location strategies (GPSAL [Câmara et al. 2001], DREAM [Basagni et al. 1998]). A good summary can be found at [Abulhasan et al. 2004].

In the following we will discuss in more detail four examples of these protocols. Later in the next section we will discuss some multi-hop communication projects in mobile ad hoc networks.

The Ad Hoc On-Demand Distance Vector (AODV) routing protocol [Perkins et al. 2002] is intended for use by mobile nodes in an ad hoc network. It offers quick adaptation to dynamic link conditions, low processing and memory overhead, low network utilization, and determines unicast between sources and destinations. It uses destination sequence numbers to ensure loop freedom at all times, even in the face of anomalous delivery of routing control messages. This solves problems, such as “counting to infinity”, associated with classical distance vector protocols. As many ad hoc routing protocols, an AODV-node informs its neighbours about its own existence by constantly sending “hello messages” at a defined interval. This enables all nodes to know the status about their neighbours, i.e. if they have gone down or moved out of reach. To resolve a route to another node in the network, AODV floods its neighbours with a route request (RREQ). A RREQ contains the sender’s address, the address of the sought node and the last sequence number received from that node if one exists. The receiving node checks if it has a route to the specified node. If a route exists and the sequence-number for this is higher than that supplied a new route is found. The node replies to the requesting node by sending a route reply (RREP). If on the other hand a route does not exist the receiving node sends a RREQ itself to try to find a route for the requesting node. If the original node does not receive an answer within a time-limit the node can deduce that the sought nodes are unreachable. Since the request was sent to all neighbours the node may end up with several routes but they are easily separated by the sequence numbers. Nodes along the route keep their routing tables updated as long as traffic flows along the route. If not, the nodes will discard the routing entries after a specified period of time. To be sure that the route still exists, the sender has to keep the route alive by periodically sending packets. All nodes along the route are responsible for the
upstream links which means that a broken link will be discovered by the closest node. This node signals the broken link by sending an error message (RRERR) downstream so that the using nodes can start to search for a new route. AODV is a very easy routing-scheme. It has low overhead and supports multicast, but requires symmetric links.

The Dynamic Source Routing (DSR) [Johnson et al. 2001a] protocol is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes. DSR allows the network to be completely self-organizing and self-configuring without the need for any existing network infrastructure or administration. The protocol is composed of two mechanisms; Route Discovery and Route Maintenance. They work together to allow nodes to discover and maintain source routes to arbitrary destinations in the ad hoc network.

The use of source routing allows packet routing to be trivially loop-free, avoids the need for up-to-date routing information in the intermediate nodes through which packets are forwarded, and allows nodes forwarding or overhearing packets to cache the routing information in them for their own future use. All aspects of the protocol operate entirely on-demand, allowing the routing packet overhead of DSR to scale automatically to only that needed to react to changes in the routes currently in use.

The Optimized Link State Routing (OLSR) [Clausen et al. 2003] protocol for mobile ad hoc networks, is an optimization of the pure link state algorithm tailored to the requirements of a mobile wireless LAN. The key concept used in the protocol is that of multipoint relays (MPRs). MPRs are selected nodes which forward broadcast messages during the flooding process. This technique substantially reduces the message overhead as compared to the pure flooding mechanism where every node retransmits each message when it receives the first copy of the packet.

In OLSR, information flooded in the network "through" these MPRs is also "about" the MPRs. Thus a second optimization is achieved by minimizing the "contents" of the control messages flooded in the network. Hence, as contrary to the classic link state algorithm, only small subsets of links with the neighbour nodes are declared instead of all the links. This information is then used by the OLSR protocol for route calculation.

As a consequence hereof, the routes contain only the MPRs as intermediate nodes from a Source to a Destination. OLSR provides optimal routes in terms of number of hops. The protocol is particularly suitable for large and dense networks as the technique of MPRs works well in this context.

In reactive routing protocols such as AODV and DSR, nodes only maintain routes to active destinations. This characteristic requires a route search operation to be completed for every new destination. Therefore, due to the time required for route search the
communication overhead is reduced at the expense of delay. In pro-active routing protocols such as DSDV, every node continuously maintains the complete routing information of the network. When a node needs to forward a packet, the route is readily available. Thus, there is no delay in searching for a route. However, for a highly dynamic topology, these schemes spend a significant amount of already restrained wireless system resources in keeping the complete routing information up-to-date. The main advantage of routing protocols that consider flat topologies, such as OLSR, is their lower complexity. There is no distinction of nearby or faraway nodes, and there is no need to maintain a hierarchical structure. Therefore, they are generally simpler to implement, both in simulations and in practical systems. However, inefficient resource management can result from treating nodes equally, regardless of their relative location. For example, it may not be necessary for a printer service node to maintain very accurate route caching information of faraway nodes or ultimately any node. Therefore, these protocols may not scale well as the size of the network increases. In contrast, routing protocols with hierarchical topologies have scalability as their main advantage. It is also important to mention that routing protocols with geographical location strategies take advantage of position information to improve performance of unicast communication.

### 2.2.2 Mobile Ad Hoc Node and Internet Integration

Mobile Computing is becoming increasingly important due to the rise in the number of portable computers and the desire to have continuous network connectivity to the Internet irrespective of the physical location of the node. The Internet infrastructure is built on top of a collection of protocols, called the TCP/IP protocol suite. Transmission Control Protocol (TCP) and Internet Protocol (IP) are the core protocols in this suite. IP requires the location of any host connected to the Internet to be uniquely identified by an assigned IP address. This raises one of the most important issues in mobility, because when a host moves to another physical location, it has to change its IP address. However, the higher level protocols require the IP address of a host to be fixed for identifying connections.

A mobile device dynamically changes its access point to the Internet while roaming from one network to another. In the meantime, the mobile device must maintain all of its existing connections and remain reachable to the rest of the world. This should be guaranteed for a mobile device, to have an access to Internet anywhere at anytime. In this section we will provide a discussion of the research efforts in the area of linking the mobile node to the Internet. This includes; Mobile IP, which gives a mobile node moving at the Internet edge a direct connection to an access point that allows it to be reachable from other nodes on the Internet. Also NEMO, MIPMANET and MEWLANA projects, where a solution is proposed to allow nodes in mobile ad hoc networks to reach the Internet.

The Network Mobility (nemo) Working Group [NEMO 2003] is concerned with managing the mobility of an entire network, which changes, as a unit, its point of attachment.
to the Internet and thus its reachability in the topology. The mobile network includes one or more mobile routers (MRs) which connect it to the global Internet. A mobile network is assumed to be a leaf network, i.e. it will not carry transit traffic. However, it could be multi-homed, either with a single MR that has multiple attachments to the Internet or by using multiple MRs that attach the mobile network to the Internet. This work is not take node as individual but the group of nodes as one mobile network which is different from our concept of node-centric.

The Mobile Internet Protocol (Mobile IP) [Perkins 1997; Perkins 1998a;1998b;1998c] is an extension to the Internet Protocol proposed by the Internet Engineering Task Force (IETF) to address the node mobility issues. It enables mobile computers to stay connected to the Internet regardless of their locations and without changing their IP addresses. More precisely, Mobile IP is a standard protocol that builds on the Internet Protocol by making mobility transparent to applications and higher level protocols like TCP.

The IP address of a host consists of two parts; the higher order bits of the address determine the network on which the host resides, and the remaining low-order bits determine the host number. IP decides the next-hop by determining the network information from the destination IP address of the packet. On the other hand, higher level layers like TCP maintain information about connections that are indexed by a quadruplet containing the IP addresses of both the endpoints and the port numbers. Thus, while trying to support mobility on the Internet under the existing protocol suite, we are faced with two mutually conflicting requirements: (1) a mobile node has to change its IP address whenever it changes its point of attachment, so that packets destined to the node are routed correctly, (2) to maintain existing TCP connections, the mobile node has to keep its IP address the same. Changing the IP address will cause the connection to be disrupted and lost [Ghosh 2000].

Mobile IP is designed to solve the problem by allowing each mobile node to have two IP addresses and transparently maintaining the binding between the two addresses. One of the IP addresses is the permanent home address that is assigned at the home network and is used to identify communication endpoints. The other is a temporary care-of address that represents the current location of the host.

Mobile IPv4 [Perkins 2002] supports mobility by transparently binding the home address of the mobile node with its care-of address. This mobility binding is maintained by some specialized routers known as mobility agents. Mobility agents are of two types; home agents and foreign agents. The home agent, a designated router in the home network of the mobile node, maintains the mobility binding in a mobility binding table where each entry is a set of a permanent home address, a temporary care-of address and an association lifetime. The purpose of this table is to map a mobile node’s home address with its care-of address and forward packets accordingly.

Foreign agents are specialized routers on the foreign network where the mobile node is currently visiting. The foreign agent maintains a visitor list which contains information about
the mobile nodes currently visiting that network. Each entry in the visitor list is identified by a
permanent home address, a home agent address, a media address of the mobile node, and
an association lifetime.

IPv6, the next version of IP, is designed to be an evolutionary step from IPv4. IPv6
addresses are 128 bits long. Mobility support in IPv6 solves many of the problems of basic
Mobile IP. Some advantages of Mobile IPv6 over Mobile IPv4 include; Route Optimization,
which is built as a fundamental part of Mobile IPv6 unlike Mobile IPv4 where it is an optional
set of extensions that may not be supported by all nodes.

Foreign Agents are not needed in Mobile IPv6 [Johnson et al. 2001b]. The enhanced
features of IPv6, like Neighbour Discovery and Address Autoconfiguration, enable mobile
nodes to function in any location without the services of any special router in that location.

In Mobile IPv4, when a mobile node communicates with a correspondent node, it puts its
home address as the source address of the packet. Thus “ingress filtering routers " used to
filter out the packets as the source address of the packet is different from the network from
which the packet originated. This problem is tackled in Mobile IPv6 by putting the care-of
address as the source address and having a Home Address Destination option, allowing the
use of the care-of address to be transparent over the IP layer.

Mobile IP for mobile ad hoc networks (MIPMANET) [Jonsson et al. 2000] is designed
to give nodes in ad hoc networks the ability to access the Internet and the services of Mobile
IP. The solution uses mobile IP foreign agents as access points to the Internet to keep track
of the ad hoc network in which any given node is located and to direct packets to the edge of
that ad hoc network. In such a scenario an ad hoc routing protocol is used to deliver packets
between the foreign agent and the visiting node. In addition a layered approach that employs
tunnelling is applied to the outward data flow to separate the mobile IP functionality from the
ad hoc routing protocol. This makes it possible for MIPMANET to provide Internet access by
enabling nodes to select multiple access points and to perform seamless switching between
them. In short, MIPMANET works as follows:

- Nodes in an ad hoc network that want Internet access use their home IP
  addresses for all communication, and register with a foreign agent.

- To send a packet to a host on the Internet, the node in the ad hoc network tunnels
  the packet to the foreign agent.

- To receive packets from hosts on the Internet, packets are routed to the foreign
  agent by ordinary mobile IP mechanisms. The foreign agent then delivers the
  packets to the node in the ad hoc network.
- Nodes that do not require Internet access interact with the ad hoc network as though it were a stand-alone network—that is, they do not require data regarding routes to destinations outside the ad hoc network.

- If a node cannot determine from the IP address whether or not the destination is located within the ad hoc network, it will first search for the visiting node within the ad hoc network before tunnelling the packet.

By using tunnelling, MIPMANET can incorporate the default route concept into on demand ad hoc routing protocols, such as AODV and DSR, without requiring any major modifications. Packets addressed to destinations that are not found within the ad hoc network are tunnelled to foreign agents. However, in MIPMANET, only registered visiting nodes are given Internet access.

**Mobile IP Enriched Wireless Local Area Network Architecture (MEWLANA)** [Ergen et al. 2002] designed two protocols called MEWLANA-TD which uses a table driven routing type and MEWLANA-RD which uses a root driven routing type. MEWLANA-TD uses Destination Sequenced Distance Vector (DSDV) routing protocol [Perkins et al. 1994] in which there is a trigger of the update either periodically or when there is a change in the routing table. DSDV enables each node to have an entry in their routing table for all other nodes. MEWLANA-RD uses Tree Based Bidirectional Routing (TBBR) as the routing protocol. TBBR is a special routing protocol designed only by using MIP entities and introduces low overhead at the expense of performance degradation for inside traffic.

MEWLANA architectures consist of FA agent discovery, Mobile Node (MN) registration, tunnelling and routing mechanisms for supporting this seamless routing problem. MEWLANA use agent advertisement and beacon (a modification of ICMP (Internet Control Message Protocol)) to interchange the network information. Although agent solicitation is also used for eliminating the overhead of agent advertisement, it is applicable only when there is a low demand for outside traffic and when some of the nodes may not want MIP service. In order to avoid unnecessary processing by MNs, FA does not advertise itself by broadcasting agent advertisement messages but a MN who searches for a FA sends agent solicitation messages and then the FA periodically unicasts the agent advertisement messages to those who are connected. But if there is a high demand for outside traffic, the overhead from agent advertisement may be balanced by the increase in agent solicitation.

### 2.2.3 Network Issues Summary

Automatic IP address assignment schemes like APIPA [Cheshire et al. 2000] and the ongoing work of ZEROCONF [Cheshire et al. 2001] working groups show the possibilities of providing on demand IP addresses to mobile devices. By having an IP address any device in an Information Space can provide and get services of TCP/IP applications. This technology...
provides a useful building block in our architecture. The main weakness of the available automatic IP address assignment methods is the fact that they both are not designed for a mobile client, especially a node in a mobile ad hoc network. The challenge of this innovation is how to assign and manage the IP address that is given to devices when they also have a capability to move.

The multi-hop communication provided by the existing routing protocols within MANET such as AODV and DSR operate in the Network-Layer. Nodes in this ad hoc network act as routers. A good example of a performance comparison of multi-hop wireless ad hoc network routing protocols can be found at [Broch et al. 1998].

In our requirements we need to have all nodes in an ad hoc network to use the same network address domain. Therefore, our research aims are to join these works together, fix the missing points and provide personal mobility devices services that are reachable from both wired and wireless sides of the network. The idea of one big ad hoc network will allow nodes to communicate with an Infrastructure network using the Access Point. The Access Point uses a Network Address Translation (NAT) functionality to replace the node address with an Internet-routable one if Internet access is needed. Therefore the node will register an access point as its gateway and still use its address obtained by its selfconfiguration to communicate with its neighbouring nodes.

### 2.3 Challenges and the Way Forward

Visualise a high street shopping area, which is a simple outdoor environment. The street is full of shops, restaurants, street vendors and other people. We pop in and out from one store to another (structured services), buy a quick snack from a street vendor (unstructured services) and greet people we know (neighbour nodes). All of these activities happen within our focal view. Mobile devices within real-world environments have to work in the same way as our shopping area analogy. This provides users of mobile devices, with the ability to interact and use services in the same way we interact with shops and people within our real environment.

Using Local Information Filtering based on GPS positioning systems [Shaw 2000; McNeff 2002], mobile equipment can perform location filters including such things as traffic and weather reports; nearby restaurants and theatres; temporal filters sent "just in time" information such as the advertising of a restaurant's nightly specials menu and discounts. This can be from the network or from a nearby services node.

Ad Hoc Networks are more flexible, being formed on demand by mobile devices. They do not have any form of a permanent infrastructure. All the nodes that appear in the network should communicate with each other and use any kind of services that are offered. We expand the idea of an Ad Hoc Network to allow a subset of nodes within the network to simultaneously access services from the surroundings and an infrastructure networks.
To be able to provide this integrated environment for an ad hoc network like in the high street shopping area scenario described earlier, we need to rethink the ability of the ad hoc network.

An area that has to be considered additionally is where the node’s own network coverage area interacts with that of an infrastructure network. The requirements of a Global Wireless IP scheme for mobile devices in the Information Space are summarised as follows:

- A mobile device should have an IP address and support protocols that enable it to communicate seamlessly with both Ad Hoc and Infrastructure Networks.
- An IP address should be automatically assigned without human intervention or any server. Each IP can be changed without rebooting or any other effect to the application (connection oriented session may need a special algorithm). A Mobile Node should have the capabilities to detect duplication and change their IP if necessary.
- Each mobile device is the centre of its own network not just part of it. This network should be able to move along with the mobile device (Node-Centric network). Interrupted service functions should be remapped with available ones in the new locations that the user migrates to.
- Communication to the mobile device can still be established through the user profile at (Mobile-IP) Home Network if other parties are not in the same environment.
- Every Access Point (nodes that act as gateways also count) should be configured to accept the connection from one large private, non-routable IP Address.
- Mobile devices can join the infrastructure network without leaving their own network.
- Communication area should be defined by MAX-HOP in repeater/bridge manner and transparent to the user.

There are four basic technical requirements that are needed to provide the integration services between an ad hoc network and the Internet. These are,

- Address Configuration,
- Multi hop Ad Hoc Communication,
- Integration with the Internet,
- Integration of Structured/Unstructured Services and its utilisation.
In the following chapter, we will consider in more detail our proposed Integrated Personal Mobility Services Architecture and its Global Wireless Framework to show how we achieved the first three goals. We will then show how this integration of services can be utilised.

2.4 Summary

In this chapter we discussed the ongoing research efforts and projects in the area of architectures for mobile computing, mobility services and mobile computing networks.

Information Space [Mingkhwan et al. 2001] is the concept of integrating information and services from the environment the user has access to. By considering the Mobile User as the centre of the surrounding information and services (Node-Centric network) we find that, in reality, the environment the user moves into provides both structured and unstructured services. The ability to select and use both of these types of services to offer the maximum flexibility for the user is of paramount importance.

The need for an integrated Information Space will require the bringing together of wired with wireless networks and their services. In particular, the challenge is to bring together services within mobile ad hoc networks and infrastructure networks like the Internet [Jonsson et al. 2000; Mingkhwan et al. 2003b]. Service nodes provide services throughout the Information Space by using Gateways that interconnect infrastructure networks and mobile ad hoc networks together.

Mobile ad-hoc communication [Frodigh et al. 2000; Chlamtac et al. 2003] has been one of the most active research areas in the past decade. From military fields, critical rescuing operations, high demand operational headquarters to personal homes and wearable devices, instantly networking many devices without a central infrastructure has made the dream of “communication anytime, anywhere” possible. The goals of anytime, anywhere, for any services, to any mobile users’ devices are set as crucial goals of many research and industrial projects [Kleinrock 2000b; Helal et al. 2001; Kleinrock 2003]. To enable services from the Internet and the environment at the same time requires construction of new protocols and techniques. Wireless nodes have to have and operate their own network and be able to communicate with other nodes in their communications range while simultaneously getting Internet information through access points. Seamless integration that allows nodes to move between Wireless Ad Hoc Networks and a Infrastructure Network is needed. In this thesis, we present the Integrated Personal Mobility Services Architecture (IPMSA) to integrate these two domains providing any available services from an Information Space to the mobile user [Mingkhwan et al. 2002a].
Chapter Three: Integrated Personal Mobility Services Architecture

Personal Mobility Services is a growing area of research aimed at supporting the convergence of Mobile Computing and Wireless technology. User requirements in this paradigm are to be able to discover and use all the needed services that are offered in their surroundings while they are moving from one place to another. In our work we propose a solution that can seamlessly support the needs of users by offering services from both the Infrastructure Networks and Ad Hoc Networks that a user simultaneously migrated to.

Most of the present day networks are “Infrastructure Networks”. Every node in our definition of Infrastructure Network is already defined to where and how to connect to the network using a specific point, password and protocols. Example nodes for this kind of network include our desktop PC that connects to the university network, our wireless notebook that connects to the wireless access point and our mobile phone that connects to the provider cell. In the literature there are two different important works on mobility in the infrastructure network.

- **Node Mobility**: In Structured Network, users can use their mobile devices from one place to another using the roaming functionality provided by the access point [Enterasys 2003]. This technology allows users to continue their work while they are moving.

- **Mobile IP** [Geiger et al. 1997; Perkins 1997; Perkins 1998a, c; Johnson et al. 2001b]: This work of IETF supports mobility by providing a function that allows users to freely move and connect to any network. Other parties can reach them by reference with the Home Agent at the Home Network.

By combining these two efforts users can freely move, connect to their nearest network and still can be reached by other parties from the other side of the Structured Network.

From the above point of view, it seems to be a perfect solution but actually, there are limitations. Let us show it from the following point of view.
- User has to connect to access points (wired or wireless) to be able to get services from any service provider from the network. This makes it too restricted for the user. In our research, we consider users as freely moving nodes and can pick any different services around them that match their needs. In our vision users can still get the benefits of the conventional infrastructure network, but at the same time they can also have many unstructured services around them available at their request.

