Computing Department - Lancaster University
United Kingdom

DISSERTATION

Submitted for the Degree of Master of Science (M.Sc.) in
Critical Software Engineering

Topic :

Location Awareness in a Mountain Rescue Domain

Panagiotis Georgopoulos
BSc (Hons) Computer Science and Telecommunications Engineer
p.georgopoulos@lancaster.ac.uk

Supervised by :
Dr. Chris Edwards

Date Submitted :
September 2007

Supporting documentation at :
http://www.lancs.ac.uk/postgrad/georgopp/mscproject/

Disclaimer

I certify that the material contained in this dissertation is my own work and does not contain significant portions of unreferenced or unacknowledged material. I also warrant that the above statement applies to the implementation of the project and all associated documentation.

In the case of electronically submitted work, I also consent to this work being stored electronically and copied for assessment purposes, including the department's use of plagiarism detection systems in order to check the integrity of assessed work.

Date:______________________

Signed:_____________________

Dedicated to…

*my **parents**,*
who always support
each of my decisions

*to **myself**,*
for persistently trying to
become a better person

*to **anyone,***
constantly fighting
for better Education

Abstract

The notion of location awareness in a Mountain Rescue domain is critical for the mission coordinator of a Mountain Rescue Team who tries to organize the team and make informed decisions for all its members. The knowledge of location of each member of the team while they are on a mission, could be provided by sending GPS coordinates from a device that each rescue worker would carry, to the server of the team located at its headquarters. The physical characteristics of the Mountain Rescue domain along with the unpredictable movement of the rescue workers during a mission prevent the deployment of a fixed network infrastructure to facilitate the transmission of the GPS coordinates and therefore alternative communication options should be defined and utilized. As a result, this project had to define a communication framework with all the viable connectivity options that seem to apply to this domain. Furthermore, an application for the device that the rescue worker will carry that would be able to transmit GPS coordinates by utilizing the defined communication framework was also required. Finally, a server application that would be able to listen for the GPS coordinates sent from the clients was required as well. The focus of this project was on the application developed for the client and its capability to identify the availability of each connectivity option and utilize the best suited one, based on their prioritization defined in the communication framework. The theoretical and practical evaluation of the developed prototype system proves that the outcome of this project satisfied the described proof of concept and successfully met all its requirements.

Acknowledgments

Firstly, I would like to deeply thank my supervisor ***Dr. Chris Edwards*** *for his guidance, support and great interest throughout this dissertation.*

Secondly, I would like to thank ***Dr. Martin Dunmore****, member of the Network Mobility Group of Lancaster University, for his ideas and time spent for this dissertation.*

Thirdly, I would like to thank my close friends, ***Serko, Sebastian and Angelos*** *for their support and ideas when suddenly things were becoming harder.*

Last, but certainly not least, I would like to thank my ***"big family"*** *back home and here at Lancaster, for their support and self-giving help during the whole year.*

Table Of Contents :

List of Figures :

List of Tables :

Chapter 1 : Introduction

This chapter introduces the topic of location awareness (Section 1.1) and describes the motivation of this dissertation (Section 1.2) regarding mobile ad-hoc networks in a Mountain Rescue Domain. In addition, it analyzes the focus of this project and pinpoints the primary aims of it (Section 1.3). Finally the structure of this dissertation is going to be presented (Section 1.4).

1.1 Introduction

In 1943 Thomas Watson (IBM's Chairman) stated that "there is a world market for maybe five computers" [1] but fortunately history proved him wrong. Also, Ken Olsen's (President of Digital Equipment Corporation) motto in 1977 that "there is no reason anyone would want a computer in their home" [2] was proved wrong by the billions of individuals who purchase a computer for their home. Nowadays, computers and technology are truly ubiquitous, being part of our every day life without even realizing it.

The rapid evolution of personal computers has forced engineers and researchers to develop ways to interconnect them making networks to facilitate the needs of every person. Huge technology breakthroughs, like Internet, wireless networks or GSM have not only helped individuals to facilitate their needs but also let them have a variety of options for doing so. All these options have been developed to satisfy the requirements of different problem domains depending on the actual need each time.

The issue of location awareness has been raised and confronted in multiple ways during the last decades. At present, the most well known and precise system for location awareness outdoors is the Global Position System (GPS). This system uses the only fully functional Global Navigation Satellite System (GNSS) to calculate the location of a specific building, vehicle or person equipped with a GPS receiver [3]. Although in the past the GPS system was used more in military services, at present it can be used successfully by any civilian in real life scenarios such as finding the best route to the closest hospital.

The issue of location awareness can undoubtedly be helpful for teams that perform critical operations like a Mountain Rescue team. For example, during a rescue mission in a Mountain domain each mountain rescue worker might carry a GPS enabled device to inform him for his exact location and help him orient himself in the rescue domain. Moreover, if a mission officer could know the exact location of each member of the team, he could remotely organize the mission of his team more accurately and effectively. Furthermore, just imagine the time that could have been gained if a wounded hiker has a GPS enabled device, which nowadays are even embedded into mobile phones [7], and directly inform the mountain rescue team for his exact location.

This project will focus on the issue of location awareness in a Mountain Rescue domain presenting a solution to facilitate the needs of a Mountain Rescue team. In particular, a fundamental requirement of such a team is the knowledge of the exact location of each member of the team in order to increase the possibilities of a successful mission. One way to satisfy this requirement is to give to each mountain rescue worker a small and lightweight device that will be able to obtain GPS coordinates from satellites and send these coordinates back to the headquarters of the team while its members are on a mission.

1.2 Motivation on the problem domain

The members of the Network Mobility Research Group at Lancaster University [4] are examining the network mobility concept in conjunction with the requirements of the Cockermouth Mountain Rescue Team [5]. The developed mobile network, referred onward as Mountain Rescue Network, identifies a complex mobile model that tries to provide data networking capabilities to mountain rescue workers [6]. In the case of an emergency incident, the Mountain Rescue Team is divided into independent search parties that are composed of a cluster of rescue workers and all-terrain rescue vehicles [4]. These parties are scattered around the valley areas of the Lake District (Northwest UK) where an incident has occurred and try to spot and aid people or animals. In order to improve the efficiency and accuracy of the work of the members of these search parties, the final goal of this project would be to equip them with a small and lightweight device which should be able to transmit each member's GPS coordinates to a centralised server located in the Mountain Rescue Team's headquarters.

The Mountain Rescue Network reveals the concept of network mobility, a relatively new concept that is being researched and developed during the last few years. This concept includes the support of whole networks (with leaf nodes) that are moving in an unpredictable fashion. This network mobility becomes apparent in various domains but has a significant importance in critical operations such as a mountain rescue mission. The concepts that need to be confronted in a mountain rescue domain are not only the constant and unpredictable mobility of the search parties of a Mountain Rescue team but also the mountain rescue domain itself.

The scale and the heterogeneity of the physical environment that a Mountain Rescue team might search in does not allow the development of a fixed wired network infrastructure to provide network services to the rescue workers of such a team. This characteristic leads to the development of a wireless network which should have the ability not only to cover a geographically large area but to provide network services to facilitate the unpredictable movements of each rescue worker in that area. This endeavor becomes more difficult considering that the case of a rescue worker moving in a coulee or an inlier is not rare and therefore a wireless network with fixed access points is insufficient. Generally, the notion of constant movement in a large and heterogeneous area introduces the concept of network mobility that could be facilitated either by a mobile wireless network or by a satellite network. By envisaging -in a simple way- that each search party of a mountain rescue team is a mobile network with the rescue workers being the end-nodes of this network, it can be realized that in a mountain rescue mission there are a lot of mobile nodes roaming among mobile networks in a large heterogeneous geographical area.

Providing network services using multiple communication technologies to such an environment is a fascinating research topic with multiple parameters. In addition the basic scheme, that is using different wireless methods to achieve network connectivity, can be applied in a number of different domains. U-2010[1], RUNES[2] and 6NET[3] are just a few of the numerous significant research projects that are focused on providing network services in mobile ad-hoc networks using cutting edge technologies and protocols such as Mobile IPv6.

1.3 Focus and aims of the project

The Mountain Rescue Network presents the ideal scenario for researching the network mobility concept with an interesting emphasis on the location awareness of each end/leaf node. This project has as a focus the mobile ad-hoc network of the Cockermouth Mountain Rescue Team and **will define and deploy a framework to transmit GPS coordinates from a device that a rescue worker will carry back to the team's headquarter**.

A basic and important starting point for this project is to identify ways to inform the administrative personnel that are located in the headquarters about the exact location of each rescue worker and its team. There are two major factors that truly influence the communication between the rescue workers and their headquarters ; the heterogeneity of the domain (mountain) that these rescue parties search in, and the undeterministic fashion that each search party and each rescue worker move. This project should examine viable connectivity options that could be used in the region that the Cockermouth Mountain Rescue team operates in and develop an application that would have the ability to seamlessly utilize them for the transmission of the GPS coordinates. The main connectivity options that seem to apply to this domain for transferring GPS coordinates are using IPv6 over a WIFI network, IPv4 over GPRS and finally, using SMS messages over the GSM network.

The system that will be developed for this project will be based on the usual 2-tier server-client communication model. A server will be located at the headquarters of the mountain rescue team and should be able to obtain real-time[4] location information that is sent from the clients regardless the connectivity option that they used. A client can be considered as the small device that each rescue worker will carry and which should be able to obtain GPS coordinates from satellites and send them to the server

[1] U-2010 "stands for" a research project regarding Ubiquitous IP-Centric Government & Next Generation Networks [8]
[2] RUNES "stands for" a research project regarding Reconfigurable Ubiquitous Networked Embedded Systems [9]
[3] 6NET "stands for" a research project regarding a Large-Scale International IPv6 Pilot Network [10]
[4] The word real-time in a mountain rescue domain does not refer to strict real-time constrains that are used in critical systems (as in hard real-time critical systems) but to the notion of identifying the location of each rescue worker in a relatively "live" manner (as in soft real-time systems).

via the best available communication option. Therefore, a basic aim for this project is that the application that would be developed for the device that the rescue worker will carry, should be able to identify and evaluate the best suited available connectivity options[5] based on some specific criteria and utilize it for the transmission of the coordinates. As a proof of concept, a PDA was used as the device that the rescue worker will carry during a rescue mission.

At this point it is essential to mention that although the term location awareness implies the use of precise GPS coordinates, this project will not deal with the precision of the coordinates that the GPS receivers obtain from the satellites. The quandary of how precise the GPS coordinates are has been researched and analyzed in detail in [3, 12, 25, 30] and will be mentioned in brief in Chapter 2. For the scope of this project, high precision GPS coordinates can be obtained by just purchasing more suitable devices with high precision GPS receivers.

1.4 Report Structure

The remainder of this report is structured as follows. Chapter 2 describes the background research domains that had to be examined regarding this dissertation. Chapter 3 describes the overall aim of this project along with the high and low level requirements set for it. Chapter 4 describes the design of the system and the reasoning for all the decisions taken. Chapter 5 describes the implementation of the defined system with all the hardware and software limitations that were confronted. Chapter 6 evaluates both in theory and practice all the parts of devised system. Finally, Section 7 suggests future improvements that could be done on the developed prototype system and makes an overall evaluation of this MSc project.

[5] *As becomes apparent the device should be able to identify the available connectivity options based on a specific subset of connectivity options that this project will identify as viable for the scope of sending GPS coordinates over a particular Mountain Rescue domain (discussed thoroughly in Section 4.2)*

Chapter 2 : Background Research

This chapter introduces the most important background research topics which were investigated during the development of this project and could be skipped if the reader is familiar with them. Section 2.1 presents the purpose of the Cockermouth Mountain Rescue Team and the networking communication model that was developed for its requirements. Section 2.2 presents the network mobility concept that becomes apparent on the mountain domain. Section 2.3 presents in brief IPv6 and its benefits and Section 2.4 presents Mobile IPv6. Finally, Section 2.5 presents the way the Global Positioning System operates.

2.1 Cockermouth Mountain Rescue Team

The Network Mobility group of Lancaster University is in collaboration with the Cockermouth Mountain Rescue Team to identify the requirements of the team and investigate ways to provide network services to such an interesting domain. Therefore, section 2.1.1 introduces the purpose of this Mountain Rescue team and section 2.1.2 presents the networking model that is enisaged to facilitate the needs of this team.

2.1.1 Purpose of the team

The Cockermouth Mountain Rescue Team (Figure 1) is composed of about 40 unpaid volunteers, three all terrain vehicles and five trained search dogs and their mission is to respond to emergency incidents around the valley areas of Lake District (Northwest UK). Their headquarters is located at Cockermouth and they mainly operate from Buttermere, Ennerdale, Lorton and Loweswater valley areas of the northwest UK (Figure 2) on a 24/7 basis throughout the whole year [5] .

Figure 1 : The Cockermouth Mountain Rescue Team (adapted from [5])

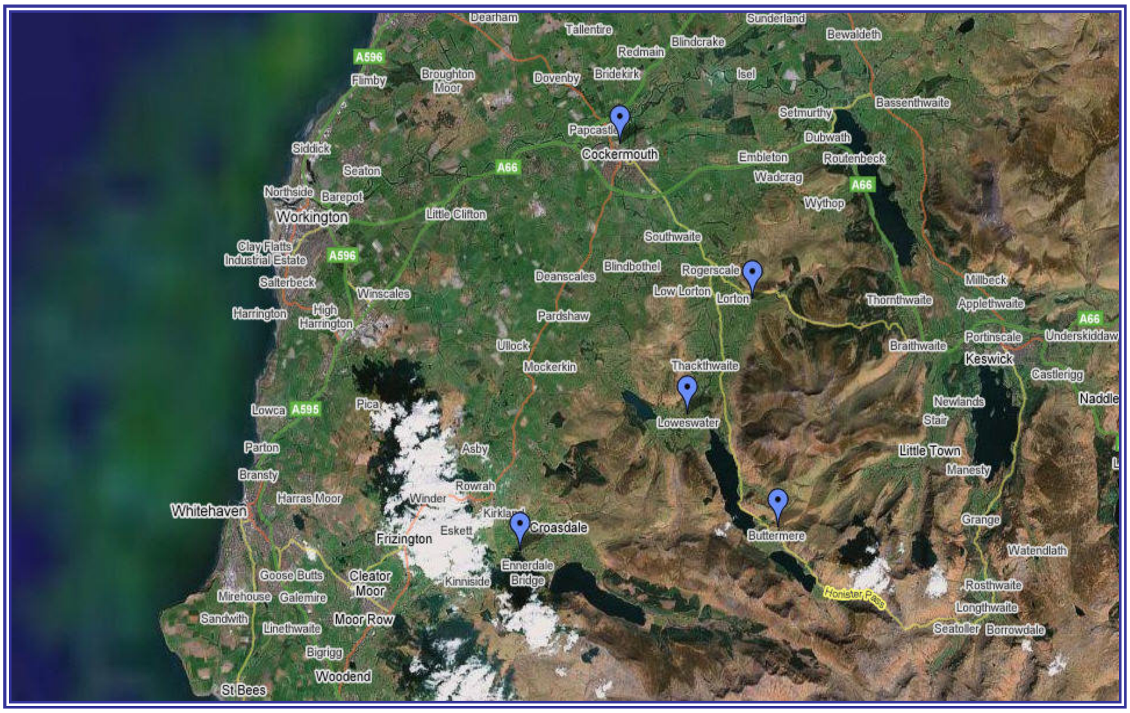


Figure 2 : The valley areas around Lake District that the Cockermouth Mountain Rescue Team operates in. The tagged areas (apart from the Cockermouth which is the team's base) are areas with the more frequent incidents.

The Mountain rescue team's responsibility is to rescue people or animals that are wounded, trapped or lost on the numerous paths in the previously mentioned region. In the case of an emergent incident usually the victim calls 999, reports its situation and the police informs the Mountain Rescue team's leader. As soon as the leader has enough information about the incident and as the team operates mainly in an ad-hoc basis, he pages -if necessary- sufficient members of the rescue team that are called to report to the headquarters. When enough members to man an all terrain vehicle have arrived at the headquarters, they immediately set off the mission by driving as close to the incident as possible without waiting for more members to arrive. If the area of the incident is not approachable by the all terrain vehicles then the mission officer requests the aid of the helicopter from the Royal Airforce.

Usually the personnel that man an all terrain vehicle composes a search party which moves independently to cover a geographical area according to the incident. As becomes apparent in such a mission there might be a few different search parties covering an area which might merge or even split into more search parties according to the requirements of the mission. The constant cooperation and organizational commands for these search parties need the awareness of the location of each member of the mountain rescue team so that the mission's coordinator can make more informed decisions. Therefore, the Network Mobility Group of Lancaster University has defined and developed an IP based network communication model that suits the nature of the mountain rescue domain and can provide powerful communication solutions while the search parties roam within the region of an incident.

2.1.2 Networking communication model

As described the personnel of the mountain rescue team is split into different search parties that generally, especially at the beginning of a mission, move independently to cover a geographical area according to the incident. These search parties could be envisaged as mobile networks and the rescue workers of each party as the leaf nodes of the network. The underlying idea for their communication is that they want to rely on a network that is specifically designed for their needs and would be financially and practically feasible to run. Therefore, the Network Mobility group has designed a wireless based (mainly 802.11 and 802.16) networking model to facilitate the requirements of this team in an ad-hoc basis (Figure 3).

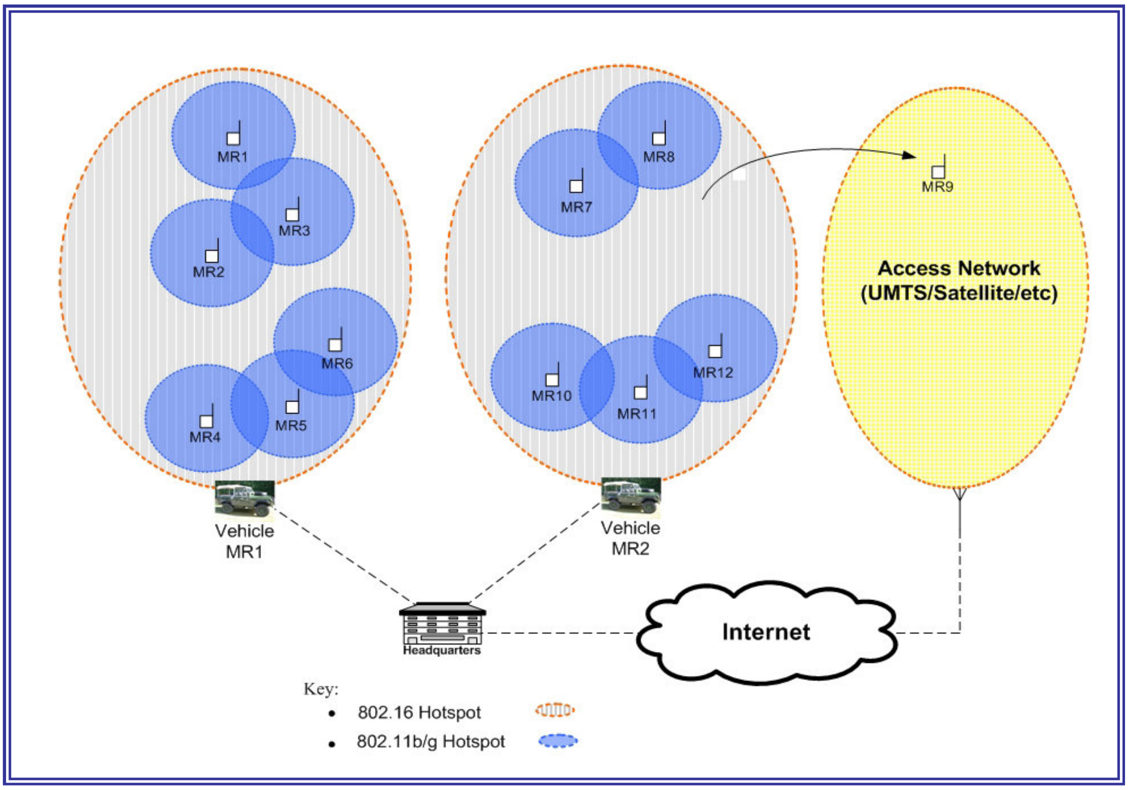


Figure 3 : Model of the Mountain Rescue Network's infrastructure (adapted from [13])

According to this model, the members of a search party should carry a simple WIFI enabled device (such as a PDA), and communication with each other is provided by a short range wireless hotspot (blue circle, Figure 3). The wireless coverage of that hotspot is projected in the area where a search party roams from a mobile router (MR, Figure 3) that a special member of the party is carrying in a rucksack [6]. The communication among these devices usually takes place using the 802.11b/g WLAN standard. The connectivity of these mobile networks is supported by a directed long coverage 802.16 wireless hotspot (gray circle, Figure 3) that is projected in the area from the all terrain vehicle that is parked closest to the incident [6].

As becomes apparent the physical characteristics of a mountain in conjunction with the constant and unpredictable movement of the rescue workers introduces many difficulties in the wireless communication in such a domain. Therefore, the WIFI based solution that was described above cannot guarantee the transmission of the data from the device that a rescue worker holds, back to the headquarters. As the right side of Figure 3 illustrates the envisaged networking model presents a vague cloud (yellow circle) to represent all the alternative communication options that could be used when the WIFI coverage is not sufficient either because a rescue worker is far away from a wireless hotspot that is projected in the area or because a physical impediment exists.

The focus of this dissertation is to develop an application that could primarily utilize the WIFI based infrastructure that is provided from the above communication model to transfer GPS coordinates from the device that a rescue worker carries back to the headquarters of the team. Moreover, the developed application should fall back to alternative communication options that could be used in such a domain when the WIFI coverage is not sufficient in order to offer redundancy in such a mobile environment. Chapter 4 (System Design) will discuss thoroughly the complementary options that the author of this dissertation suggests and the exact reasoning for their prioritization and usage. Moreover, it will discuss the hardware and software that will suit to the previously described scope.

2.2 IPV6

IPv6 is a network layer protocol which was finally defined by S. Deering and R. Hinden in 1998 [20] in RFC2460 [22]. IPv6, or IPng (IP Next Generation), is the successor of IPv4 which is the dominant Internet Protocol facilitating the communication over the Internet. IPv4 was designed in 1981 with the initial intention to provide unique global addressing so that computers over the Internet can uniquely address one another [18]. As the years were passing by, the number of users demanding connectivity increased vastly leading to approaching the depletion of the available IPv4 addresses. This was the initial motivation for designing IPv6.

This section describes IPv6 and the general need for its deployment to current networks. The remainder of this section is organized as follows. Section 2.2.1 identifies the reasons why networks do need IPv6, Section 2.2.2 mentions in brief the key characteristics/benefits of it, Section 2.2.3 discusses about IPv6 addressing and finally Section 2.2.4 elaborates on Mobile IPv6 that applies to our domain.

2.2.1 Why is IPv6 needed?

During the last two decades the Internet has experienced an exponential growth and more users globally need Internet connectivity. In addition, many new devices such as mobile phones and PDAs (not to mention devices used in IP telephony), need Internet access and therefore demanding an IPv4 address. As IPv4 was not initially designed to cope with this tremendous number of devices that require Internet connectivity (or network connectivity in general) its address range is about to be depleted. Various techniques have been used to hold off the depletion of IPv4 addresses such as NAT[6], DHCP[7] or CIDR[8] but each of them introduces new problems [19].

Although the previously mentioned techniques can serve the server-client model efficiently, they fail to effectively satisfy network and user mobility, as well as the new requirements that are formed from new cutting-edge applications demanding more bandwidth [21]. Existing intermediate network equipment need a significant amount of time to perform IP translation and fail to respond to the requirements of more bandwidth [19]. Moreover, the non-hierarchical allocation of the IPv4 addresses expands the size of the routing tables on the routing devices which need more time to identify the routing path for each packet. Furthermore, DHCP and NAT require to keep states of the IP addresses and the ports of the clients and thus not only they break the integrity check of the packets which are now being encapsulated and decapsulated multiple times but also introduce complexity on the management and the performance of the edge devices [19].

As becomes apparent the new requirements that are formed from newly developed cutting-edge applications in conjunction with the numerous new devices that require Internet connectivity (mobile phones, PDAs, etc.) cannot be confronted by adding features to the current IPv4 implementation. Although IPv6 was initially designed to confront the IPv4 address depletion problem it includes various useful features to facilitate more devices that require network connectivity, using either a wired or a wireless network infrastructure, along with the new developed applications.