- From the services point of view, we should enable the system to provide services from the mobile node environment point of attachment instead of from the infrastructure domain. By doing this services around the mobile nodes including all those in the infrastructure network side will seamlessly be integrated [Mingkhwan et al. 2001, 2002a].

The aims of our work are to find solutions to give mobile devices the ability to use an infrastructure network’s services without joining the Network, i.e. without changing their IP addresses. By not being part of the infrastructure network, mobile devices can get the services that are offered to users from both the Wired-Site Networks and Ad Hoc Networks that they have migrated into simultaneously. This will allow a node to maintain its network with its surrounding nodes while these nodes are moving with it. This is instead of finding an Access Point to connect to the Infrastructure Network and discover its neighbours.

From a different point of view to Mobile Ad Hoc Network (MANET), our Node-Centric network is not to form the network that can interact within a group of nodes but to communicate with every nearby node and infrastructure network in its range of communication. The capabilities required for such an architecture are to be able to provide support to the new type of wireless mobile node that is working in the node-centric perspective like the shopping scenario discussed before. As a Node-Centric in a Global Wireless framework, a node is moving through the Information Space without joining any of the infrastructure access points.

In this chapter we will discuss in detail how we put all the ideas together for the proposed Integrated Personal Mobility Services Architecture. Starting from the requirements, we give our overview of the architecture and the key components functions. This will give an insight into how the architecture operates to satisfy the discussed vision. Finally we will end with a summary and enforce the main contribution of our research towards the goals of the proposed architecture.

### 3.1 Architecture Requirements

It is a reality that nowadays network and Internet systems are widely used for the growing commercial and public interest. Wireless technologies for public use like Wi-Fi ZONE™ [Wi-Fi 2003] bring Internet services to every mobile user on their move. But on the other hand
the services that are available to the user are mostly those services available on the Internet. As discussed in Chapter One the possibilities of implementing a high street shopping area analogy to wireless network services provision brings to our attention the limitations of the current technologies, this hugely inspired our motivation.

As proposed, services provided by current systems can be classified into two main groups:

- **Structured Services**: we define Structured Services as those provided with the help of a third party server, e.g. Directory Services, Name Server and Proxy. These kinds of services typically have complex structures, such as network connectivity, database access and multimedia functions.

- **Unstructured Services**: these kinds of service nodes provide services to the client independently, without the help of any other server. This concept is based on a simple services definition, like a radio or television remote control.

Service nodes in our system can provide either or both types of services. The main important idea here is how we can provide services to the Information Space anytime they are required. It should be up to the mobile node to discover and make use of the services available in the Information Space whenever the user migrates to the new environment.

The proposed Integrated Personal Mobility Services Architecture (IPMSA) allows clients in Personal Mobility environments to use concurrent services from both types of networks, either Infrastructures or Ad Hoc Networks. From the services provider point of view, the architecture also allows the flexibility to put any kind of services into the Information Space by means of open services architectures. The services variety will vary from the high-end structure services like secure payment to the very simple services like P2P digital content exchange.

The requirements of the architecture for mobile devices networking in the Information Space can be described as follows:

- Mobile devices should have the ability to communicate in both Ad-Hoc and Structured Networks simultaneously. Mobile devices should have an automatic IP address configuration ability which can be used to communicate in both Global Wireless and Infrastructure Networks. This will make the node able to communicate with both types of networks at the same time. Because the node is moving and may get into an area where other nodes get the same random IP, a Mobile Node should have the capabilities to detect duplication and negotiate to change of its IP if necessary. An IP address is automatically assigned without human intervention or any server. This address should be assigned from one large, auto-assigned, private, non-Internet-routable IP address space to provide anytime, anywhere services to any nomadic end...
user devices in the Information Space. Each IP can be changed without 
rebooting or any other effect to the application.

- Each mobile device should be the centre of its own network, we define this as 
a Node-Centric Network, and this network should move along with the mobile 
device. Consequently mobile devices should be able to join the Infrastructure 
Network without leaving their own network. Dropped service functions should 
be able to re-map with available ones in the new locations that the user 
migrates to.

- All Access Points are required to be configured to accept the connection from 
the mobile nodes IP Addresses. The address spaces only need to be unique 
in the reachable area of one node and can be reused in another space if it is 
not related. Network Address Translation techniques should be used for the 
cross communication of the node in the ad hoc wireless and the infrastructure 
domains. Communications from the infrastructure network side to the mobile 
devices should be able to establish through the user profile at the Home 
Network if other parties are not in the same environment as defined by 
Mobile-IP [Perkins 1998a].

- The architecture should give the user the ability to compose and use the 
services that are available in their surroundings to perform the task they need.

These requirements give us an insight of what the architecture should support and the 
scope of which algorithm and technology to support that. In the next section we give an 
overview of the proposed architecture later in Chapter Four we describe its components in 
more detail.

### 3.2 Architecture Overview

Being motivated by the vision and requirements of using mobile devices in true and free 
mobility emulating the high street shopping analogy, we put together the necessary pieces of 
the components required to satisfy our needs. The full Integrated Personal Mobility Services 
Architecture (IPMSA) is illustrated in Figure 9.
As shown, the service composition in user devices is connected to service available space via either the Global Wireless or the Infrastructures Network or both simultaneously. We define our architecture assuming that a user interface application gives the ability to associate with many different services that are provided within the Information Space. This will be achieved by the provision of the main components of the architecture which can be divided into two parts:

a) Information Space is the joining of the Global Wireless and Infrastructure networks. Node from both sides can communicate through the gateway. The Global Wireless component [Mingkhwan et al. 2002b; Mingkhwan et al. 2003a; Mingkhwan et al. 2003b] provides the framework of the Node-Centric Network. This Global Wireless Framework is the main contribution in our work and it is the key to deliver the higher layer functions of the architecture.

b) Services Integration as shown in the architecture has two layers on top of the Information Space, Services Available Space and Services Composition. Services Available Space represents the services from the Information Space as equally behaved components. It provides a heterogeneous level that allows the Services Composition Layer to treat the services equally in user application. The Services Composition Layer provides the user with the ability to dynamically create the needed tasks from the available services in the Information Space [Mingkhwan et al. 2004].
In the two main parts of the architecture, there are a number of key components working and relating to one another. The details of the symbols and their functions in the architecture are as follow:

- **Services Composition Layer** is the core function that allows users to freely design their tasks and take care of that task.
- **User Task** is the profiles that are predefined or created by the user. These are created by composing the available services from the Information Space. Task in our architecture is defined as a group of services that are functioning together, not a series of execution as in programming workflow.
- **Service Function** is the definition of services offered in the Information Space. Service functions can be both Structured and/or Unstructured Services (e.g., search, print, and map). From the user point of view, these services can be associated and provided by another node providing the same function if the current node is out of service or out of the communication range.
- **Services Available Space** is a definition perspective of places where a service node is present. This services space can be a place nearby or a services node far away that provides services through the network. It provides a coherent repository of the available services.
- **Services Node** is a node that is connected within the Information Space and provides services for personal mobility equipments. Types of services can be either free or charged. Each services node has a services definition that can be discovered by the user equipment if they are nearby or can be located through the network. An example of this services node could be a printer at an airport that can print and charge for service.
- **Control/Communication** is representing the path used to communicate between user applications that use the services and the services within the Information Space. In this architecture the communication paths are lie between two Services Agents. By the help of the agents, this path will provide a secured tunnel to send the code or name between network and user equipment. This enables a user to establish the connection with the network in either visible or invisible modes. The invisible mode connection allows the users movements to be untraceable. The algorithm and design of this path including the following Server/Client User Agent will be considered for future work.
- **Server User-Agent** is an intelligence component that represents and works on behalf of the user on the network side to provide services, negotiate and carry on end-user work while they are off-line. This agent can be implemented at the gateway, the user home network or a mobile node moving between them.
- **Client User-Agent** looks after a services connection for the user and remaps it again if the user moves to a new environment to keep the services available. This agent will also take care of the housekeeping work of the mobile user application.

As shown in Figure 9, every client has a services composition ability at their application layer. The services composition has predefined tasks (default or created by the user). The services on offer are not limited, e.g. Location Information, Map, Printing, Storage. All the functions will map to a set of services from the service nodes in the Information Space (Structured/Unstructured Services servers).

### 3.3 Key Components of the Architecture.

As we described in the previous section the achievements of Integrated Personal Mobility Services Architecture are formed by the combination of the two main parts; the Global wireless Framework and the Services Integration. As shown in the architecture the Information Space is the integration between the proposed Global Wireless Framework and the Infrastructure Network. The Infrastructure network is the normal conventional network like the Internet. This means that the key contribution of the Information Space is the Global Wireless Framework. Here we will describe in more detail the Global Wireless Framework and Services Integration components to show how they will operate in our architecture.

#### 3.3.1 Global Wireless Framework

In this section, we discuss our solution to the addressing problem – the Global Wireless Framework – to suit the needs of the nomadic end user from the services point of view. By extending the capabilities of Ad Hoc networks this framework can provide services to a nomadic user from Ad Hoc and Infrastructure Network at the same time. A user does not have to become part of the infrastructure network to get the services.

From the concepts of the Integrated Personal Mobility Services Architecture, we separate network nodes in our paradigm into two main groups; one is nodes that work in the Internet domain (including wireless nodes that always connect when switched on) and the other is nodes in the Global Wireless Network. These two groups of nodes are connected by the Access Point which has a Network Address Translation (NAT) [CISCO 2003] functionality as illustrated in Figure 10. The nodes in the second group perform mainly in the ad hoc environment but also can get information from the Internet if they are in an access point communication range. The Access Point will use NAT functionality to replace the ad hoc node private IP address with a public one before sending their packet to the Internet.
Global Wireless can be viewed as one of the wireless modes that a node will go to as a default when it is switched on. Every node participant joins the same default ad hoc network name called “Global Wireless”. Access points participating in this framework also join their wireless interfaces as part of the Global Wireless network. This makes the entire Global Wireless Framework nodes able to connect with the Infrastructure Network through the access points. The concept of Global Wireless Framework will also change the coverage from centred by an access point to the node itself as we mention in Node-Centric Network definition.

To achieve the requirements of the Integrated Personal Mobility Services architecture as described in the previous section we have to overcome three main obstacles to the Global Wireless Framework:

- Automatic Address Configuration
- Multi-hop Communication in Global Wireless Framework
- Integration with the Internet

These problems have mainly been addressed by groups of researchers such as Zeroconf [Guttman 2001b] and Automatic Private IP Addressing (APIPA) [Cheshire et al. 2000] for the automatic address configuration, MANET [MANET 2003] are looking at the multi-hop communication, Mobile IP, MEWLANA [Ergen et al. 2002] and MIPMANET [Jonsson et al. 2003] for the integration with the Internet.
are working on the Internet reachability. However, none of them addressed the whole solution for mobile users as described in the requirements of Global Wireless Framework.

### 3.3.1.1 Automatic Address Configuration

The Global Wireless Framework is based on the concept of Node-Centric network. In a Node-Centric network the node itself is the centre of its own network. Every node in the Framework can allocate its IP address without the aid of any established infrastructure or centralized administration and without any user-initiated configuration actions. The network time frame of using that IP address can vary from seconds to hours. IP addresses can change without a major effect to the application and does not require reboot.

By scaling down the network of the mobile devices to be just the communication ranges of the node itself, as described in the Node-Centric concept, we suggest that one large non Internet-routable IP address space (such as Class A Private) [Rekhter et al. 1996] is enough for all the nodes in that one node communication area. Outside the communication area of a node the IP address space may be reused. A protocol to detect and resolve duplicated IP addresses was defined in our paper [Mingkhwan et al. 2002b] and is detailed in Chapter Four. All the nodes that appear in the range of the Node-Centric network of one node should communicate with each other and be able to use any kind of services that are offered in its Information Space.

Using the node MAC Address and time as random parameters to avoid a duplicated IP allows conflicts to be resolved by using modifications of an ARP like protocol. If the same address was selected, the node that holds the IP will reply that it is using that IP address (also reply on behalf of other parties known to it or are still using its services), the client will then try a different IP address and so on until there is no conflict or it exceeds the maximum trying value. The main differences between the Global Wireless automatic configuration compared to zeroconf are:

- A bigger IP address space and
- The IP randomize-pick-up algorithm will use the MAC address and time to pick a random IP number. This will prevent the nodes from picking up the same address.

### 3.3.1.2 A GWF Ad Hoc Bridge

The aims of our work are to find a solution that can provide an integrated multi-hop communication for wireless devices working in one large network (same Network Address) as defined in the requirements. We proposed the Global Wireless Framework to provide the new way of connection to nomadic end users from the services point of view [Mingkhwan et al. 2002b]. We also need to extend the communication range of the node in our wireless network which can provide a bigger area of services to a nomadic user. Our early introduction of the framework in the previous section was more concerned with IP automatic
configuration. However since the Global Wireless Framework defines one large network, multi-hop communication can no longer be provided in the network layer and integration with the Internet can no longer be performed according to MIPMANET [Jonsson et al. 2000].

Therefore, to complete the vision of the Global Wireless Framework, that is, all devices in a large coverage area only use one network address and seamlessly exchange and offer their services, meant that we have to tackle two major challenges. Firstly, the internetworking of the different wireless networking protocol specifications available for the nomadic end user, for example IEEE802.11 WLAN and IEEE802.15.1 [IEEE-SA 2002] (Bluetooth). In our research laboratory other projects are looking into introducing interoperability solutions that interwork the Wireless Local Area Network (WLAN) and Wireless Personal Area Network (WPAN) domains into one single domain within the MAC services boundary [Abuelma'atti et al. 2002b]. The results of this work will help complete our vision.

Secondly, the need to provide this extended network with a multi-hop communications capability beneath the universal Network Layer. The whole idea of our multi-hop communication in Global Wireless framework is based on the philosophy of having a single networking domain where any device can join in its communications and offer its services to its extended neighbours, when the service node becomes offline its services also disappear while others are introduced to the network and so on. However adding a multi-hop functionality to these nodes will allow them to extend the wireless network range providing more services to more users in the environment. An implication of this is that we cannot implement any of the currently developed routing protocols and algorithms like AODV, DSDV within the established MANET group [MANET 2003] as all of them are tackling the problem from the networking layer point of view, i.e. assuming an integration between different networks.

Figure 11 illustrates a Global Wireless Framework that reflects these requirements. It shows integration between three major networks, an IEEE802.15.1 (Bluetooth) network, an IEEE802.11 ad hoc network and the structured Internet. IEEE802.15.1 and IEEE802.11 are considered as the main protocol specifications used by the nomadic end users devices, they also represent two domains in wireless networking, the WPAN and WLAN respectively. This extended framework shows two major points of integration:

a) The integration between the two wireless domains, WLAN (IEEE802.11 ad hoc) and WPAN (IEEE802.15.1 Bluetooth), where the WLAN is shown as the major network as it can accommodate more nodes,

b) The integration within the networks themselves, i.e. providing a multi-hop communication between their nodes. This integration needs to be deployed within the MAC services boundary in order to support the framework Global Network Address. Considering IEEE802.11 and IEEE802.15.1, the latter has no limitations since the scatter net topologies along with the master/slave switches already solve
the problem. However, the case of the IEEE802.11 ad hoc network is a major challenge since its protocol specifications has no definitions of multi-hop communication.

c) The integration between this whole Global Wireless Network and the Infrastructure Network will be described in the next section.

Figure 11 Multi-Hop Communication in Global Wireless Framework

As shown in Figure 11, the Global Wireless network may consist of one group of GWF Ad Hoc Bridge nodes and one group of Bluetooth Nodes linked together by the 802.11 AD HOC/Bluetooth Gateways. There are only two of GWF Ad Hoc Bridge nodes that are in the communication range of the access point. The rest of the nodes in Global Wireless network will need a multi-hop communication to get to the access point which is the gateway to the Internet.

3.3.1.3 Global Wireless Network and Internet Integration

The last key to achieve the requirements of the Global Wireless framework is to be able to communicate with the Internet. We focus here on the scenario of a mobile node that is moving while in the mean time connected to another node in its covering area and also get the information from the Internet.
Since the Global Wireless framework is just a one big network, to communicate with an infrastructure network, a node in the Framework needs to use the Access Point (AP) services. The Access Point will use a Network Address Translation (NAT) functionality to replace the node address with an Internet-routable one if Internet access is needed. Therefore the node will register an access point as its gateway and still use its same address for the Global Wireless Network communications.

On the other hand, a Node from the Internet can reach one in the Global Wireless network by using existing research efforts like Mobile IP [Perkins 1998b]. This is possible because nodes in the Global Wireless network are just one-network hop away from the access point.

### 3.3.2 Services Integration

As we join the networks around the mobile node to form the Information Space as mentioned in the early sections, the advantage from that is we can discover more services. How we utilise the new dimension of the available services in the Information Space is initially exposed in this part of the architecture. The Services Integration in our architecture is shown as two layers, Services Available Space and Services Composition, on top of the Information Space. The Services Available Space representation means that services from the Information Space are equally treated which makes no difference in the user applications at the Services Composition layer. Service nodes providing the services are either in the Global Wireless network or the Infrastructure network. The Services Composition Layer shows the ability that a user can create the task they need from the available services in the Information Space.

Before we can distribute and discover the available services, it is paramount that our framework addresses a number of key requirements. The very nature of a mobile node in Global Wireless network is that nodes will come and go over time; therefore we cannot guarantee the availability of services from the mobile node at any given moment. The challenge is also to enable both Structured and Unstructured services to exist in such a non-deterministic environment and allow mobile nodes to effectively expose and discover all the available services. Our initial work of utilising the integrated services proposed that:

- **Services Availability**: Services in Information Space must be available for both infrastructure and Global Wireless networks.
- **Dynamic environments**: Services have to be able to work in dynamically changing environments.
- **Platform Independence**: To support a variety of services and devices, a service has to be lightweight and capable of communicating across different platforms for example by using W3C standards [W3C 2003].
- **Ubiquity**: The deployment of services must include a variety of mobile devices ranging from workstations to mobile phones.

- **On-demand Services**: Services will not run continuously and will be invoked as and when they are required.

The availability of highly independent services offered by for example appliances in the home domain enables us to map different types of services to form an Implicit Function. The services within this environment can be both Unstructured and Structure services. This depends on how complicated the services themselves are. For example a DVD player service might be a Structure service while a lighting control is offered as an unstructured service. The architecture of such service integration network is illustrated in Figure 12.

![Home Appliances Services Composition Concept](image)

*Figure 12 Home Appliances Services Composition Concept*

In this home networked appliances service integration scenario, the Service Integration Controller (SIC), which would reside in a Home Appliances Integration Unit (HAIU), is the central component. The HAIU has the ability to connect to all the appliances in the home network by incorporating all the appliances physical network transceivers. The design allows for future extensions, widening the possibilities of providing high definition interoperable networking. The appliances can discover and advertise their services and the SIC coordinates the relations between the services in accordance to the user’s requirements. This scenario and its implementation is discussed in more details in the coming chapters.
3.4 Summary

In this chapter we discussed our proposed Integrated Personal Mobility Services Architecture. Since the Information Space is based on networks, which are a combination of the information from surrounding networks, and information from Infrastructure networks (which includes the local wired network), all the service nodes that represent their services in our work are providing their services through our Information Spaces by using Gateway Service nodes (access point) that interconnects infrastructure and global wireless together.

We clearly define in our architecture that the user interface application is able to associate with many different services that are provided within the Information Space. This will be achieved by the main components of the architecture divided into two main parts. First, the Information Space is the joining of the Global Wireless and Infrastructure networks. Nodes from both sides can communicate through the gateway. Second, the Services Integration, as shown in the architecture is divided into two layers, Services Available Space and Services Composition on top of the Information Space. Services Available Space representation means that services from the Information Space are equally treated and make no difference in the Services Composition layer in regard to user applications. The Services Composition Layer shows the ability of the user to create the task they need from the available services in the Information Space.

To complete the provision of our architecture for providing services from both surrounding and infrastructure networks to mobile devices in the information space, we have to achieve the important task of seamlessly joining surrounding network and infrastructure network. The Global Wireless Framework will provide those requirements for the architecture.