2.2.2 Key Characteristics / Benefits

IPv6 is a revised version of IPv4 and is designed to meet the requirements of the huge Internet expansion not only by introducing a larger address space but also by including numerous interesting features for the development of new applications. In brief, IPv6 introduces the following new key characteristics [19, 20, 21, 22] :

- **Larger address space :** IPv6 addresses are composed of 128bits as opposed to the 32bits of an IPv4 address offering a huge address range so that any device could have a global IP address. Studies show that there are 1000 IPv6 addresses available for every living person on the planet [19, 21]. By being able to give to any network device a global IP address, IPv6 eliminates the need for NAT and provides end-to-end reachability and scalability. In addition, this broad range of IPv6 addresses are organized in a hierarchical manner improving the routing efficiency and simplifying the network management [19].

Simplified header format : IPv6 working groups have revised the structure of the IPv4 header format for improving packet handling on both the intermediate network devices and the end hosts. This revision includes the elimination of six out of twelve IPv4 header fields, the modification of the name and type of existing header fields and

[6] *NAT stands for Network Address Translation [38]*
[7] *DHCP stands for Dynamic Host Configuration Protocol [39]*
[8] *CIDR stands for Classless Inter-Domain Routing [40]*

the addition of new fields for introducing new features [19]. For example, IPv6 includes a Traffic Class header field which is equivalent to the IPv4 Type of Service header field, but also includes a Flow Label header field. This new field enables IPv6 routers to identify the flow of each packet from the IP layer significantly improving the performance of the network compared with the IPv4 case of having to dig to the Transport Layer of the TCP/IP stack. Moreover, IPv6 can include at the end of the header field a daisy chain of optional extensions that allows the addition of new features. This chain does not decrease the performance of the intermediate devices because there is no demand for all the routers to be able to process all the additional header fields [19]. Generally, the IPv6 header format is simpler and more efficient as it has a fixed length and less number of fields than the IPv4 header [19, 20].

- **Embedded security** : The IPv6 suite has been designed with mandatory IPSec implementation that can be enabled in every node potentially aiding the security of the whole network [19]. IPSec is a suite of protocols for securing IP communication, basically operating in the Internet Layer of the TCP/IP stack (or the network layer of the OSI model) and was initially designed to be optionally used in IPv4 [18, 23, 24].

- **Improved multiple hosts reachability** : IPv6 has eliminated the use of broadcast packets which were downgrading the network performance by causing a CPU interrupt to every node disregarding if a packet was of its interest [19]. However, services that require broadcast packets (such as ARP) can now be facilitated by IPv6 multicast which has been designed to increase network performance and stream packets efficiently [19, 20, 34].

- **Built in mobility** : One of the headers that can be included into the chain of optional extension header fields is the mobility header which is used from nodes that implement Mobile IPv6 [6, 19]. The mobility header includes information that is used by mobile nodes, correspondent nodes and home agents to facilitate the communication among them guaranteeing session continuity while a mobile node roams from one network to another [14, 16]. IPv6 mobility will be discussed thoroughly in Section 2.2.4.

The development of IPv6 has introduced a lot of innovative characteristics and many current services such as ICMP, DHCP and DNS have also been redesigned based on the advantages of IPv6. These services include new functions in order to take advantage of the new features that IPv6 includes such as stateless autoconfiguration, neighbor discovery (NDP) and IPv6 multicasting[19]. IPv6 services are not only more efficiently in the terms of network performance but can also be more secure by using IPSec and grant authentication and encryption to every packet [19, 20]. Concluding, IPv6 is being carefully designed for more than a decade and can introduce a great benefit to current networks.

2.2.3 IPv6 Addressing

As stated an IPv6 address is composed of 128 bits which are usually represented in eight four-digit parts in hexadecimal format separated by semi-columns (Table 1, a) [36]. These addresses are not case-sensitive and can be compressed by following two simple principles. Firstly, a field composed of zeros can be represented by just one zero (Table 1, a to b) and secondly, successive fields of zeros can be represented by double colon (Table 1, b to c) [19, 34]. IPv6 addresses can also be written with a prefix length notation (as the CIDR notation in IPv4) (Table 1, d) basically to characterize a group of addresses allocated for a network [19].

TABLE 1 : IPv6 Addressing Examples	
a	2001:0000:a215:0000:00000:c5ff:feaf:d42b
b	2001:0:a215:0:0:c5ff:feaf:d42b
c	2001:0:a215::c5ff:feaf:d42b
d	2001:0:a215:495a / 64

Generally IPv6 interfaces in network devices may have more than one IPv6 address for different scope and of different type. There are three major types of IPv6 addresses [19, 20, 34, 35, 36] :

a) A ***Unicast*** address identifies a single interface of a single network device and any packet destined to a unicast IP address is routed only to that single network device [36].
b) A ***Multicast*** address identifies a group of interfaces usually belonging to different nodes. A packet sent to a multicast address will be replicated along the path from the sender to the receiver so that it can be delivered to all the nodes which have subscribed to a specific multicast address [19]. IPv6 multicast implementation successfully eliminates the need for broadcasting IP packets to all the nodes of a subnet, improving the overall performance of the network [34].
c) An ***Anycast*** address identifies a set of interfaces which typically belong to different nodes. Any packet destined to an anycast address will be routed to the closest interface with that address [20]. This proximity can be defined by the routing protocols used and can be valuable when requesting a specific service [19]. For example, when a node makes a DNS query to an anycast address its packet will be routed to the closest DNS server which can reply to that specific request [35].

RFC3587 [36] defines unicast addresses as being additionally split into the following four categories with different scope:

a) **Global** unicast addresses are the equivalent of IPv4 public addresses and thus their scope is the whole IPv6 Internet, meaning that they can be reached by any network device [35].
b) **Link-local** unicast addresses are the equivalent of IPv4 APIPA[9] addresses (169.254.0.0/16) and thus are used by nodes communicating with nodes on the same link with no intermediate routers. A link-local unicast address is

[9] *APIPA stands for Automatic Private IP addresses and are used from devices running Microsoft Windows operating systems in order to acquire an IP address in a network without a DHCP service.*

automatically given to any device as part of the "address autoconfiguration" feature of IPv6 and always start with the prefix "fe80:: / 64" [35].

c) **Site-local** unicast addresses are the equivalent of IPv4 private addresses (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16) and their scope is a site which is either an organizational network or a portion of one with a defined geographical location [35]. Site-local addresses are not automatically configured and must be assigned to devices either through stateless or stateful address configuration processes [34, 35, 36]. Formally, site-local unicast addresses have been deprecated by RFC 3879 [37] but are unofficially used from existing IPv6 implementations.
d) **Unique-local** unicast addresses have been presented to address the problem that was identified in a corporate network when two nodes of different sites were communicating using the same site-local unicast address. These addresses are still considered to be private addresses within an organization, but unique across it [36, 37].

Finally, IPv6 uses the unspecified address "::" to identify that a node does not have an IP and the loopback address "::1" to enable a node to send packets to itself.

2.2.4 Mobile IPv6

Mobile IPv6 is derived from Mobile IPv4 but takes advantages of a number of the features of IPv6. Therefore, and contrary to Mobile IPv4, Mobile IPv6 can be used in an ad-hoc fashion by any node that uses IPv6 and "needs mobility". As described in Section 2.2.2, IPv6 built-in mobility feature can by effectively used by adding specific extension header fields to the IPv6 packets that will be used to facilitate the mobility of nodes. These extension header fields not only manage to seamlessly support the mobility of a node but they also introduce benefits in the routing mechanism. For example, the Destination Options header field aids the efficiency of the routing mechanism by avoiding triangle routing [19].

In mobile IPv6 when a mobile host moves from its home network to any other subnet it "acquires" a new IP address called the care-of address (c/o IP) [19]. When packets are destined to its home address, that is the IP address that is assigned to its home network valid on the Internet, they are seamlessly forwarded from an entity called a home agent to its care-of address, which is the temporary IP address that the mobile node currently has on a "foreign" network [41].

Mobile IPv6 overcomes the need for the intermediate entity (that is the foreign agent), by making the binding of the home address of a mobile node with its care-of address with the use of special binding messages. Thus, when a mobile node roams to another network it will send a binding update message to its home agent informing him about his new care-of address and will wait for a binding acknowledgment message from his home agent to confirm this update (Figure 4). Consequently, every packet destined to the mobile node will be directed to its care-of address which is retained with the use of specifically designed IPv6 destination header options, such as a Binding Request Option and a Home Address Option, and the use of a binding cache which stores previously acquired care-of addresses [41].

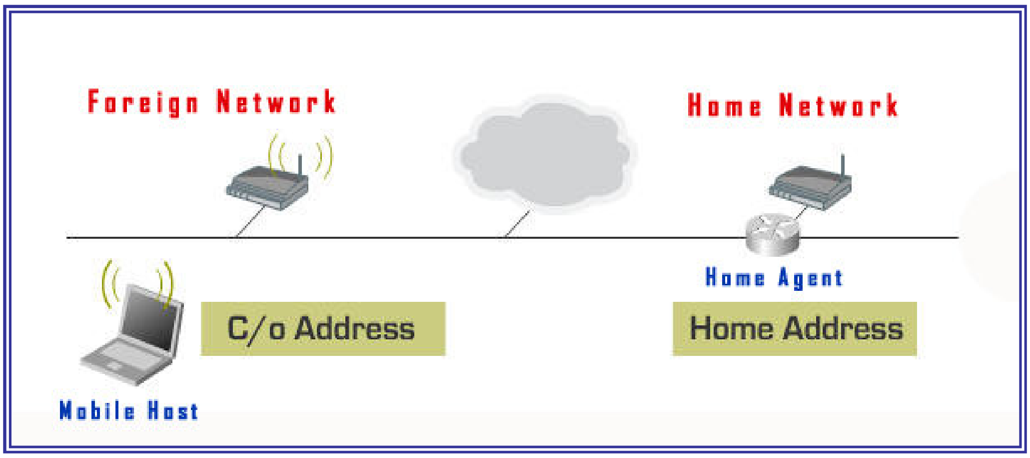


Figure 4 : A mobile Host roaming to a foreign network supported by Mobile IPv6 (modified from [19])

Generally, Mobile IPv6 benefits from inherent features of IPv6 such as autoconfiguration, the neighbor discovery protocol and authentication and encryption mechanisms. Autoconfiguration in IPv6 includes both stateful and stateless configuration mechanisms and, as becomes apparent, the stateless autoconfiguration feature is of great value in a mobile environment. Thus, all hosts and routers that use the stateless autoconfiguration feature will generate their own link-local IPv6 address for all their interfaces by appending to the 64-bit local link prefix (fe80::/64) their link layer address (EUI-64). This procedure is significant as it eliminates the manual configuration of hosts, and the configuration of routers becomes minimal [19]. In conjunction with the autoconfiguration mechanism the neighbor discovery feature enables hosts (either mobile or "fixed") to find neighboring routers and keep track of the neighbors that they can reach. Moreover a host can use locally available information and information advertised by routers to automatically configure its default router, the default hop limit setting, the timers used in network discovery process and many other options which have significant importance in a mobile environment as they simplify the communication [19, 41, 42].

To sum up, Mobile IPv6 allows a mobile host not only to transparently maintain connections while moving from one subnet to another, but also to be reached with the same IP address while moving to different locations. Moreover, mobile IPv6 optimizes routing compared with Mobile IPv4 because it eliminates the need for an intermediate foreign agent between the home agent and the mobile host, and removes the triangular routing through the home agent by instantiating a bi-directional tunnel between the correspondent node and the mobile host.

2.3 Network Mobility

During the last few years, the need for also having mobile networks in conjunction with mobile hosts became apparent and thus the concept of network mobility gained attention. The Mountain Rescue domain includes the concept of network mobility, along with the concept of hosts' mobility, and thus this section will describe it in brief. Therefore, Section 2.3.1 introduces the Network Mobility concept in theory and Section 2.3.2 discusses the efforts of the Network Mobility group at Lancaster University to apply network mobility "techniques" to the Mountain Rescue Team.

At this point it should be mentioned that the focus of this project is neither to investigate the routing mechanisms in such a scenario nor to find ways to confront their inefficiencies. The author of this project intents to focus on the IP layer and above, by designing and implementing an application that will be able to utilize the apparent network routing mechanism on the WIFI mobile network and swap to alternative solutions when the WIFI coverage is not sufficient. However, this section pinpoints that future work could be done to give to the developed application the ability to identify inefficiencies due to the network mobility concept, and swap to alternative communication options when needed.

2.3.1 Network Mobility in theory

The network mobility concept that is apparent to the mountain rescue domain is a research topic that is in the process of being examined thoroughly by the research community in an effort to identify efficient ways to confront its new requirements. This need was captured by the Internet Engineering Task Force (IETF) which has formed the Network Mobility Charter (NEMO) to research the concept of network mobility as opposed to the individual hosts mobility [15].

This charter focused its interest in facilitating the physical mobility of entire networks, such as networks in public transportation or personal area networks [6], and therefore they designed the NEMO Basic Support protocol [16]. According to the NEMO Basic Support protocol a bi-directional tunnel is instantiated between a mobile network's Mobile Router (MR) and the Home-Agent (HA) of the mobile router that facilitates all the traffic destined to a node of the mobile network with a corresponding node [6]. This is the equivalent of the mobile node roaming that was described in Section 2.2.4.

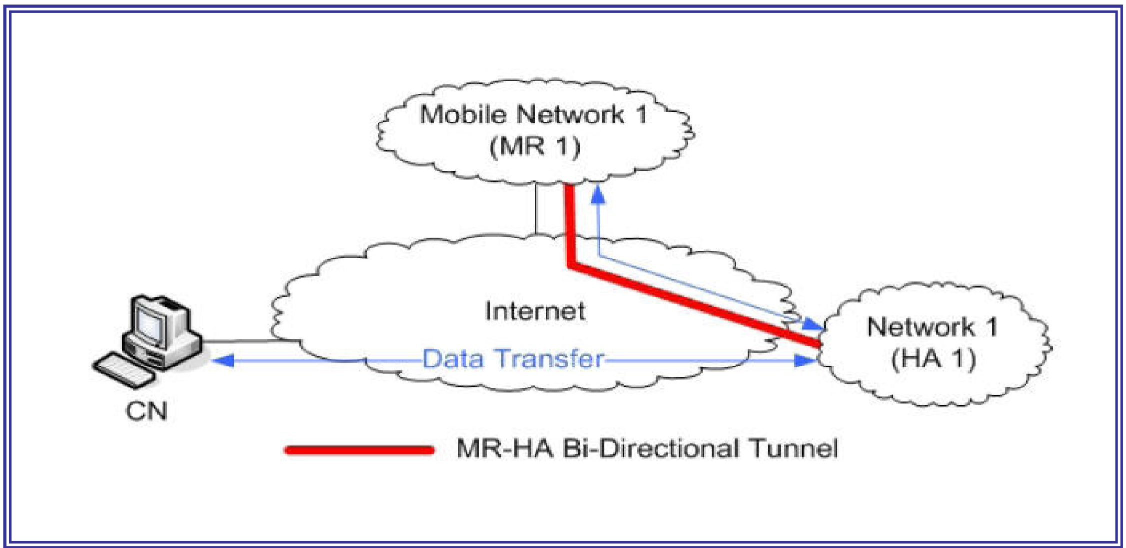


Figure 5 : Communication imposed by the NEMO basic support protocol (adapted from [6])

For example, as Figure 5 illustrates when a corresponding node (CN) wants to send data to a leaf node of a mobile network (e.g Mobile Network 1, Figure 5) while it is roaming, it can do so by sending the packets to the home agent of the mobile network that belongs to another network (Network 1, Figure 5). This home agent has established a bi-directional tunnel to the mobile router (MR1, Figure 5) of the mobile network 1 that roams and therefore all the traffic destined to a leaf node of the mobile network will be transferred through this bi-directional tunnel. In addition, every packet sent from a leaf node to any other node will also pass through this bi-directional tunnel and be routed to the suitable node.

The NEMO Basic Support protocol presents a basic milestone in the concept of network mobility and guarantees the continuous communication of the nodes of a single mobile network that is roaming across different access networks [5]. Along with this session continuity, the NEMO Basic Support protocol takes advantage of IPv6 guaranteeing that every node is reachable with a unique IP address and provides communication in a way that the mobility of the network is transparent to the nodes inside it [4].

2.3.2 Network Mobility in practice

The network mobility group of Lancaster University has applied the NEMO basic protocol to facilitate the communication of the search parties that roam during their mountain rescue missions [17]. This effort has presented a number of essential research points which pinpoint that although NEMO provides basic communication coverage for a mobile network it introduces a significant overhead on the communication model that needs to be efficiently confronted [17].

A key characteristic of a mobile network in a mountain rescue domain is that although the whole network is moving and its communication should be effectively supported, its nodes are in fact relatively static in respect with one another. This characteristic in conjunction with the physical mobility of the network introduces a variety of scenarios which define new requirements that cannot be confronted with a single “mobility solution” such as the NEMO basic protocol [14]. For example, as Figure 6 illustrates, a network may move out of the coverage of an 802.16 hotspot leading either to a total lack of connectivity for the nodes of the network or the enforcement of another type of connectivity, such as the nested case of the MR1’s and MR2’s networks (Figure 6). Moreover, as a mobile network moves, a node may have the chance to establish multiple connections to send packets either for load balancing or for redundancy purposes forming a case of a Multihomed NEMO [14]. Therefore, other techniques, such as the MIPv6-style Route Optimization mechanism or MANET centric solutions should be used to counteract inefficiencies that become apparent when a network roams in such a domain and complement the NEMO basic protocol [6, 17].

However, although some of these additional techniques manage to maintain the connectivity of a mobile network they introduce a lot of overhead in terms of bandwidth and CPU cycles in the intermediate network devices. For example, every time a packet is transferred along an extra mobile router-home agent bi-directional tunnel it should be encapsulated adding 40 bytes (the size of an IPv6 header) to the original packet header. Moreover, this addition increases the general packet size and reduces the maximum transfer unit that is usable from the application, increasing also the chances for fragmentation [6]. Furthermore, the encapsulation and decapsulation procedures introduce increased delay in the delivery of the packets, inefficiency in terms of bandwidth and an increment in the CPU cycles in the tunnel end-points.

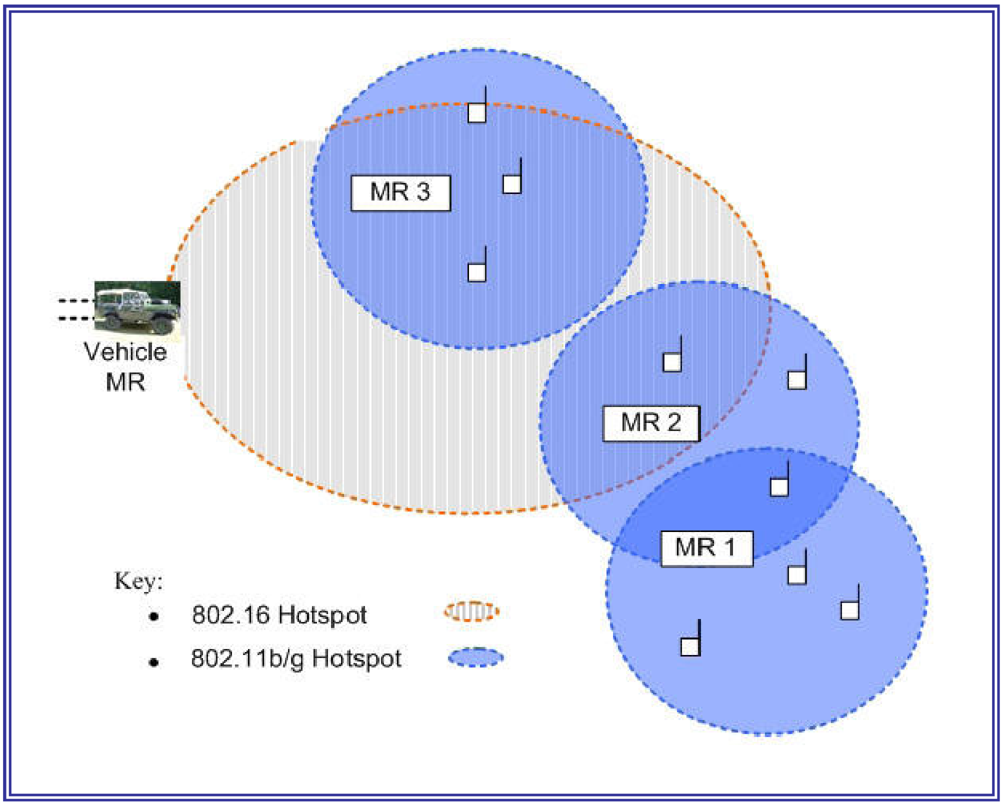


Figure 6 : Example of a Mountain Rescue Network's infrastructure (adapted from [6])

To summarize, the above arguments illustrate that the network mobility concept on a mountain rescue domain can be initially confronted with the NEMO Basic Support Protocol, but there are numerous issues that become apparent in the terms of the efficient routing of the packets being sent. Moreover the NEMO basic support protocol with any additional feature/protocol, can be applied if there is sufficient wireless coverage to support the device that a rescue worker carries during a mission. These issues identify the need for an application which should be able to recognize when a mobile node is out of the wireless hotspot's coverage and use alternative mechanisms to transmit packets, in our case packets containing GPS coordinates, to the server that is located at the headquarters of the Mountain Rescue Team. Moreover, future work could be done to enable the application to identify and counteract the inefficiencies and overhead that are presented in the routing protocols if the network is unable to confront them with inherent mechanisms.

2.4 GPS

The Global Positioning System is mainly composed of a satellites' constellation that provides suitable information to GPS receivers to calculate their exact position on earth. As GPS is going to be used for the location awareness of the mountain rescue workers this section discusses GPS. In particular, Section 2.4.1 describes the structure of the Global Positioning System, Section 2.4.2 explains how it works and finally Section 2.4.3 identifies causes that introduce errors in the calculation of the coordinates and some ways to improve accuracy.

2.4.1 System Structure

The Global Positioning System (GPS) is a navigational system primarily composed of a network of satellites placed in orbit around earth that enables a GPS receiver to determine its location speed and direction [25, 26]. The U.S. Department of Defense initially designed GPS for military purposes and launched the first GPS satellite in 1978, but two years later it was decided to be given free for civilian use without subscription fees or setup charges [25]. Up to now (September 2007), GPS is the only fully functional Global Navigation Satellite System (GNSS), managed by the United States Air Force 50th Space Wing, and "demanding" for maintenance, research and development 750 million dollars per year [26].

Essentially GPS is composed of three major segments; the **Space Segment** (SS), a **Control Segment** (CS) and a **User Segment** (US) [25, 26]. The Space Segment consists of 24 GPS satellites distributed equally around the earth (Figure 7) in an approximate altitude of 12.600 miles. These satellites are traveling at speeds of 7.000 miles an hour, which allows them to pass over the same location of earth twice a day. Satellites are distributed in a fashion where a GPS receiver can theoretically "see" at least six of them from almost everywhere on earth's surface [26]. The uniform distribution of the 24 satellites' constellation was changed in 1994 by adding more satellites to increase the redundancy of the system. As of April 2007 there are 30 actively broadcasting satellites that are used to transmit signal information to GPS receivers.

The **Control Segment** of GPS is mainly formed from all the agencies that are responsible for the maintenance of the satellite network and the overall system. Major monitoring stations are placed around the earth (Hawaii, Kwajalein, Ascension Island, Diego Garcia, Colorado Springs [26]) tracking the orbit of the satellites and contacting them regularly with navigational updates. These updates synchronize the atomic clocks on every satellite in a microsecond precision and adjust their orbital model with emphemeris information (described in the next Section) by taking into account various inputs such as weather condition and predefined orbit [29, 26].

The **User Segment** of the GPS is, apparently, composed of the GPS receivers equipped with three logically distinct parts; an antenna tuned to the signal frequencies that the satellites use to transmit information, a small-processor and a highly stable clock to maintain the time. Nowadays, these parts can be found integrated in a GPS module (such as the SIRF Star III chipset, Figure 8) which can be used as inner parts of non-navigational devices such as PDAs and mobile phones [26].

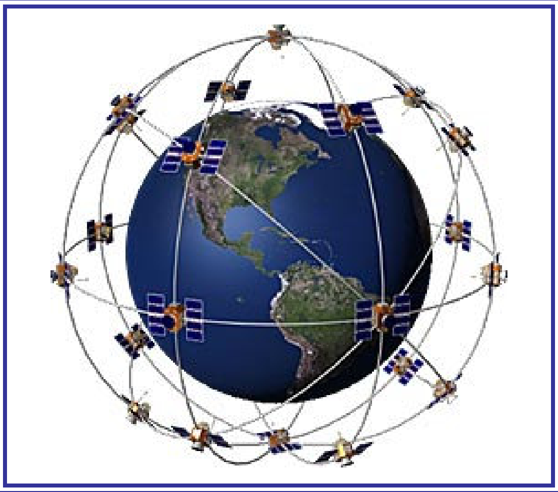


Figure 7 : GPS Satellite Constellation (adapted from [25])

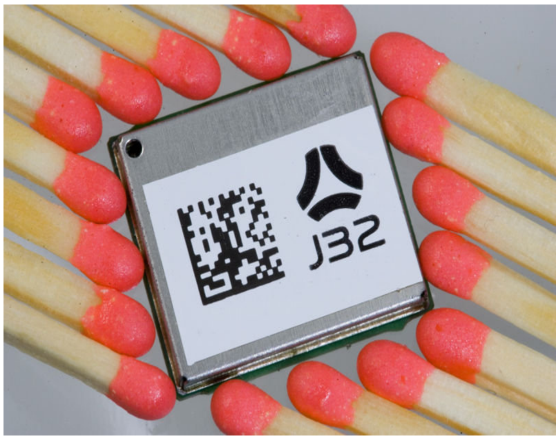


Figure 8 : GPS integrated module (adapted from [26])

A significant characteristic of a GPS receiver is the number of channels that it can simultaneously control, eventually representing the number of satellites that it can concurrently monitor. Recently developed GPS receivers can monitor from 12 to 24 satellites simultaneously increasing the redundancy of the data that they provide. Another characteristic of a GPS receiver is its ability to display the retrieved data to the user on a screen, or whether it can transmit them, regularly using the NMEA protocol, to another device using either a serial, a USB or a Bluetooth connection [26].

2.4.2 How does it work?

As described the GPS core system comprises a satellite constellation which is responsible for providing suitable information to GPS receivers located on earth. To explain simply the procedure a GPS receiver follows to calculate its location based on the information retrieved from the satellites, let us consider the following example.

Let us envisaged that an individual is lost in the area around Birmingham (UK) and a policeman informs him that he is 200 miles away from London (in a straight line). Although this would not phenomenally help him it would though inform him that his location is on the perimeter of a circle centered at London (red circle, Figure 9). By asking another person he also manages to acquire the information that he is 100 miles away from Manchester, information that enables him to identify two points on the map where the circles drawn, based on the information retrieved, intersect (two red balloons, Figure 10). If he is totally unaware of the region that he is walking in, he just needs one more information about his location that will enable him to define which of the two previously identified points is correct. Consequently, by acquiring the information that he is within a 50 miles radius from Worchester he is able to identify the point with his location (red balloon, Figure 10).

Figure 9 : A 200miles radius circle centered at London (red circle)

Figure 10 : A 200miles radius circle centered at London (red circle) intercecting with 100miles radius circle centered at Manchester (blue circle)

Figure 11 : Identify a unique location point based on the 2D equivalent procedure of the GPS.

The equivalent of the two dimensional example given above to the three dimensional space is the procedure GPS receivers use to calculate their location. The difference is that each circle drawn on the 2D map is represented by a 3D sphere having as a center a satellite that orbits around the earth. Moreover, in this real world case, two intersecting spheres mark a region with possible "location" points, three intersecting spheres mark two points and a fourth sphere identify a unique location spot. Thus, the basic principle is that a GPS receiver can calculate its location by locking the signal from four satellites, but generally, it can calculate its position by using signals from only three satellites that determine two possible location points and by opting out the point that is not located on the earth's surface.

The mathematical procedure described above is called trilateration and determines the relative position of an object based on its exact distance from others whose location is already known [31]. For our 2D example, our passenger determined his exact position by learning the exact location of other towns. In the 3D example a GPS receiver will try to determine its location based on the known location of the satellites that are within its site. The two remaining queries are how a GPS receiver can identify its exact distance from each satellite and how it is aware of the exact location of each satellite each time.

A GPS receiver monitors information that is transmitted from satellites that are within its sight. Satellites transmit data that contain three important pieces of information; **time**, **almanac** and **ephemeris**. Each satellite has a high precision atomic clock that is updated from ground units every day and a GPS receiver updates its clock with this precise timing information that it retrieves from satellites [30]. Almanac contains information that describes the orbit that a specific satellite is supposed to follow and finally, ephemeris contains the slight modifications that a satellite had to commit in order to follow the predescribed orbit [30]. Therefore, when a GPS receiver obtains information from a specific satellite, it is aware of the exact time, the exact location and the future orbit of a satellite.

The only remaining required data for the GPS receiver is to be able to identify its precise distance from satellites with known locations so that it can use trilateration and find out its own location. This can be calculated following the basic principle that "a distance from a known location A to a point B equals the velocity of the transmitted signal from point A to point B multiplied by the time it takes for the signal to reach one point from the other" (Figure 12) [30]. The signals that are sent from satellites or from GPS receivers are wave signals that travel at the speed of light[10]. Therefore, a GPS receiver can calculate the distance between its location and a satellite by subtracting the time a signal was sent from the satellite, from the time the signal was received. This subtraction is possible only because the satellite and the GPS receivers have already synchronized their clocks from previously transmitted signals.

[10] *The speed of light is approximately 3×10^8 m/s or 1foot/nsec*

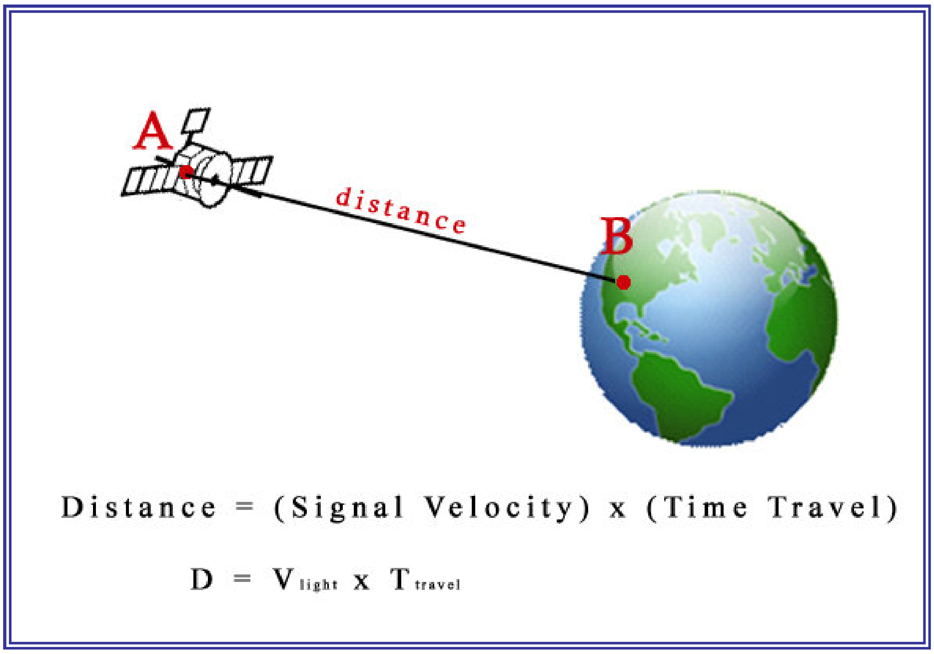


Figure 12 : Calculating the distance between a point with a known location (point A) and a point with unknown location (point B) by sending wave signals.

To sum up, a GPS receiver initially receives "establishing" information from a satellite that includes time, almanac and ephemeris information and stores this data in a local memory so that it can be aware of the exact location of a satellite that it monitors [26, 30]. Afterwards, it calculates its distance from satellites within its site and uses trilateration to identify its location. A GPS receiver is able to mathematically define its location by monitoring three satellites, but every additional satellite provides redundancy and accuracy to the data provided.

2.4.3 How accurate it is?

As defined a GPS receiver is able to calculate its location based on the current time, the position/orbit of the satellite and the measured delay of the received signal [26]. Each of these essential parameters may include an error which might lead to a location that might be inaccurate. The most important causes of error in the previously mentioned parameters are [25, 26, 30, 32] :

- **Atmospheric effects :** The velocity of the signals that travel through the atmosphere is affected from weather conditions such as humidity, but most importantly their velocity is reduced as they pass through the Ionosphere and Troposphere. The delay that this deceleration imposes can be confronted by a GPS built-in model that calculates an average amount of delay to correct the error. Due to these atmospheric effects an inaccuracy of ± 6meters can be introduced [26].

- **Multipath effects :** As GPS signals travel through the atmosphere they can be reflected off objects such as tall buildings, hard ground or canyons and arrive at a GPS receiver with delay. Various techniques have been developed to confront the errors imposed by multipath effects (such as the narrow correlator spacing [26]) and they can effectively recognize and discard signals with reflection delays (especially the ones that suffer with long delay reflections). Another technique that was developed during the last decade is the DGPS which uses ground stations at specific known locations that also pick up the reflected signals, as usual receivers do, but they broadcast corrections for the delay that these signal carry [33].

- **GPS receiver clock errors :** The GPS receivers are equipped with a small and less accurate clock compared to the atomic clock that the satellites have and although they obtain timing updates from satellites, they can still introduce very slight timing errors [30].

- **Orbital/ephemeris errors :** As described, satellites send navigational updates with ephemeris information to GPS receivers so that they can be aware of the satellites' locations and orbits. However, ephemeris information are sent out from a satellite every 12.5 minutes [26] which might introduce an error in the calculation of the location of a GPS receiver, especially in the case of a satellite being boosted to a proper orbit and a GPS receiver having not received this update.

- **Geometric Dilution of Precision** : When satellites are positioned at wide angles relative to each other then the geometry presented is ideal for a GPS receiver to use trilateration and calculate its location [30]. The additional launch of satellites may "place them" in a relative line or a tight grouping regarding the receiver's point of view and this might impose slight erroneous calculations of the location. Moreover, if a GPS receiver does not have accurate information of a satellite's orbit (due to orbital/ephemeris error described above) then the presented geometry is not sufficient for the GPS receiver to do trilateration because the satellites' spheres might not intersect at all.

- **Selective Availability** : The U.S. military department has initially imposed an intentional degradation of the signals sent by the satellites in order to block hostile military units to obtain accurate GPS coordinates and launch missiles against U.S. troops [26]. These signals were carrying the suitable information to correct the error imposed, but it was encrypted and could only be decrypted by the U.S. military or other authorized units. In 2000 the error presented by Selective Availability was set to zero, basically because the Federal Aviation Administration Department of the U.S. Government did not want to continue spending millions of dollars every year to preserve and maintain the system to eliminate the error from selective availability [26, 30].

Concluding, nowadays GPS receivers can provide accurate location information, when used outdoors, with a ± 15meters precision. Higher precision GPS receivers can be acquired by anyone who is willing to spend a couple of hundred pounds.

Chapter 3 : Requirements Analysis

This chapter presents the overall aim of this dissertation and analyzes this aim into high and low level requirements for the overall system.

3.1 Overall aim of the project

As presented in Chapter 1, this project will focus on the issue of location awareness in a Mountain Rescue domain. In particular, the author of this project would like to develop an application with the following overall aim :

- ***Equip a mobile device with the capability of sending GPS coordinates to a server via the best suited wireless communication option among the ones that can be applied in the Mountain Rescue Domain.***

In an effort to make the above primary aim clearer and although the system architecture will be discussed thoroughly in Chapter 4, it should be mentioned that the primary intention of this project is to develop an application for the mobile device that the rescue workers will carry during their missions. This application should be able to send GPS coordinates to the server that is located at the headquarters of the team. The author of this dissertation will try to achieve the primary aim mentioned above according to the requirements of the Cockermouth Mountain Rescue Team (Section 2.1), its mission and the specific region that this team operates in.

3.2 High level requirements

Providing network services in a Mountain Rescue mission is a complex task primarily affected from the mobility of the end leaves of the mobile network (rescue workers), the mobility of the mobile network itself (search party) and the physical characteristics of the domain (mountain). Therefore, in order to achieve the described overall aim of the project, the following high level requirements have to be achieved :

a) ***Identify, assess and prioritize communication options*** *that could be used in a mountain rescue domain for transmitting GPS data, referred onwards as communication framework.*

b) ***Implement a client application for a mobile device*** *that can recognize which of the defined communication options are available and efficiently utilize them for transmitting GPS coordinates according to the defined criteria.*

c) ***Implement a simple server application** that can receive the GPS coordinates that are sent from the clients, regardless the communication option that a client used.*

3.3 Low level requirements

The high-level requirements defined in the previous section regard three different aspects of this dissertation; identifying a framework regarding the communication options used, developing an application for the mobile device that the rescue worker will carry and finally, developing the application for the server located at the team's headquarters.

The first (a) high-level requirement described in the previous Section, demands a theoretical and practical study of the available connectivity options that can be used in the mountain domain that the Cockermouth Mountain Rescue team operates in. This study should evaluate the connectivity options that could theoretically be used regarding the specific scope of sending the GPS coordinates, namely a fixed length string. This evaluation should also prioritize the theoretically viable connectivity options on the mountain rescue domain in order to provide the grounding for the application that will be developed on the mobile device. The underlying principle is to provide a basic connectivity framework with the options used and the reasoning for them, that could be used to aid the development of other networking services in such a domain.

The second (b) high-level requirement regards the application that will be developed for the mobile device that the rescue worker will carry. This high-level requirement defined in the previous Section can be split into more detailed requirements that this project should satisfy regarding the developed client's side application :

1) *It should be able to **acquire real-time GPS coordinates.***

2) *It should be able to **identify which of the predefined connectivity options are available**.*

3) *It should be able to **utilize the best suited connectivity option** and **seamlessly swap among them** based on some criteria.*

4) *It should be able to **support IPv4 and IPv6**.*

5) *It should be able to **transmit GPS coordinates in defined intervals**.*

Additional secondary requirements for the client's application regarding the rescue worker's point of view would be that the chosen mobile device should be small and lightweight in order to be carried easily. Moreover, the developed application for that device should be easily used without any configuration from the rescue worker. Furthermore, the application should inform the rescue worker of its exact GPS coordinates, the wireless communication used at a time and the number of packets/messages sent using each communication method.

The third (c) high-level requirement described in the previous Section regards the application that will be developed for the server located at the headquarters of the team at Cockermouth. This high-level requirement can be split into the following more detailed requirements that this project should satisfy regarding the developed server's side application :

1) *It should be able to* ***listen for GPS coordinates regardless the connectivity option*** *that the client used to sent them*

2) *It should be able to* ***identify the exact node*** *that sent a packet and the* ***exact connectivity option*** *used*

3) *It should be able to* ***identify packets received in the wrong order****.*

The main secondary requirement for the application developed for the server side regarding the mission coordinator's point of view would be that the application should be able to inform him about the information retrieved. In particular, the server's application should present at least the GPS coordinates received, the ID of the node that sent them, the communication method that was used and the time that these were sent. Moreover, the server application should be able to count and present the content of packets received from different communication methods and present these data to the coordinator.

Finally, it is critical to clarify that the focus of this project is on identifying and using the available communication option to send GPS coordinates from a mobile device, acting as a client, to a server. Keeping in mind the heterogeneity of the physical domain (mountain) and the unpredictable movement of the mountain rescue workers it becomes of great importance to develop an application that could be flexible to identify, seamlessly swap and utilize the available connectivity options according to the needs.

Chapter 4 : System Design

This chapter presents the architecture of the system discussing all the design decisions that were taken regarding all the components of the system. Section 4.1 presents an overview of the system architecture and Section 4.2 discusses and evaluates the connectivity options that better suit the envisaged system architecture on the specific mountain domain. Section 4.3 presents the "client component" commenting on both hardware and software decisions and finally, Section 4.4 present the "server component", again, commenting on both hardware and software decisions.

4.1 System Architecture Overview

As described in the previous section the main aim of this project is to try to transmit GPS coordinates from a mobile device that a rescue worker carries, to the server located at the team's headquarters via a number of potential connectivity options. Figure 13 illustrates a high level system architecture overview of the envisaged system to enable us to set the grounding for this chapter.

As Figure 13 depicts the system architecture is composed of three conceptually distinct parts; the **mountain rescue domain**, that is the region that an incident can occur, the **headquarters** located at Cockermouth (Figure 2) and all the **connectivity options** that can interconnect these two parts. In the mountain rescue domain there are a lot of "rescue worker components", each one of them representing in general a rescue worker with all the appropriate equipment needed in a mission regarding this project. The connectivity options part includes all the connectivity components that represent every possible connectivity option that can be used to transfer the GPS coordinates. Finally, the server component represents the rescue team's server with all the appropriate equipment to be able to listen for the GPS coordinates that are sent using the different connectivity components and informing the team's coordinator of the received messages.

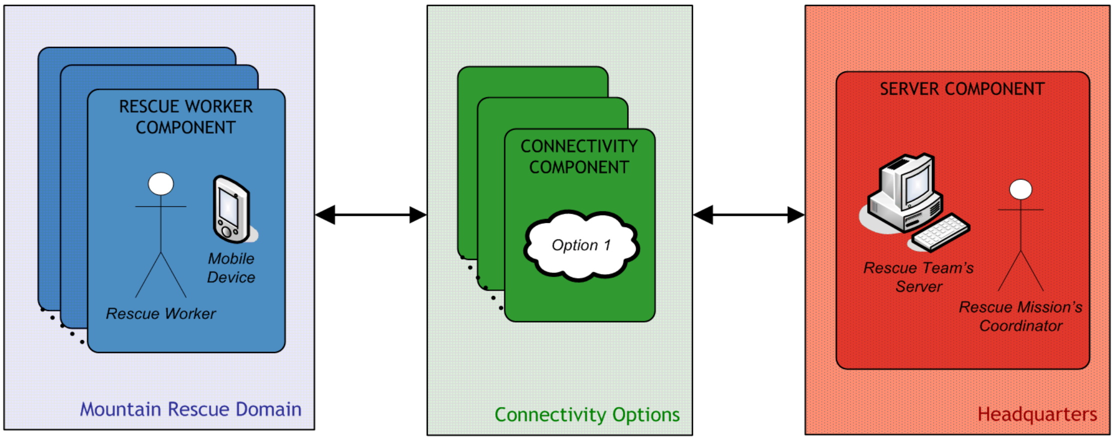


Figure 13 : High level system architecture overview

4.2 Connectivity Options

The connectivity options that could be used to facilitate the communication in a mountain rescue domain are usually limited due to the physical characteristics of the domain. Moreover, since the area is not permanently inhabited the telecommunication companies have not developed a fixed network to provide full coverage at all the parts of the region, a characteristic that makes the decision for the potential connectivity options harder.

Figure 14 depicts again the architecture of the system focusing on the three significant communication options that seem to apply to this domain in order to provide the transmission of the GPS coordinates effectively. The first one is to transmit the coordinates via IPv4 or IPv6 over an 802.11 and/or 802.16 WIFI network, the second one is via IPv4 using GPRS over the GSM network and the third one is to transmit them via SMS over the GSM network. The following sections present the arguments for the described connectivity components that were decided to be used regarding this project's aim.

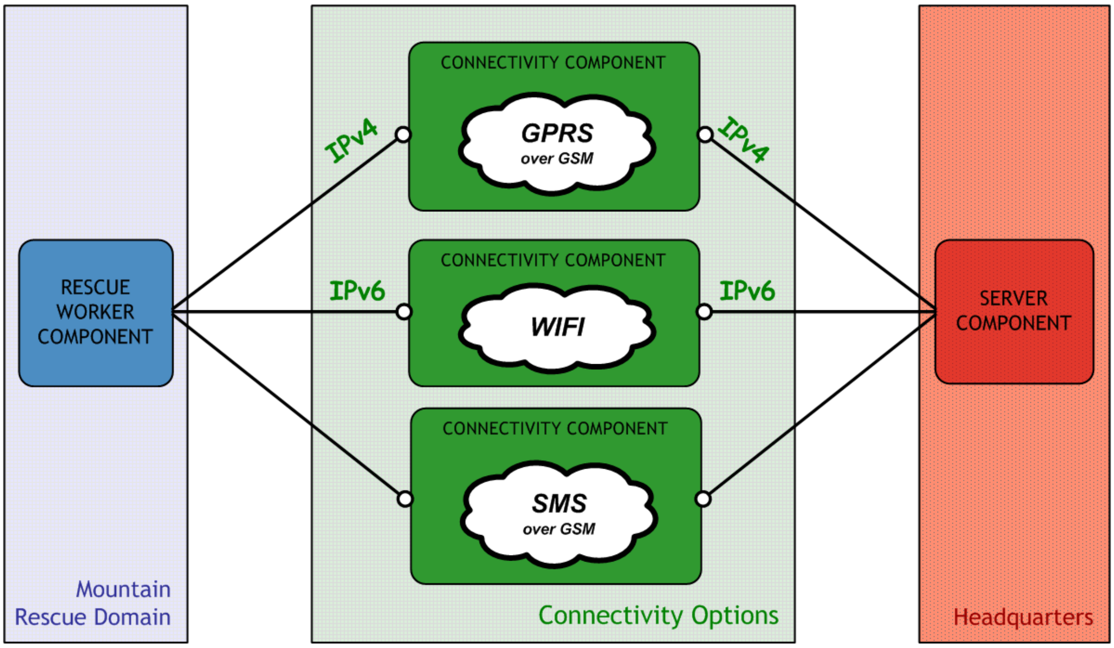


Figure 14 : System architecture focused on connectivity options

4.2.1 WIFI

As the mountain rescue domain is characterized by heterogeneity, physical impediments and lack of fixed network infrastructure, the primary connectivity option should be wireless and provided from the members of the rescue team themselves. Thus, as discussed in Section 2.1.2 the Network Mobility group at Lancaster University has developed a network communication model that includes a mobile WIFI based network infrastructure to provide coverage in the areas that the search parties roam. This WIFI coverage is provided primarily by directed long coverage 802.16 wireless hotspots that are projected in an area from the all terrain vehicles that are parked closest to an incident. This coverage is extended from the mobile routers that a member of each search party carries to provide a shorter range wireless hotspot to facilitate -as possible- the communication among the members of the search party.

The WIFI connectivity option and the way it is used according to the above described communication model presents some significant advantages over other communication methods. Firstly, it can be considered as highly flexible and functional as the long coverage wireless hotspots are projected from moving all terrain vehicles that follow an incident as close as possible. Moreover, considering the fact that the shorter coverage wireless hotspots are projected from the mobile routers that are carried by members which follow the general movement of their teams, the provided WIFI coverage can be considered at least as promising. Secondly, the WIFI based mobile network equipment is owned by the members of the team and thus it can be configured and managed according to the exact needs of the team. Consequently, the routers of this team can be properly configured to take advantage of Mobile IPv6 (described in Section 2.2.4) and the NEMO basic support protocol (described in Section 2.3.2) without having to depend on the configuration of the network equipment of third-party telecommunication companies. Being able to use cutting edge technology and protocols for the network mobility cases in the mountain rescue domain is of great importance as the case itself requires innovative solutions. Thirdly, the WIFI connectivity option and the way that it is provided in the mountain domain, is considered to be less expensive than any other option because it is not dependant on any third-party companies and does not include any billing policy upon usage. WIFI's primary cost is on purchasing and setting up the equipment with almost no maintenance cost.

On the other hand, the presence of the WIFI connectivity option cannot fully confront the unpredictable movements of the members of each search party and cannot provide coverage overcoming all the physical impediments. As a result, the author of this project has defined two additional connectivity options that can provide redundancy on the transmission of the GPS coordinates and improve the effectiveness of the overall communication model. At this point, it should also be mentioned that the defined communication model is a work in progress and has not been deployed yet due to technical restrictions and legal regulations, especially regarding the 802.16 network. However, even though the communication model has not been deployed yet, there is a massive need for alternative communication options that could be used in the Mountain Rescue domain.

4.2.2 GPRS over GSM

One of the alternative options that have been defined to complement the primary WIFI connectivity option is using GPRS over the GSM network. The General Packet Radio Service (GPRS) is a mobile data service that can be used to transmit data over the GSM network [43]. Usually, GPRS is used from individuals to have access to WAP[11], MMS[12] or the Internet from mobile devices such as PDAs and mobile phones

[11] *WAP stands for Wireless Application Protocol, and it is mainly used to provide Internet to mobile phones and PDAs [83].*

[12] *MMS stands for Multimedia Messaging Service and enables sending messages that include multimedia objects [84].*

in a packet-switched fashion, meaning that they are charged only by the amount of kilobytes they transmit. Older GPRS implementations, which are still provided by telecommunication providers, are using the circuit switched data (CSD) standard billing the user per minute independent of whether he\she is actually transferring data [43].