The provision of the architecture bring us to the clarify the problems of achieving the provision of Integrated Service as the requirements the Global Wireless Framework have to be implemented.
Chapter Four: Architecture Design

In the previous chapter we introduced our complete solution to an Integrated Personal Mobility Services Architecture. Within this architecture, our main research contribution will be to the problem of Global Wireless Framework. As shown in the architecture, services available from both sides of the network have to provide the ability to be discoverable by any node in the network seamlessly and equally.

In this chapter we give details of the main design of the Global Wireless Framework to show how it can be achieved. We detail the algorithms and techniques used to deliver our framework. We also give the initial design of the Integrated Services Composition component that builds on the advantage of having a homogeneous domain of the services available in our framework. The main requirement to provide services available to the user in a node-centric scenario is to be able to integrate the infrastructure and global wireless networks together.

4.1 Global Wireless Framework Design

The main idea of Global Wireless Framework is to allow the node to freely expose the entire surrounding environment and make most of the benefit from it. With this motivation in mind we came up with the Node-Centric concept where the node itself behaves just like a human. When we look around us, we find many people and many services that are ready for us to contact and use. We are the centre of our surrounding we make the decision and we see all of that as far as our eye sight can allow. We proposed our Global Wireless Framework to allow network nodes to be able to perform just like that. This brings a challenge; to extend the range of services available to the nomadic end user in this framework. To provide this we need to, a) include other networks in the environment, and b) provide a multi-hop communications to extend each node’s covering range.

Wireless network nodes are classified into two main groups; Infrastructure and Ad Hoc. The Global Wireless Framework node can be distinct from mobile nodes in an infrastructure and ad hoc mode as follows:
• **Infrastructure Mode**: Nodes are connected to the access point, receive and transmit their communication packets through the access point (BSS mode) or under supervision of the access point (IBSS mode).

• **Ad Hoc Mode**: in this mode, a group of nodes form the network to communicate with each other for a specific task at one period of time.

• **Global Wireless Mode**: Global Wireless is a working mode that we propose to give the freedom to the node to work in both ad hoc and Infrastructure networks at the same time. A Global Wireless node has the address and operation like ad hoc network but also use access point to provide its communication to/form the Internet.

As we stated earlier in Chapter 3, there are three main obstacles to the Global Wireless Framework:

- Automatic Address Configuration
- Multi-hop Communication.
- Integration with the Internet.

The following sections in this chapter will describe in detail the design of the algorithms and protocols that we use to support the Global Wireless Framework. At the end of the chapter we present our design of the Services Composition framework which builds on the advantage of the availability of services provided in the Global Wireless Framework to compose dynamic services to the Node-Centric user.

### 4.1.1 Automatic Address Configuration

As nodes migrate from one network to another, it is essential that they maintain contact with their services. To do so, nodes need their IP addresses to be automatically assigned and if migration causes duplication in addresses, an algorithm is needed to detect and resolve them. In our work we designed an algorithm to assign and maintain the Automatic configuration of IP addresses. It consists of two main functions:

- The Address Assignment Protocol, and
- The duplicated address detection.

As detailed in our published work [Mingkhwan et al. 2002b], a mobile client in the Global Wireless Framework is set to initiate an IP address by itself. By default this means that when the client boots in the network it uses a random IP address from a predefined IP list and automatically reconfigures itself. Predefined lists of IP addresses that can be used for any private network are assigned by Internet Assigned Numbers Authority (IANA) [Rekhter et al. 1996]. As shown in Table 1, in our experiments we used the private class A IP address
10.0.0.0 – 10.255.255.255 that is equal to 16,581,375 available addresses. This avoids any conflict with any other network that is using this same private IP in their Structured Network because their router has to eliminate this IP. This IP will be released after finishing the communication (one time use).

<table>
<thead>
<tr>
<th>Private Address Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 10.0.0.0 - 10.255.255</td>
</tr>
<tr>
<td>B 172.16.0.0 - 172.31.255.255</td>
</tr>
<tr>
<td>C 192.168.0.0 - 192.168.255.255</td>
</tr>
</tbody>
</table>

Table 1 Private IP Address Space

The client also configures itself with a default class A subnet 255.0.0.0. The client will continue using this IP address until it switches off or discovers any conflict. The use of class A IP address will provide the maximum number of client devices and reduce the number of conflicts.

The algorithm of picking up and assigning IP addresses in the Global Wireless node is similar to the work by zeroconfig [Guttman 2001b]. Potential IP initial-pick-up conflicts are resolved by using modification of an ARP like protocol. For example if the first client that switches on wants to assign itself an IP address of 10.10.10.1, it sends out an ARP broadcast, if it receives no answer, it keeps the address. When a second client boots up, it randomly picks the address 10.10.65.5. To avoid duplication on the network, it too sends a gratuitous ARP to ensure that no other clients in its area have selected the same address. Next, the third client switches on and picks 10.10.10.1, the same address as that of the first client. Once the first client receives the ARP broadcast, it will reply back to tell the third client that it is already using that address (also reply on behalf of other parties that still use its services). Then, the third client tries a different IP address and so on until there is no conflict. The algorithm is designed to allow the client to try self-configuration for up to 10 times (This number is arbitrary and would be optimised for real situations).

This ARP broadcast will work within the communication range of the node itself; referring to Figure 4 in section 1.1.2, this network will only be in a node communication area defined in the grey area. The neighbours in this area will response to the ARP broadcast if they use or know that the same address is being used.

Since each node in the Global Wireless Framework is mobile, the IP address that it was assigned may not be valid if it encounters another device using the same IP whilst on the
move. In Figure 13, we present a state diagram of the algorithm to detect conflicts and maintain IP addresses in our framework.

After obtaining an IP that can be used, the node is now able to discover/use/provide the services in the Information Space. This IP can be used without worry until it is disconnected from the network or detects that another node in its communication range is using the same address. Because the node is moving, the environment and the neighbour nodes are always changing. When a node finds the duplicate IP address, it should switch to the conflict detection state and solve the conflict to resume communication.

Figure 13 Flow Diagram of IP Configuration in GWF

As shown in Figure 13, there are four main processes in the Global Wireless Framework auto address configuration. These four processes can be describes as:

- **Assigned New IP**: In this state, the node will randomly choose an IP address from the IP address private pool using its MAC address and time as a random hash function parameter to avoid the duplication. The random IP address will be checked for duplication before assigned to the node.

- **Check Duplicated IP**: There are two different states of checking duplication in our algorithm. First, checking for the assigning process where the node will broadcast the ARP packet to check the duplication with other nodes in its area. Second is the checking for the IP address that it already has; and whether it collides with another node having the same IP address.
- **Used IP**: In this state, a node is using an IP address after it got it from the assigning process and already checked with its neighbours that this IP address is not duplicated. The IP address will be continually used until the node is disconnected from the network or found another node with the same IP address.

- **Release IP**: The first case of this state is when the node collides with another node that has the same IP addresses. In this circumstance the node will release the IP that it has been using and try to obtain another one that is unique in its communication range. The second case that the node releases its IP address is when it is disconnected from the network.

In this section we detailed the self-configuration process that the node needs to perform in order to have an address that allows it to participate in the communication of a Global Wireless domain. Once this is achieved the node is now ready to communicate with its surroundings. In the next section we detail a GWF Ad Hoc Bridge that we designed to support extended multi hop communications in the domain.

### 4.1.2 A GWF Ad Hoc Bridge

Founded on the challenges identified in the previous chapter where a node in the Global Wireless Framework requires a multi-hop communication between its wireless neighbours and therefore this multi-hop communication needs to be able to operate beneath the network layer to make all the nodes in Global Wireless Framework able to use the same network address. We implemented our algorithm to initialise and maintain the multi-hop communication for an ad hoc wireless network in a way that corresponds to the Global Wireless Framework and complies with the defined IEEE802.11 protocol specifications. We name it “GWF Ad Hoc Bridge”. In this section, we start by explaining the algorithm and protocol definitions to clarify our solution and show how the bridging system works. Next, we show the frame definitions that redefine the ARP frame where an example of the constructed bridging table is given, we also explain how it works, how to resolve the route and finally how to maintain the bridging table.

Providing a MAC layer multi-hop communication means that we need to redefine some capabilities of the involved protocols to allow them to support this multi-hop communications, we focus our work here on the main challenge of the IEEE802.11 WLAN. The standard specification does not specify a method for multi-hop Ad Hoc networking. However, in several experimental networks, MANET-based IP routing has been used [Elhakeem 2001]. Nonetheless, the experiments did not employ automated host configuration, that is, static IP addresses were assumed. The problems of the multi-hop communications within the same network domain (Global Wireless Framework) are considered in our work.
Within our research group other members are providing solutions to interwork IEEE802.11-Bluetooth. A Wireless Networked Appliances MAC Bridge has been proposed [Abuelma'atti et al. 2002b] as an interoperability solution that interworks the WLAN and WPAN domains below the MAC service boundary. It creates a single interoperable domain that allows applications to run seamlessly in the upper layers. That work defined the bridge’s Internal Sub-layer Services and its protocol primitives. The solution proposed a buffering and filtering functionality to dissolve the boundaries between the protocols operations and create interoperable frames according to the bridging-protocol relay entities. Ongoing work will also propose to link other standards such as Infrared and HomeRF. With this promising solution to provide multi hop communications between the other wireless standards we can now concentrate our work on providing multi-hop communications within the Link-Layer of IEEE802.11.

Apart from the forwarding packets from one node to another that is performed by the access point [IEEE-SA 1999], the IEEE802.11 WLAN protocol specifications does not define any support for multi-hop communications that is performed by the node itself in an ad hoc manner, especially in the link layer. This constitutes a challenge for the integration between the nodes of the ad hoc network in the Global Wireless Framework. Hence, we propose a GWF Ad Hoc Bridge that allows the nodes to extend their range up to MAXHOP hops away and therefore offer and use more services in the environment and accordingly extend the overall range of the Global Wireless Framework. In the following we define the requirements of a GWF Ad Hoc Bridge,

a) It should follow the Link-Layer operation, a suitable bridge protocol should not recommend a change to the Link-Layer operations or add any frames.

b) In our work we consider a MAXHOP = 4. This is a value that we arbitrary picked for experimental purposes. A higher MAXHOP value will affect the path reachability in a dynamic situation.

c) Based on the Global Wireless Framework, every node in the MAXHOP communication area can send and receive data to each other as if they are in the same network. The bridging protocol makes sure that the frames can be bridged in the link layer, thus allowing applications on the upper layers to communicate as if they are in the same domain.

d) The bridge should maintain a bridging table listing all the routes it can support in the MAXHOP range.

e) In a dynamic ad hoc network the status of an out-of-range node should be automatically updated in the bridging table. This will work using the time of the last communication of that node.
f) The protocol should provide the mechanism of not repeating a frame through the same path if that path is already marked as unreachable.

All of the above requirements have lead to our work on developing the proposed GWF Ad Hoc Bridge. In the IEEE802.11 Direct Sequence Spread Spectrum (DSSS) communications, the same frequency is used for transmitting and receiving. A bridge or repeater using this scheme must use some type of Store And Forward, in which frames to be repeated are recorded and played back a fraction of a seconds later. But how this wireless bridge will differ from the normal wired network bridge, seems to be the first question of our implementation.

In a normal wired network, a bridge is a link between two different collision domains. The bridge will learn which addresses of the nodes are in which side and only forward the packets if the address of the packet is in the other side of the bridge. Packets communicated between nodes within the same side of the bridge will be filtered.

Since communication of nodes in a wireless ad hoc network does not behave like the wired one, a bridge in a wireless ad hoc network can not be physically forced to sit between two sides of groups of nodes like in the wired network, where we cut the wire and put a bridge between two sections. Every node in a wireless ad hoc network will receive other node’s packets if the node itself is within the transmitting node communication range. This means that the wireless ad hoc bridge has to be more sophisticated than a normal wired bridge to be able to do the same job.

Unlike a conventional bridging system in a wired network, our GWF Ad Hoc Bridge operates by treating every neighbouring node in its communication range as a bridge port that can forward its frames to the next hop. Once a node sends a frame, every other node in its communication range will receive that frame. According to these special characteristics, to implement a multi-hop communication in an IEEE802.11 ad hoc network, we have to employ a new technique to filter the frames from the neighbouring nodes if the node is not in the correct route to the destination address.
In Figure 14 we show the node communication range and the MAXHOP communication range provided by a GWF Ad Hoc Bridge. The MAXHOP communication range is shown for the range of 4 hops away from the source node. In this section, we will use node-communication-range and MAXHOP-communication-range as defined in Figure 14 to explain GWF Ad Hoc Bridge algorithms.

From the IEEE802.11 specifications [IEEE-SA 1999], we note that it defines four different address fields in its MAC frame of the Wireless Distributed System (WDS), i.e. when the To DS and From DS bits in its header Frame Control field are equal to 11. In this case, the frame is initiated from another node in the bridged network and Address1, Address2, Address3 and Address4 represent the Receiver Address (RA), Transmitter Address (TA), Destination Address (DA) and Source Address (SA) respectively. The WDS frame is shown in Figure 15. We use this frame to send data form one GWF Ad Hoc Bridge to another over a multi-hop communication.

In our initial work, we defined a special control frame called “802.11 Ad Hoc Bridge Beacon Frame” to be a frame that carries neighbours information [Mingkhwan et al. 2003a]. Experiments have shown that this early design has excessive overhead since beaconing is repeatedly sent at specific periods of time and in our early algorithm this frame was repeated.
throughout the network even if the node itself does not want to send any data at all. Realising this problem, we proposed to replace the beacon frame by using an Address Resolution Protocol (ARP) [Plummer 1982; Hornig 1984] frame instead. Since ARP frames are only initiated by the source node when it wants to communicate with the destination node, this will reduce the unnecessary overhead compared to using the beacon frame. The ARP protocol also maintains its ARP table over a period of time and we can also use this to maintain our bridging table.

The proposed GWF Ad Hoc Bridge is a routing path in the Link Layer of the Mobile node. We provided this ability using the advantages of the ARP protocol. The ARP frame as shown in Figure 16 is the frame that works in the Link Layer for the purpose of getting the MAC address of a specific target IP address. By modifying the ARP frame as shown in Figure 17, a frame would be initiated by the source node in the network and forwarded by its neighbours until it reaches the MAXHOP. Every node will learn about its neighbours and therefore builds and updates its own bridging table.

![Figure 16 Normal ARP Frame Format.](image)

In the Ethernet Header of a normal ARP request frame, as shown in Figure 16, sender MAC address is the MAC address of the host sending the ARP request, target MAC address is the Ethernet broadcast address (FF:FF:FF:FF:FF:FF), frame type field is 0x806. For an ARP reply, sender MAC address is the MAC address of the host replying to the ARP request, target MAC address is the MAC address of the host that sent the ARP request, and the frame type field is 0x806. The ARP frame details are:

- **Hardware MAC Type** - Type of the hardware MAC address which is being mapped.
- **Protocol Address Type** - Type of the protocol address to which the MAC address is mapped. For IP address the value of this field is 0x800.
- **Hardware Address Size** - Size of the hardware MAC address.
- **Protocol Address Size** - Size of the protocol address. For IP, the value of this field is 4.
- **Operation** - Type of operation being performed. The value of this field can be 1 (ARP request), 2 (ARP reply).
- **Sender MAC Address** - The hardware MAC address of the host that is sending the ARP request or reply. This is the same as the source MAC address present in the Ethernet header.

- **Source IP Address** - The IP address of the host sending the ARP request or reply.

- **Target MAC Address** - The hardware MAC address of the host that is receiving the ARP request or reply. This is same as the destination MAC address present in the Ethernet header.

- **Target IP Address** - The IP address of the host receiving the ARP request or reply.

---

![Figure 17 Global Wireless Framework ARP Frame Format.](image)

In our work we modified the ARP frame; we call it GWF-ARP. This frame is used to carry the forwarding node information for up to 4 hops away as shown in Figure 17. The details of the added fields are:

- **No of Hop** – The number of hops that this frame has been forwarded. This field is increased by the value of one every time the frame is forwarded. In our work we set MAXHOP to 4. This means that the forwarding will stop at the fourth hop away from the original node.

- **F-1 MAC Address** - The hardware MAC address of the first host forwarding the GWF-ARP request.

- **F-1 IP Address** - The IP address of the first host forwarding the GWF-ARP request.

- **F-2 MAC Address** - The hardware MAC address of the second host forwarding the GWF-ARP request.

- **F-2 IP Address** - The IP address of the second host forwarding the GWF-ARP request.

- **F-3 MAC Address** - The hardware MAC address of the third host forwarding the GWF-ARP request.
- **F-3 IP Address** - The IP address of the third host forwarding the GWF-ARP request.

Founded on the details of the neighbouring nodes collected using our GWF-ARP frame, a node is able to pick up the information of its neighbours while put the information of itself into the frame and forward the frame to its neighbours. This forwarding will continue until reaching the MAXHOP value.

The information in the GWF-ARP frame can be read and transferred to an entity of the bridging table directly without adding any information to it. For example the node that receives the packet when the Hop Number field equals MAXHOP will not forward the frame. It will just read the information from the frame and discard it. As it is the last node, the information put into its bridging table will be that of MAXHOP entities equal to four. This mapping of the GWF-ARP frames to the bridging table is shown in Table 2. We call this bridging table GWF-ARP table.

<table>
<thead>
<tr>
<th>IP</th>
<th>MAC</th>
<th>FW-MAC</th>
<th>NO OF HOP</th>
<th>TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sender IP</td>
<td>Sender MAC</td>
<td>FW-3 MAC</td>
<td>4</td>
<td>T1</td>
</tr>
<tr>
<td>FW-1 IP</td>
<td>FW-1 MAC</td>
<td>FW-3 MAC</td>
<td>3</td>
<td>T1</td>
</tr>
<tr>
<td>FW-2 IP</td>
<td>FW-2 MAC</td>
<td>FW-3 MAC</td>
<td>2</td>
<td>T1</td>
</tr>
<tr>
<td>FW-3 IP</td>
<td>FW-3 MAC</td>
<td>-</td>
<td>1</td>
<td>T1</td>
</tr>
</tbody>
</table>

*Table 2 The GWF-ARP Frame to Table Mapping*

As an example of the GWF-ARP table, Figure 18 shows the bridging table of a node (Node5) after the node has learnt its neighbour’s information from the received/forwarded GWF-ARP frames.
The GWF-ARP table as show in Figure 18 shows that node 5 knows that nodes 7, 6, 4, 8, 2, 1 and so on are its MAXHOP neighbours. We can spot from the table that if node 5 need to send or forward the frame to node 1 it will take 3 hops and need the help of node 4 as forwarder.

Like all the other nodes of this sample topology, every entry in the bridging table of Node5 consists of the following fields:

- **IP**: This field stores all the known neighbours’ addresses within the MAXHOP communication range of the node.

- **MAC**: This field stores the MAC addresses of the IP node stored in the previous field. It also periodically get/update the TIME field of the same known node with the Source Address of the GWF-ARP frame and the received frames.

- **FW-MAC**: This field stores the MAC addresses of the next node in the routing path where the frame should be forwarded to reach the intended destination or null if no hop is necessary. The node gets this address from the latest Bridge Address entry of the GWF-ARP frame; null value is used in the field if the node is the first bridge (direct neighbour).

- **NO OF HOP**: By counting the number of the Bridge Address entries in the forwarded GWF-ARP frame, this field records the number of hops that are needed to reach the intended destination.
\textbf{TIME}: This is the information from the Time Stamp field of the GWF-ARP request/reply frame of the last communication with that particular node. This is used to keep the table updated. Old entries will be removed from the table if the time difference is more than the Expired-Time value that is previously set up.

For an example of how a particular node will use the information in the GWF-ARP table, let us follow these steps. When a node needs to send data to another node or forward a frame, it looks up the entries of the specific Destination Address. The Bridge recorded entries are ordered by the NO OF HOP (fourth column) to the intended destination. These along with the latest time of update are used to automatically pick the shortest and most reliable path. As an example, for the same destination address we first select the shortest path (fewer hops) from the entries that are not expired. If we still have more than one entry, we will select the entry that is most updated. Since there would be no entry of the same address that has the same time value, we will have unique entry information of the next hop that the node will forward or send data to.

The frame type used to send data in the Global Wireless Framework is a MAC frame of the Wireless Distributed System (WDS) as show in Figure 15. To send or forward a frame, a node will put the next hop forwarder in the RA field and put its address into the TA field. The frame is now ready to be sent and thus should reach the destination as detailed by our implementation of the GWF Ad Hoc Bridging protocol described in Chapter 5.