Regarding this project's aim, GPRS is a powerful connectivity option because it can be used wherever there is sufficient GSM coverage. According to [47] and [48] UK major telecommunication providers claim that they have networks covering over 99% of the population. However, as the region that we are interested in is not a permanently inhabited area and as the presented percentage is usually a by-product of their marketing policy, further facts are needed.

Therefore, Figures 15 and 16 present the GSM coverage of two major telecommunication companies, O2 and Orange, in the region that the Cockermouth rescue team operates. It is important to mention that the area with the non-white solid color on the maps (blue for O2 in Figure 15 and orange for Orange in Figure 16) represent the area that can support high quality voice calls and 3G services, whereas the areas with the white color provide variable quality voice and standard data [46]. Areas without any GSM coverage would have been noted on the maps in yellow. As can be inferred from these indications, there is not any area without any GSM coverage in the region that the Mountain Rescue team operates, a fact that provides a promising grounding for the GPRS connectivity option that is used for this project where the WIFI coverage is not sufficient. However, the question whether the GSM coverage is sufficient for establishing and utilizing a GPRS connection in that region, can only be answered with real case studies.

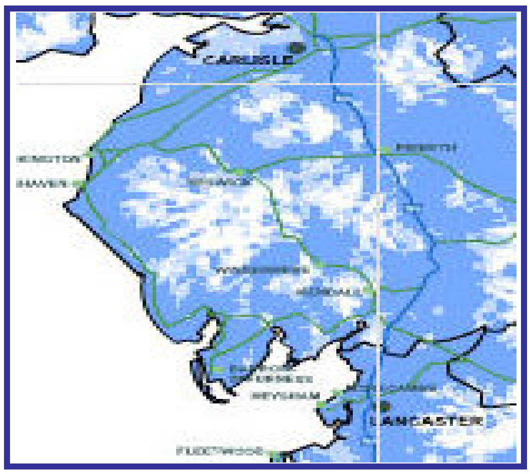


Figure 15 : O2 coverage in the Cockermouth Rescue Team's domain (adapted from [46])

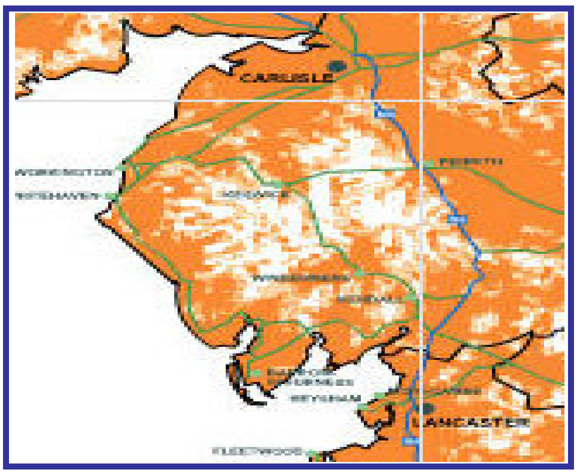


Figure 16 : Orange coverage in the Cockermouth Rescue Team's domain (adapted from [46])

A significant advantage that the GPRS connectivity option presents is that it is an IP based solution that –hopefully- follows the "evolution" of IP based networks. Therefore, although current UK GSM telecommunication companies provide IPv4 support over their GPRS network, we carry expectations that they will migrate to more cutting edge solutions like IPv6. This movement would inherent the benefits of

Mobile IPv6 and provide greater support for the network mobility concept of our domain. Moreover, as the GPRS connectivity component is confronted as an IP based method, any adaptation of any newly developed IP based connectivity solution that a telecommunication company might follow will easily suit this project's architectural structure. For example, if the telecommunication provider of our choice upgrades its GPRS network to EGPRS (EDGE) [44] or UMTS [45] to provide higher bandwidth, this will not require any change to the design of our system as it will continue using the new connectivity component as an IP based option.

Another critical aspect of this connectivity option is the potential bandwidth that it can provide. GPRS was developed to provide bandwidth from 10 to 100kbits/sec based on the class of the GPRS equipment of the client and the quality of the GSM signal. At this point it should be mentioned that GPRS packets usually encounter high latency and round trip time which often reaches 1 second, basically because they are prioritized below voice [43]. Moreover, the bandwidth offered is not too steady as it gets affected by the GSM signal. However, for the focus of this project, that is transferring GPS coordinates, the previously mentioned bandwidth is more than enough as the payload that our application wants to transfer is a string with less than a hundred characters (thoroughly discussed in Section 5.2.1). For that reason, the proposed GPRS connectivity option is also considered to be financially affordable and practical as the Mountain Rescue team will be billed only for the kilobytes of the data being transferred.

4.2.3 SMS over GSM

The well known Short Message Service (SMS) is presented as the third connectivity option that can complement the envisaged scenario in a mountain rescue domain. The data that the developed application transfers can be effectively represented in a defined character format of a fixed length that can easily be sent from a mobile device that a rescue worker carries back to the headquarters.

The usage of the SMS connectivity option depends generally on the availability of the GSM network. At first, someone could support that as both GPRS and SMS are dependant on the GSM network, there is no point providing the SMS connectivity option as an additional method to improve the redundancy of the system. However, as stated in the previous chapter, GPRS is an IP based solution that gets affected from the GSM's signal strength. Therefore, the provided bandwidth fluctuates a lot, a characteristic that affect the network protocols used and imposes delay on the transmitted packets. Moreover, a GPRS connection needs at best 10 to 15 seconds to be set up. On the other hand, the text message option (SMS) is used simply and does not include the complexity of a GPRS connection. Furthermore, an SMS can generally be sent in occasions when the GSM's signal strength is really poor, where a GPRS connection might not have been established at all.

Concluding, the SMS option is generally simple and can be used to increase the redundancy of the developed system. In addition, it is a method that does not require any set up fees and can be relatively cheap if is used as a back up method.

4.2.4 Prioritizing the options

The previous sections have presented the three basic connectivity options that have been chosen to interconnect the mountain rescue domain with the headquarters; WIFI, GPRS and SMS. Although arguments have been given for each method, this section will sum up by prioritizing them and defining the decisions taken upon their characteristics regarding this project.

TABLE 2 : Evaluating the connectivity options for the Mountain Rescue Domain			
	WIFI	**GPRS**	**SMS**
Availability	Very Good	Very Good	Very Good
Cost to purchase/set up	High	Low/None	None
Cost to run	Low/None	Usage dependant	Usage dependant
Extensibility for the provided service	High	High	None
Ability to follow evolution	High	Good	Poor
Practicality	Low	Very Good	High

Table 2 compares and contrasts the three chosen connectivity options. **Availability** is considered to be very good for all of them, judging by the fact that WIFI coverage is provided by special equipment that "follows" the rescue workers' mobility and that telecommunication companies claim to provide GSM coverage on the specified region. However, WIFI availability is dependant on how well the communication model is applied to the Mountain Rescue domain and if the mobile equipment is in "place" at every time. At this point it is important to say that no method can certainly confront the physical impediments of the region or the unpredictable movement of each rescue worker and only real case studies can prove the correctness of the claimed availability.

Moving on, WIFI coverage can be provided by mobile routers and directed antennas, equipment that is generally expensive to purchase but cheap to run. On the other hand GPRS can have low or zero **cost** to set up while its running cost is dependant upon usage. A fact that can be stated without any real use case testing is that the defined type of data that is transferred can keep the running cost for the GPRS connectivity option to low levels. The SMS connectivity option does not need any setting up fees and its running cost is also usage dependant. However, as one SMS is way more expensive than one GPRS packet sent, although they contain the exact same information, the running cost of the SMS service can be relatively high if it is not used with caution[13].

[13] *At this point it has to be said that the cost criteria is relatively subjective. On the one hand, someone could state that the importance of the mission of the Mountain Rescue Team is not comparable with, for example, the 500 text messages that might be sent during a mission. On the other hand, as WIFI packets are provided during a mission without any cost, and as GPRS packets have a low cost there should be a way compare their cost.*

The WIFI and GPRS connectivity options provide –in theory- great **extensibility on the type of the service** that they can support. Although the defined data that is transferred is a fixed set of characters, as both methods are IP based they can truly support any type of network service that can be facilitated from the IP protocol. For example, both communication options can generally support the transmission of photos or complex data types if such a service is needed for the team in the future. The SMS connectivity option cannot provide any extension as the only data type/service that it can provide is transferring a specific amount of characters.

The WIFI connectivity option provides **great ability to follow networks' evolution** and newly developed cutting edge protocols. The WIFI based network can use IPv6 and can have all the advantages that Mobile IPv6 and the NEMO basic support protocol can provide. As the WIFI network is provided with equipment owned by the Mountain Rescue Team it can be configured and tweaked to the exact needs of the team. GPRS, as an IP based connectivity option, has the potential to provide the foremost great advantages, but as it is managed and provided from third-party telecommunication companies its evolution is dependant upon them and cannot be configured for the needs of the rescue team. Finally, the SMS connectivity option cannot follow any such evolution.

The WIFI connectivity option is the least **practical** method as it is provided from mobile equipment (routers) that is carried from members of the team and demands a significant power supply. As this supply cannot be provided for many hours the practicality of this connectivity option is considered to be low. On the other hand, both GPRS and SMS connectivity options are highly practical as they do not require any special and cumbersome network equipment apart from a mobile device that can utilize both options. However, GPRS is considered to be less practical than the SMS option because it demands a larger power supply.

To sum up, as WIFI can be provided from the members of the Rescue Team themselves, can provide great network mobility benefits and is the cheapest option, it is considered to be the primary option that the mobile device that the rescue worker carries, will use. The GPRS connectivity option will be the next possible option (when WIFI coverage is not sufficient) as it can potentially provide greater networking services than the SMS option, without being non affordable to run. Finally, SMS would be used as the least prioritized connectivity option to primarily back-up the two IP-based options when there is insufficient coverage.

4.3 Client

The rescue worker component that was included in the design of the proposed system (Figure 13) represents a rescue worker with all the appropriate equipment needed for its mission. The underlying idea is that each rescue worker will carry the defined equipment and will be able to acquire GPS coordinates according to its location that will be sent in specified intervals to the server component located at the headquarters. This process will aid the Mountain Rescue Team's coordinator to make more informed decisions about the mission of the team because he will be constantly aware of the location of each rescue worker.

Consequently, each rescue worker component should -in general- include a mobile device with the ability to acquire GPS coordinates and transmit them according the best suited connectivity option available at that time. Therefore, the mobile device should be WIFI enabled, GPRS enabled and GMS enabled to be able to utilize the defined connectivity components.

Figure 17 illustrates the system architecture focusing on the final form of the rescue worker component as it was defined in this project. The mobile device used, referred onwards as client, is a HP Ipaq 6915 PDA with WIFI, GPRS, GSM and GPS capabilities. As it can be noticed in Figure 17 an external GPS receiver is used to calculate the GPS coordinates (according to the procedure described in Section 2.4.2) and transmit them via Bluetooth to the PDA which will transmitted them again via the best suited connectivity option. The following sections will discuss the reasoning why an external GPS receiver is used although the PDA has GPS capabilities and will also mention the reasoning behind the decisions taken for the defined rescue worker component criticizing both the hardware and the software used.

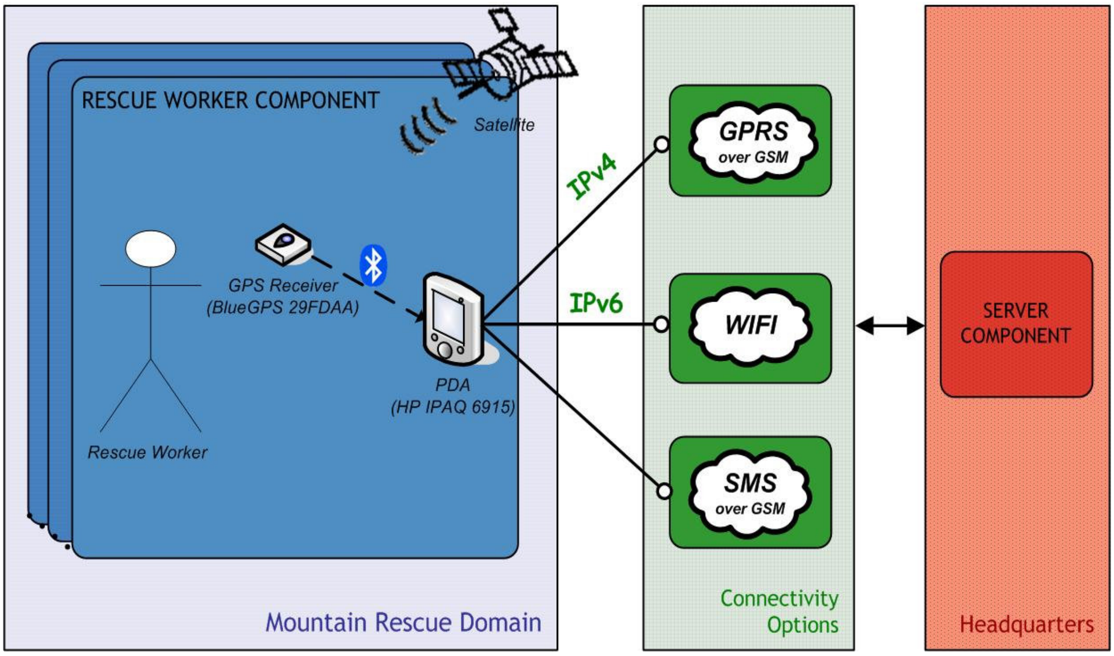


Figure 17 : System architecture focused on the Rescue Worker Component

4.3.1 Hardware

The requirements that the Cockermouth Mountain Rescue team has from the mobile device that the mountain rescue worker will carry on a mission are numerous and cannot be fully satisfied in a project with focus on the better suited connectivity option during a mission. For example, the team requires devices that are lightweight, water proof, durable, "accident resistant", wearable and can be operated easily without any configuration needed from the rescue worker. Moreover, due to the fact that rescue workers usually wear big gloves, requirements such as a big display with large buttons that could be operated easily are generally useful.

By evaluating and prioritizing the above requirements according to the focus of this project it has been decided that a small and lightweight device was needed. Moreover, that device should be able to utilize the defined connectivity components (discussed in Section 4.2) and thus it should have WIFI, GPRS and GSM capabilities. Furthermore, the device should have a nice screen and either an internal GPS receiver or the ability to communicate with an external GPS device. For all the foremost reasons a PDA was decided to be the most suitable solution.

The Network Mobility group had at its disposal a HP Ipaq 6315 PDA (Figure 18) and a Pretec GPS device (Figure 19), that obviously does not meet all the above criteria but could facilitate the needs of this project. The HP Ipaq 6315 PDA device has a 3.5inch screen, WIFI, GPRS and GSM capabilities and could establish a Bluetooth connection with the Pretec GPS device in order to receive GPS coordinates over a serial communication[14]. The specified GPS device is small, lightweight, can concurrently monitor 12 satellites and could send the data over Bluetooth to the PDA.

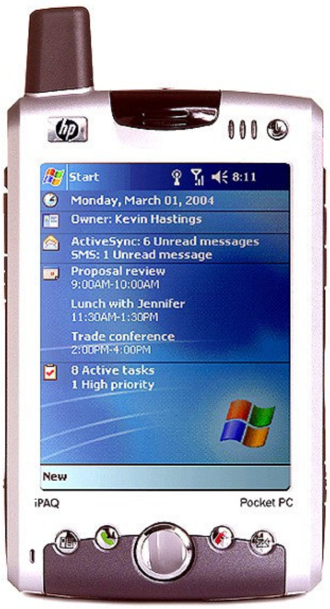


Figure 18 : IPAQ 6315 (adapted from 52)

Figure 19 : **GPS device** *(adapted from 53)*

According to the plan of this dissertation (presented in the Gantt chart of the Research Proposal, Appendix) the first four weeks were devoted to research regarding

---

[14] *At this point it is essential to mention that the PDA could establish a Bluetooth connection with the GPS external device but had no capabilities of showing the streaming data that the GPS device was able to send. This feature, as it is going to be described in the next chapter, was implemented in the application designed for the PDA.*

the hardware, the software and the communication options that could be used for the sake of this project. During this period thorough research was committed on examining the technical capabilities of such devices, their operating system, the available programming language that could be used, the available API and their potential for expanding the outcome of this project.

The outcome of this period defined that although the Ipaq 6315 PDA suited the needs of this project due to its hardware capabilities, its Operating System and the available programming framework for it were found inadequate. In particular, Ipaq 6315 runs Windows Mobile 2003 and the available SDK for developing an application for this project was limited (analyzed thoroughly in the following Section) and thus a brand new device was purchased.

Figure 20 : IPAQ 6915 (adapted from [54])

The purchased new device was an HP Ipaq 6915 (Figure 20) offering new features and providing greater flexibility for developing an application to facilitate the transmission of the GPS coordinates. This new device runs Windows Mobile 2005 and offers an API with greater potential than the previous one. Moreover, Ipaq 6915, apart from all the wireless adapters that it could offer, has an internal integrated GPS receiver that could obtain GPS coordinates, a feature that eliminates the need for the external GPS receiver in a mountain rescue mission. Table 3 presents a subset of the specification of the two PDA devices that were used during this project[15], basically identifying that the new purchased PDA can offer all the features that the older one could, plus the integrated GPS feature and the better SDK that comes with the new operating system.

Concluding, the only equipment/device that a rescue worker should carry during a mission is the Ipaq 6915 that could obtain GPS coordinates and use one wireless

[15] *During the first month of the project the development of the client's application was done on the Ipaq 6315 and it was then that the decision for a new device was taken.*

connectivity option (WIFI, GPRS or SMS) to transmit these coordinates to the server of the team. However, as Figure 17 illustrated the Rescue Worker component includes the use of the Ipaq 6915 with the external GPS device basically because the external device aided a lot the debugging of the developed application[16] and because in some occasions it offered more accurate GPS coordinates.

TABLE 3 : Comparing IPAQ 6315 vs IPAQ 6915 [50, 51]		
	IPAQ 6315	**IPAQ 6915**
Operating System	Microsoft Windows Mobile™ 2003 Software for Pocket PC- Phone Edition	Microsoft® Windows Mobile™ 5.0, Phone Edition with Messaging and Security Feature Pack 5.0
Processors Available	TI OMAP 1510	Marvell PXA270 CPU 312 MHz
Memory	64-MB SDRAM (55 MB user accessible)	64-MB SDRAM (45 MB user accessible)
Display	3.5 inch (89 mm) transflective color TFT, 240 x 320 pixels, 0.24 mm dot pitch, 64K-color support, touch screen	3.0 in (75 mm) diagonal, transflective TFT color, 240 x 240 pixels, 0.24mm dot pitch, 64K-color support, touch screen
Wireless	Integrated quad band GSM/GPRS, Wi-Fi 802.11b, Integrated Bluetooth 1.1, Integrated IRDA	Integrated Quad band GSM/GPRS/EDGE wireless radio with automatic band transition; Integrated Wi-Fi (802.11b); Integrated Bluetooth 1.2 wireless technology; Integrated IrDA SIR
Dimensions (w x d x h)	2.95 x 0.73 x 4.68 in (75.0 x 18.7 x 119.0 mm)	2.8 x 0.71 x 4.65 in (71 x 18 x 118 mm)
Weight	190 g	179.45 g
Battery	Removable/rechargeable 1800 mAh, 3.7 Volt, Lithium	Removable/rechargeable 1200 mAh Lithium-ion
Extras	-	GPS, Built-in VGA Camera

4.3.2 Software

As the hardware device that would satisfy the Mountain Rescue Team's requirements regarding the focus of this project have been defined, the remaining part is to identify the exact programming language, API and/or SDK that would be more suitable for the needs of this project.

[16] *As mentioned in Section 2.4 the GPS receivers do not work indoors and thus in order to debug the GPS feature of the developed application we should either have worked outdoors or have the external GPS device placed next to a window and establish a Bluetooth connection between it and the PDA in order to receiver the GPS data. Obviously, the latter method was used during the software development phase.*

Developing an application for a PDA does not provide a wide range of options regarding the available programming languages especially when there is need for use of numerous different connectivity options and technologies. Moreover, as PDAs have limited CPU, memory, storage and display capabilities the chosen programming language should be able to comply with them. Furthermore, the requirements of the project demand full access to the WIFI and GPRS network adapters, to the Bluetooth stack, as well as ways of using the GSM network. Finally, although presenting a GUI to the rescue worker or the mission coordinator was never a basic requirement for this project, an API with display capabilities in such devices would be desirable. Research based on these requirements defined that the two most suitable and powerful programming languages for developing an application for a PDA are Java and Visual C#.

TABLE 4 : Various Java based solutions for Pocket PC (modified from[55])

	Sun Personal Java	Esemertec Jeode	IBM Websphere Studio	Blackdown J2RE (ARM Port)
Supported OS	WinCE 2.11	WinCe 2.11, Pocket PC, Linux	Win CE 2.11, Pocket PC	Linux
JVM Compatibility	Personal Java Compiant	Personal Java Compiant	J2ME MIDP 1.0	Java 2
Speed	Fast	Fast	Fast	Slow Start Reasonable Execution
Supported Hardware	MIPS, SH3	Dell Axim X5	PocketPC,WinCE, PalmOS,Windows	IPAQ H36xx (or above)
Cost	Free – Sun	$49.95	(depends on the exact Micro Environment used)	Free

Starting from Java, Sun has correctly identified that there was a need for providing a subset of J2SE for mobile devices and thus it initially published the Sun Personal Java which was later renamed to J2ME and then to JavaME (Java Micro Edition). JavaME successfully dealt with the constraints the small devices present and tried to support Sun's motto "write once, run everywhere" [56]. However, quoting from Sun's web portal, "Java ME platform is a collection of technologies and specifications that can be combined to construct a complete Java runtime environment to fit the requirements of a particular device or market" [56]. Therefore, this runtime environment is totally device dependant and requires specific configurations and profiles not only for each device but also for each implementation of every feature of any device. The author of this report believes that this might be the main reason that Sun does not provide any official support and Java Virtual Machine for mobile phones and pocket PCs (PDAs). Consequently, the evolution of JavaME was based on various Virtual Machines, configurations and profiles that many third party companies have developed to either support their mobile devices (e.g Nokia phones [57]) or specific implementations of a feature such as the Bluetooth stack 2.0 or the Wireless Industry specification (JTWI) [58]. Table 4 presents only four significant

Java based “solutions” that could be used for developing an application for a mobile device, that actually pinpoint the variety of properties that should be considered[17].

The alternative programming language that could be used to develop the application for the PDA that a rescue worker would carry was Visual C#. Microsoft Visual .NET framework was created to provide programmers with a variety of libraries that could be used to create new applications easier especially when combined with Visual C# [61]. Imitating Sun, Microsoft has also defined a subset of the .NET programming framework for mobile devices called .NET Compact Framework (.NET CF) which includes libraries especially tweaked for use in devices with limited resources [60]. Although .NET CF can be used in devices that run only Microsoft’s Operating Systems, it provides a great advantage of calling unmanaged code from managed code in such devices [60]. According to this characteristic embedded C++ or C code could be used in the .NET CF environment to have low level access to native DLL files that the operating system uses, potentially giving access to all the features that a mobile device has.

Comparing the foremost options regarding the programming language and the SDK that could be used for developing the application for the client’s side, it can be stated that the pair of Visual C# and .NET Compact Framework presents significant supremacy against Java based solutions. The fact that the HP Ipaq 6915 PDA which was purchased was running Microsoft Windows Mobile 2005 aided the decision for using the .NET Compact Framework 2.0 and Visual C#. At this point it is essential to mention that the .NET CF has seriously restricted capabilities from the .NET framework used on PCs and lacks essential functionality that someone could expect to see in a framework. On the other hand, the benefit of having access to native DLL files and embedding C code to gain low level access presented an advantage that could not be disregarded. Moreover, the fact that the freely distributed and used .NET CF was supported by Microsoft against any small or big third party company trying to sell their developed Java Virtual Machines for only a small subset of devices and features, was also considered as essential making the decision for our chosen programming language and framework easier.

4.4 Server

The server component of the envisaged system according to the networking communication model for the Mountain Rescue domain should include a PC located at the headquarters of the team, with the server application and the ability to receive the GPS coordinates that a client sends regardless of the connectivity option used.

Figure 21 depicts the system architecture focusing on the server component located at the headquarters of the team. This component includes a PC with network interfaces able to listen to IPv4 and IPv6 packets and a “GSM module” listening for incoming text messages (SMS). Finally, the server application should be able to present the received information to the Rescue Mission coordinator.

[17] *Two extensive comparisons of the many Java based solutions for PDAs can be found in [55] and [59].*

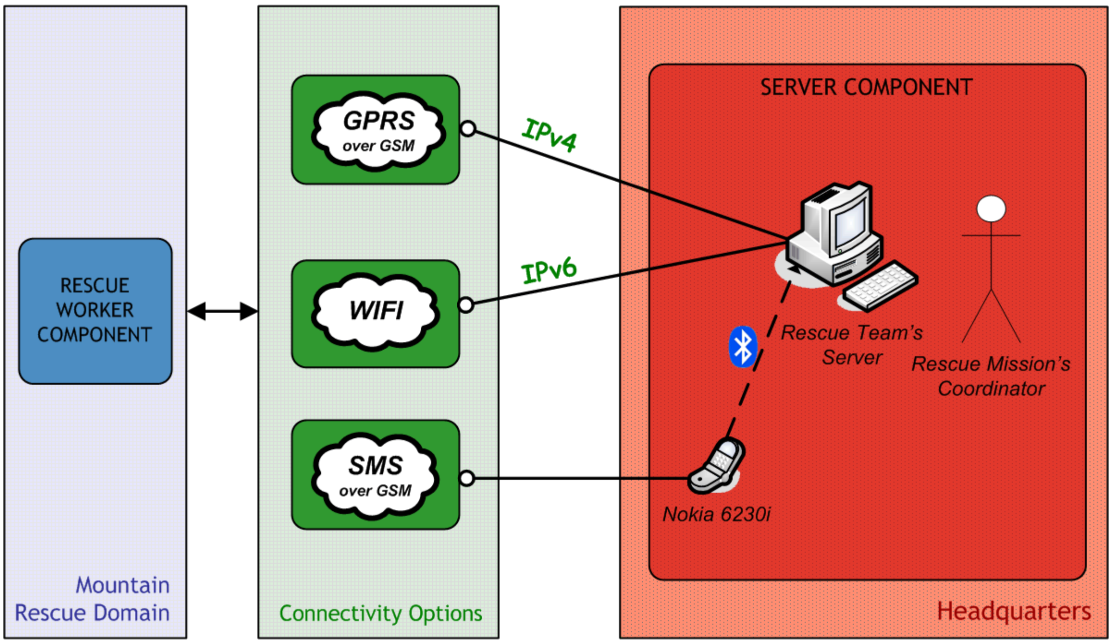


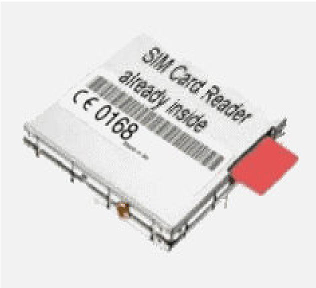
Figure 21 : System architecture focused on the Server Component

4.4.1 Hardware

The hardware requirements of the server component are simpler to satisfy as the server is located at a fixed location, has no limited resources and its only necessity is to be able to listen to all the packets that the clients sent.

Therefore, a regular PC with a broadband connection should be used as a server with a network interface listening to packets sent from the clients either via WIFI or GPRS. The decision whether a PC should have one network interface listening for both IPv4 and IPv6 packets (using two sockets) or two different network interfaces listening individually for IPv4 and IPv6 packets is just an implementation issue (discussed in Section 5.2.3). An important design decision for the network interface(s) that the server would have is that each IP address used should be either a global IPv4 address or a unique global IPv6 address so that packets destined to that IP address would be able to reach the server via any intermediate network (usually the Internet).

Apart from the IP based connectivity options that were defined for this project, a client has also the ability to send GPS coordinates via SMS and thus the server should have a GSM module to receive these packets. Generally, there are various GSM modules (some of them depicted in Figure 22) that could be used to provide the SMS service to the server application but for the shake of this project a standard Nokia 6230i mobile phone was used for this purpose. As Figure 21 illustrates a Bluetooth connection between the Nokia 6230i mobile phone and the server was established, with the aid of a Widcom USB-Bluetooth dongle on the server's side, so that the server application would have access to the SMS messages sent from the clients to that mobile device. As becomes apparent, such a solution cannot be considered as a permanent one for the Mountain Rescue Team but it was a flexible solution for the proof of concept of this project.



GSM Module 9/18 ([62])

Falcom C2D-SI-GPRS ([63])

Nokia 12i ([64])

Figure 22 : GSM Modules

4.4.2 Software

The application on the server side could be actually developed with almost any programming language suitable to handle network sockets and the connection with the mobile phone. For compatibility reasons and although this was not mandatory, the author of this dissertation decided to use the .NET framework and Visual C# for the development of the server application. Although the GUI part for neither the server's nor the client's application was a main focus of this project, the visual implementation of C# was preferred against the plain one (C#) so that the server's application would be able to have a GUI in order to present the retrieved data to the Mountain Rescue coordinator in a nice and clear format.

Chapter 5 : System Implementation

This chapter discusses the system implementation based on the architectural design that was presented in the previous chapter. Therefore, Section 5.1 introduces the chapter and the tools used, Section 5.2 discuses the important implementation decisions for the defined connectivity options, Section 5.3 discusses the implementation on the client's side (PDA) and finally, Section 5.4 discusses the implementation on the server's side.

5.1 Introduction and tools used

This chapter discusses the implementation of the system and all the main issues that arose during the development phase and the ways that they were confronted. The IDE that was used for both the client (PDA) and the server applications was Visual Studio 2005 and the main SDKs were .NET Compact Framework 2.0 for the client and .NET Framework 2.0 for the server. Many additional libraries were also used to add functionality to the developed applications and will be mentioned throughout this Chapter. Finally, the programming language used for both the development of the main functionality of the applications and for having access to low level native DLL files on the PDA side using Platform Invoke methods was Visual C#.

5.2 Communication & Connectivity Options

The communication of the clients with the server is based on the 2-tier model, basically including clients roaming in the Mountain Rescue domain and communicating with the server located at Cockermouth over the best suited connectivity option. The communication among the clients and the server would be "established" using one of the predefined best suited connectivity options, namely IPv6 over WIFI, IPv4 via GPRS over GSM, and SMS over the GSM network (discussed in Section 4.2). These connectivity options would be used upon their availability and their prioritization discussed in Section 4.2.4. An important decision was taken defining that the communication among the clients and the server should be unidirectional, from the clients to the server, based on the simplicity that this model present and based on the networking protocol used (discussed in Section 5.2.2).

The remainder of this Section is organized as follows. Section 5.2.1 discusses the uniform defined format of the packet being sent regardless the connectivity option used. Section 5.2.2 discusses the implementation regarding the IP based connectivity options and the networking protocol used. Section 5.2.3 discusses the way that the IPv6 connectivity option was implemented and the networking restrictions that became apparent on the testbed that was used during the development phase. Finally, Section 5.2.4 discusses implementation issues regarding the use of the GSM network.

5.2.1 Payload of packet sent

The application that was developed for the clients (PDA) should be able to utilize the different connectivity options that were defined for this project. It was decided that it would be better if the payload of the packets being sent, either via WIFI or GPRS or SMS would have a uniform format and thus ease the functions of composing the fields to packets on the client side and decompose them on the server side.

Figure 23 illustrates the fields of the packets being sent. As it is depicted the defined uniform format of each packet is composed of eight different fields separated with the character "#". The use of a special character to separate the fields instead of defining a fixed length for each field and pack them one after the other was preferred for two basic reasons. Firstly, some fields, such as the "Type of Connectivity Option", "Timestamp"[18] and "Possible extensions" have not a fixed length and the use of the special character eases their separation on the server side without having to also send the size of each field. Secondly, all the packets being sent and received are logged on both sides and the use of a special character eases their readability, instead of having them being packed one after the other.

Figure 23 : Format and example values of the payload of the packets sent regardless of the connectivity option used

The defined format of each packet consists the following fields :

1) **Type of Connectivity Option :** This field describes the connectivity option used for this packet to be send. Nominal values for this field are WIFI, GPRS and SMS.
2) **Node ID :** This field includes the Node ID of the client that sent a packet. The Node ID is a three digit number that uses the first (from the left) digit to describe the ID of the search party of the client with ID described by the two remaining digits, belongs to.

[18] *The Timestamp field could be either 21 or 22 characters depending on the length of the milliseconds of the current time (3 or 4).*

3) **Security Code :** This field includes a four digit number that is unique to every node and is used as a security code to prove to the server the authenticity of the node that sent the packet. This security code can be changed from the GUI that each client has[19].
4) **Sequence Number :** This five-digit field describes the sequence number of the packets being sent from one client. The server can use this number in conjunction with the node ID of a packet to identify packets received in wrong order.
5) **Timestamp :** This field describes the time and the date that a packet was sent from the client. The time value follows the format "HH:mm:ss:msec" and the date the format "dd/MM/yy". These values are acquired from the Operating System of the PDA.
6) **Latitude :** This field includes the latitude part of the GPS coordinates that the client sends in the NMEA format that will be discussed in Chapter 5.3.2. An interesting point is that the latitude in NMEA format should be of the type, for example "54°00.3137"N", but it was found that the character "°" could not be sent in an SMS message. Therefore, the workaround that was developed was to replace, on the client's side, the character "°" with "@" to all the transferred packets and modified it back on the server side to keep a uniform format.
7) **Longitude :** This field includes the longitude part of the GPS coordinates that the client sends in the NMEA format which should be of the type, for example, "002°47.3053". Again the character "@" was used instead of character "°" to preserve a uniform format.
8) **Possible extensions :** This field can include any additional information that the client decides to send and is basically there for any future use. The current use of the field includes the value "normal" to describe that the packet was sent using an available communication option and was not withhold due to no availability of none connectivity options.

Concluding, the above defined format for the payload of each packet will be used from the client regardless of which of the best suited connectivity option will be used at any point in time.

5.2.2 IP based connectivity options and network protocol

Sections 4.2.1 and 4.2.2 described the two IP based connectivity options; WIFI and GPRS respectively. The mobile network infrastructure of the Mountain Rescue Team is responsible for providing the WIFI coverage in the domain, while the GPRS could be used based on the GSM signal coverage of the mountain rescue domain. Both IP connectivity options was decided to be implemented using network sockets and transferring packets over them. A basic dilemma was which of the most used

[19] *At this point it is essential to mention that in the networking world of IPSEC, complex encrypting and hashing functions, ciphers with 1024bit keys and IPv6 with embedded security feature no one could claim that an unencrypted password field could add security to the described communication model. The Security Code field just adds a tiny notion of security in the developed application.*

protocols, TCP or UDP should be chosen to facilitate the communication over the IP layer.

Table 5 presents a basic comparison of TCP versus UDP, a well known and thoroughly examined issue in various studies [65, 66, 67]. Mainly, the basic characteristics of TCP is that it is a connection oriented protocol that guarantees delivery of packets in the correct order basically due to the error correction, flow control and congestion control mechanisms [65]. On the contrary UDP is a connectionless protocol that does not guarantees delivery, lacks of error correction mechanisms and is generally fast by being aggressive.

TABLE 5 : TCP vs UDP

TCP	UDP
Connection oriented	Connectionless
Guaranteed delivery	Non guaranteed delivery
Error correction (*sequence numbers, acknowledgments, in order delivery*)	No error correction mechanism
Congestion Control Mechanism	No mechanism for identifying congestion
Monitors network status	Aggressive

By carefully examining the above characteristics it has been decided that TCP should be the chosen protocol to facilitate both IP connectivity options, namely sending IPv6 packets over WIFI and IPv4 packets over GPRS. The main reason for choosing TCP was that its inherent mechanisms for error correction, acknowledgments and sequence numbers would basically remove the necessity for developing such features on the application layer. These benefits, that the lower layers of the TCP/IP stack present, could keep the communication at the application layer simple and unidirectional. The need for having to implement acknowledgements for each packet sent on the application layer would introduce a lot of complexity in the envisaged communication framework, judging from the fact that the mountain rescue domain lacks a constantly available connectivity option.

On the contrary, although TCP is using inherent sequence numbers for the packets being sent, the developed application should maintain sequence numbers for the packets that a client sends using different connectivity options. This was needed because, for example, at a given time, a client might send an SMS and then swap to the WIFI connectivity option and send an IP based packet. Since the GSM network gives to SMS messages the lowest priority, the SMS message might arrive later to the server than the WIFI packet. Therefore, the application layer should be able to maintain and check the validity of the sequence numbers that are included in the packets being sent from clients and thus identify occasions when packets arrive in wrong order.

5.2.3 "Implementing" IPv6

One of the interesting parts of this project was to be able to use IPv6 for the communication of the clients with the server over the WIFI network. Section 2.2 presented that IPv6 and especially Mobile IPv6, introduces a lot of benefits in the concept of networks and hosts mobility and thus there was a challenge to utilize it and give to all the IPv6 enabled intermediate and boarder network devices the ability to take advantage of it.

In order to utilize IPv6 in the purchased Ipaq 6915 both, the Operating System of the device and the programming framework that was decided to be used for the development of the client's application should be able to support IPv6. As IPv6 was initially thrown as a requirement in this dissertation, both Windows Mobile 5.0 [68] (that the purchased Ipaq runs) and the .NET Compact Framework [69] that was used for the development of the client's application could theoretically support IPv6 as this was one of the reasons that they were chosen.

Unfortunately, Windows Mobile 5.0 do not provide neither a GUI form for configuring IPv6 nor the usual ipv6.exe tool to manually configure IPv6 on the PDA. Moreover, the IPv6 address of the WIFI network adapter of the Ipaq could not be found neither on the equivalent of the registry that PDAs have, nor programmatically. After a lot of experiments and methods trying to configure the IPv6 feature on the device a workaround[20] was used to identify the link-local address that the device obtained during the autoconfiguration and the neighbor discovery phases. By having the link-local IPv6 address of the PDA we were able to establish a communication with the server by using IPv6 TCP sockets over the WIFI network.

At this point of discussing the use of IPv6 it should be mentioned that the fist testbed that was used during the development of the applications for this project, included an ad-hoc wireless network between a laptop configured with IPv6 and running the application of the server and the Ipaq 6915 running the application of the client. Consequently, the transmission of IPv6 packets over a WIFI network was successfully implemented and tested on this testbed in conjunction with the SMS over GSM connectivity option that has also been developed at that time. In the process of developing the functionality to utilize the GPRS connectivity option it has been decided that the server machine should have a global IPv4 address so that it could be addressed from clients over the GPRS network and an IPv6 address so that it could be addressed from clients via WIFI. Therefore, and keeping in mind that a more generic testbed was needed, the server application was deployed in a machine placed in the Computing Lab (Room A27, Infolab21) with both a global IPv4 address and an IPv6 one. In addition, at that time it was also decided that the WIFI network of the campus would be more suitable to provide the infrastructure for the WIFI connectivity option.

The problem that was apparent in this new testbed was that none of the University's wireless networks was able to provide IPv6 support in order to transfer

[20] *The only way of identifying the IPv6 address of the device was by capturing WIFI packets when the wireless network adapter of the PDA was turned on and thus notice the IPv6 address that the device was publishing as part of the network discovery phase.*

the packets that the client would sent to the server. A meeting with an ISS[21] representative suggested the following workaround. The ISS would provide a global IPv4 address to the PDA for its wireless network adapter and then this address would be used to create a 6to4 IPv6 address in order to successfully utilize the IPv6 over WIFI connectivity option defined in our project. Unfortunately, until the end of August the ISS was unable to provide the above mentioned global IPv4 and thus, keeping in mind the evaluation phase of this project, it was decided to also utilize IPv4 over the "eduroam" wireless network of the University's campus. Therefore, the remainder of this Chapter will discuss the implementation based on the fact that eventually IPv4 was used for both the WIFI and GPRS connectivity options.

5.2.4 Using the GSM network

In order to be able to utilize the GPRS and SMS connectivity options over the GSM network two O2 pay as you go sim cards were purchased based on the fact that O2 has a good GSM coverage over the Mountain Rescue Domain that the Cockermouth Mountain Rescue Team operates in. Unfortunately, O2 could not be able to provide GPRS with the packet-switched option to pay as you go customers but was offering instead the circuit-switched option (CSD) charging each client per minute of being connected even when it was not using the GPRS connectivity options. Therefore an additional Orange pay as you go sim card was purchased to utilize the GPRS connectivity option and the SMS connectivity option on the client side and one of the two O2 sim card was used for listening to SMS packets on the server side.

5.3 Client

The Client application was developed mainly in Visual C# with the aid of many libraries and some extern calls to native DLL functions in order to have lower level access to the network functionalities of the PDA. Generally, the client's application was developed having in mind that the Ipaq 6915 has limited resources and should be used with caution.

Figure 24 presents a high level class diagram and their abstract interconnection for the application developed for the client. Mainly the `Program` static Class initiates the application and creates an object of the `GUI` class which provides the main functionality. The `GUI` class includes a GUI form to hold all the elements that are presented to the rescue worker and also instantiates one object of the `GPSConnector` class and one of the `IPConnector` class. The `GPSConnector` class provides the connection with the GPS device and instantiates an object of the `NMEAInterpreter` class. Basically, the GPSConnector initiates a connection with either the external GPS device via Bluetooth or the internal GPS module that the Ipaq 6915 has and continuously retrieves data using a serial communication with one of

[21] *ISS is the Information Systems Service Department of Lancaster University.*

them. These data are passed to the NMEAInterpreter object which fires events upon the information retrieved.

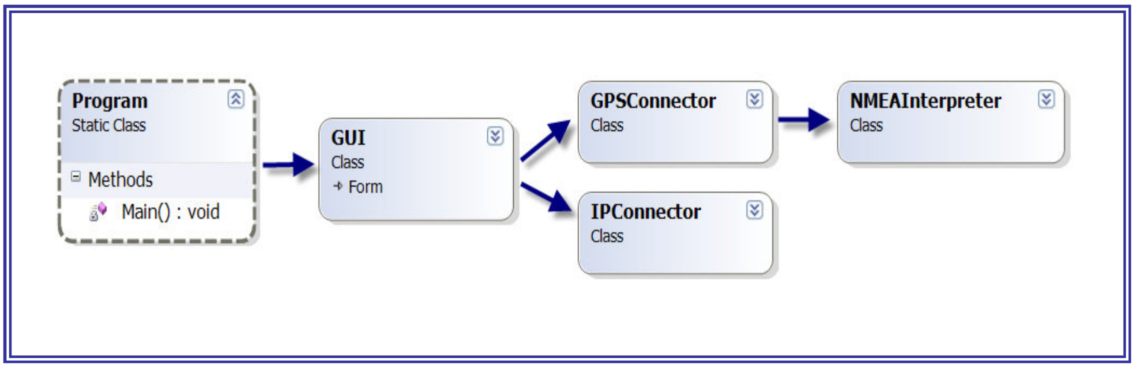


Figure 24 : High Level Class Diagram illustrating their interaction for the Client's application

The remainder of this Section describes the main Client components. In particular, Section 5.3.1 describes the `GUI` Class, Section 5.3.2 the `GPSConnector` class, Section 5.3.3 the `IPConnector` class, Section 5.3.4 the Map feature, Section 5.3.5 the Log feature and finally, Section 5.3.6 describes two main restrictions that the device presented.

5.3.1 GUI

The GUI object of the `GUI` class is instantiated from the main method of the `Program` class and holds all the elements that the user could see, along with the main functionality of the client's application. This huge class, composed of about 3500 lines of code, is split into two distinct logical and practical parts taking advantage of the partial class implementation feature of Visual C#. The first part of the `GUI` class is the `GUI.Designer.cs` (automatically named by the Visual Studio) including all the GUI elements that the rescue worker can see and interact. The second part is the partial implementation of the `GUI` class including the main functionality of the application developed for the client.

At this point it should be mentioned that this automatically done separation was followed in order to keep the functionality of the program as effective and simple as possible, since the `GUI.Designer` and the partial `GUI` main functionality implementation are constantly passing arguments to each other. An alternative solution could be to create a totally distinct class for the main functionality of the program but this would impose the implementation of many more background threads and methods to pass arguments from one thread to another, a process that would require much more of the valuable limited resources of the PDA. The two aforementioned distinct parts are going to be examined in the following Sections.

5.3.1.1 GUI Designer

The GUI Designer includes 75 variables and 37 methods to represent, update and handle the elements of the GUI Form that is presented to the rescue worker. Moreover, these functions are handling the interaction of the user with the GUI and the events that are triggered when GPS data have been received.

The GUI of the client's application is mainly composed of five different tab pages to separate distinct parts of the application, namely Main Tab, Client Tab, Server Tab, Map Tab and Info Tab (identified in Figure 25).

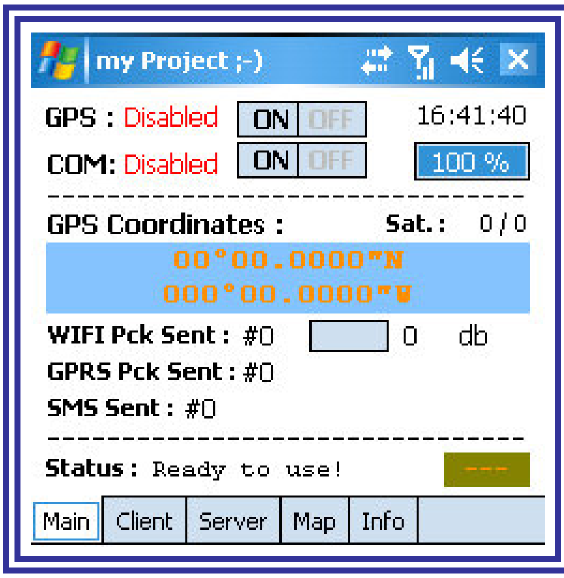


Figure 25 : Main tab-page of the client's application not in use

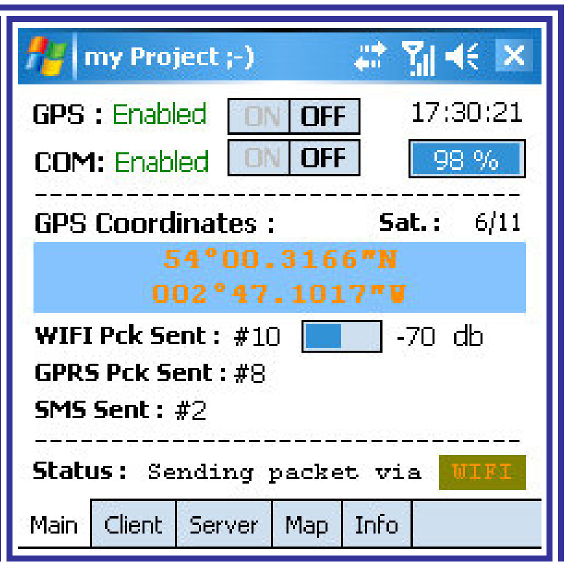


Figure 26 : Main tab-page of the client's application in use

Figures 25 and 26 present the Main tab of the application which is the main screen that a rescue worker should look during a mission. Figure 25 presents the application after it has been executed and being ready to use, whereas Figure 26 presents the Main tab while the application is functioning.

In the upper left corner of the Main tab two basic functionalities are presented to the user; the GPS and the communication (COM), which could be turned on and off using the suitable button. If the user turns the GPS functionality on then a connection is going to be established with either the external GPS device or the internal GPS module that the PDA has, based on the options chosen on the Client tab (presented later in this Section). If the user turns on the communication functionality, which can be done only if the GPS functionality is on, then the client will start transmitting GPS coordinates using the best suited connectivity option every 15 seconds. The upper right corner of the Main tab presents the current time and the capacity of the battery of the PDA.

The central part of the Main tab presents the longitude and the latitude of the GPS coordinates received from the GPS module in NMEA format. These values are updated every time an appropriate signal is received from the GPS module and transmitted to the application based on the anonymous delegates, an inherent functionality of Visual C# that enables methods to be called when events are fired. On the right side of the central part of the Main tab the satellites that are used (locked) from the ones that can be viewed are presented.

Below the GPS coordinates, distinct counters for the packets that are sent using each connectivity option are illustrated to the user. Moreover, next to the WIFI packets sent counter, the WIFI signal strength is presented in both graphical and numerical format (db).