At this point nodes in the Global Wireless Framework should have the GWF-ARP table built up and ready to process the incoming frames from the surrounding nodes. The flowchart of Figure 19 details how the node will process the received frames.
The Bridge protocol algorithm flowchart as illustrated in Figure 19 shows the operation of a GWF Ad Hoc Bridged node. As discussed, after receiving the frame the system will update the status of the sender in the GWF-ARP table by putting the time stamp of the receiving frame in the TIME field of that node. The destination address (DA) will be checked against the node own address, and if that is the case, this frame will process as a normal received frame. If not, the system classifies the received frame into two categories:

1. **GWF-ARP Frame**: The GWF-ARP frame is categorised into two types GWF-ARP-Request and GWF-ARP-Reply. The different of how to process these two types of frames are:
- **The GWF-ARP-Request FRAME**: The MAC Destination address of this type of frame will be the same as the normal ARP-Request Ethernet frame which is a broadcasting address. When this type of frame arrives, the system will update its GWF-ARP table according to the information contained in the GWF-ARP frame. The information updated can be read from the frame as show in Table 2. The node will also update the time stamp of all relevant addresses. Then, if the frame has not reached the MAXHOP value, the node will enter its MAC address, IP address and re-broadcast the frame in its communication range. If it is the MAXHOP or the frame has already reached it, the node will drop the frame. This frame will flood the Global Wireless Network up to MAXHOP away in every direction.

- **The GWF-ARP-Reply FRAME**: This frame is generated to reply the MAC address of the requested IP. This frame will not be broadcasted but sent directly to the requesting node. It will also collect information of the forwarding nodes on the way back. When this type of frame arrives, the system will update its GWF-ARP table according to the information contained in the frame. The information updated from this type of frame will be the same as the GWF-ARP-Request information. Then, if the frame has not reached the destination, the node will enter its MAC address, IP address and forward the frame to the next hop or destination node. If this node is the destination, now the node knows the MAC address of the target IP node. This frame will not be forwarded and the node will start to send the data to the target IP using the MAC address received.

2. **All Other Frames**: when any other frame arrives, the Bridge algorithm will first check whether it is a GWF-ARP frame or not. If the node is the destination, the frame will be normally processed and sent to the upper layers. If it is not, it will check for the Receiver Address (RA), if its MAC is not equal to RA, it will drop the frame. If RA is equal to its MAC Address, it will look at the next hop MAC address for the Destination Address (DA), put the NEXT HOP MAC in the Receiver Address, put its MAC address as the Transmitter Address (TA) and forward the frame. If it can not find the DA in the GWF-ARP table it will drop the frame as well. After a timeout period of not receiving an ACK, the sender node will then try another path if it has or re initiate the GWF-ARP request frame again.

Using the Global Wireless Framework GWF Ad Hoc Bridge algorithm and protocol will provide a multi-hop communication within the Link Layer. This will result in a better range of services in an interoperable heterogeneous domain offering services from both the network side and the local environment. In the next section we will look at integrating these global wireless nodes with the Internet to integrate even wider range of services.
4.1.3 Global Wireless Network and Internet Integration

As discussed and shown in the previous section, the main aim of the Global Wireless Framework is that all nodes will belong to the same network address domain. In this section, we show how Mobile IP [Perkins 1997; Perkins 1998a; Perkins 1998b; Perkins 1998c; Jonsson et al. 2000] can be used to reach a Visiting Node (VN) in Global Wireless Network. The Foreign Agent (FA) will reside at the gateway to take care of VN.

To communicate with distant networks the node in the Global Wireless Framework needs to use the access point services. The Global Wireless Framework node will register an access point as its gateway (but would not have to change address when connecting to the Infrastructure Network). A Global Wireless Framework node will still use the same IP address that it was using when it was switched on and negotiated its address.

As shown in Figure 20, the access point has Network Address Translation (NAT) functionality [Egevang et al. 1994]. Communications through this access point with the Internet will use the public address picked up from the access point IP pool.

![Figure 20 Global Wireless Framework & Mobile IP](image)

The integration with the infrastructure network can be considered in two major cases, communication with a node in the Internet and a connection attempt from the Internet.

a) **Communication with a node in the Internet**: To communicate with distant networks (through the Internet), a node in the Global Wireless Network needs to use the gateway services (Access to the Internet Access Point). This will be
just the same as a normal infrastructure mode. The node will register an access point as its gateway. The node does not have to change its address when connecting to the Structured Network since the wireless interface of the access point that offered the service is also part of the Global Wireless Network. The access point will be assigning an Internet routable public IP for the GWF-node using the NAT function. It will also support the necessary parameters like DNS services and Proxy. Multi-hop communication is handled transparently in the Link-Layer by a GWF Ad Hoc Bridge protocol in the case of nodes not directly connected to a gateway. All the services would work in an on-demand basis and those available for the node will be mapped by Services Types / MAC Address or Internet ID of the node that provides the service, including those IP level services (services identification and provision is an ongoing work in our research group).

b) **Connection attempt from the Internet**: After a GWF-node connects to the access point and registers with its Home Agent. A request by a Correspondent Host (CH) at the Internet side to the VN is handled in the same way as in Mobile IP systems. As all the gateways also act as GWF-nodes and have bridging tables, each of which designates all the nodes in its communications range, they can notice the requested node by sending an RARP to ask that node for its current IP address if the TIME field in the GWF-ARP table is expired. A Node will then automatically configure itself to be ready for communication with the CH who requested the connection. The FA will perform the normal function after this point.

From the above process we can see that the integration to the Internet of the Global Wireless Framework is very straight forward. The foreseen properties of the Global Wireless node in the internet integration point of view are as follows.

- **Connect when needed**
  - A Mobile node will register with a preferred GW/AP in their communication area and update the table at the Home Agent (HA) (default profile of gateway provider can be set for an always-used, other GW/AP will appear as a choice).
  - For a gateway provider, MAC addresses can be used to enable for an Automatic registration and updated Home Network table, e.g. pre-paid for connection.
  - Nodes will accept a call from a Correspondence just like a phone call.

- Since Mobile Nodes may not hold an IP Address at that moment of connection, RARP is used to notify the Mobile node to configure their IP for the connection.
- Every time a Mobile Node gets an IP, it will update the table at FA.

*Chapter Four: Architecture Design*
Any user can be a GW/AP if they are willing to, e.g. user switch on their notebook with dial-up connection and BT/802.11 interface.

The advantage of our achievement to integrating Global Wireless Framework with the Internet is the elimination of the need for extra protocols like in MIPMANET [Jonsson et al. 2000] where it needs to run on top of the ad hoc routing protocol like AODV. The Global Wireless Network is only one Network-Layer hop away from FA, which reduces network overhead compared to MIPMANET.

4.2 The Proposed Use of Services Integration

The Integrated Personal Mobility Services Architecture provides a solution to the personal mobility services integration problem. The novelty is in bringing the new idea of a Services Space that consists of many service nodes from both sides of the network that provide services to users.

After achieving the Integration of services from both sides of the network by using the Global Wireless Framework we also proposed a few aspects of how we can use the Integrated Services that are found in the Information Space. In this section, we will first explain the idea of how a node can map the structured and unstructured services from both sides of the network. We then detail the Distributed Semantic Unstructured Services (DiSUS) framework that shows the idea of discovering and using unstructured services in the P2P environment. DiSUS [Fergus et al. 2003a] was developed in collaboration with another researcher to provide a demonstration of an application of IPMSA. Finally, we detail our novel work on utilising the available services in a dynamic services composition networked appliances environment.

4.2.1 Structured and Unstructured Services

After integrating the Infrastructure Network and the Global Wireless Network together, the Information Space will have many services available. In the previous chapter we classified services into Structured and Unstructured types. In this section we will show the details of how these two types of services can be discovered, mapped and used by the client node.

Our service mapping system as shown in Figure 21 is developed based on the combination of JINI and SLP to satisfy the needs of the client terminal [Mingkhwan et al. 2002c]. The service registry will advertise itself to a new node on introduction to the network. If there is no registry reply, the new node will use on-demand discovery services mode. The discovery and mapping services are maintained by the Client. A step by step of how the discovery and services mapping scheme works is as follows:

1. When a user switches on a client node, it will introduce itself to the Information Space.
2. All the Services Registry in the Information Space will list the available Structured Services that have been registered with it to the new client.

3. Structured Services that match the client’s needs are mapped to the predefined service profile functions in the client, which has a local registry for required services.

4. The rest of the user’s needed services will be assumed as an on-demand service. All these on-demand services will be discovered by a broadcast protocol.

5. The Unstructured Services that match the needs of the client are mapped and registered in the client registry.

6. Structured Services nodes that are introduced to the Information Space will register with the Services Registry. This node may appear either on the Infrastructure or Global Wireless Network.

7. New Structured Services information will be updated to Clients. Clients can check their previously unavailable required services with the new updated information. If new services become available they will be mapped as in step 3, and the status will change to ‘available’.

8. A new Unstructured Service that is introduced will advertise itself throughout the Information Space. Clients that need this kind of service will register it and change the function status to available.

**Figure 21 Client Discovery and Services Mapping**

After all services are mapped, the Client will maintain the availability of the service nodes that were mapped with the client function. The availability of the Unstructured Service nodes
is not as reliable as it can just come and go. Therefore nodes have to remap again with the similar services. There are two types of situations an unavailable service:

- Unstructured Services unavailable: Client User-Agent will send the query again for the same types of services.
- Structured Services unavailable: This information is updated from the Services Registry. When available the service will be switched to another node providing the same service.

### 4.2.2 Distributed Semantic Unstructured Services (DiSUS) Framework

From the services point of view and after implementing the Global Wireless framework, nodes will see each others as individuals who provide services. This can fit into the idea of having Peer-to-Peer network (P2P) services, where services can come and go in an ad hoc manner; i.e. provided for a specific period of time. In collaboration with another researcher in our Networked Appliances Lab, we developed a DiSUS Framework [Fergus et al. 2003a] to provide a demonstration of an application of IPMSA. This framework implements several subsystems to distribute and semantically discover services within a P2P network. It provides a description of subsystem interaction and overall framework capabilities. This system also sits on top of the Global Wireless Framework where there is no border between Infrastructure and wireless ad hoc networks nodes.

The DiSUS framework consists of three core components. These three components implement a P2P network, a knowledge base with inferential capabilities that allows us to semantically represent and query distributed peer services and a Web container capable of dynamically hosting Web Services. The integration of these components comprises the DiSUS framework as illustrated in Figure 22.

The framework consists of two types of peer services that can choose to implement any, all or none of these components dynamically. These peers in turn can be a Simple-Peer or a Specialised-Peer.

The DiSUS framework describes three peer services, which communicate with each other using standardised XML messaging wrapped in FIPA-ACL [Schalk et al. 2002] compliant objects, as illustrated in Figure 22. The two peers labelled Specialised-Peer-A and Specialised-Peer-B are peer services that have the ability to host Web Services, semantically match queries with local services on the peer machine and propagate queries within the P2P network. The peer labelled Simple-Peer-C is a simple peer and has limited capabilities. This type of peer service does not have the capability to host Web Services or respond to queries propagated from neighbouring peers. This peer service propagates its own queries and invokes discovered Web Services.
In this framework Web Services are hosted locally on peers. They are described within the peer’s knowledge base best describing, conceptually, what the Web Service means. This process enables the peer to capture the capabilities the service offers. Information about the Web Service itself is not entered into the knowledge base automatically, but rather a knowledge engineer relies on his/her understanding of the services capabilities and manually adds this information to the knowledge base. This should be a temporary measure, which has tested successfully to prove our solution. Research within our group is currently undergoing developments on algorithms to automate this process [Fergus et al. 2003b]. The challenge is to automatically create these representations and enable mapping processes between information contained in the knowledge base and the service itself.

The Web Services themselves are developed and hosted within distributed Web Service Containers. Invocation of Web Services is achieved in conformance with the WSDL 1.0 specification [W3C 2001] – this ensures that any compliant Web Service can be invoked by a peer irrespective of where the service is located, providing it has been conceptually described within a peer’s knowledge base.

Knowledge base servers reside on each Specialised-Peer, which ensures that the knowledge within the peer network is not centralised. This requirement allows us to move away from centralised registries such as JINI and UDDI and enables Web Service descriptions to be distributed within knowledge bases throughout Specialised-Peers connected to the P2P network. This ensures that there is no central point of failure and that
service descriptions are distributed throughout the network – if a peer becomes unavailable you only lose the services provided by that peer.

We believe this framework to be very flexible and it provides distinct advantages over other areas of research such as the Service Location Protocol (SLP) [Guttman 1999] and Avancha, S. et al. [Avancha et al. 2002]. Our framework has the ability to extend its reach outside an organisational LAN and access services contained within a peer network and semantically match queries with distributed services. Furthermore the DiSUS framework is protocol agnostic and we envisage that we will be capable of enabling interoperability between different protocols such as Bluetooth, WiFi and HomeRF. Our understanding is that [Guttman 1999] and [Avancha et al. 2002] are incapable of achieving this level of functionality.

Figure 23 describes the interaction between the user and two peer services. It describes the systematic interaction between subsystems in the DiSUS framework. It shows how services are semantically discovered, selected and invoked. The process begins when the user selects a service category from the user interface. After selection a query message is retrieved from an XML configuration file and wrapped in a XML message. This message is propagated to all peers on the network. When a Specialised-Peer receives a message it extracts the queries from the XML message and passes it to its knowledge base to determine semantically if it has a service corresponding to the received query.

**Figure 23 DiSUS Framework Operation**

It does not matter what name we give to services because services are not retrieved by keyword but rather on their capabilities. Services that have the same capabilities, but are
named differently should be conceptually equivalent, therefore they will be returned irrespective of what they are called.

If the peer has a service, the abstract name of the service is retrieved from the knowledge base and used to extract the Web Service’s relative URL from an XML file as illustrated in Figure 23. In the implementation we programatically retrieve the IP address dynamically from the peer machine at runtime and construct the URL by combining the IP address and Port number with the relative URL extracted from the XML file.

Before a Web Service is invoked, the peer service retrieves the WSDL file and programmatically determines what functions the service offers. Typically, the functions might be displayed to the user who then selects the required function (method) they want to use, or the program might intelligently invoke the most suitable service. This is outside the scope of our research. Before the service method is invoked the user is prompted to enter the required parameters. The Web Service processes this method invocation and a query response is created and wrapped within an XML message. This message is returned to the peer that initiated the query and presented to the user.

### 4.2.3 Dynamic Services Composition

The Dynamic Services Composition is a novel concept of how to utilise the hidden services that are redundant while an appliance is performing one function. This idea was detailed in section 1.1.4. In our published work [Mingkhwan et al. 2004], we proposed the idea of reengineering the appliances. That is to expose all the capabilities and services of one appliance and make them independently available from each other. We then put them in the network and let the user play the part of composing the tasks that they want. The possibilities of deploying such systems and creating the on demand virtual appliances are only limited by the user’s own imagination.

Our demonstrated system ensures the integration of services provided by service-enabled appliances within the home environment. The service integration process is based on Implicit Functions, which discovers and integrates services to form a specific configuration as shown in Figure 12 (in section 3.3.2). The Implicit Functionality is devised from the user’s requirements at any given time to form an abstract integration of services. To further ground our idea we defined the abstraction itself as a profiled list of services, which are combined in such a way as to define a high-level concept, such as a ‘Theatre.’

The system defines a common service description that must be implemented by participating service-enabled appliances. Each appliance situated within the environment must conform to description rules that specify how the device registers itself and how the services it offers are described.

An appliance provides one or more services, which are described using metadata. The metadata must provide information about the appliance the service it belongs to, version
information and the references to the compulsory components needed to describe the service as a whole – these components are the Service Profile, Service Process Model and the Service Grounding.

The metadata must also provide information to describe the type of service being offered, i.e. an audio or video service and information regarding the endpoint the service has, for example an IP address and Port number. This information is then used to publish the service once the device is registered.

![Figure 24 HAIU Operation.](image)

The operation of the protocol is illustrated in Figure 24. The Services Integration Controller (SIC) acts as the central point of contact for appliances and the Implicit Function formulator, to register services and invoke profiles. The formulator is defined as any appliance capable of viewing and interacting with the user interface provided by the SIC to install and view profile templates or create user defined profiles. The Home Appliances Integration Unit (HAIU) will provide commonly used profile templates, however the user should have the ability to create their own templates from scratch or download previously created profiles from the Internet.

In a real world situation an appliance will be placed within the home environment and switched on. This enables it to discover and connect to the SIC and initiate a request to register its services.
Before the services are registered the SIC ensures that the vocabulary is standardised and conform to the structure of the service descriptions, which are described using a variation of DAML-S [Ankolekar et al. 2002], can be understood and processed by any other appliance within the home environment. If the service description is valid, the appliance’s services are registered. The registration process itself involves adding the information about the appliance and its associated service descriptions to a centralised lookup table. Each appliance added to the environment will perform this process and register its existence and service capabilities.

The second function performed by the SIC is to process profile invocations received from the formulator. Again the service requests must conform to pre-defined vocabularies, to ensure the SIC understands the profile descriptions. The SIC processes the request and extracts all the services found in the profile contained in the request. In Figure 24, the formulator asks the SIC to run the ‘Theatre’ profile, which results in a message being sent to the ‘Player’ service. The ‘Player’ service receives the message and processes the information contained within it, which specifies that the player should begin to broadcast the audio and video information it has. In parallel the SIC also discovers that the ‘Theatre’ profile needs two additional services capable of processing the streamed audio and video information. In this process the SIC connects to the ‘Audio’ and ‘Video’ service defined in the profile and instructs them to receive the audio and video broadcast. Typical interactions between the Implicit Function formulator, the SIC and the services are illustrated in Figure 24.

Service profiles are stored on the SIC and used to resolve information about services, i.e. their location and how they are composed in a particular configuration. The profiles must use a standardised vocabulary, which enables the SIC to pre-determine the semantics of a profile and determine what services found in the profile match descriptions contained in the SIC registration table.

4.3 Summary

In this chapter we gave the detailed design of the IPMSA architecture. The main key components of the architecture are the Global Wireless Framework and the Services Integration.

Global Wireless Framework is a framework that uses one large, auto-assigned, private, non-Internet-routable IP address space to provide anytime, anywhere services to any nomadic end users’ devices in the Information Space. In such a network mobile devices need to communicate in an ad hoc manner and have the ability to perform a multi-hop coordination. However, the IEEE802.11 protocol specifications with its lack of multi-hop support definition constitute a challenge for the delivery and sharing of services in the integrated extended Framework. Our GWF Ad Hoc Bridge algorithm allows a GWF Ad Hoc Bridge client in the Global Wireless Framework to provide a multi-hop communication to
other nodes in the network and therefore extend its communication range and provide more services accordingly.

In this chapter we also presented the published ongoing work of how to utilise the available service from the Information Space in a few different situations. The works included Interoperability of the Structured and Unstructured Services, which is concerned about how client nodes can discover and map these two types of services to the required function from the Information Space. In our collaboration with other researchers in our Network Appliance Lab, a DiSUS Framework was developed to provide a demonstration of an application of IPMSA. DiSUS also investigated the semantic discovering of the required services which is still an ongoing work [Fergus et al. 2003b, c]. The last piece of the services integration is the Dynamic Services Composition which presents our novel idea of freely combining the services in our Information Space to perform user imaginary tasks.
Chapter Five: Simulation and Prototype Implementation

As presented earlier, the Global Wireless Framework is the key component of the Integrated Personal Mobility Services Architecture. It delivers the goal of integrating both sides of network services. To be able to study the framework we implemented its protocols and algorithms using the network event simulator NS-2 [ns-2].

A prototype that makes use of the integrated services was also implemented to demonstrate the benefits and novel concepts that can be harnessed from the new surrounding properties. The prototypes show inspiring ideas such as how to provide and discover the unstructured services and utilise the services in the Information Space by combining them into an imaginary virtual appliance.

This chapter will present in detail the Global Wireless Framework protocol simulation which is the key part of the IPMSA architecture and the work on the prototypes of the services integration that we implemented.

5.1 Global Wireless Framework Implementation

Following the presentation of requirements in Chapter 3 and the detailed design in the previous chapter, we need to study the performance and ability of the framework. We therefore implemented the Global Wireless Framework in the network event simulator NS-2 which is a widely accepted simulation platform within the research society.