At the bottom of the screen, above the tab options, an informative message is given according to the functionality of the application at any given time. For example, messages like “Sending packet via (…)”, “Changing networks”, “Found eduroam” or “Trying to reconnect” are likely to be presented there. Finally, at the bottom right corner of the Main Tab the current connectivity option used is presented, namely WIFI, GPRS or SMS.

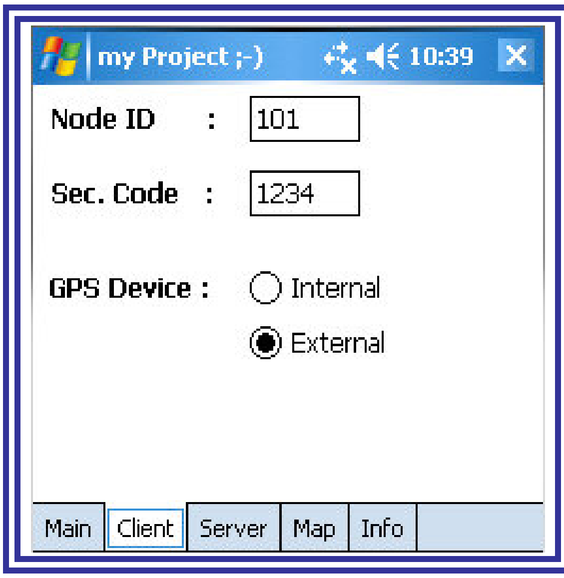


Figure 27 : Client tab-page ;
presenting the settings for the client’s side

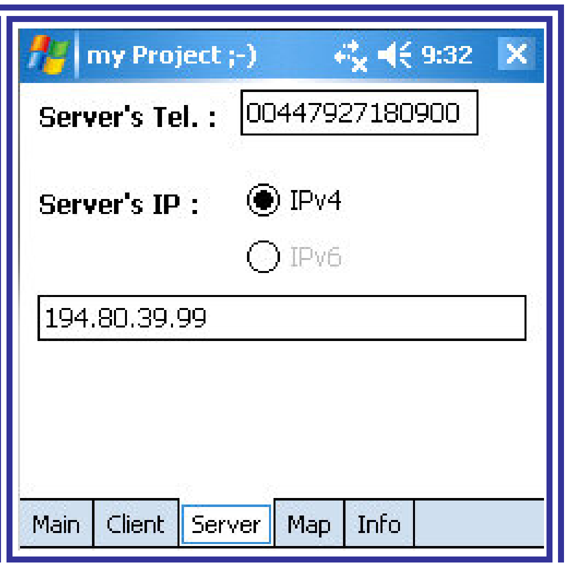


Figure 28 : Server tab-page ;
presenting the settings for the server’s side

Figure 27 presents the Client tab page, illustrating the settings for the client’s side of the application. On this tab, the rescue worker can change the Node ID of the device and its security code. The application does some “sanity” checks for the Node ID and the Security Code values in order to be sure that they are always numbers and their length is 3 and 4, respectively. Moreover, on this tab there is an option defining if the application will receive the GPS coordinates from the external Bluetooth device (Pretec GPS device, Figure 19) or the internal GPS module that the Ipaq has integrated.

Figure 28 presents the Server tab page which includes settings regarding the server. In particular, the telephone number that should be used to send SMS messages to and the IPv4 address that the IP packets should be addressed to are presented.

Again, sanity checks are also done to both the server telephone number (a 10 digit number) and the IP address field. As it can be noticed in Figure 28 the IPv6 option is currently unavailable based on the reasoning presented in Section 5.2.3.

Figure 29 presents the Map tab page that gives to the rescue worker the ability to see his\her exact location on a map based on the GPS coordinates that are presented in the Main tab page (discussed in Section 5.3.4) . Finally, Figure 30 illustrates the Info tab page which presents to the rescue worker data retrieved from the Satellites, such as the current date, time, bearing and speed of the person who holds the PDA as they are retrieved from the GPS module.

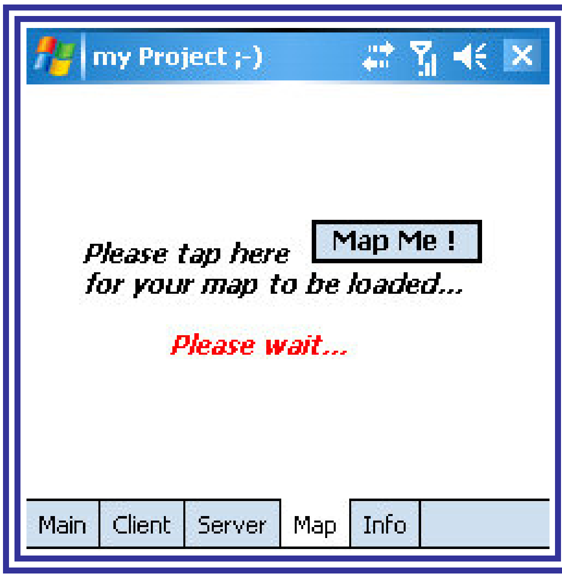


Figure 29 : Map tab-page
(the button "Map Me!" has been tapped)

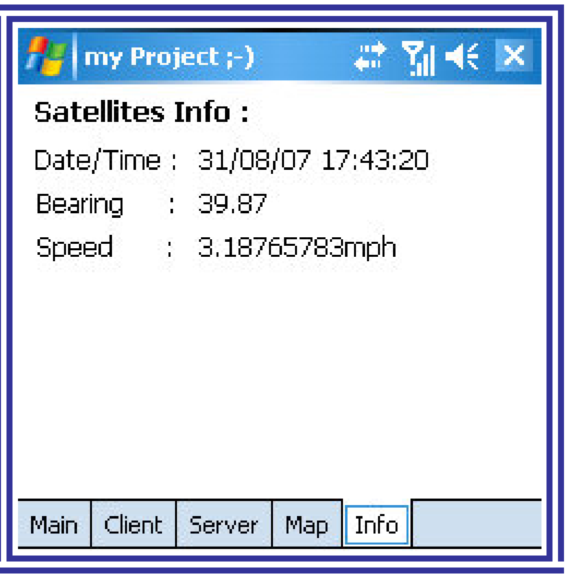


Figure 30 : Info tab-page ;
presenting information retrieved from the Satellites

5.3.1.2 GUI Main Functionality

This Section describes the main functionality of the application which is the partial implementation of the `GUI` class of the client. Upon the creation of the object of the `GUI` class, done on the Main method of the `Program` class, the following five actions are taken :

1) **Initialization** of all the **GUI elements**
2) **Initialization** of all the **GUIDs** that are the references that the class will hold to refer to either the WIFI or the GPRS connection
3) **Initialization** of the **log file**
4) **Creation** of a **GPSConnector object** that will "represent" the connection with the GPS module
5) **Creation** of an **IPConnector object** that will "represent" the best suited IP connectivity option at any time

When all the above actions are done successfully the application waits for the rescue worker to firstly enable the GPS functionality and then to start the communication by taping the ON button presented on the screen for both of them (Figure 25). If the user enables the GPS functionality, and has not modified the default option that is using the external GPS device, he will notice a screen with all the available Bluetooth devices that the PDA could see, and therefore he should choose the Pretec GPS device to establish a Bluetooth connection with the external device. If the user has chosen the internal GPS module from the Client tab (Figure 27) then he should also wait for the PDA to initialize the internal GPS module. From that point on, the application will display to the user GPS information that will be received from the GPS module (discussed thoroughly in Section 5.3.2).

Consequently, the rescue worker will tap the ON button to start the communication phase and instruct the device to transmit GPS coordinates to the server. At this point, the application performs various checks to identify if the WIFI connectivity option is available[22]. If the WIFI connectivity option cannot be used at that time, then the device will try to initiate a GPRS connection. If either of the previous IP-based connectivity options is successful (priority given to WIFI) then the device will try to open a socket with the server over the successful connectivity option and inform the user that he has been connected to the server. Whether the previous procedure has been successful or not, in the case of both the WIFI and GPRS connectivity options failed to be initialized or obtain a TCP socket, the application will initialize two different timers which will be used for the onwards communication.

The first timer, named as **timerWifiSignalStrength**, ticks every two seconds and is responsible for updating the WIFI signal strength indication on the Main tab page on the screen. In order to achieve this update it performs the following actions in the described order :

1) Acquires a reference of all the network interfaces
2) Identifies the wireless network interface
3) Checks if the PDA has been connected to the defined WIFI network[23]
4) Checks if the PDA has obtain a valid IP address[24]
5) Acquires the signal strength (in db) of the wireless adapters and updates the WIFI signal indication on the screen.

The acquisition of the WIFI signal strength is implemented with the use of the OPENNETCF library [70] based on the indications that the WIFI network driver can provide to the developed application. This WIFI signal strength indication can obtain one of the following distinct levels ; excellent signal (higher than -50db), very good signal (lower than -50db and higher than -70db), good signal (lower than -70db and higher than -80db) and poor signal (lower than -80db).

[22] *These checks are identical with the actions that the timerWifiSignalStrength performs that are described below.*

[23] *As described in Section 5.2.3 the eduroam wireless network of the University's campus was used in the testing and the evaluation phases of this project.*

[24] *Although the PDA may have managed to connect to eduroam, it may have for a significant number of seconds a non valid –in this context- IP address (0.0.0.0) basically due to the authentication mechanisms that need to take place before the acquisition of a valid IP address. From the application's point of view, when the PDA is connected to eduroam but is using the 0.0.0.0 IP address, the WIFI connectivity option is considered to be unavailable.*

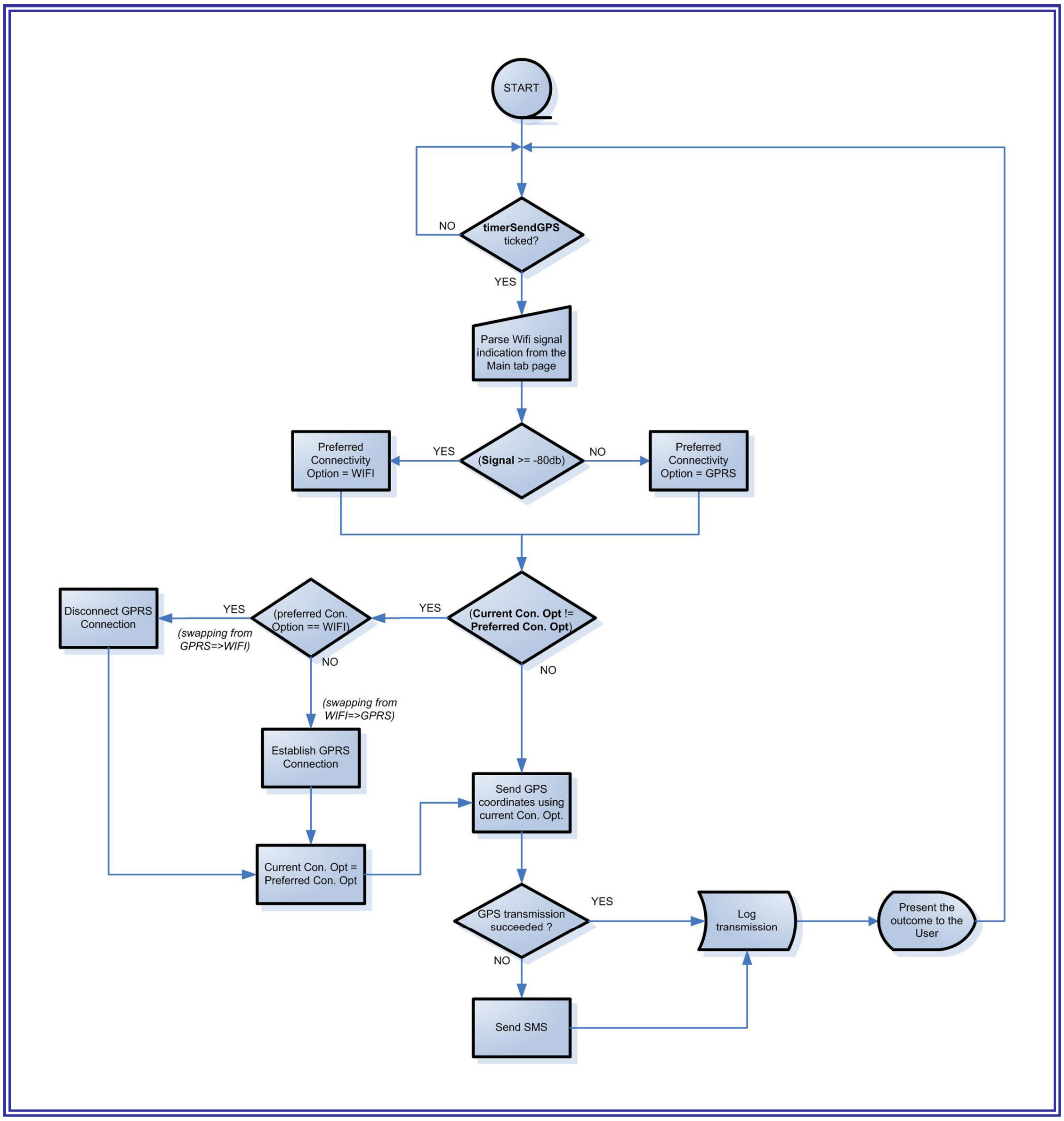


Figure 31 : Flow Chart of transmitting the GPS coordinates

The second timer is the **timerSendGPS** and is actually responsible for sending the GPS Coordinates. It is initialized when the user has started the communication, and is ticking every 15 seconds and sends the current GPS coordinates presented on the suitable field, to the server. Its function, described in the flow chart of Figure 31, is mainly done with the following actions in the described order :

1) **Parses the current WIFI signal strength** from the field of the Main tab page that has been updated from the timerWifiSignalStrength. If this value is higher than -80db then it means that the WIFI signal strength is considered to be at least good and then the preferred connectivity option for sending GPS coordinates is set to WIFI. If the WIFI signal value is lower than -80db or has not been set, then the WIFI coverage is either poor or none and then, the preferred connectivity option for sending GPS coordinates is set to GPRS.
2) If the preferred connectivity option is different than the one that is used this means **that the application should swap** to the other IP-based connectivity option. If the used connectivity option was WIFI and the application should

swap to GPRS, then it tries to establish a GPRS connection based on the reference that it has for that connection and swaps to it. If the used connectivity option was GPRS and the application should swap to WIFI, then the GPRS connection is disconnected[25] and the application swaps to WIFI.

3) The application tries to **send the GPS coordinates** with the use of the IPConnector object over the preferred connectivity option that has been previously identified. If this transmission fails, then the application swaps to the backup alternative option, that is SMS over GSM, and therefore tries to send the coordinates via SMS.

The aforementioned high-level described procedure is done every 15 seconds, which is the set interval for the application to send coordinates. Due to hardware and Operating System's restrictions that are going to be examined in Section 5.3.6 this interval might not be followed precisely, mainly in the cases of swapping from one connectivity option to another. For example, when the application swaps from WIFI to GPRS it has to wait for the GPRS connection to be fully established before it tries to send GPS coordinates and also it can not send SMS messages in the background basically due to hardware restrictions (explained in Section 5.3.6.2).

5.3.2 GPSConnector

The `GPSConnector` class is used to represent the connection with the GPS module either the external GPS device or the internal GPS module of the PDA is used. An object of the `GPSConnector` class is created on the `GUI` class so that the `GUI` class will have a reference to that GPS connection. The GPSConnector object is mainly responsible for continuously obtaining data from the GPS module and pass it as NMEA phrases to the object of the `NMEAInterpreter` class that has created. Consequently, the NMEAInterpreter object parses the data that are received and fires the appropriate events upon the validity of the data. These events and handled by EventHandlers of the GUI that are responsible for updating the GUI Form that is presented to the rescue worker and some global variables that are needed for the main functionality of the `GUI` class. The high-precision NMEAInterpreter implementation used is written by Jon Person (author of the "GPS.NET") and slightly modified by the author of this dissertation to make it more stable in our context. Moreover, the `GPSConnector` class was conceptually based on Marian Mohr's implementation.

In more detail, the GPSConnector object retrieves as an argument from the GUI object the GPS module that will be used, internal or external, and tries to initiate a connection with it. If the internal device is used, then by trying to poll the COM7 serial port of the device at 56kbps baud rate it will initiate the GPS integrated module of the device. If the external device is used a polling procedure on COM6 serial port will pop-up the Bluetooth console to initiate the connection with the Pretec GPS external device. When the serial communication is successfully established then an

[25] *Based on the software restrictions that the device has (presented on Section 5.3.6.1) the GPRS connection cannot remain active in the background but has to be completely disconnected so that the device would be able to obtain WIFI connectivity and "inform" the application that it could swap to that.*

NMEAInterpreter object is created and all the appropriate EventListeners are assigned to events that could be fired from the NMEAInterpreter object. The serial connection is passed as an argument to a new thread that will continuously retrieve data from that connection. When all the sanity checks for the retrieved data are successful then these are passed to the NMEAInterpreter object for parsing.

As a parenthesis, the NMEA format is the usual format used from GPS devices to report the data that they have received from the satellites. According to the NMEA protocol, there are 62 defined code phrases with an average of 15 arguments each, that include data from satellites varying from GPS almanac data, to wind direction, water speed and satellites' time [74]. Most of the GPS devices that are relatively cheep to purchase could interpret the 19 most useful and frequently used NMEA phrases that are presented in Table 6 [72, 73]. Table 6, also includes in bold the four significant NMEA phrases that the NMEAInterpreter object of our implementation could parse and are of use for the developed client's application.

TABLE 6 : Most frequent NMEA phrases used (adapted from [73])
-the NMEA phrases used in the application are marked with bold-

NMEA CODE	DESCRIPTION
$GPBOD	Bearing, origin to destination
$GPBWC	Bearing and distance to waypoint, great circle
$GPGGA	**Global Positioning System Fix Data**
$GPGLL	Geographic position, latitude / longitude
$GPGSA	**GPS DOP and active satellites**
$GPGSV	**GPS Satellites in view**
$GPHDT	Heading, True
$GPR00	List of waypoints in currently active route
$GPRMA	Recommended minimum specific Loran
$GPRMB	Recommended minimum navigation info
$GPRMC	**Recommended minimum specific GPS/Transit data**
$GPRTE	Routes
$GPTRF	Transit Fix Data
$GPSTN	Multiple Data ID
$GPVBW	Dual Ground / Water Speed
$GPVTG	Track made good and ground speed
$GPWPL	Waypoint location
$GPXTE	Cross
$GPZDA	Date & Time

When the NMEAInterpreter object parses a known phrase it fires the suitable events which trigger the appropriate EventHandlers. These EventHandlers are responsible for updating the `GUI` main class and global variables that are stored for the main functionality of the application.

5.3.3 IPConnector

The IPConnector object that the `GUI` class holds represents the IP based connectivity option that is used at a specific time. Since both WIFI and GPRS connectivity options are eventually implemented in IPv4, the server listens on one IPv4 address that the client is using to send packets either via WIFI or GPRS. Although the current implementation of the application uses only IPv4, the `IPConnector` class can facilitate both IPv6 and IPv4 TCP sockets.

Upon the creation of the object of the `IPConnector` class, a socket is opened with the server based on the IPaddress that the GUI class passes as argument to the constructor of the IPConnector class. Consequently, this object is used to send GPS coordinates to the server over the instantiated socket and is responsible for informing the `GUI` class if the coordinates have been sent successfully[26]. In addition, if there has been a failure in the transmission of the GPS coordinates, meaning that the socket was "broken", then this class uses a background thread to force reconnection with the server and the creation of a new socket for future use under a specific connectivity option. Finally, this class also informs the Main tab of the GUI Form for network related issues, such as the IP address of the client, failures in obtaining a socket or failures in the reconnection procedure.

5.3.4 Map

Although presenting a map to the rescue worker was not a main part of this dissertation it has been decided that it would be a useful feature. Presenting a map with a marker on a certain location based on the coordinates obtained from the satellites informs the rescue worker for his location in respect with a region. Moreover, the Map feature was found to be valuable during the testing phases of the project, because it enabled us to be aware of the exact point on the map that the GPS coordinates that are sent from the client to the server correspond to.

The Map functionality is added on the application using the Google Maps API v2.0 [75]. Basically, an html page that includes javascript is created and is placed on a specific webspace[27]. When the rescue worker opens the Map tab and taps the appropriate button (Figure 29), the client retrieves the current latitude and longitude and converts them into their decimal equivalent. This procedure is mandatory because the Google Maps API cannot parse latitude and longitude in the NMEA format. Consequently, a request is done on the described html file passing as arguments the latitude and longitude in decimal format along with the node ID of the client that makes the request. This request is performed over the current IP-based connectivity

---

[26] *The described implementation does not use acknowledgements in the application layer but can programmatically monitor the TCP socket that is used for the transmission of the coordinates. Therefore a boolean value is returned to the GUI whether the transmission was successful or not. Based on the fact that the TCP protocol is used, this indication was found to be accurate for the scope of this project.*

[27] *For the proof of concept of this project the html file that is using the Google Maps API can be found at : http://www.lancs.ac.uk/postgrad/georgopp/maps/indexcf.html*

option. The online file parses the arguments of the request and embeds a Map with the region that the coordinates correspond to. Moreover, a “balloon” marker is placed on the exact GPS coordinates that the client sent on his request. As a result a browser is loaded (Figure 32) on the client side presenting the map with the marker on the defined location of the rescue worker (Figure 33).

Figure 32 : Loading Minimo web browser

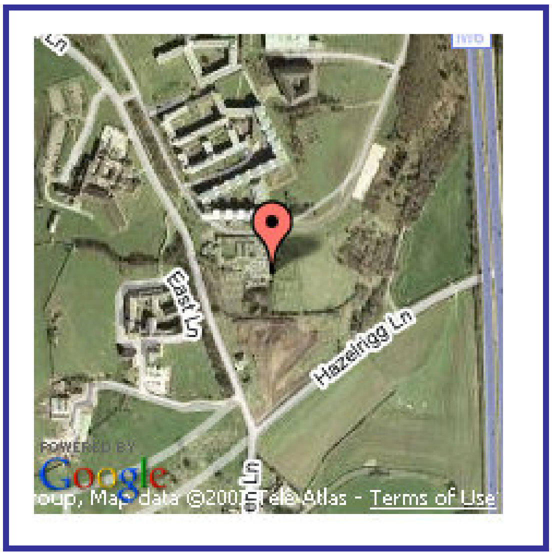


Figure 33 : The presented Map with the location of the rescue worker

Although the Map functionality can be used from the application that was created for the client, it cannot clearly be stated that it is an integrated part of it since it launches an external web browser that makes the request for the map. Unfortunately, the webBrowser container that the .NET Compact Framework provides and could be totally integrated on the developed application uses the default web browser of the Operating System, namely the Pocket Internet Explorer 3.0. This version of pocketIE supports a very limited set of javascript [76] that cannot provide the demanded functionality using Google Maps and thus the Minimo web browser, developed from Mozilla Corporation, was used. At this point it should be mentioned that Minimo, which is currently the only free web browser for mobile devices that fully supports javascript, is using a significant amount of the resources of the PDA, in terms of memory and CPU workload, and therefore it was implemented to display the location of the rescue worker only on demand and not to be automatically updated when new GPS coordinates are obtained.

5.3.5 Log

The client includes a powerful Log function that appends to a log file a lot of information regarding user input, data retrieved from the satellites and network status along with a timestamp. Moreover, this log file gets updated with information regarding the current connectivity option used, the payload of the packets sent, the IP address of the client, the WIFI signal strength at specified intervals and many more useful information.

The Log function has been implemented in such a way so that it could be examined offline and give important information for the connectivity options used and the network conditions that aided the decision at each time. The underlying idea for the Log feature of the application is to record the available connectivity options and network status at the specific coordinates that the rescue worker roams in the Mountain Rescue domain. Hopefully, this log file would provide a useful grounding for the availability of the connectivity options at specific regions and could aid the future work that could be done regarding the client's application or the WIFI network coverage that the team provides in the domain.

5.3.6 Main difficulties and restrictions

This section presents two main restrictions that the device presented, one in software and one in hardware, primarily affecting the implementation that was developed for the client's application.

5.3.6.1 Operating System's restriction

Regarding the software restriction, the initial design of the system included two different classes, namely the `WIFIConnector` and the `GPRSConnector` to provide connectivity for the WIFI and the GPRS connectivity options, respectively. The underlying idea was that each one of them would address the server over its own socket to the same IPaddress[28] and to different port. This way, when there was an indication of poor WIFI signal strength, the `GPRSConnector` would establish a GPRS connection, acquire a socket with the server and be on hot stand-by when the WIFI signal would be lost completely. Likewise, if the GPRS connectivity option was used and when sufficient WIFI signal would have been detected, the `WIFIConnector` would initially try to acquire the socket with the server and then instruct the GUI class to swap the transmissions of the GPS coordinates to it.