NS-2 is an event driven, object oriented network simulator enabling the simulation of a variety of local and wide area networks. It implements different network protocols (TCP, UDP), traffic sources (FTP, web, CBR, Exponential on/off), queue management mechanisms (RED, DropTail), routing protocols (Dijkstra) etc. NS-2 is written in C++ and Otcl to separate the control and data path implementations. The simulator supports a class hierarchy in C++ (the compiled hierarchy) and a corresponding hierarchy within the Otcl interpreter (interpreted hierarchy). The reason why NS-2 uses two languages is that different tasks have different requirements: For example simulation of protocols requires efficient
manipulation of bytes and packet headers making the run-time speed very important. On the other hand, in network studies where the aim is to vary some parameters and to quickly examine a number of scenarios the time to change the model and run it again is more important.

The modifications implemented within the NS-2 network simulator to support our Global Wireless Framework will be described in details in the following sections.

5.1.1 NS-2 Wireless Simulator Module

In order to evaluate a GWF Ad Hoc Bridge protocol and algorithm of the Global Wireless Framework we had to implement our concept as an extension to the NS-2 simulator. We choose NS-2 as our network simulator for the Global Wireless Framework because NS-2 is a widely accepted in the network research society.

NS-2 is a discrete event simulator targeted at networking research. NS-2 provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. Using NS-2 simulator also save us a huge amount of developing time since the entire basic module simulation module is already provided. The part that we have to do is to implement our GWF Ad Hoc Bridge Protocol.

In NS-2, C++ is used for detailed protocol implementation and in general for such cases where every packet of a flow has to be processed. For instance, if we want to implement a new queuing or protocols discipline, then C++ is the language of choice. Otcl, on the other hand, is suitable for configuration and setup. Otcl runs quite slowly, but it can be changed very quickly making the construction of simulations easier. In NS-2, the compiled C++ objects can be made available to the Otcl interpreter. In this way, the ready-made C++ objects can be controlled from the OTcl level. In our implementation we had to deal with the 802.11, ARP and Link-Layer modules which mean that we had to implement our module in C++.

In this section we describe in detail the extensions we made to NS-2 in order to implement a GWF Ad Hoc Bridge protocol. To be able to control a new protocol like a GWF Ad Hoc Bridge we have to make modifications to the existing structure of the NS-2 wireless header. The structure was designed to carry the information of all the protocols that are in use in the simulator. This is the place where we should start to define our protocols and move the information form one mobile node to another. But most of all we have to understand what, where and when should we put our information to this header structure.
As illustrated in Figure 25, the header definition of the NS-2 simulator is a set of complicated structures separated into header and data. The special part that we are interested in is the wireless headers which consist of the Link-Layer header, MAC 802_11 header and the ARP header. This structured data of the frame header can be accessed through the header inline functions. Figure 26 shows an example of the inline function of the Link-Layer header in part of the NS-2 code. Here we have an inline function that return the value of the record of the structure hdr_ll such as seqno() which will return the value of the sequence number of the Link-Layer header.

```c
struct hdr_ll {
    LLFrameType lltype_;  // link-layer frame type
    int seqno_;           // sequence number
    int ackno_;           // acknowledgement number
    int bopno_;           // begin of packet seqno
    int eopno_;           // end of packet seqno
    int psize_;           // size of packet
    double sendtime_;     // time the packet is sent

    static int offset_;   
    inline int& offset() { return offset_; }
    static hdr_ll* access(const Packet* p) {
        return (hdr_ll*) p->access(offset_);
    }

    inline LLFrameType& lltype() { return lltype_; }
    inline int& seqno() { return seqno_; }
    inline int& ackno() { return ackno_; }
    inline int& bopno() { return bopno_; }
    inline int& eopno() { return eopno_; }
    inline int& psize() { return psize_; }
    inline double& sendtime() { return sendtime_; }
};
```

The other part of the simulator that we have to understand is how the simulator works and the schematic of each type of the relevant nodes that we will use in our simulations. This should enable us to understand how the data passes from one module or agent to another.
and where we can tap and modify it. We start the investigation by looking at the normal wired node schematic.

![Figure 27 NS-2 Schematic of a Wired Node](image)

The structure of the wired node as illustrated in Figure 27, it is not very complicated. Packets enter the node at the entry point then demultiplexed based on their destination address. This address demultiplexer corresponds to the routing table of the ordinary IP routing. If their destination address matches the node’s own address they are demultiplexed once again, this time based on their destination port number, and delivered to their respective application shown as an agent in the schematic diagram. If their destination does not match the node’s own, the packets are delivered to the link that corresponds to the correct next hop of the route. From the simple structure of the wired node we move to the more complicated structure of the MobileNode.

The wireless model essentially consists of the MobileNode at the core with additional supporting features that allows simulations of multi-hop ad-hoc networks, wireless LANs etc. The MobileNode object is a split object. The C++ MobileNode implemented in NS-2 is derived from parent class Node as described in the ns Manual [NS-2]. A MobileNode thus is the basic Node as shown in Figure 28 with added functionalities of a wireless and mobile node; like the ability to move within a given topology and the ability to receive and transmit signals to and from a wireless channel etc. A major difference between them, though, is that a MobileNode is not connected by means of Links to other nodes or mobile nodes. In the following we will describe the internals of the MobileNode, its routing mechanisms, the routing protocols, creation of network stack allowing channel access in MobileNode, brief description of each stack component, trace support and movement/traffic scenario generation for wireless simulations. We will follow on by detailing our extensions to the node structure.

Chapter Five: Simulation and Prototype Implementation
The schematic of a MobileNode is more complicated than the wired node. As shown in Figure 28, many changes are concerned with the layers below the network layer; the features presented like Link-Layer, ARP, MAC, radio interface, and propagation model can be seen in the figure. One major change is that the address demultiplexer is no longer used to decide which next hop to use. If the destination address matches the node’s own address the packet is delivered to the port demultiplexer, otherwise it is handed on to the routing protocol. It is then up to the routing protocol to decide which next hop to use and then hand the packet down to the lower layers.

The network stack for a MobileNode consists of a Link-Layer (LL), an ARP module connected to LL, an interface priority queue (IFq), a mac layer (MAC), a network interface (netIF), all connected to the channel. These network components are created and plumbed together in OTcl. The details of the MobileNode components are given below:

- **Routing Agent:** The Routing Agent is a special module developed to deal with the multi-hop communication of the MobileNode. An example of well known ad hoc routing protocols is AODV, DSR, TORA, etc. There is also one NS-2 routing
agent that does not route anything at all called “DumbAgent”. This agent is used when the node is not required to do any multi-hop communication. This agent is considered in more detail in our implementation since our MobileNode also does not do any routing in the network layer.

- **Link-Layer**: The Link-Layer (LL) object is responsible for simulating the data link protocols. Many protocols can be implemented within this layer such as packet fragmentation and reassembly, and reliable link protocol. Another important function of the link layer is setting the MAC destination address in the MAC header of the packet. In the current implementation this task involves two separate issues: finding the next-hop-node's IP address (routing) and resolving this IP address into the correct MAC address (ARP). For simplicity, the default mapping between MAC and IP addresses is one-to-one, which means that IP addresses are re-used at the MAC layer. The LL used by MobileNode is the same as that of the wired node. The only difference being the link layer for MobileNode. It has an ARP module connected to it which resolves all IP to hardware (MAC) address conversions. Normally for all outgoing (into the channel) packets, the packets are handed down to the LL by the Routing Agent. The LL hands down packets to the interface queue. For all incoming packets (out of the channel), the MAC layer hands up packets to the LL which is then handed off at the node_entry_point.

- **ARP**: The Address Resolution Protocol (implemented in BSD style) module receives queries from the Link-Layer. If ARP has the hardware address for a requested destination, it writes it into the MAC header of the packet. Otherwise it broadcasts an ARP query, and caches the packet temporarily. For each unknown destination hardware address, there is a buffer for a single packet. In case an additional packet to the same destination is sent to ARP, the earlier buffered packet is dropped. Once the hardware address of a packet's next hop is known, the packet is inserted into the interface queue.

- **IFQueue**: The IFQueue is implemented as a priority queue which gives priority to routing protocol packets, inserting them at the head of the queue. It supports running a filter over all packets in the queue and removing those with a specified destination address.

- **MAC**: The MAC layer handles collision detection, fragmentation, and acknowledgements. The IEEE802.11 MAC protocol has been implemented. This MAC protocols uses CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). One important feature of IEEE802.11 is that it allows operation in an ad hoc mode, apart from an ordinary Wireless LAN with access point. The MAC object simulates the medium access protocols that are necessary in the shared medium environment such as the wireless and local area networks. Since the
sending and receiving mechanisms are tightly coupled in most types of MAC layers, it is essential for the MAC object to be duplex. On the sending side, the MAC object is responsible for adding the MAC header and transmitting the packet onto the channel. On the receiving side, the MAC object asynchronously receives packets from the classifier of the physical layer. After MAC protocol processing, it passes the data packet to the link layer. The IEEE802.11 distributed coordination function (DCF) MAC protocol has been implemented by CMU [Carnegie Mellon University 2002]. It uses an RTS/CTS/DATA/ACK pattern for all unicast packets and simply sends out DATA for all broadcast packets. The implementation uses both physical and virtual carrier sense.

NetInterface: The Network Interface layer serves as a hardware interface which is used by the mobilenode to access the channel. This interface, subject to collisions and the radio propagation model, receives packets transmitted by other node interfaces to the channel. The interface stamps each transmitted packet with the meta-data related to the transmitting interface like the transmission power, wavelength etc. This meta-data in the packet header is used by the propagation model in the receiving network interface to determine if the packet has minimum power to be received and/or captured and/or detected (carrier sense) by the receiving node. The model approximates the DSSS radio interface (LucentWaveLan direct-sequence spread-spectrum).

Propagation: The radio propagation models are used to predict the received signal power of each packet. At the physical layer of each wireless node, there is a receiving threshold. When a packet is received, if its signal power is below the receiving threshold, it is marked as error and dropped by the MAC layer.

Channel: The Channel module simulates the actual transmission of the packet at the physical layer. The basic Channel implements a shared medium with support for contention mechanisms. It allows the MAC to carry out carrier sense, contention, and collision detection. If more than one transmission overlaps in time, a channel raises the collision flag. By checking this flag, the MAC object can implement collision detection and handling. Since the transmission time is a function of the number of bits in the packet and the modulation speed of each individual interface (MAC), the Channel object only sets its busy signal for the duration requested by the MAC object. It also schedules the packets to be delivered to the destination MAC objects after the transmission time plus the propagation delay.
5.1.2 An Implementation of a GWF Ad Hoc Bridge

From the description of the NS-2 MobileNode schematic above, we can locate the parts of the node that we need to extend and implement in our Global Wireless Framework.

![Figure 29 NS-2 Schematic of GWF Node](image)

The schematic diagram of our GWF-Node (Global Wireless Framework Node) is a modification of the MobileNode. As shown in Figure 29, the parts modified are concerned with the layers below the network layer. The main modification was carried out at the Link-Layer module and the ARP class. The replacements of the Global Wireless Framework module to the original NS wireless node are GWF-Link Layer and GWF-ARP for the Link-Layer and ARP respectively. We implemented our protocol in the following parts of the NS-2 MobileNode.

**DumbAgent**: The first change of the GWF-Node is to use the DumbAgent. As shown in the diagram, we use DumbAgent as the routing agent. This was done because we do not require any routing agent at the upper layers. This function is implemented in NS-2 to support non network layer multi-hop simulations. DumbAgent does not perform any packets forwarding process. Figure 30 shows the details of its receive-packet function.
void DumbAgent::recv(Packet *p, Handler *h=0)
{
    hdr_cmn *ch = HDR_CMN(p);
    hdr_ip *iph = HDR_IP(p);
    if (ch->direction() == hdr_cmn::UP) { // in-coming pkt
        if ((u_int32_t)iph->daddr() == IP_BROADCAST) {
            printf("Recvd brdcast pkt\n");
            demux_->recv(p, 0);
        } else {
            assert(iph->daddr() == here_.addr_);
            printf("Recvd unicast pkt\n");
            dmux_->recv(p, 0);
        }
    } else { // out-going pkt
        target_->recv(p, (Handler*)0);
    }
}

Figure 30 NS-2 DumbAgent Received

The receive-packet function of DumbAgent will check the direction of the incoming packet either going up to the upper layer or going down. In the case of going down it will pass it to the lower layer. If the packet is sending up it will check if it is a broadcast or a unicast address but either of that it will not do any routing process only pass it to the address de-multiplexer receive function.

struct hdr_arp {
    u_int16_t arp_hrd;
    u_int16_t arp_pro;
    u_int8_t arp_hln;
    u_int8_t arp_pln;
    u_int16_t arp_op;
    int  arp_sha;
    u_int16_t pad1;  // so offsets are correct
    nsaddr_t arp_spa;
    int  arp_tha;
    u_int16_t pad2;  // so offsets are correct
    nsaddr_t arp_tpa;
    int    arp_hop;  // Number of hop forward
    nsaddr_t arp_f1pa;  // Forward_1 IP Address
    int    arp_f1ha;  // Forward_1 Hardware Address
    nsaddr_t arp_f2pa;  // Forward_2 IP Address
    int    arp_f2ha;  // Forward_2 Hardware Address
    nsaddr_t arp_f3pa;  // Forward_3 IP Address
    int    arp_f3ha;  // Forward_3 Hardware Address

    // Header access methods
    static int offset_; // required by PacketHeaderManager
    inline static int offset() { return offset_; }  
    inline static hdr_arp* access(const Packet* p) {
        return (hdr_arp*) p->access(offset_);
    }
};

Figure 31 NS-2 GWF-ARP Header Detail

GWF-ARP: The GWF-ARP is our modification of the NS-2 ARP class to allow the information of the ARP protocol to be carried forward up to MAXHOP away from the requesting node. As detailed in the design discussed earlier in section 4.1.2, Figure 31
shows some of the added header fields that are used to get the next-hop information for the GWF-ARP Table.

Three forwarding node information fields have been added to the ARP header to collect the necessary information while the packet is forwarded up to the MAXHOP (4 hops in our simulation). This information is used to form the GWF-ARP table that is later used to determine the path (node) that the bridge will forward to in order to send the packet to its final destination.

```c
int ARPTable::gwfnexthop(int next_hop) {
    ARPEntry *a, *temp;
    int no_hop = MAXHOP+1; // GWF set initial hop number value
    for(a = arphead_.lh_first; a!= NULL; a = a->nextarp()) {
        if(a->macaddr_ == next_hop) {
            if((a->no_hop_ <= no_hop)) {
                no_hop = a->no_hop_;
                temp = a;
            }
        }
    }
    if (temp->macaddr_ == next_hop) return temp->next_hop_mac_;
    return 0;
}
```

*Figure 32 NS-2 GWF-ARP gwfnexthop() Function Detail*

One of the key functions added to the GWF-ARP extension to NS-2 as shown in Figure 32 is gwfnexthop(). The gwfnexthop() function is the function that look-up in the GWF-ARP Table for the next hop of the given destination address. This function looks for the fewer hops to the destination. This is determined by initiating the no_hop to MAXHOP+1 and looking in the table for the record that has the least hops to the destination address. The first found record of the least hop is the latest that was updated to the table. This will give us the most reliable path since this path is the latest and shortest path to reach the destination.

**GWF-Link Layer**: Since our Global Wireless Framework is adding new protocols and also a new packet type. The Link-Layer class has also to be modified to handle these new protocols. The GWF-Link Layer has to handle the multi-hop packet that was identified by the address type value of the incoming packets. We defined NS-AF-GWF address type to make the Link-Layer recognise and process our new packet type. This packet type is defined in the Packet class.
switch(ch->addr_type()) {
    case NS_AF_GWF:    // Added to support GWF mode
        mac_->hdr_gwf_ta((char*)HDR_MAC(p),arptable_->gwfnexthop(dst));
        break;
        
    
}  
if (tx == 0) {
    Scheduler& s = Scheduler::instance();
    // let mac decide when to take a new packet from the queue.
    s.schedule(downtarget_, p, delay_);
}  


Figure 33 NS-2 GWF-Link Layer Packet Classification

The code in Figure 33 shows how the GWF-Link Layer handles GWF-Packet (NS_AF-GWF). Every GWF-Packet that the destination address is not the node’s own will be checked for the next hop MAC address of the next forwarding node using gwfnexthop() function of the GWF-ARP class. The address of the next hop returned from the gwfnexthop() function will be put into the MAC header of the packet in the receiver field using the inline function of the MAC header from GWF-MAC. This function is defined to take care of MAC protocols. We will give more detail of this module in the following section. The ready to forward packet with the next hop address will be scheduled and sent down to the GWF-MAC Layer module as shown at the end of the code in Figure 33.

GWF-MAC: The GWF-MAC module is the extension to the MAC Layer of NS-2, in the original MAC Layer, every node assumes that the communication in this level will reach every node in the same network address. This should be changed if multi-hop communications are to be implemented in this layer.

Figure 34 GWF-MAC Multi-hop Handler

In the Global Wireless Framework, GWF Ad Hoc Bridge we extend the range to up to MAXHOP hops away which is a change the nature of MAC protocols especially in the part that is concerned with node communications. Applying our new algorithms, nodes have to recognise that packets are not only sending to the destination as in the original MAC layer, they are now sending to the forwarding node as well. The code in Figure 34 shows that the conditions of sending RTS not only check the address in the destination field but also the forwarder field as well.
struct hdr_mac {
    MacFrameType ftype_; // frame type
    int macSA_; // source MAC address
    int macDA_; // destination MAC address

    // GWF- added address to match with IEEE802.11
    int mac_gwf_TA_; // GWF next hop MAC address
    int mac_gwf_FA_; // GWF forwarder MAC address
    u_int16_t hdr_type_; // mac_hdr type
    double txtime_; // transmission time
    double sstime_; // slot start time
    int padding_; //

    inline void set(MacFrameType ft, int sa, int da=-1) {
        ftype_ = ft;
        macSA_ = sa;
        if (da != -1) macDA_ = da;
    }

    inline MacFrameType& ftype() { return ftype_; }
    inline int& macSA() { return macSA_; }
    inline int& macDA() { return macDA_; }
    inline int& mac_gwf_TA() { return mac_gwf_TA_; }
    inline int& mac_gwf_FA() { return mac_gwf_FA_; }
}

virtual inline int hdr_gwf_ta(char* hdr, int gwfta = -2) {
    struct hdr_mac *dh = (struct hdr_mac*) hdr;
    if(gwfta > -2) dh->mac_gwf_TA_ = gwfta;
    return dh->mac_gwf_TA();
}

virtual inline int hdr_gwf_fa(char* hdr, int gwffa = -2) {
    struct hdr_mac *dh = (struct hdr_mac*) hdr;
    if(gwffa > -2) dh->mac_gwf_FA_ = gwffa;
    return dh->mac_gwf_FA();
}

Figure 35 GWF-MAC Header File

Figure 35 shows the GWF-MAC header extension in the NS-2 code. We added two fields to the MAC header to match the address of the IEEE802.11 MAC. This is to make the code of the Link-Layer able to operate the same way for both MAC packets because it will reference to the same memory space. The same structure is also defined in IEEE802.11. We also show our inline function in the MAC class that is used to access the value of the MAC packet header for the value assigned by a GWF Ad Hoc Bridge Protocol.

The implementation of GWF Ad Hoc Bridge extensions as detailed in each part above enables the Global Wireless Framework to handle multi-hop communications in the lower layers in the same manner as the implemented routing algorithms operate in the upper layers. The advantage of our framework protocol algorithm is that all nodes in the Global Wireless Framework are able to use the same network address. The analysis and evaluation of the results will be discussed in the next chapter.

5.1.2 Global Wireless Framework and Internet Integration

To simulate the Internet integration of the Global Wireless Framework we need nodes with both wired and wireless interfaces to operate as gateways to the infrastructure network.
These nodes have also to support the Mobile IP functions. Packets coming from the wired interface will still enter the node at the entry point. The routing protocol used on the wired network will put entries in the address de-multiplexer. The wired-cum-wireless extension of the wireless model has paved the way for supporting wireless Mobile IP in NS-2.

Here we briefly describe the wireless Mobile IP implementation within NS-2. The mobile IP scenario consists of Home-Agents (HA) and Foreign-Agents (FA) and have Mobile-Hosts (MH) moving between their HA and FA nodes. The HA and FA are essentially base-station nodes while MH nodes are basically the mobile Nodes described in the previous section. There are two types of mobile IP enabled nodes in the simulator:

- **MobileNode/MIPBS**: MIPBS stands for Mobile IP Base Station. The HA and FA nodes have a registering agent (RegAgent) that sends beacons out to the mobile nodes, sets up encapsulator and decapsulator as required and replies to solicitations from MH nodes.

- **MobileNode/MIPMH**: MIPMH stands for Mobile IP Mobile Host. The MH nodes also have a RegAgent that receives and responds to beacons and sends out solicitations to HA or FA nodes.

Figure 36 illustrates the schematic of a MobileNode/MIPBS node. A typical such node type is the base station and the node that provides the gateway services.

The MobileNode/MIPMH node is very similar to this node except for the fact that it does not have any encapsulator or decapsulator. The MobileNode/MIPBS node routinely broadcasts beacons or advertisement messages out to MH nodes. A solicitation from a mobile node generates an advertisement that is sent directly to the requesting MH. The address of the base-station sending out the beacons is heard by MH and is used as the COA (care-of-address) of the MH. The COA of an MH node will change when it moves from its native to foreign domains. Upon receiving a register-request (as reply to advertisement) from a mobile host, the base-station checks to see if it is the HA for the MH. If not, it sets up its decapsulator and forwards the register-request towards the HA of the MH.

In case the base-station is the HA for the requesting MH but the COA does not match its own, it sets up an encapsulator and sends register-request-reply back to the COA (address of the FA) who has forwarded the register-request to it. All packets destined to the MH reaching the HA would be tunnelled through the encapsulator which encapsulates the IP packet header with an IPinIP header, now destined to the COA instead of MH. The FA’s decapsulator receives this packet, removes the encapsulation and sends it to the MH.
If the COA matches that of the HA, it just removes the encapsulator it might have set up (when its mobile host was roaming into foreign networks) and sends the reply directly back to the MH, as the MH have now returned to its native domain.

The mobile host sends out solicitations if it does not hear any advertisements from the base-stations. Upon receiving advertisements, it changes its COA to the address of the HA/FA it has heard the advertisements from, and replies back to the COA with a request for registration (register-request). Initially, the MH maybe in the range of the HA and receives all packets directly from its COA which is HA in this case. Eventually as the MH moves out of range of its HA and into a foreign domain of a FA, the MH nodes COA changes from its HA to that of the FA. The HA now sets up an encapsulator and then tunnels all packets destined for MH towards the FA. The FA decapsulates the packets and hands them over to the MH. The data from MH destined for the wired world is always routed towards its current COA.

The normal simulation for mobile IP in NS-2 consists of a MH moving between its HA and a FA. The HA and FA are each connected to a wired domain on one side and to their wireless domains on the other. TCP flows are set up between the MH and a wired node.
The schematic of the Global Wireless Framework Mobile IP Node (GWF/MIPBS) as illustrated in Figure 37 uses the same structure of the upper layer as with the MobileNode/MIPBS. Mobile IP algorithms are implemented at the network layer; they will not affect nodes in Global Wireless Framework. Since all mobile nodes in the Global Wireless Framework have the same network address and only one network hop away from the base station. Global Wireless Framework nodes do not need any network layer routing algorithms to perform a multi-hop communication.

After a GWF-Node connects to the access point and registers with its Home Agent. A request by a Correspondent Host (CH) at the Internet side to GWF-Node is handled the same way as in Mobile IP systems. [As all the gateways also act as GWF Ad Hoc Bridge nodes and have a GWF-ARP table, which designates all the nodes in its communications range.]

To communicate with distant networks (through the Internet), a node in the Global Wireless Network needs to use the gateway services (Access to the Internet Access Point). This will be just the same as a normal infrastructure mode of the IEEE802.11 WLAN. The
node will register an access point as its gateway. The node does not have to change its address when connecting to the Infrastructure Network since the wireless interface of the access point that offered the service is also part of the Global Wireless Network. The access point will be assigned an Internet routable public IP for the GWF-node using the NAT function. It will also support the necessary parameter like DNS services and Proxy. Multi-hop communication is handled transparently in the Data Link Layer by GWF Ad Hoc Bridge protocol in the case of nodes not directly connected to a gateway.

The implementation of Global Wireless Framework within NS-2 network simulator is in accordance to our design detailed in Chapter Four. This implementation module will be used for the simulation study detailed later in Chapter Six.

5.2 Service Integration Prototype

Our work in services integration as part of the architecture is an ongoing work that takes advantage of the availability of more services by integrating the ad hoc network and the infrastructure network by the Global Wireless Framework. This prototype is to demonstrate the vision of the Integrated Personal Mobility Services Architecture in the services integration component.

In this section we will cover some of our early prototypes proposed to show how we are going to use and provide services in the Information Spaces. We start by describing our services integration working scenario, then we detail our unstructured services framework and finally end with the dynamic services composition prototype that demonstrates the use of the architecture in the home appliances networking environment.

5.2.1 An Early Scenario of Services Integration

In our initial experiment of providing services, we constructed a services system that provides services to client devices according to a user interface definition. The implementation prototype uses a modified JINI services architecture to provide a structured network services situation. We also simulated Unstructured Services by using a WWW server to represent TV and Radio control.
Figure 38 Experimental Environment of IPMSA Prototype

Figure 38 shows the combination of the environments we used in our experiment. The main components of our Information Space are:

- **Services Registry:** This server takes care of information on all Structured Services available in the Information Space.

- **Profile Server:** This server is the user information storage that is used to support and keep the user data.

- **Structured Services Nodes:** In this experiment, we put some examples of structured services nodes including Home Control System Services (Light, Temperature and other Home sensors), Applications Services and Printing Service.

- **Unstructured Services:** Television, Radio and Information Display are used as examples of the Unstructured Services.

- **Client:** We provide three types of clients in this experiment; they are a Desktop PC, a Notebook and a PDA to simulate the different kinds of Client User Interfaces.
As shown in Figure 39 the IPMSA client framework was developed in our experiment based on a JiniClient with the addition of the functionality to handle Unstructured Services along with the Structured Jini Services. The IPMSAClient class combines the integrated services and client properties to implement the core features of the Integrated Personal Mobility Services Architecture (IPMSA). The IPMSAClient class provides the following client functions:

- Services discovery functions, Structured/Unstructured Services, mapping and maintaining availability as describe in section 4.2.1.
- Device capabilities management functions, these are used to provide an appropriate function interface to different kinds of device properties.
- Log-on functions, these provide capabilities to connect and retrieve users’ data from their information storage on the network.

5.2.2 Unstructured Services Prototype

The unstructured Services Prototype is designed to provide the unstructured services in a P2P network. After both the ad hoc and infrastructure networks are joined in the Global Wireless Framework, nodes are able to reach and be reached throughout the Information Space. The idea here is to show how nodes can interact with each other in an ad hoc manner. That is to provide come and go services as pleased like the street vendor described earlier in the shopping scenario (section 1.1).

In our prototype we have created a node that hosts several Web Services. These services are distributed and discovered within three Windows XP peers connected in a JXTA [Wilson 2002] P2P network on a Wireless LAN (IEEE802.11b) network. Two Notebook
Computers were used as mobile nodes and one Desktop Computer was used as a fixed wireless node. Figure 40 illustrates the prototype system configuration.

![Diagram](image)

*Figure 40 DiSUS Prototype System Configuration*

Although the machines used within our prototype are fairly high powered we envisage deployment of our prototype to much smaller devices such as PDAs and Mobile Phones, however our current research has not yet addressed this issue.

All software was constructed using Java based on the 1.4 JRE. We chose three toolsets to construct our framework – JXTA was used to implement the P2P network, OpenCyc [OpenCyc.org 2002] was used to implement the knowledge base and Inference Engine, and GLUE was used to implement the Web Service container. In conjunction with the three toolsets described, the FIPA-ACL [Schalk et al. 2002] standard was used to standardise message passing between peer services.

The test environment consisted of two Specialized-Peers that host three Web Services each and provide OpenCyc services to semantically processed queries received from the peer network. The third peer is a Simple-Peer and provides no services except the minimal functionality required, as described in section 4.2.2.

In the test environment, the Specialised-Peer-A hosts the following Web Services; PF Book service; a Restaurant Finder service and a Postcode Checker service. The Specialised-Peer-B hosts a AM Book service; a ABC Restaurant Finder service and a Thesaurus service.

The Web Service descriptions in our prototype are represented in the OpenCyc Knowledge Base using the built-in language provided by OpenCyc, which is CycL [Cycorp 2002]. This is a language based on Description Logics and allows information to be represented taxonomically based on classifications obtained from specific domains. As well
as a means of representing information, OpenCyc provides a built-in Inference Engine. This enables peers to make inferences about information in order to determine semantically what services it hosts and matches queries received from the peer network.

Queries are represented as CycL and have a similar syntax to Common Lisp [Guy L. Steele 1982]. These queries are wrapped in FIPA-ACL messages, which are propagated around the peer network. When a peer receives a FIPA-ACL message it extracts the CycL query and sends it to the knowledge base using the OpenCyc API. This API is developed in Java and allows the peer to connect to the OpenCyc server and invoke a number of functions to directly manipulate the knowledge base.

Information retrieved from the peer’s knowledge base will be the abstract names of URLs for services that conceptually match the query received. Service URLs are returned to the querying peer and once a peer service has the URL, the Web Service can be invoked. The peer service dynamically discovers the methods supported by the Web Service along with associated parameters by interrogating the WSDL file received when a binding to the Web Service is made. This is achieved by using the APIs provided by the GLUE [GLUE 2003] framework to interact and retrieve all required information from the WSDL file.

5.2.3 Dynamic Services Composition Prototype

The Dynamic Services Composition prototype is the latest novel approach that we proposed on how the services and its properties in the Information Space should be organised and used. Our published paper [Mingkhwan et al. 2004] presented the novel idea of services composition in a home networked environment scenario. The propositions are based on the idea of breaking the services of home networked appliances into the smaller units and recompose them back to the needs of the user in an as, when and how needed basis. This idea reduces the redundancy of the functional services in the appliances and allows users to compose a virtual appliance from the available services.

According to the design as described in detail in section 4.2.3, we implemented an early prototype to demonstrate the general scenarios of Implicit Functionalities using dynamic service composition. In this prototype we scope the services environment to the home environment and used a home entertainment theatre system as a case study. The prototype enables a user to invoke profiles stored on the Services Integration Controller (SIC) and start the services that stream audio and video to receiver’s service-enabled appliances – service profiles are processed by the SIC using the Jena toolkit [McBride 2001].
In our implemented prototype, we were using four wirelessly connected notebook computers to simulate service-enabled devices, which are based on IP networking. An example configuration of one node is illustrated in Figure 41. Each notebook is capable of hosting a visual service, an audio service and a player service. This means that our home entertainment system has the capabilities to provide four TV like capable services, four audio services and four DVD player services. All these services are independent and can be invoked separately as individuals. Once the services are registered to the Services Integration Controller (SIC) they can be composed and configured into a desired profile using the Implicit Functional Formulator. For example the audio and visual services offered by a TV appliance could be combined with the player service offered by a DVD appliance to form a ‘Theatre’ Implicit Function. Furthermore the audio service offered by a Hi-Fi appliance could be combined with the visual service offered by a TV appliance and the player service offered by a DVD appliance. The prototype offers the flexibility of combining any of the services available into any specified user desired configuration.
Using the profile creation user interface illustrated in Figure 42, the user can create Implicit Function profiles. The creation process begins by submitting a request to the SIC controller in order to discover all available services within the home environment. This results in a list of service-enabled appliances, which the user can select to compose a profile based on a desired configuration – this profile is then stored on the SIC controller for future use.

The process of invoking a pre-configured profile simply requires the user to load the profiles and select the required profile to run. Figure 43 illustrates how profiles can be loaded and selected. The invocation of individual services is achieved using the Simple Object Access Protocol (SOAP) [Englander 2002; Newcomer 2002].

![Figure 43 Profile Selection and Invocation User Interface](image)

Each appliance in our prototype hosts several services, which enables other appliances within the home environment to connect to it and invoke the operations it supports. The audio and video appliance services are implemented using the Java Media Framework (JMF).

Communications between all the appliances are achieved via standardised XML messages. We tested the prototype by combining the audio and visual services provided by a TV appliance with the player service provided by a DVD player appliance. When the services are run they connect to the SIC controller, register themselves and listen on a TCP/IP port for incoming service requests. When the ‘Theatre’ profile is invoked on the SIC controller, the ‘Player’ service is instructed to load an MPEG video file and begin broadcasting the audio and video stream data.

In parallel the SIC controller connects to the ‘Audio’ and ‘Video’ services and instructs them to begin receiving the video and audio data. The video and audio services achieve this by using the Real-Time Protocol (RTP).
The prototype demonstrates that we are able to integrate the services provided by appliances within our home entertainment system. It shows that our framework provides the flexibility to compose services into any desired configuration and store them as profiles. We are able to invoke these profile configurations, redirect the broadcast stream from one device to another, which demonstrates the ability to dynamically compose and control multiple appliances.

Our system also allows appliances outside a profile configuration within the same network to join the configured appliances system and receive video/audio broadcast streams in come-and-join manner by just picking up the broadcast stream. We are able to configure a device to listen on a well known broadcast port to receive information without having to join a particular profile configuration. This functionality brings with it additional considerations, more notably, security which is only provided by the existing wireless standard. In [Askwith et al. 2002] the authors proposed a framework for dynamic security analysis of software components in Networked Appliances systems. Such a framework would be of great use in our future work to securing Implicit Functions.

5.3 Summary

The implementations of the Global Wireless Framework show exactly how and where we implemented our protocols and algorithms within the network instance simulator NS-2. Our implementation of a GWF Ad Hoc Bridge as detailed in section 5.1 enables the Global Wireless Framework to handle multi-hop communications in the lower layers in the same manner as the implemented routing algorithms operate in the upper layers such as AODV, DSR and DSDV. The advantage of our framework protocol algorithm is that all nodes in the Global Wireless Framework are able to use the same network address. The implementations also show that our algorithms can be seamlessly implemented and work with other protocols already existing. The analysis and evaluation of the results will be shown and discussed in the next chapter.

This chapter also presented a services integration environment prototypes showing how we can use and provide services in the Information Spaces. We described our services integration working scenario. We also detailed our unstructured services framework and finally presented the dynamic services composition prototype that demonstrates the use of the architecture in the home appliances networking environment.
To allow evaluation of the Global Wireless Framework, we have implemented our solution in the network simulator NS-2 v2.27 [NS-2]. Chapter 5 provided details of this simulation implementation. In this chapter we present the results of carrying out these simulations on a limited set of scenarios to verify that Global Wireless Framework behaves as expected.

The simulations of the Global Wireless Framework are aimed at showing the validity of the developed protocols and algorithms that we proposed. The major mechanism within our framework that we have studied is the proposed GWF Ad Hoc Bridge that provides multi-hop communication in an ad hoc network. Aspects of multi-hop and adaptability in both static and dynamic scenarios are studied. The simulation results given in this chapter are conducted in the following order:

- A GWF Ad Hoc Bridge
  - Static Scenario
  - Dynamic Scenario

The objective of the simulations is to prove that a GWF Ad Hoc Bridge is able to support multi-hop communications within the Link-Layer. Proving this also implies that we can implement the Global Wireless Framework and as a result allowing all ad hoc nodes to use the same network address.

In this chapter we will describe our simulation environment, how we setup our simulation, and finally present the results of the simulations by comparing against the existing models of two of the ad hoc routing protocols discussed in the literature, being; AODV and DSR. This comparison is only aimed to verify the general idea of how the proposed GWF Ad Hoc Bridge operates and how it compares to others. This is not for the performance comparison objective, although obviously we are interested in achieving at least comparable performance.
6.1 A GWF Ad Hoc Bridge Simulation

As we already stated in our design, a GWF Ad Hoc Bridge aims at supporting different purpose requirements compared to Ad Hoc routing protocols. However to give an idea of how it performs we presented our results in comparison to the existing models of ad hoc routing protocols. The ad hoc routing protocols that we compare against are Ad Hoc On-Demand Distance Vector (AODV) and Dynamic Source Routing (DSR). Both AODV and DSR are widely implemented and available within NS-2. This also makes it easier for us to evaluate our protocol since we only have to develop models for our own proposed solutions. The different design approach also means that a GWF Ad Hoc Bridge forwarder node will only process packets up to the Link-Layer while AODV and DSR process up to Network-Layer. A general discussion of the processing and comprehensive simulations statistics are also presented in section 6.2.

All the data and values that we have studied in our simulations are captured within the Link-Layer. They are recorded by the trace agent in the NS-2 simulator. The Link-Layer data are indicated by “MAC” within the trace file records.

6.1.1 Static Scenario Simulation

The first studied simulation scenario is the static scenario of mobile nodes. Its simulation parameters are shown in Table 3. These static scenario simulations are designed to evaluate the static mobile nodes, i.e. mobile nodes that do not change their location in a short time of communication, including for example people in an ad hoc meeting or networked home appliances. It also covers moving nodes that do not change the communication shortest path, i.e. moving simultaneously.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmitter Range</td>
<td>250 Metres</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>5 Mbit</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>50 Sec</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>10</td>
</tr>
<tr>
<td>Environmental size</td>
<td>1000 x 1000</td>
</tr>
<tr>
<td>Traffic type</td>
<td>Constant Bit Rate (CBR)</td>
</tr>
<tr>
<td>Antenna</td>
<td>Omni Directional</td>
</tr>
<tr>
<td>Packet size</td>
<td>512 bytes</td>
</tr>
<tr>
<td>Number of flows</td>
<td>3</td>
</tr>
<tr>
<td>Traffic Time Interval</td>
<td>5 Sec</td>
</tr>
<tr>
<td>Multi-hop Protocols</td>
<td>AODV, DSR and GWF Ad Hoc Bridge</td>
</tr>
</tbody>
</table>

Table 3 Static Scenario Simulations Parameters
In this scenario we set up 10 wireless mobile nodes (0 to 9) in a flat grid 1000 x 1000 square metres area as show in Figure 44. Every node is equipped with an omni directional antenna which can communicate in 360 degree around the node. The transmission ranges of each node are 250 metres and as show in the figure every node is set up at 250 metres distance from its neighbour.

Figure 44 Static Scenario Simulation

As detailed in the commands shown in Figure 44, there are 3 communication flows within the simulation; all of them are a constant bit rate, they are:

- From node 0 to node 4. This is a 4 hops away communication flow that starts at time 0.0 seconds and stops at time 50 seconds. Thus this communication takes the 0-1-2-3-4 path which is the shortest path calculated by the algorithm detailed in Chapter Five.
- From node 5 to node 9. This is also a 4 hops away communication flow. It starts at time 10.0 seconds and stops at time 50 seconds. Same as 0-4, the communication path of 5-9 will take the 5-6-7-8-9 which is the shortest path.
- From node 7 to node 2. This is only one hop away path. It starts at time 15.0 seconds and stops at time 50 seconds. We choose the 7-2 communication path as these two nodes are also in the path of the 0-4 and 5-9.
The constant bit rate communication in each communication flow will send one CBR packet of 512 bytes every 5 seconds (time interval).

The results of the simulations are logged to a trace file by the trace agent of NS-2 and have been analysed by filtering only the activities of the Link-Layer.

<table>
<thead>
<tr>
<th>Static Scenario</th>
<th>Multi-Hop Protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GWF Ad Hoc Bridge</td>
</tr>
<tr>
<td>Avg. CBR Send Time MAC-MAC (Sec)</td>
<td>0.004608833</td>
</tr>
<tr>
<td>Avg. CBR Process Time Receive-Send (Sec)</td>
<td>0.001378922</td>
</tr>
<tr>
<td>Avg. Reach Time (Sec)</td>
<td>0.017886029</td>
</tr>
<tr>
<td>Start CBR sending time In path 0-4 path</td>
<td>2.502550553</td>
</tr>
</tbody>
</table>

*Table 4 Static Scenario Experiment Statistic*

The statistic results of the simulation analysis of a GWF Ad Hoc Bridge compared against the AODV and DSR protocols are shown in Table 4. They are compared in terms of the Average CBR Send Time, the Average CBR Process Time, the Average Reach Time and the Start Sending CBR Time.

The **Average CBR Send Time** is the average time taken to send a packet by the MAC agent of one node until it is received by the MAC agent of the next node. As we can see in the table a GWF Ad Hoc Bridge takes less time to reach the destination while AODV and DSR take longer time. By analysing the simulation results data we find that this fraction of time difference is cause by the size of the sent packet.

The **Average CBR Process Time** represents the average processing time taken by a node from receiving the CBR packet until it sends it out. In this experiment we found that a GWF Ad Hoc Bridge takes more processing time than AODV and DSR to process the packet before forward it to the next node.