Unfortunately, from the application's point of view, Windows Mobile 5.0 and basically all the Microsoft's mobile Operating Systems that are based on the CE platform, is using two different network components that are included into the cellcore dynamic library of the Operating System to expose access to network resources [79]. The first component is the Connection Manager that, according to the Windows CE Networking Team Weblog [78], "manages all network resources" and claims to "provide access to different destination networks". The second component is the Connection Planer "that is responsible for choosing one or a set of connections

[28] *To be precise, the initial concept was that the WIFI would address the server to its IPv6 address, and the GPRS to its IPv4 address. Unfortunately, as Section 5.2.3 described, that concept was modified based on the fact that the wireless network of the campus could not provide IPv6.*

that will satisfy a connection request" [78]. To put it simply, in theory, if an application demands a connection to the Internet, the Connection Planer will evaluate all the predefined connection settings that the device has and instruct the Connection Manager to initiate a certain connection to facilitate the application's request.

Unfortunately, in practice there are times when the Connection Manager will not follow the Connection Planner's "proposal" based on some specific criteria. Quoting again from Microsoft Windows CE networking team's blog [78], Connection Manager might "restrict network resources to different connection requests from applications based on their priority and resources' security properties". For example, if both a WIFI and a GPRS connection are UP and the WIFI connection is used from an application, the Connection Manager will not swap to the GPRS connection even if the Connection Planner might instruct it to do, based on some defined criteria. One of these criteria is that a WIFI connection generally provides more bandwidth than a GPRS connection and thus the Connection Manager will not swap from WIFI to GPRS when the Connection Planner instructs it to do it. Moreover, there are occasions when the Connection Manager will not swap immediately from a GPRS connection to WIFI based on the fact that a GPRS connection is costly in terms of time while establishing the connection and therefore the Connection Manager waits for several seconds to swap from GPRS to WIFI.

To make things worse, the Connection Manager cannot support packets being sent and received from more than one active IP-based connection at the same time. To be precise, two or more IP-based connections could be "UP", for example being connected to both a WIFI network and have a GPRS connection established at the same time, but packets can be transmitted only over one of them. Therefore, our initial design for having two different classes (`WIFIConnector` and `GPRSConnector`), one active and one on hot-stand by, was replaced from the `IPConnector` class described in Section 5.3.3 due to the restrictions of the Connection Manager. As stated, there was no chance for having one socket transmitting packets and another one being connected and waiting on stand-by.

For our scenario in the Mountain Rescue Domain, and in order to be able to swap from WIFI to GPRS a hybrid implementation was used to access functions in the coredll.dll and cellcore.dll libraries. The implementation developed for the client's side totally overcomes the Connection Planner and is able to directly instruct the Connection Manager to which specific network it should connect[29]. Unfortunately, as described above the Connection Manager was not following immediately our instruction as it did not follow the instruction of the Connection Planner.

There are two apparent swapping procedures, namely swapping from WIFI to GPRS and swapping from GPRS to WIFI. In the first case, although the application is able to identify poor WIFI signal strength, establish a GPRS connection (handled as a dial up connection with the telecommunication provider) and instruct the Connection Manager to swap to it, the Connection Manager does not follow our instruction until it looses the WIFI signal completely. This functionality increases the possibility of loosing packets transmitted over a WIFI socket when in fact the WIFI signal is too

[29] *This was developed with the use of a library from OPENNETCF Corporation [70] and by platform invoking functions to native DLL files.*

low. In the second case, swapping from GPRS to WIFI, although a WIFI network might have been available, the Connection Manager could not follow our instruction to immediately swap to the WIFI network when the GPRS connection was active. To confront this inefficiency the following workaround was implemented. When the application needed to swap from GPRS to WIFI, a method was implemented to iterate through the RAS [30] connections and shuts down the GPRS manually, with the aid of the coredll.dll library, so that the Connection Manager will be "forced" to identify that there is no other valid connection apart from the connection to the WIFI network and eventually swap to that.

5.3.6.2 Hardware's restriction

The Ipaq 6915 is equipped with a GPRS class B module which has some restrictions in terms of connectivity [81]. Class B GPRS modules can be connected to GPRS and GSM service but are able to use only one at a given time [42]. When a request is done for one service, then the other one is temporarily suspended until the service is fully completed. Regarding our project, the described limitation presented significant restrictions in the two following cases.

When the application was able to identify that the WIFI signal strength was poor or zero, it initiated a GPRS connection so that it would swap the transmission of the GPS coordinates to that. However, the GPRS connection needs at least 10 seconds to be established and in addition, the application should firstly acquire a TCP socket and then swap to GPRS. Many tests during the development phase prove that if the client was sending SMS messages during the procedure of establishing a GPRS connection, the GPRS connection could not be set up at all probably because the GPRS module was suspended during the transmission of SMS messages. The workaround that was found in order to initiate the GPRS connection was to actually wait until the connection is set up and then start sending SMS messages. The impact of this workaround was that the application was not able to send GPS coordinates in fixed intervals but was –at least- making the GPRS connection available for future use.

The other case that the class B GPRS module is affecting is the acquisition of the TCP socket over the GPRS connection that has just been initialized. When the GPRS connection has been successfully initialized, and as there is no WIFI connectivity, the application will swap to SMS messages in order to be able to send GPS coordinates until the application will be able to acquire a TCP socket. Again, during the transmission of the SMS messages, the GPRS service is suspended and thus the GPRS socket might be more difficult to acquire or might timeout.

[30] *RAS stands for Remote Access Service.*

5.4 Server

The application that was developed for the server that is located at the headquarters of the Cockermouth Mountain Rescue Team was developed in Visual C# using the .NET framework v2.0 in Visual Studio 2005. The requirements of the server application mainly demanded that the server should be able to listen for packets sent from the clients regardless the connectivity option used. The server application consists of a multi-threaded implementation that listens to a global IPv4 address for packets sent either via WIFI or GPRS and has a Bluetooth connection with a mobile phone to listen for SMS messages (Figure 21). The remainder of this section will describe the GUI of the server and its main functionality (Section 5.4.1), the GSM module (Section 5.4.2), the Map functionality (Section 5.4.3) and its detailed log file (Section 5.4.4).

5.4.1 Main Form Class

The `MainForm` Class of the server is composed of two distinct logical and practical parts, similar to the ones that the client had, developed with the partial class implementation feature of Visual C#. The first one is the `MainForm.Designer.cs` partial class which holds all the GUI elements of the form that the rescue coordinator notices on the server application and handles all the interaction with the user. The second one is the partial `MainForm` class that includes the main functionality of the server. The two aforementioned parts are going to be described in the following sections.

5.4.1.1 GUI

Figure 34 presents the GUI of the server application that has just been initialized. The `MainForm.Designer` partial class includes a main central area presenting two tab pages, the Map and the Log, and also two panels on the right side, presenting the server status and information for the last packet received.

Starting from the tab pages of the main central area, the Map tab page presents a map and a marker on the exact location of the client that sent the last packet. If the marker is clicked then the exact longitude, latitude and the node ID of the client that sent the packet is presented on the screen (this is illustrated in Figure 34). The Log tab page (Figure 35) presents an extensive Log with information regarding all the functionalities of the server and a lot of information for all the retrieved packets. Basically, the rescue mission coordinator monitors the Map tab along with the two side panels that present all the information needed for a mission.

The panel that is located on the upper right side of the server application presents the status of the server regarding the three different connectivity options that are supported. Therefore, the mission coordinator can see which of them are enabled and the server currently monitors for packets. Appropriate buttons are also presented on the panel so that the rescue coordinator could turn ON and OFF the listening functions for each one of them. In addition, the server status panel presents a counter for each connectivity option so that the user could be aware of the number of packets received from each connectivity option. Furthermore, this panel presents additional information such as the IP address and port that the server is using for listening for IP packets and the serial port that the server uses to communicate with the GSM module, which for the proof of concept of this project, is a Nokia 6230i mobile phone.

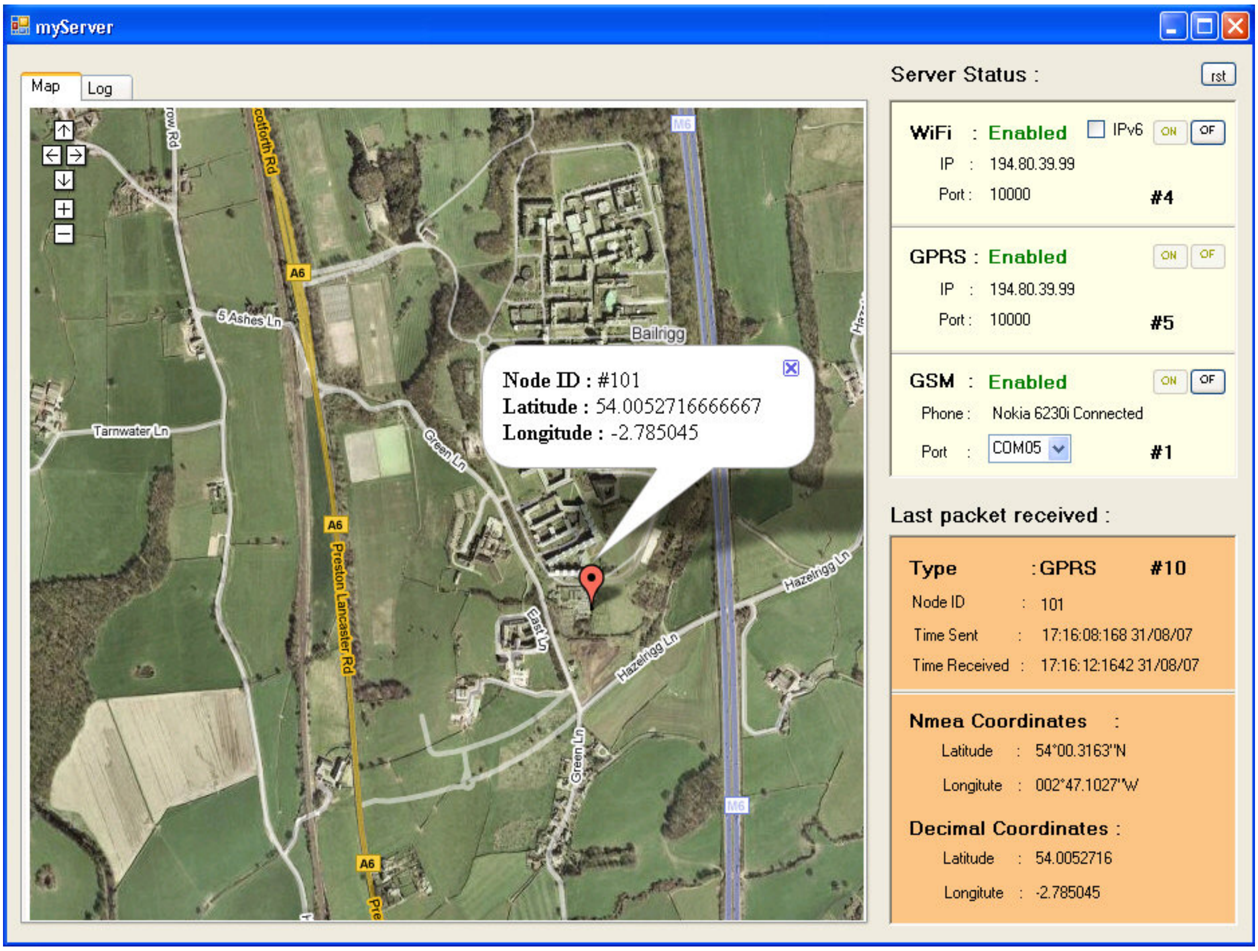


Figure 34 : The server application monitoring a "mission"

The panel that is located on the lower right corner of the GUI presents information for the last packet received. In detail, it displays the connectivity option that was used to sent the last retrieved packet, the node ID of the client that send it, the time that the packet was sent and the time that was received[31]. Furthermore, the coordinates of the packet that has just been received are presented in both the NMEA and decimal format. Finally, a counter on the upper right corner of this panel presents the total number of packets received.

[31] *At this point it should be mentioned that the time sent and received that are presented for the last packet received should not be considered as fully accurate as the time for the client and the server were synchronized manually.*

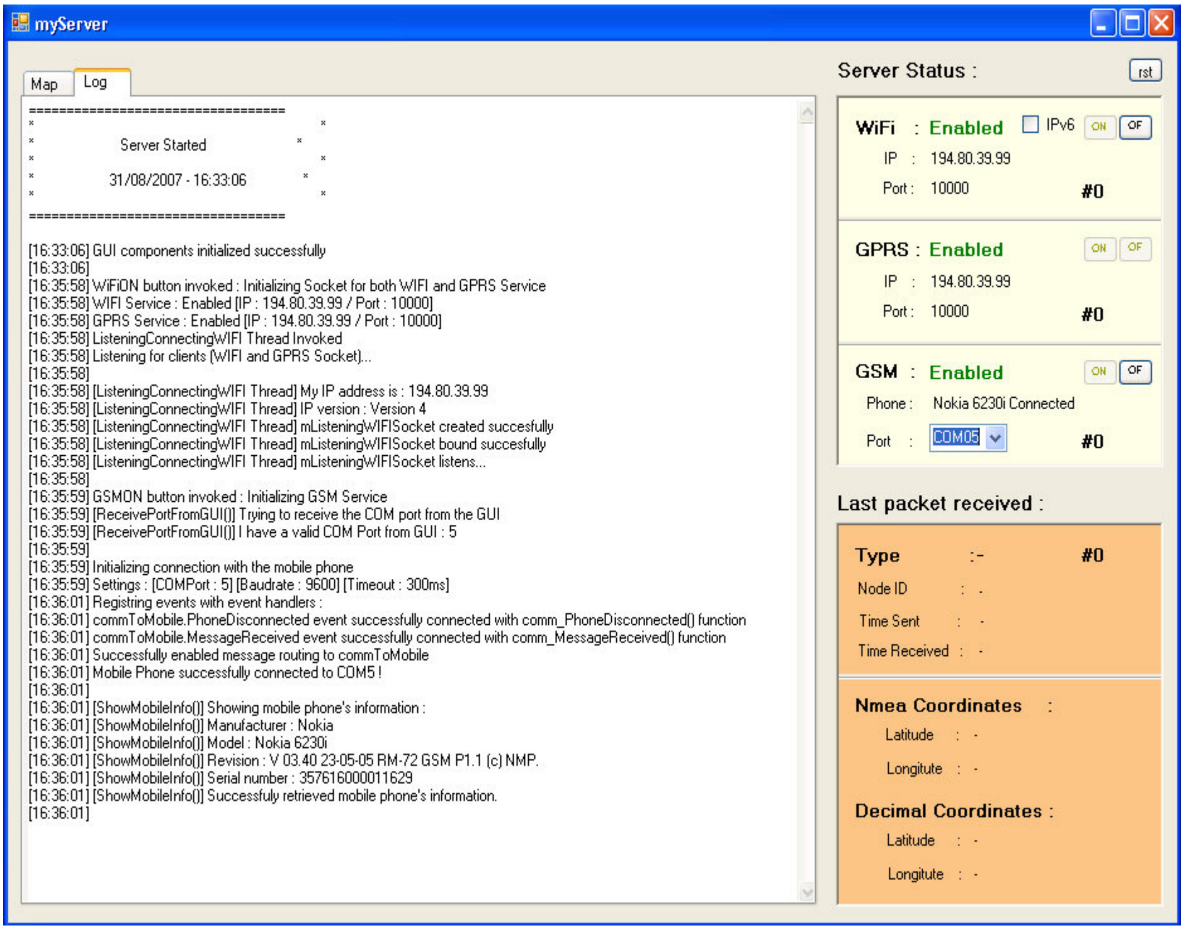


Figure 35 : The server application listening for packets and presenting the Log tab page

5.4.1.2 Main Functionality

The server application starts listening for packets sent from clients when the rescue worker coordinator enables the suitable listening functions. Therefore, when a user clicks on the “ON” button of the WIFI connectivity option (Figure 35) the server creates a socket bound to its global IPv4 address and listens for packets on port 10.000. This user’s action also enables the GPRS module as the server listens only to one IP address for packets sent using either IP based connectivity option. When the user enables the GSM module then a serial communication is initiated over Bluetooth with the Nokia 6230i mobile phone.

Regarding the implementation for listening to IP packets, the server uses a separate thread to listen for clients that try to acquire a TCP socket and initiate a connection. When such a request is identified it is passed over a different thread to facilitate the connection with that client and receive data over the instantiated socket. Therefore, the initial thread is able to listen for more clients’ requests. The thread that holds the socket with a client continuously retrieved packets from it and passes them to a specific method, entitled as ParseReceivedString(…), which parses the payload of the packets retrieved. This method is used for every received packet (even for SMS messages) as every message has a defined uniform format.

Initially, the ParseReceivedString(…) method splits the received payload into "words", based on the delimiter character "#", and performs numerous checks to be assured of the validity of the retrieved data. In brief the following actions are performed in the defined order :

1) Check if every word has a valid format based on the defined length and on the fact that some words should include some special characters. For example the longitude and latitude values should include the character "@".
2) Check if the node ID and the security code of the client that sent the packet are valid.
3) Check if the sequence number of the packet received is valid.
4) Identify the connectivity option that was used and update the suitable counter.
5) Update the panel that holds the information for the last retrieved data (node ID, time sent, time received, GPS coordinates in NMEA format)
6) Call a function to transform the GPS coordinates from NMEA to decimal format.
7) Check that the converted decimal coordinates correspond roughly to a position in the UK.
8) If the coordinates are not the same with the ones previously retrieved then a request is made to update the Map on the suitable tab page.

The above described procedure is done in every packet that is retrieved from a socket until the client gets disconnected. If an SMS message is retrieved (discussed thoroughly in the following Section) the ParseReceivedString(…) method is also called with the payload of the SMS as an argument. Generally, the server application continuously listens for incoming packets until its functions are disabled by clicking the OFF buttons for each functionality.

5.4.2 GSM Module

For the proof of concept of this project, the server is able to listen to SMS messages by monitoring the SMS messages retrieved on a Nokia mobile phone over a Bluetooth connection. This functionality is written with the aid of the GSMComm library written by Stefan Mayr [82].

Basically, and before the server initiates the GSM functionality, both the server and the Nokia Mobile phone should have turned on their Bluetooth adapters. When the user enables the GSM functionality on the server's application, a serial connection is created over Bluetooth between the mobile phone and the server. From this point on, the server application is able to query the mobile phone with AT commands addressed to its internal modem that is exposed over the Bluetooth interface.

When the Bluetooth connection is initiated the server application assigns a few EventListeners to listen for events that are fired based on the status of the mobile phone. Moreover, the server application enables a functionality that forwards every arrived SMS message from the mobile phone to the server and deletes the message from mobile phone. The most significant EventListener that is assigned for the GSM functionality is the comm_MessageReceived(…) that is fired when a new SMS message is forwarded to the server application. This method decodes the received

SMS to an SMSPdu unit and strips the actual data that the SMS carries (the text) from all the other information. Consequently, all the information regarding the sender of the SMS and the time sent are presented on the Log tab page, and the actual string is passed as an argument to the ParseReceivedString(…) method that is responsible to analyze the content of the message as was described in Section 5.4.1.2.

5.4.3 Map

The Map tab page, illustrated in Figure 34, is responsible for presenting a Map to the rescue mission coordinator and is automatically updated when new GPS coordinates arrive regardless the connectivity option that was used for their transmission. There are three distinct occasions were the server application does not automatically update the map, namely when the GPS coordinates are composed of zeros, when the GPS coordinates seem not to correspond to location points inside the UK[32] and when they have not changed from the previously received coordinates from the same client.

The Map tab page includes a webBrowser container that acts as the one that was described for the client side, with the only difference being that the functionality for the server side is completely integrated into the server application.

5.4.4 Log

The server application includes two different types of logging functionality, namely the Log tab page and a log file. Both logging features are almost identical with some exceptions when the application starts and closes because at that time the log file has to be treated differently. Every time the server application starts and closes appropriate entries with the date and time are written to both of them, and from this point on, every entry includes a timestamp.

The logging functionality includes extensive information for each action taken on the server's side and details for each packet received, the way that it was retrieved and its payload. Generally, the log feature is implemented in a way so that it would enable a study of the chosen connectivity option that each client used in each location in the Mountain Rescue Domain, hopefully, enabling better network support for the Mountain Rescue Team.

---

[32] *A rough decimal estimation for coordinates that correspond to location points inside the UK notices that the received latitude should be higher than 50 and lower than 60, and the received longitude should be higher than -6 and lower than 2. This rough estimation is used because there are times when the GPS unit on the client's side is giving erroneous information. The decision for making this sanity check on the server side and not on the client side, is based on the fact that the server application needs to receive data from the client in order to have an indication that the client is "alive" and does transmit data regardless of their validity.*

Chapter 6 : Evaluation

This chapter evaluates the designed and implemented prototype system that was developed to provide location awareness in a Mountain Rescue Domain. Section 6.1 sets the scene and the scenario that is going to be used for the evaluation of this project and Section 6.2 describes the actual evaluation process, refereed onwards as the Experiment. Section 6.3 analyzes the result of the Experiment and finally, Section 6.4 evaluates thoroughly the developed applications based on the initial requirements and the analyzed results from the conducted study.

6.1 Setting the Scene and the Scenario

The overall system that was designed and developed for this project is composed of three main parts, namely the client, the server and the communication framework focused on the Mountain Rescue region that the Cockermouth Mountain Rescue team operates in. The application that was developed for the client equipped it with the ability of obtaining GPS coordinates and identifying the best suited communication option based on their availability and prioritization defined in the communication framework. Moreover it enabled the client to utilize the defined connectivity option for transmitting the GPS coordinates to the server of the Mountain Rescue team located at the headquarters in Cockermouth. The application that was developed for the server was able to listen for messages regardless of the connectivity option that was used for their transmission.

In order to evaluate the designed system it was decided that a region with heterogeneity in the availability of the communication options defined in the communication framework, was required. Moreover, as the primary connectivity option is to use the WIFI network for the transmission of the GPS coordinates, an environment with heterogeneous WIFI coverage was also required. As the communication networking model developed from the Network Mobility group (described in Section 2.1.2) has not been fully deployed in the Mountain Rescue domain of our interest, and as such an evaluation would be completely unfeasible and unpractical to run, the campus of Lancaster University was chosen to be the suitable scene for the evaluation of this project. Figure 36 presents a map of the WIFI coverage of the Lancaster University's campus, proving that it is an ideal environment to evaluate the outcome of this project, as it provides heterogeneity in the WIFI coverage imitating the availability of the WIFI coverage in the Mountain Rescue Domain.

The devised scenario for evaluating the developed prototype of this project was to run the server application on a PC located in room A27 of InfoLab21 imitating the server of the Cockermouth Mountain Rescue Team. To imitate the rescue worker's mobility it was decided that we should roam across campus holding the PDA running the client's developed application and the external GPS device.