The **Average Reach Time** is the overall average of the time taken in the simulation to send the CBR packets from the MAC agent of the sender to the MAC agent of the receiver node. The results show that a GWF Ad Hoc Bridge average reach time is less than that of the other two protocols.

The **Start Sending CBR Time** indicates the time in the simulation at which node 0 of the first starting communication path (0-4) starts sending the CBR packets. The in-depth discussion and characteristic comparisons of each protocol in our simulations are explained in more details in Section 6.2.
From the simulation results we extracted all the related records of the CBR packets and presented them as a graph to show how a GWF Ad Hoc Bridge will perform when compared against AODV and DSR. The graph of Figure 45 represents the End2End Cumulative Time of the simulation.

The End2End cumulative time is measured from the time that a MAC agent of a sender starts sending a CBR packet until it is received by the destination node MAC agent. This End2End Cumulative Time is the summation of the times used by all the CBR packets sent in the simulations.

In Figure 45 we present a comparison of the End2End cumulative times graphed against the percentage of the successfully received CBR packets that have been sent by each protocol in the static simulation scenario. From the percentage of the CBR packets that have successfully reached the destination, the graph shows that:

- In the first 15% of the packets the performances of each protocol are very close to each other. This is the time that each protocol performs its route search mechanism. Note that the communication flows started at this time are only 0 to 4 and 5 to 9 which does not cross each other.

![End2End Cumulative Time Static Scenario](image)
• It is clear from the End2End cumulative time of the CBR packets sent, that DSR uses more time to reach the destination than a GWF Ad Hoc Bridge.

From the overall End2End Cumulative time in our static scenario simulation we can note that a GWF Ad Hoc Bridge takes slightly less time than AODV and DSR for the packets to reach their destinations. It is important also to note that the End2End cumulative time performance of AODV is nearly the same as that of a GWF Ad Hoc Bridge.

In the next graph we present the simulation throughput of each protocol in each of the three communication paths. This data is generated from the total number of packets sent in each communication path by each protocol.

![Communication Path Throughput Comparison](image)

*Figure 46 Communication Path Throughput Comparisons in Static Scenario*

From the data shown in the graph of Figure 46 we can compare the throughput of a GWF Ad Hoc Bridge against the other protocols.

- In the 0-1-2-3-4 path: GWF and AODV have the same number of packets. DSR has the highest throughput in this communication path.
- In the 5-6-7-8-9 path: The results show that the performance of AODV and DSR is the same. GWF is only one packet less.
- In the 7-2 path: The AODV throughput is better than that of DSR and GWF which have the same throughput.
From the overall view of the throughput comparison results and the End2End time in this static scenario simulation we can conclude that a GWF Ad Hoc Bridge is able to conduct a multi-hop communication task. This proves that our assumption of allowing multi hop communication in ad hoc networks below the network layer is valid. This is a crucial requirement of our Integrated Personal Mobility Services Architecture. However, a GWF Ad Hoc Bridge throughput needs to be improved to enhance its performance. This is caused by the long time taken to discover the path before the first packet in a communication path can be sent out.

6.1.2 Dynamic Scenario Simulation

A dynamic scenario of our simulation is designed to test the performance of our GWF Ad Hoc Bridge when the mobile nodes are moving and the topologies and therefore the communication paths are changing.

This will allow us to compare the performance of the protocols adaptation to the new communication paths. We will also study the effects of this movement on the characteristic of each protocol by comparing its throughput.

<table>
<thead>
<tr>
<th>Dynamic Scenario Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmitter Range</td>
<td>250 Metres</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>5 Mbit</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>150 S</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>10</td>
</tr>
<tr>
<td>Environmental size</td>
<td>1000 x 1000</td>
</tr>
<tr>
<td>Traffic type</td>
<td>Constant Bit Rate</td>
</tr>
<tr>
<td>Antenna</td>
<td>Omni Directional</td>
</tr>
<tr>
<td>Packet size</td>
<td>512 bytes</td>
</tr>
<tr>
<td>Number of flows</td>
<td>3</td>
</tr>
<tr>
<td>Nodes Speed</td>
<td>5 M/Sec</td>
</tr>
<tr>
<td>Multi-hop Protocols</td>
<td>AODV, DSR and GWF Ad Hoc Bridge</td>
</tr>
</tbody>
</table>

*Table 5 Dynamic Scenario Simulations Parameters*

As shown in Table 5 our dynamic scenario simulations use almost the same parameters as in the static scenario. An addition is the characteristic of the node movement being a node speed of 5 M/Sec. We also extended the simulation time to 150 seconds to give the nodes enough time to move to the destination.
In this simulation the communication patterns are also the same as in the static scenario with 3 CBR communication flows from 0 to 4, 5 to 9 and 7 to 2. We allow the nodes to move in this simulation in the following manner:

- Node 4 will start moving from position (0,750) to (350,200) at time 20.00 Sec of the simulation.
- Node 6 will start moving from position (250,500) to (500,750) at time 20.00 Sec of the simulation.

The node position, communication pattern and the movement direction within this dynamic scenario are shown in Figure 48.

The movements in our dynamic scenario are designed with the aim to test 3 different communication situations within one scenario. These communication situations are:

- The static node behaviour within a movement scenario. We represent this by the communication between 7 and 2. This is a simple, one hop communication but still considered within the dynamic environment since 7 and 2 are also acting as forwarding nodes to the other communication paths.
- The movement that does not change the topology but only change its shortest path. We represent this by the communication between 5 and 9. The movement
of node 6 will not have an effect on the route of 5-6-7-8-9 which should still be able to achieve throughout the simulation period. But its movement to position (500,750) will offer the shortest path of 5-6-9.

- The movement that change the topology. We represent this by the movement of node 4. The movement of this node will change the topology of the communication between 0 and 4 from 0-1-2-3-4, 0-1-2-4, 0-1-4 and finally 0-4.

This change in the scenario should allow us to learn how each protocol performs in a dynamic situation. To analyse the simulation data we first present some of the statistics obtained from the trace tables of the dynamic scenario. Table 6 shows these statistics.

<table>
<thead>
<tr>
<th>Dynamic Scenario</th>
<th>Multi-Hop Protocols</th>
<th>GWF</th>
<th>Ad Hoc Bridge</th>
<th>AODV</th>
<th>DSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start CBR Send Time the 0-1-2-3-4 path</td>
<td>2.5026</td>
<td>0.0445</td>
<td>0.0850</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start CBR Send Time the 0-1-2-4 path</td>
<td>44.4568</td>
<td>N/A</td>
<td>51.5059</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start CBR Send Time the 0-1-4 path</td>
<td>83.1142</td>
<td>N/A</td>
<td>91.9691</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start CBR Send Time the 0-4 path</td>
<td>87.6984</td>
<td>146.9534</td>
<td>98.2024</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start CBR Send Time the 7-2 path</td>
<td>15.0118</td>
<td>15.0053</td>
<td>15.0013</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start CBR Send Time the 5-6-7-8-9 path</td>
<td>19.1617</td>
<td>12.0436</td>
<td>12.1234</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start CBR Send Time the 5-6-9 path</td>
<td>94.4466</td>
<td>N/A</td>
<td>97.9193</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. Reach Time (Sec)</td>
<td>0.0114</td>
<td>0.0158</td>
<td>0.0135</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table 6 Dynamic Scenario Experiment Statistic*

From Table 6 statistics analysis comparing GWF, AODV and DSR protocols, we can note the following:

- **Start CBR Send Time in the 0-1-2-3-4 path**: The start CBR send time in the 0 to 4 communication path is the time that each protocol uses to discover the route of the requested communication. The results show that a GWF Ad Hoc Bridge takes a very long time to finish this first route discovery as the time it uses is 2.5026 seconds. However, it takes less time than DSR to discover the new shortest route of 0-1-2-4, 0-1-4 and 0-4 while AODV does not discover the first 2 routes at all until it is actually possible to do a 0-5 communication. AODV is also the last protocol to discover the 0-4 rout which it has discovered at nearly the end of the simulation.
• **Start CBR Send Time in the 7 to 2 path**: The start CBR send times in the 7 to 2 path shows that all the protocols start at nearly the same time. It is almost the same time that we set the CBR to start the communication in the simulation scenario.

• **Start CBR Send Time in the 5 to 9 path**: In this communication path, the route 5-6-7-8-9 remains available until the end of the simulation. AODV only use this path to send its data. However, the data also show that a GWF Ad Hoc bridge is about 3 seconds faster than DSR to discover the new shortest path 5-6-9.

• **Average Reach Time**: The average reach time is the overall average of the time taken in our dynamic scenario simulation to send the CBR packets from the MAC agent of the sender to the MAC agent of the receiver node. The results show that a GWF Ad Hoc Bridge average reach time is 0.0114 seconds.

From our dynamic scenario simulations statistics we can conclude that in terms of the adaptation to the change in the communication path. AODV is the fastest protocol to start sending the data. However, it does not react well to the path changing. Our GWF Ad Hoc Bridge is the fastest to adapt to the path change but it takes longer time to start sending the first packet.

We extracted CBR records from the NS-2 simulation trace files of each protocol and studied the time it uses to reach the destination noted as End2End time. We present the analysis of the results in terms of End2End cumulative time in the graph of Figure 48.

![Figure 48 End2End Protocols Dynamic Scenario Comparison](image-url)
Figure 48 shows the comparison of the End2End cumulative time of our dynamic scenario simulation which is measured from the time the MAC agent of the sender starts sending a CBR packet until the packet is received by the destination node. From this graph we can note the following:

- From the percentage of the packets that have successfully reached the destination, the graph shows that in our simulation scenario a GWF Ad Hoc Bridge has overall taken less time than AODV and DSR for the packets to reach their destinations.

- A GWF Ad Hoc Bridge provides a significant increase of a better End2End cumulative time over AODV and DSR.

- The End2End cumulative time performance of AODV also includes the times of 13 packets that have not arrived to the destination in the route 0 to 4. These failed packets are the reason that AODV performance is worst than GWF and DSR.

The graph of Figure 49 presents the simulations throughput of each of the protocols in each communication path. This data is generated from the number of packets sent in each communication path which changes according to the node movement.
The results of the simulation shown in the graph of Figure 49 can be analysed as the following:

- **In general:**
  - The route search for the new path makes the start-sending time of our GWF Ad Hoc Bridge slower than the AODV and DSR algorithms. This results in a few packets less than the other algorithm in the simulations.
  - In AODV and DSR, nodes only maintain routes to active destinations, so a route search is needed for every new destination.

- **The topology changing path (0-4):**
  - In the 0-1-2-3-4 path, a GWF Ad Hoc Bridge has less throughput than AODV and DSR. This is the result of the route search.
  - In 0-1-2-4, a GWF Ad Hoc Bridge discovers the path faster than DSR. This is results in a better throughput. In the mean time, AODV does not discover this path at all.
  - In the 0-1-4 path which is offered by the simulation scenario in a very short time as a result of the movement of node 4, both DSR and a GWF Ad Hoc Bridge are only able to send 1 packet. This path is also not discovered by the AODV.
  - In the 0-4 path, the AODV only discover this path when it is nearly the end of the simulation time. A GWF Ad Hoc Bridge has however just one packet more throughput than DSR.
  - In addition we have also found that there are 14 packets of the 0-4 communication of AODV that only reach 0-1-2-3. This kind of event does not appear in either the simulation results of GWF or DSR.

- **The unchanged topology movement that only offer shortest path (5-9):**
  - The communication from 5 to 9 using the path of 5-6-7-8-9 is able to be used until the end of the simulation.
  - From the simulations we can see that AODV only use this path and does not do a route search for the shortest path if it is still able to send the data to the destination. Therefore, it is not surprising that AODV has the best throughput in this path.
  - For the 5-6-9 path, which becomes the shortest path after the movement of node 6, a GWF Ad Hoc Bridge performs better as it has the fastest route search. The results show that GWF throughput is better than DSR. Of course AODV has no packets in this path.
6.2 Comparison and Discussion

From the results and data generated from our simulations, we shall provide in-depth discussions of the three protocols compassion results as presented earlier in this chapter.

As presented in our design addressed in Chapter Four, a GWF Ad Hoc bridge is an implementation of multi-hop communications within the Link-Layer of an ad hoc network. It was designed to support the main concept of the Global Wireless Framework (GWF). A GWF main objective is to allow every ad hoc node in the network to have the same network address by allowing them to operate in one network domain. The concept of the design is making a significant advance in the processing model of an ad hoc forwarder node.

The different processing models of a Network-Layer multi-hop communication and a Link-Layer multi-hop communication concept are shown in Figure 50. As shown in the figure, the multi-hop communications performed by the Network-Layer of a forwarder node process the received packets up to the Network-Layer. In the mean time a GWF Ad Hoc Bridge protocol implemented in a Global Wireless Framework forwarder node resolves the next node within the Link-Layer.

Figure 50 GWF Communication Path
To study the difference in the characteristics of the processing algorithm of each of the protocols that we have used in our simulations, we study the logged events in the trace files that have been recorded from the simulations by the NS-2 Simulator trace agent. We can conclude some interesting characteristics of the compared communication protocols, these are:

- The data proves that the multi-hop communication of a GWF Ad Hoc Bridge is performed within the Link-Layer; this is in compliance with our design.
- The RTR agent used in the network layer of a GWF Ad Hoc Bridge is a DumbAgent that is defined by NS-2 to do no processing in the network layer apart from passing the packet to the upper layers and vice versa. In the sender node at the beginning of the process, this dumb process is shown by having no processing time difference. We can also note that the RTR agent of AODV and DSR also records no time. This can only be explained this that NS-2 does not record the fraction of time taken in the process, as can be proven later in the process where a forwarder node in AODV and DSR is actually processing the packet as the status of the action is changed from r to f.
- As mentioned, there is also no difference in the processing time of the RTR agent of the forwarder node.
- There are 34 events logged for GWF, 40 events for AODV and 41 for DSR in the studied 4 hops communication.
- In DSR, the last node passes the CBR packet to an RTR agent. While AODV does not behave the same, this makes DSR has one more processing event than AODV.
- Apart from the RTR processes in the forwarder nodes of AODV and DSR the rest of the communication processes are the same as of a GWF Ad Hoc Bridge.

<table>
<thead>
<tr>
<th>Multi-Hop Protocols</th>
<th>GWF Ad Hoc Bridge</th>
<th>AODV</th>
<th>DSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTR-RTS Time</td>
<td>0.0004</td>
<td>0.0004</td>
<td>0.0006</td>
</tr>
<tr>
<td>RTR-MAC Process Time</td>
<td>0.0010</td>
<td>0.0011</td>
<td>0.0013</td>
</tr>
<tr>
<td>CBR-Packet Size When send</td>
<td>576</td>
<td>584</td>
<td>612</td>
</tr>
<tr>
<td>CBR-Packet Size When received by MAC</td>
<td>512</td>
<td>532</td>
<td>560</td>
</tr>
<tr>
<td>CBR-Send Time</td>
<td>0.0046</td>
<td>0.0047</td>
<td>0.004784834</td>
</tr>
<tr>
<td>CTS-Packet Size</td>
<td>38</td>
<td>38</td>
<td>38</td>
</tr>
<tr>
<td>CTS-Send Time</td>
<td>0.0003</td>
<td>0.0003</td>
<td>0.0003</td>
</tr>
<tr>
<td>RTS-Packet Size</td>
<td>44</td>
<td>44</td>
<td>44</td>
</tr>
<tr>
<td>RTS – Send Time</td>
<td>0.0004</td>
<td>0.0004</td>
<td>0.0004</td>
</tr>
<tr>
<td>ACK-Packet Size</td>
<td>38</td>
<td>38</td>
<td>38</td>
</tr>
<tr>
<td>ACK – Send Time</td>
<td>0.0003</td>
<td>0.0003</td>
<td>0.000</td>
</tr>
</tbody>
</table>

*Table 7 Protocol Statistics Comparison*
The statistics of the simulations snapshot of the compared communication protocols are presented in Table 7, from the table we can note that:

- The RTR-RTS time is the time that a node uses to process the CBR packet. In the snapshot tables it is the time from the RTR send event until the RTS packet send-off time.

- The RTR-MAC processing time is the time that a sender node uses from receiving the CBR packet by the RTR agent until it sends it out by its MAC agent.

- The CBR packets in our simulation are the data packets. We can note that the packet size of a GWF Ad Hoc Bridge is the smallest. Also the size when received by a MAC agent is only 512 bytes which is only the size of data itself.

- The CTS, RTS and ACK packet size and processing time of every protocol is the same.

We can conclude that an advantage of a GWF Ad Hoc Bridge is its smaller size of overhead in the packets. This also improves the performance of the communication as the time used to send the packet is also reduced.

The route discovery process of a GWF Ad Hoc Bridge takes longer time (about 2.5 sec) for the first path. In our simulations the 0 to 4 communication is the first path that have to do the route discovery.

In our algorithm a GWF Ad Hoc Bridge node that wants to send data to a node whose route does not exist in its GWF-ARP table will broadcast a GWF-ARP packet to its neighbouring nodes. This is repeated until it reaches the MAXHOP value. This flooding of the network is a drawback to a GWF Ad Hoc Bridge and should be a main focus of improvement in the future work.

The simulation studies also show that a Global Wireless Framework implementation is in compliance with the main objective of the IPMSA where it allows all the ad hoc nodes in the network to use the same network address. This will be a solution for providing an integrated multi-hop communication for wireless devices working in one large network (same Network Address) as defined by our requirements.

Since a Global Wireless framework provides a one big network concept, to communicate with an Infrastructure network, a node in the Framework needs to use the Access Point (AP) services. The Access Point uses a Network Address Translation (NAT) functionality to replace the node address with an Internet-routable one if Internet access is needed. Therefore the node will register an access point as its gateway and still use its same address for the Global Wireless Network communications.

To communicate with a distant network (through the Internet), a node in a Global Wireless Network needs to use the gateway services (Access to the Internet Access Point).
This will operate in just the same way as a normal infrastructure mode. The node will register an access point as its gateway. The node does not have to change its address when connecting to the Infrastructure Network since the wireless interface of the access point that offered the service is also part of the Global Wireless Network. The access point will assign an Internet routable public IP for the GWF-node using its NAT function. It will also support the necessary parameters like DNS services and Proxy. Multi-hop communication is handled transparently in the Link-Layer by a GWF Ad Hoc Bridge protocol in the case of nodes not directly connected to a gateway.

After a GWF Ad Hoc Bridge connects to the access point and registers with its Home Agent. A request by a Correspondent Host (CH) at the Internet side to a GWF Ad Hoc Bridge node is handled the same way as in Mobile IP systems. As all the gateways also act as GWF Ad Hoc Bridge nodes and have a GWF-ARP table, which designates all the nodes in its communications range, they can note the requested node by sending an RARP to ask that node for its current IP address if the TIME field in the GWF-ARP table is expired. A Node will then automatically configures itself to be ready for communication with the CH who requested the connection. The Foreign Agent (FA) will perform the normal function after this point.

6.3 Summary

From the results presented in this chapter, we can conclude that a GWF Ad Hoc Bridge is able to support multi-hop communications within the Link-Layer. The simulation studies have shown that Global Wireless Framework can be deployed and used in real environments along with the existing protocols. The comparison of the performance of a GWF Ad Hoc Bridge against nodes implementing AODV or DSR algorithms shows acceptable performance. Nevertheless, in accordance with the proposed Integrated Personal Mobility Services Architecture a GWF Ad Hoc Bridge is the only one that complies with its requirements.

From the static scenario throughput results and End2End time simulations, we can conclude that a GWF Ad Hoc Bridge has the smallest data packet size which results in the fastest time to successfully pass data packets from one node to another. Accordingly, this reduces the time of communication and increases the throughput.

In terms of adaptation to the change in the communication path in the dynamic scenario simulations, we can conclude that in comparing all the start times of AODV, DSR and a GWF Ad Hoc Bridge, AODV is the fastest protocol to start sending the data; however it does not perform well with the path adaptations. A GWF Ad Hoc Bridge has provided the fastest path adaptation performance regardless of taking longer time to start sending the first packet from the first node of the multi-hop path.
Within our static and dynamic simulation scenarios we have successfully identified a number of key issues that need to be improved to allow for better performance of a GWF Ad Hoc Bridge, these are:

- The discovery time of the path before starting to send the first packet. From the simulation a GWF Ad Hoc Bridge takes 2.5 seconds which is about 30 times higher than DSR and 50 times higher than AODV.

- The processing time of the packets. A GWF Ad Hoc Bridge also take more time than AODV and DSR.

Using the proposed GWF Ad Hoc Bridge and the integration provided by the Global Wireless Framework will provide a seamless implementation of multi-hop communication within the Link Layer. This should result in a better range of services in an interoperable heterogeneous domain by offering services from both the network side and the local environment. It would also make it more flexible to achieve both kinds of services based on a node own self-centred network in the Information Space.
Chapter Seven: Conclusions and Future Work

This thesis has presented our work that illustrates a new perspective in exploring the world of Personal Mobility Services. The convergence of wireless technologies and the great expectation of the ubiquitous use of mobile networks in our mobile life style bring more needs of more choices of services available not solely from the network provider. An Integrated Personal Mobility Services Architecture aims at providing mobile users with new and more choice of services achieved by seamless integration of the services from both the ad hoc and infrastructure networks simultaneously, i.e. the environment.

In this chapter we finally come to the conclusion of our work, contribution and the future work discussions will be carried out. These will summarise the presented vision and the key parts of that vision which we implemented. Also included in this chapter is the early work proposed on how to utilise the advantages of the integrated services to enhance the nomadic user life style.

7.1 Thesis Summary

Chapter One of this thesis introduced the context and provision of the work, namely personal mobility and the availability of services in different Information Spaces. We identified the need of a system that supports the needs of nomadic users. The aims of our work are to deliver the freedom and variety of using and/or providing services in the mobile computing surroundings. From top to bottom analysis of our vision based on the existing mobile computing model and the mobile node requirements from the services point of view, we found that the services are greatly separated into two sides. These are services from the provider and services from neighbouring ad hoc nodes. The key challenge to our vision resides in the fact that users connected to the infrastructure are not able to get any other type of services apart from what is available through their own provider. This fact led us to the idea of bringing both types of services to the user seamlessly. Such a type of node that
can accomplish this ability will need to be able to seamlessly connect to both sides of the network simultaneously.

In **Chapter Two** we presented the background and related work. We started by examining the services provided, and then we discussed service discovery and how other research proposed how to do it. We gave details of well known active works such as Universal Plug and Play (UPnP), Services Location Protocol (SLP), JXTA and OGSA. This inspiring research work brings together the technology, requirements and ideas which moved us onto our proposed vision of Integrated Personal Mobility Services Architecture (IPMSA). This vision and its requirements necessitated the need to understand the wireless ad hoc network segment. We studied the widely used IEEE802.11 standard and the automatic address configuration of the multi-hop communication as presented in MANET and related works.

**Chapter Three** of this thesis presented the overall idea of how the architecture of the personal mobility will look in order to support the vision of Chapter 1. The Integrated Personal Mobility Services Architecture put together the pieces of protocols and algorithms to show the possibility of how we can integrate the services from both the ad hoc network and the infrastructure network. The two main parts of the proposed architecture consist of:

- Global Wireless Framework which allows users in an ad hoc network to seamlessly connect to those at the infrastructure network and vice versa.
- The second part of the architecture shows the novel prospective of the way forward of how to utilise the services available in the Information Space when the ad hoc and the infrastructure networks are joined together.

The provision of the architecture allows us to clarify the problems of achieving the provision of Integrated Services as the requirements the Global Wireless Framework have to implement.

**Chapter Four** gave the detail of the proposed protocols and algorithms designed to deliver the objectives of the proposed Integrated Personal Mobility Services Architecture. Yet, our research proved that the crucial part of the work giving the time limit allowed is to seamlessly join the two different networks being the ad hoc network and the Infrastructure network together. The design was carried out to ensure that we archive our main requirements being:

- To give the mobile devices the ability to communicate in both Ad-Hoc and Infrastructure Networks simultaneously.
- To propose a system that can provide the network and ensure that each mobile device is the centre of its own network which moves along with the mobile device.
- To provide a concept of how the services will be organised to efficiently use the architecture.
We put most of our efforts to implement the Global Wireless Framework and discuss in detail its design. At the end of the chapter we also show some design aspects of the services integration part.

Chapter Five presents the implementation. We choose to implement our protocols and algorithms in the network event simulator NS-2. In this chapter we showed exactly how and where we implemented the protocols and algorithms of the Global Wireless Framework within the simulator. The schematic diagrams and the important parts of the code are also provided to give more understanding of our implemented algorithms. This chapter also presented the early published prototypes of the distributed unstructured services and the dynamic services composition of the available services in the home network environment.

Chapter Six detailed the evaluation study and the results of the implemented framework. We start with the result of the Global Wireless Framework from the multi-hop communication model of a GWF Ad Hoc Bridge that we proposed. We compare our results against simulations of the existing ad hoc routing protocols like AODV and DSR to demonstrate / evaluate the proposed solution that shows how the network can overcome the requirements and the visions.

Finally, Chapter Seven, this chapter, presents the summary of the thesis, our contributions, the future works within the boundaries of our vision and the thesis final remarks.

To summaries our work as presented in this thesis, we can conclude that we have offered a more realistic vision for mobility services; they will not be exclusively Internet-based or ad hoc, but a mix of the two – IPMSA dissolves this distinction for the user where a user is in the centre of the network and surrounded services. The work presented also verifies the proposed architecture, its underlying support framework and its services integration which is a significantly achievement of our goals.

7.2 Contributions

Within our work we made some of the major contributions in the area of wireless network communication and mobile computing services. The lists of those contributions are classified into the following topics:

- Our first major contribution is to identify the requirements of the system that will provide services for Personal Mobility by analysing the needs of mobile users in Information Spaces [Mingkhwan et al. 2001].
- Next, we proposed a novel architecture that shows how to provide IP application services for mobile devices in a seamlessly integrated Infrastructure and Ad Hoc
Network environment, we call it an Integrated Personal Mobility Services Architecture (IPMSA) [Mingkhwan et al. 2002a].

- The proposed Integrated Personal Mobility Services Architecture allows clients in Personal Mobility environments to use two different classes of services, either Structured or Unstructured [Mingkhwan et al. 2002c]. This will allow flexibility to put any kind of services into our Information Space by means of open services architecture. The experimental environment scenarios have shown that it is more flexible to have both kinds of services in the Information Space.

- We proposed the Global Wireless Framework [Mingkhwan et al. 2002b] to provide a seamless integration between the infrastructure and ad hoc networks. This novel framework has a different perspective to MANET which uses routing protocols to extend the coverage area of mobile clients. In contrast, the Global Wireless Framework offers multi-hop communication within the Link-Layer which provides a transparent multi-hop communication to the Network Layer and enables all the nodes in the framework to have the same network address. By using our proposed Global Wireless Framework the mobile devices can achieve the following benefits:
  
  - They can use their initial auto-configured IP Address to connect to the Internet. Address auto configuration is achieved by adopting the existing work of zeroconf.
  
  - Mobile devices do not have to deal with the routing function. We can provide multi-hop communications by using our GWF Ad Hoc Bridge protocols [Mingkhwan et al. 2003a].
  
  - Nomadic end users can get services from both the Ad-Hoc and the Infrastructure Networks at the same time. The Internet integration is proposed in our work by using Global Wireless Framework with the Mobile IP scheme [Mingkhwan et al. 2003b].

- Our proposed Dynamic Services Composition framework [Mingkhwan et al. 2004] which is the latest work, makes three novel contributions: Functionality Utilisation, where complex appliances like TVs and HiFis can expose their resources for simultaneous use; Dynamic Composition, which provides the ability to use the services of better performing appliances when they are readily available; Virtual Appliances, that provides the ability to create an appliance that does not physically exist from the available services.

- The challenges of utilising the available services in the Information Space after the integration of the networks have been studied within our research group. We join the idea of providing Unstructured Services with semantic services representation to make better discovered and use of the services. Distributed Semantic Unstructured Services (DiSUS) was developed in collaboration with another researcher to provide
a demonstration of an application of IPMSA. In the work we have done with other research [Fergus et al. 2003a], within this work we developed an algorithm to achieve the semantic knowledge ability by using evolutionary programming techniques [Fergus et al. 2003b, 2003c].

![Diagram](image)

**Figure 51 Contributions of our IPMSA work**

In Figure 51 we map the contributions that we made to our architecture, the picture shows the parts of the IPMSA architecture that we published in international conferences. This should provide a clear picture of how the contributions are related.

### 7.3 Evaluation

The problems of mobile ad hoc networking and reachability to the Internet have been addressed by different groups of researchers. Zeroconf [Guttman 2001b] and Automatic Private IP Addressing (APIPA) [Cheshire et al. 2000] are dealing with automatic address configuration. MANET [MANET 2003] looks at multi-hop communication. And Mobile IP [Geiger et al. 1997; Perkins 1998b; Johnson et al. 2001b] and MIPMANET [Jonsson et al. 2000] are working on the Internet reachability. However, none of these proposals have addressed a complete solution for mobile users as described in the requirements of a Global Wireless Framework.

Apart from the forwarding of packets from one node to another that is performed by the access point [IEEE-SA 1999], the IEEE802.11 WLAN protocol specifications does not define
any support for multi-hop communications that is performed by the node itself in an ad hoc manner.

Our proposed Global Wireless Framework main objective was to implement a GWF Ad Hoc Bridge. It defines a multi-hop communication protocol and algorithms that works below the Network-Layer in order to join the entire wireless ad hoc nodes seamlessly in one network. This will therefore allow them to use the same network address domain. This aim of our GWF Ad Hoc Bridge is different from the existing work on ad hoc routing protocols that work in the Network-Layer and are concerned only with the multi-hop communication performances.

Our implementation of a GWF Ad Hoc Bridge as detailed in section 5.1 enables the Global Wireless Framework to handle multi-hop communications in the lower layers in the same manner as the implemented routing algorithms operate in the upper layers such as AODV [Perkins et al. 2002], DSR [Johnson et al. 2001a]. The advantage of our framework protocol algorithm is that all nodes in the Global Wireless Framework are able to use the same network address. The implementation also shows that our algorithms can be seamlessly implemented and work with other protocols already existing.

The simulation study results obtained using our developments in NS-2 (Discussed in Chapter Five) were given in Chapter Six. The results have shown that the Global Wireless Framework concept is valid. The simulations also show the comparisons within the following aspects:

- From the static scenario throughput results and End2End time simulations, we can conclude that a GWF Ad Hoc Bridge has the smallest data packet size which results in the fastest time to successfully pass data packet from one node to another. Accordingly, this reduces the time of communication and increases the throughput.

- In terms of adaptation to the change in the communication path in the dynamic scenario simulations, we can conclude that in comparing all the start times of AODV, DSR and a GWF Ad Hoc Bridge, AODV is the fastest protocol to start sending the data; however it does not perform well with the path adaptations. A GWF Ad Hoc Bridge has provided the fastest path adaptation performance regardless of taking longer time to start sending the first packet from the first node of the multi-hop path.

The advantage of our achievement in integrating Global Wireless Framework with the Internet is the elimination of the need for extra protocols like in MIPMANET to run on top of the ad hoc routing protocol like AODV. The Global Wireless Network is only one network-layer hop away from FA, which reduces network overhead compared to MIPMANET. The advantage of Global Wireless Framework over MIPMANET is that a system implementation
of Global Wireless Framework will have one network address for all the ad hoc nodes in the network. This enables a Global Wireless Framework node to easily identify the network destination whether it is within the Global Wireless Network or the Internet. In MIPMANET, a node has to check in its routing table.

In conclusion to the Global Wireless Framework simulation study, we can state that Global Wireless Framework can be deployed and used within the real environments along with the existing network technologies. In addition, it is more flexible to achieve both kinds of services available in the Information Space from the node point of view based on our Node-Centric network concept.

In our thesis we have also addressed the wide range of issues in mobile computing and services provision. Some of the previous research has focused on reliable connections [Kim et al. 1999], some, for example MAP [Duda et al. 1997], provides a middleware layer that works with a mobile host to collect results and mange the data transfer path. AOAC [Fleming et al. 2000; Helal et al. 2001], HP Cool Town project and JINI work on an automatic discovery of services and other devices from the physical surroundings using mobile equipments. Furthermore partitioning application services has been proposed as a method to enable personal mobility [Koskimies et al. 2000]. However, none of them proposed a solution to carry on services from an Ad Hoc network to an Infrastructure Networks and vice versa. In summary, the available research presents many very interesting and useful ideas; however, this work does not meet our requirements for an Integrated Personal Mobility Services Architecture, introduced in section 3.1, individually or collectively.

Furthermore, from the Services point of view each existing services discovery technology has its own model for key features such as device and service description, scoping of service advertisements, interaction with devices and notification of events. One of the reasons for this heterogeneity is that the technologies were constructed with different domains in mind, and in the spirit of experimentation. For example, Service Location Protocol (SLP) is a very scalable discovery protocol, intended to serve enterprise networks; UPnP targets home and small office computing environments; while HAVi was designed to enable interoperability in home AV networks and supports multimedia streaming and service reservation. The in-depth comparison of the services discovery technologies are discussed in [Bettstetter et al. 2000; McGrath 2000]. None of these technologies are aimed at the integration of services like the high street scenario we discussed before. Due to these different aims, we believe that our Personal Mobility Services Architecture that considers specifically the characteristics of integrating the services from both Infrastructure and mobile ad hoc networks is important especially for those Unstructured Services type, Implicit Function Services and Dynamic Services Composition [Mingkhwan et al. 2002a; Fergus et al. 2003a; Mingkhwan et al. 2004].

Our proposed services integration solution has the ability to extend a node’s reach outside an organisational LAN, and access services contained within an ad hoc network,
taking care of both Structured and Unstructured Services and also semantically matching 
queries with distributed services. Furthermore the DiSUS framework is protocol agnostic 
and we envisage that we will be capable of enabling interoperability between different 
protocols such as Bluetooth, WiFi and HomeRF [Abuelma'atti et al. 2002b; Mingkhwan et al. 
2003a]. Our understanding is that [Guttman 1999] and [Avancha et al. 2002] are incapable 
of achieving this level of functionality.

Finally for the properties of the service itself we have also shown the novelty of 
revolutionising and utilising the network appliances functionalities services [Mingkhwan et al. 
2004]. This takes a totally opposite direction from today’s industries that make devices 
sophisticated by adding more functions into them such as the new generation of mobile 
phone and home appliances.

Overall the framework offers a more realistic vision for mobility services; they will not be 
exclusively Internet-based or ad hoc, but a mix of the two – IPMSA dissolves this distinction 
for the user.

7.4 Future Work

To date, our current work results show that we can join the ad hoc and infrastructure 
networks together. This will bring together the services from both networks. As a result 
mobile users will have more choice of the services available. From a service providers’ point 
of view, a provider can vary from major companies that provide services through the 
Infrastructure network to one with a mobile device that provides an unstructured service(s) 
through the ad hoc network.

As shown in the services integration part of our work we already introduce some of the 
services that take advantage of these integrated services. In this section we discuss the 
works that are left to be carried out. We also discuss the problems, related technologies and 
research that can be blend to complete our vision.

7.4.1 Architecture Issues

During this thesis we have come across many interesting issues. The aim of the research 
was to investigate and propose a solution to integrate and utilise services from ad hoc and 
infrastructure networks. We have provided encouraging results and a prototype. However, 
some open questions remain interesting for future work. These include:

- **Ad-Hoc Parties Identification**: How to identify parties in an Ad-Hoc environment is 
  not easy since there is no fixed ID that can apply to this scheme so that a user can 
  initiate contact from any device available. This ID information is needed to identify 
  users to their communications partners and initiate the connection with their Home 
  Network.
- **Internet Integration:** The results of our simulation demonstrated that our proposed Global Wireless Framework algorithms are valid, which is the main objective of this work. However, there are still many aspects of experiments and simulation that need to be undertaken since our Global Wireless Framework is still far from perfect. The parts that would be very interesting to study in more detail include:
  
  - The hand off performances within the Link-Layer. This will be the important factor for the connection oriented services such as multimedia streaming.
  
  - The performance and effects of other communication protocols such as FTP, TCP and RTP operating within the system.
  
  - IP Hand-Over Problem: Handover of a connection from one access point to another is already taken care of by Layer 2 protocols. Nevertheless, the IP address that the access point lends to a client terminal also needs hand over from one access point to another. This problem can be solved using Home Agent in Mobile IP Scheme. However, the time taken by this procedure may cause trouble with real-time multimedia applications. Buffer techniques may have to be used to avoid the delay in this case. The other situation that has to be considered is that the IP can change when the user moves to a new environment that already has some devices using the same IP address. The system has to switch to the new IP without effect to the application running on top of it. The other parties that give services to this changing node also have to be notified about this change.

- **The Distributed Unstructured Services and Semantic Capability:** In our thesis we explained how this framework can support unstructured services in Mobile Ad hoc Networks. We defined the requirements which a framework needs to address and described the prototype we developed. Within our research into the Distributed Unstructured Services framework our primary goal was to implement a prototype that could ground our ideas and provide a framework on which to build. We believe we have achieved this and have successfully implemented a solution that enables us to semantically discover distributed Web Services in an ad hoc network. However, there are four particularly important areas that we have still not implemented within our framework, which is secure communication and authentication, continuous services provision, screening services to eliminate duplicate and unwanted services and copyright and royalty of the provided content. Within our early work of services integration especially the unstructured services, we acknowledge that we need to address a number of issues.
7.4.2 Dynamic Services Composition

In this thesis, we have shown that the availability of highly independent services offered by appliances enables us to map different types of services to form an Implicit Function. In our work we proposed a framework that dynamically incorporates appliances into a Service Integration Controller which is located in a Home Appliances Integration Unit (HAIU).

For the future work, we believe that the HAIU unit needs to incorporate “intelligence” that enables it to conceptually understand the meaning associated with a particular appliance. Furthermore appliances can be situated anywhere within the home environment and therefore the HAIU needs to consider location based semantics and posses real-world knowledge to ensure highly sophisticated service negotiation and composition. The HAIU has also to have the ability to abstract the complexity, associated with computer control, from the user.

In providing services of digital content, another research strand in our Networked Appliances Laboratory is dealing with payment of compensation for the distributor and the copyright owners of the content if and when they are used by another home user [Arora et al. 2003]. This is to ensure that resource providers are compensated for the services their appliances offer – this feature will be implemented in the HAIU.

7.4.3 Security & Privacy

We realised that security and privacy are one of the important key issues are needed to considered. In our implemented work we try to achieve a first goal of making the framework work as it is designed. Implementing or applying the security part will be the next important step.

For the better understanding of the security need for the architecture, we set about defining the problem so as to gain a set of requirements. These requirements were classified in to three parts:

- First, nodes have to be able to perform mutual authentication with the local and home networks to achieve location registration whilst maintaining location and identification privacy.

- Second, to be able to receive services from any provider within the Information Space without compromising privacy whilst also being able to maintain accountability for the services received.

- Finally, we need to be able to communicate with any host in the network in an anonymous way.

There are lots of research going on to tackle these problems which can be applied into our architecture and evaluated for the suitable situation. One of the researchers within our
research group is dealing with these security aspects in a mobile network privacy architecture (MNPA) [Askwith et al. 2000].

7.5 Final Remarks

Mobile computing is gradually being integrated into our daily life styles. To understand and implement such a system that makes the most of these new technologies we have to look at it as a whole. Architectures like the one described in this thesis addresses the aspect of many problems and proposes a solution that helps in achieving some of these goals by providing a framework for experimentation and the suggestion of how we can benefit from it. Also from these simulation studies one can see how the system will work and be able to evaluate it against other models in terms of performance, problem, requirements, and complexity. We clearly define in our architecture that the user interface application is able to associate with many different services that are provided within the Information Space.

The Integrated Personal Mobility Services Architecture (IPMSA) is a promising architecture for the support of mobile computing. This is achieved by the main components of the architecture divided into two main parts. Firstly, the Information Space is the joining of the Global Wireless and Infrastructure networks; nodes from both sides can communicate through the gateway. Secondly, the Services Integration, as shown in the architecture is divided into two layers, Services Available Space and Services Composition on top of the Information Space. Services Available Space representation means that services from the Information Space are equally treated and make no difference in the Services Composition layer in regard to user applications. The Services Composition Layer shows the ability of the user to create the task they need from the available services in the Information Space.

From the results presented in the Global Wireless Framework simulation study, we have show that the framework can be deployed and used within the real environments with the existing network protocols. Our proposed framework is more flexible to achieve both kinds of services available in the Information Space from the node point of view based on our Node-Centric network concept.

Finally, it is the author’s belief that the development of this architecture, it underlying support framework and its services integration, represent a significant early contribution in this important field of network and mobile computing.


References