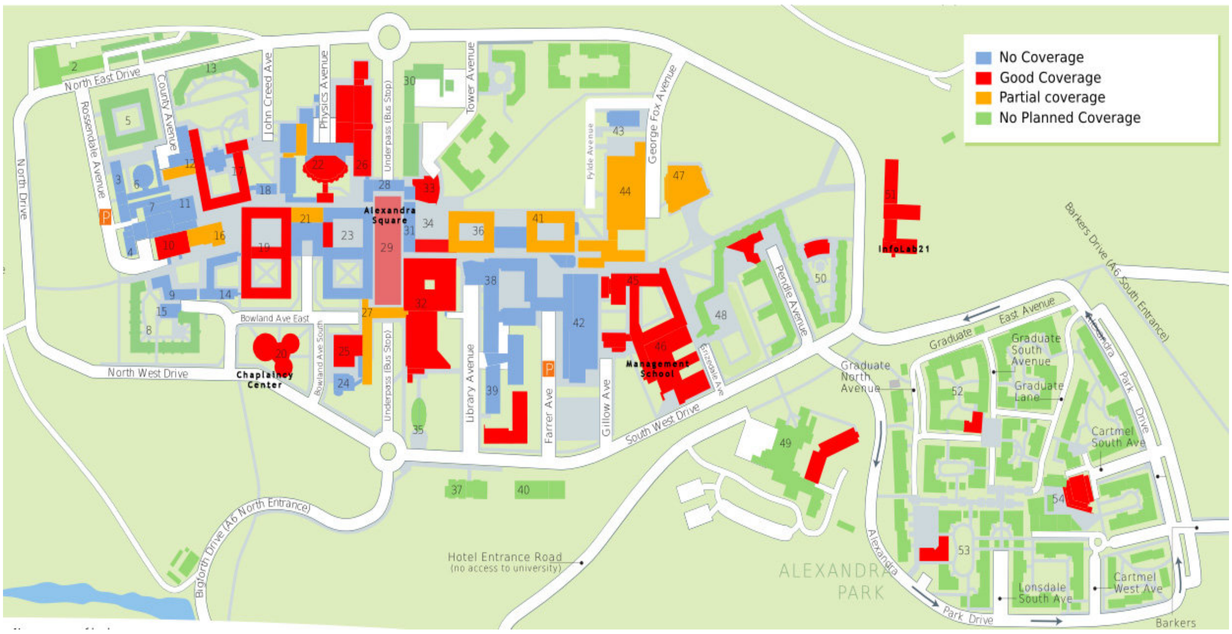


Figure 36 : Map of the Lancaster University's campus identifying the WIFI coverage

The wireless infrastructure of the campus' network provides three different wireless networks, namely the eduroam, the eduroam-web and the lancaster. The eduroam wireless network is considered to be secure using WPA network authentication and TKIP data encryption mechanisms. Once a client has been configured to use eduroam, then the login procedure is automatically handled by its Operating System. The other two available networks, eduroam-web and lancaster, are insecure and every time a client wants to utilize them it needs to login using a web authentication procedure.

Since our intention was to roam across campus it was decided to utilize the eduroam network so that we did not have to web authenticate the client every time we were in the vicinity of another access point. Moreover, an ISS network specialist informed us that by using eduroam the client will obtain the same IP address and will be able to roam with it across campus. Therefore, the Ipaq 6915 PDA that we had in our disposal was set up to be able to login to eduroam automatically using the credentials of the author of this dissertation. Unfortunately, as Windows Mobile 5.0 did not provide the suitable options for setting up the settings for eduroam, the SecureW2 program was used to set up the WIFI functionality for this network on the PDA [85].

6.2 The Experiment

In order to follow the above described scenario a specific route was defined illustrated in Figure 37. The server application was set up at a PC located at InfoLab21 and all the listening features were enabled. To be precise, the server was listening to its global IPv4 address 194.80.39.99 and port 10.000 for IP packets being sent either via WIFI or GPRS, and was also listening for SMS messages over its Bluetooth connection with the Nokia 6230i mobile phone.

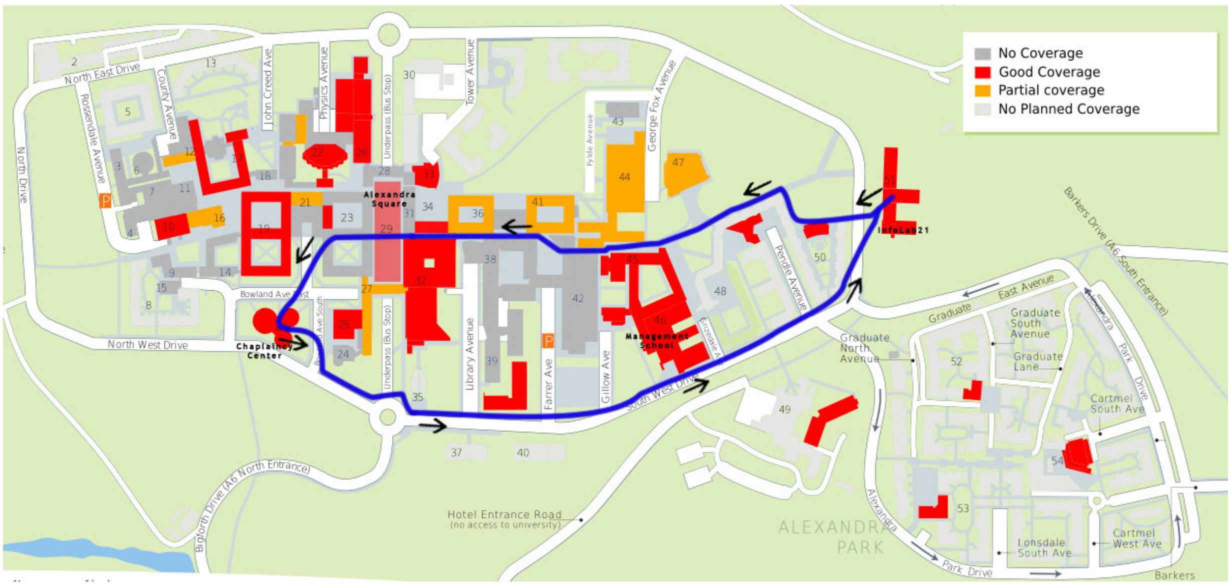


Figure 37 : ***Map of the Lancaster University's campus identifying the route and direction of our experiment***

On the client's side the application was executed and the GPS functionality was enabled using the external GPS device [33]. Consequently the communication functionality was also enabled and we decided to follow the specific route that is illustrated in Figure 37. Basically, our starting point was at Infolab21, in the room where the server was located, and we decided to walk along the spine till the Chaplaincy Center and then walk back to the InfoLab21 following the road that passes next to the Ruskin Library and the University's Hotel.

In brief, while we were still inside the Infolab21 the client was sending GPS coordinates via WIFI. While exiting the Infolab21, the application identified that the WIFI signal strength was poor and established a GPRS connection. While we were drewing away the InfoLab21 the application tried to acquire a TCP socket over GPRS and send three SMS messages as there was no WIFI coverage and as the TCP socket over GPRS was not initialized yet. When the application managed to acquire the TCP socket over GPRS it swapped to it and started sending GPS coordinates over it. From that point on, GPS coordinates were sent only via GPRS through the whole route until we reached back the InfoLab21. While entering InfoLab21 the application detected WIFI connectivity disconnected the GPRS connection and tried to acquire a TCP socket over WIFI. During this swapping procedure one SMS was sent and then the application started sending GPS coordinates over WIFI.

Figure 38 presents the status of the client's application just after the end of the experiment. As it is illustrated the client sent 9 packets using the WIFI connectivity option, 77 packets using the GPRS connectivity option and 4 SMS messages. Figure 38 also illustrates the last pair of GPS coordinates sent and the last WIFI signal strength indication before the end of the experiment.

---

[33] *Before our experiment, the external GPS device was turned on and left outside the InfoLab21 for a couple of minutes. This workaround was done in order to be able to initialize the communication with the satellites and "lock" the signals from the ones that were on its sight. With this procedure, the client was able to send valid (not zero) coordinates while we were inside Infolab21 and also was able to acquire live GPS coordinates really fast from our first steps outdoors.*

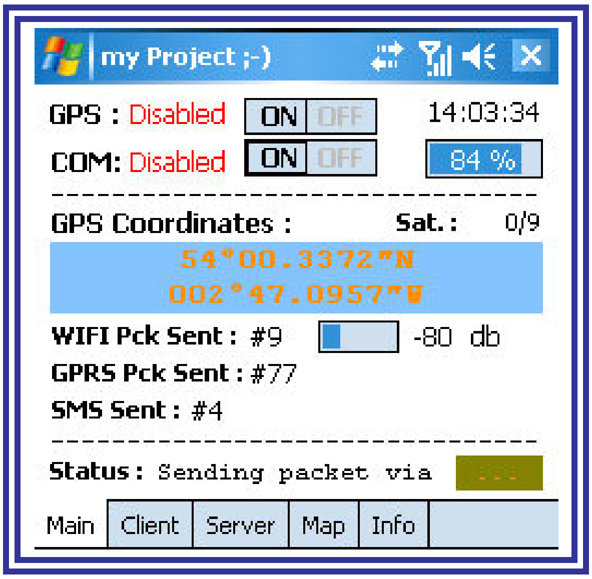


Figure 38 : Status of the Main tab-page of the client's application after the end of the experiment.

Figure 39 presents the status of the server's application after the end of the experiment. As it is illustrated the server identified 7 packets sent over WIFI, 77 packets sent over GPRS and 4 SMS messages retrieved during the experiment.

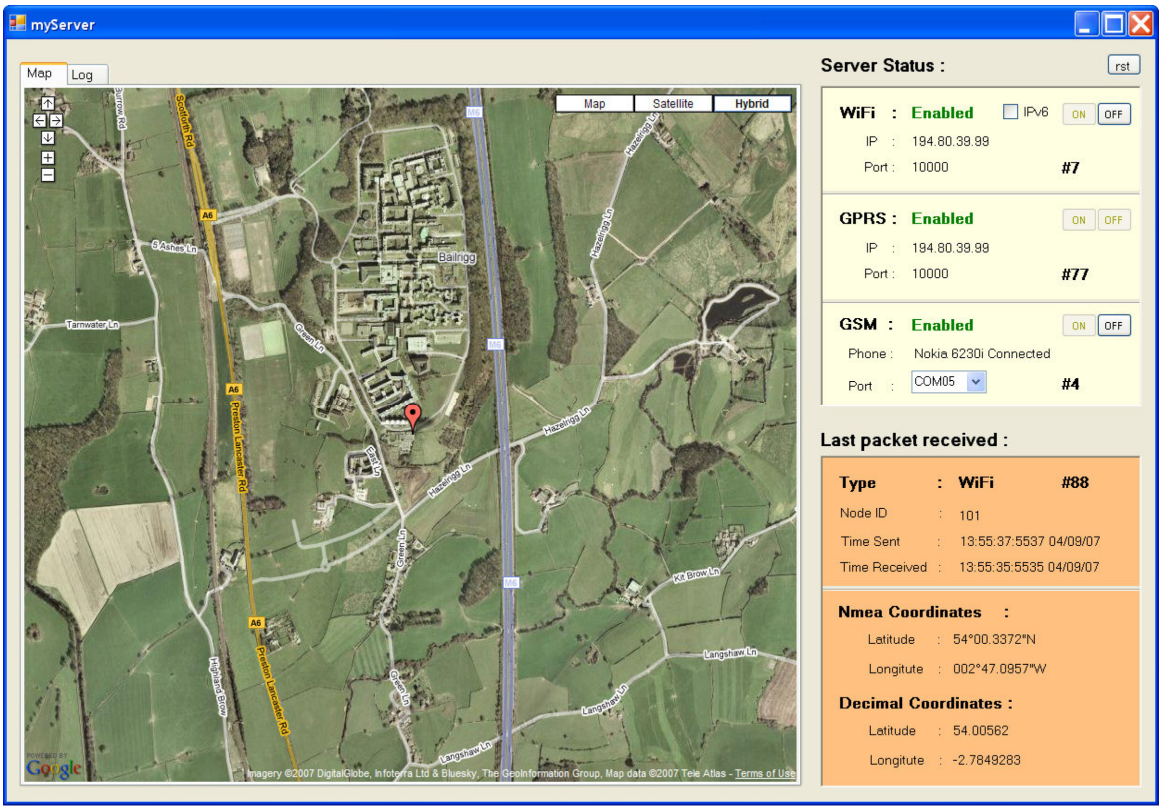


Figure 39 : Status of the server's application after the end of the experiment

The log files of both the client and the server during the described experiment could be found at the website that accompanies this dissertation[34]. In addition, since

[34] *The URL for the mentioned website is* *http://www.lancs.ac.uk/postgrad/georgopp/mscproject/*

the server application was recorded during the whole experiment, the aforementioned website includes a video with what a person sitting in front of the server's application could see during the conducted experiment.

6.3 Analyzing the Results

In order to evaluate the developed system against the defined requirements, the results of the described experiment should be analyzed thoroughly. The analysis presented below is the outcome of a thorough examination of the log files.

Regarding the main scope of this project, that is to be able to identify the best suited of the available communication options, the results presented on the previous Section illustrate that the client used all the connectivity options and swap from one to the other based on their prioritization and availability. Figure 40 graphically represents the location of the retrieved GPS coordinates and the connectivity option used for their transmission during our experiment.

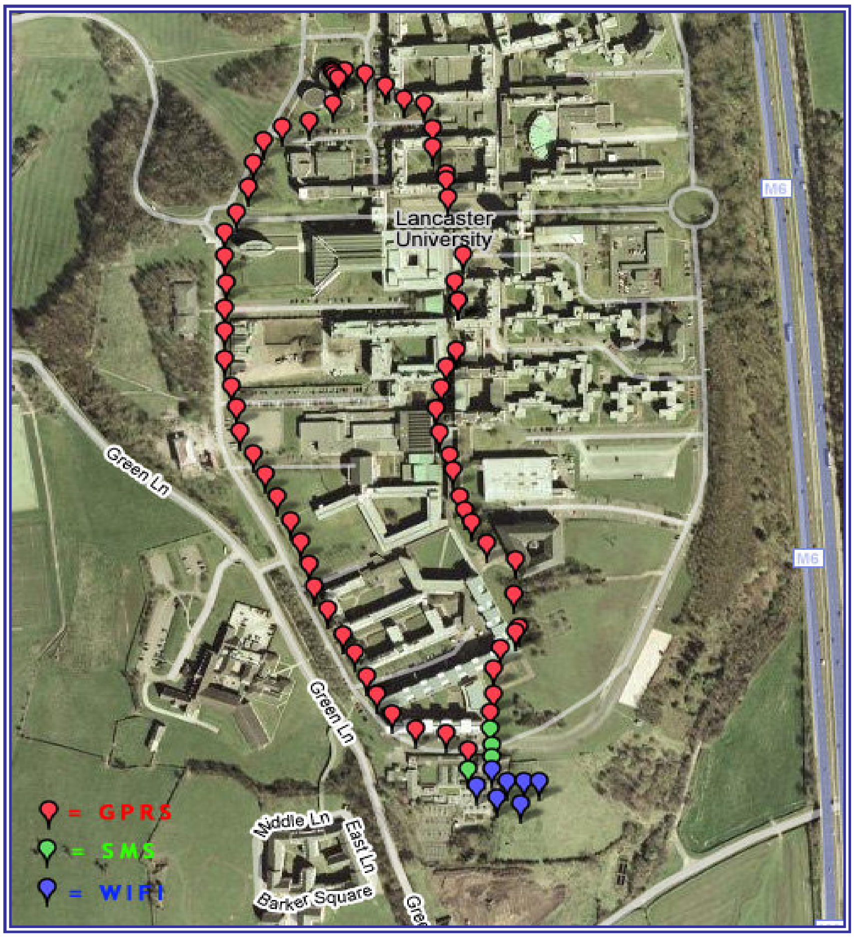


Figure 40 : Graphical representation of the GPS coordinates that were sent and the connectivity option that was used during our experiment[35]

[35] *Figure 40 was created using a modified implementation of the html (with javascript) file that the server and the client uses for presenting a Map to the user. The initial and final location points (around InfoLab21) are manually spread a bit so that they can be recognized.*

However, it was expected that the application would utilize the WIFI connectivity option a lot more during the experiment as there are specific areas along our route that present, based on Figure 37, good WIFI coverage. By carefully examining the log file of the client it was identified that our application did not get any indication from the Operating System that the device was connected to eduroam. By tracing more this indication we found out that eventually the device was not able to connect to eduroam at all during this route. Many similar experiments were conducted to identify the cause of this problem and it was found out that when the device was using WIFI and then was establishing a GPRS connection, the Operating System could not connect to the eduroam WIFI network from access points along campus even when our application was closed. Furthermore, the device was unable to connect even when the wireless adapter was restarted.

As the inability of the device to connect to the eduroam WIFI network along the defined route was not a problem of our application we conducted an ISS network specialist to investigate the issue and find out if there are similar issues reported with PDAs running Windows Mobile 5.0. After thorough investigation of the problem[36] and many repetitions of the experiment the following statements were identified :

1) The WIFI signal strength along the route was not sufficient for a device that has a small antenna and thus the PDA was “logically” unable to connect to eduroam.
2) The WIFI signal strength along the route was fluctuating a lot and thus the PDA was not able to perform the EAP authentication procedure in such an environment.

However, the above described statements can not explain why the device could not be able to connect to eduroam when we were inside the Chaplaincy Center where there is sufficient WIFI coverage. On the other hand, the PDA was able to connect again to eduroam when the experiment was about to finish and we have come back to Infolab21. Discussing this issue with the ISS network specialist he stated that there might be an issue with the DHCP[37] service, and to be precise, that the DHCP lease might not expire soon enough in order to “push” the device to establish a connection to eduroam using different access points. Though, such an issue, could not be examined as the DHCP service is used by thousands of users on the University’s campus.

Moreover, a research on the Internet identified that there are a lot of similar issues reported for PDAs that run Windows Mobile unable to connect to WIFI after having initialized a GPRS connection. In addition, the author of this dissertation contacted Peter Foot, a Microsoft Device Application Developer, who clearly stated that “having GPRS and WiFi connections concurrently is not a well supported scenario” in devices running Windows Mobile 5.0 or 6.0 [80]

---

[36] *We conducted the experiment again with the aid of the ISS network specialist holding a laptop and walking next to us holding the PDA. During this repetition it was found out that indeed the signal strength along the route was not sufficient in some occasions neither for the laptop nor for the PDA to connect to eduroam. However, there were times when the laptop was able to connect to eduroam but the PDA could not.*

[37] *DHCP stands for Dynamic Host Configuration Protocol and enables a client device to acquire a valid IP address from a server in a specific network [26].*

To sum up, the developed application was able to identify the best suitable connectivity option at any given time and send GPS coordinates via it. The fact that the Operating System was not able to connect to eduroam either due to poor signal or after initializing a GPRS connection is beyond the scope of this project as the thorough examination of the problem made us confident that there was not a problem in the application's side.

Another interesting point in the results of the conducted experiment is that the client application sent 9 packets over WIFI but the server managed to receive only 7 of them. By carefully examining the log of the server it was identified that the packets with sequence number 00005 and 00006 have not been received. The log snippet from the client that is presented below identifies that these packets "were sent" when we were exiting the InfoLab21. As can be noticed in the log the WIFI signal indication is really poor (-90db) and therefore the application identifies that it should swap from WIFI to GPRS, according to the Flow Chart presented in Figure 31. According to the implementation of this procedure our application instructs the Connection Manager to swap from WIFI to GPRS but as it can be noticed on the log the device continues transmitting packets over the WIFI network. This behavior was discussed in Section 5.3.6.1 (Operating System's restrictions) as, unfortunately, it is known that the Connection Manager operating component does not always follow the application's connection request.

```
[13:34:19] Found eduroam, IP:148.88.249.252, Signal:-90
[13:34:21] Found eduroam, IP:148.88.249.252, Signal:-90
[13:34:22] Timer Ticked - Sending Coordinates
[13:34:22] Current signal : -90
[13:34:22] Current way : WIFI
[13:34:22] I will send coordinates with : GPRS
[13:34:22] Current way is different from the one that i want to
use
[13:34:22] Changing networks
[13:34:34] I will send this : #WIFI#101#1234#00005#13:34:34:3434
04/09/07#54@00.3187"N#002@47.0920"W#normal#
[13:34:34] Sending packet via WIFI
[13:34:34] WIFI pcks : 5
[13:34:35] Found eduroam, IP:148.88.249.252, Signal:-90
[13:34:35] Found eduroam, IP:148.88.249.252, Signal:-90
[13:34:37] Timer Ticked - Sending Coordinates
[13:34:37] Current signal : -90
[13:34:37] Current way : WIFI
[13:34:37] I will send coordinates with : GPRS
[13:34:37] Current way is different from the one that i want to
use
[13:34:37] Changing networks
[13:34:37] I will send this : #WIFI#101#1234#00006#13:34:37:3437
04/09/07#54@00.3187"N#002@47.0920"W#normal#
[13:34:37] Sending packet via WIFI
[13:34:37] WIFI pcks : 6
[13:34:37] Found eduroam, IP:148.88.249.252, Signal:-90
[13:34:39] Found eduroam, IP:148.88.249.252, Signal:-90
```

Analyzing more the results of the conducted experiment, it could be stated that the application followed the 15 second interval among the transmission of the GPS coordinates when it did not have to swap from one connectivity option to another. In the case of swapping from WIFI to GPRS, there was a 45 seconds gap when the client was not transmitting any coordinates. This gap is explained from the fact that the application waits for the GPRS connection to be established and cannot send SMS messages on the background due to the hardware restriction described in Section 5.3.6.2. Moreover, after the initialization of the GPRS connection the application needs more time to acquire the TCP socket

Concluding the analysis of the results from the conducted experiment it can be stated that the developed application for both the client and server side behaved as expected. Although, the inability of the Operating System of the PDA did not provide the opportunity, during the described experiment, to evaluate how the application behaves during multiple demands of swapping from one connectivity option to another, tests in the vicinity of InfoLab21 show that the developed prototype is able to swap when the Operating System provides the means for doing so.

6.4 Evaluation against Requirements

This section evaluates the outcome of the project against the requirements that were defined and described in Chapter 3. Mainly, the requirements of this project were set regarding three distinct parts of the developed system; the application on the client side, the application on the server side and the envisaged communication framework that defined the connectivity options that could enable the transmission of GPS coordinates from the clients to the server.

At this point, it should be highlighted that the focus of this project and the overall aim of it was to equip a mobile device with the capability of sending GPS coordinates to a server via the best suited wireless communication option among the ones that could be applied in the Mountain Rescue domain. This aim was successfully achieved and an Ipaq 6915 PDA was equipped with the ability of identifying the available communication option (WIFI, GPRS or SMS) and utilizing it to send GPS coordinates to the server.

6.4.1 Communication Framework vs Requirements

The first part of this project was to theoretically and practically study and define a communication framework that could provide the grounding for the communication between a client and a server. This communication framework succeeded in defining three different connectivity options that could be applied on the Mountain Rescue Domain considering the transmission of GPS coordinates. The first connectivity option was to transfer GPS coordinates via IPv4 or IPv6 over a WIFI network, the second one was via IPv4 using GPRS over the GSM network and the third one was to transmit them as SMS messages via the GSM network.

The defined connectivity options were carefully identified, assessed and prioritized based on main criteria, namely their availability, cost to be purchased and ran, the extensibility that they can provide, their ability to follow technological evolution and their practicality. Although their thorough evaluation could only be done with studies on the Mountain Rescue Domain the author of this dissertation believes that the devised communication framework meets its requirements as it successfully provided the grounding for the application developed for the client. Moreover, the properties that were used to define the communication options included in the communication framework, are general enough to provide a promising framework that could be reused or extended for other network services in such a domain.

6.4.2 Client vs Requirements

The high-level requirement that was set for the application that was developed for the client side was met as the conducted experiment proved that the application was able to identify the best suited connectivity option and utilize it for transmitting GPS coordinates. This high level requirement was split into more detailed ones that should be evaluated one by one. Therefore, the detailed requirements for the application developed for the client side were :

- *It should be able to* ***acquire real-time GPS coordinates.***

□ The developed application was able to acquire real-time GPS coordinates either from an external GPS device or from the internal GPS module of the PDA. The conducted experiment proved the use of the external GPS device.

- *It should be able to* ***identify which of the predefined connectivity options are available****.*

□ The developed application was able to identify the availability of each connectivity options that was defined in the envisaged communication framework. The conducted experimented pointed out that there were times when the Operating System of the PDA used was unable to connect to the wireless network and therefore the application was correctly identifying that the WIFI connectivity option was not available and should not be used.

- *It should be able to* ***utilize the best suited connectivity option*** *and* ***seamlessly swap among them*** *based on some criteria.*

□ The developed application was able to identify and swap to the best suited connectivity option without any intervention from the user. An Operating System limitation regarding the Connection Manager networking component (described in Section 5.3.6.1), prevented the immediate swap from WIFI to GPRS unless the WIFI coverage was completely lost. Even at these occasions, the application was able to identify the loss and swap to another connectivity option after a while.

- *It should be able to* ***support IPv4 and IPv6****.*

□ The developed application was able to support both IPv4 and IPv6. The IPv6 support was fully designed and implemented but was not evaluated thoroughly in the testbed that was used for the evaluation of the project as the wireless network of the University's campus could not support IPv6.

- *It should be able to **transmit GPS coordinates in defined intervals**.*

□ The developed application was able to transmit GPS coordinates according to the defined 15 second intervals when a specific communication option was set. Due to a hardware limitation of the GPRS class B module (described in Section 5.3.6.2) the specified interval was not followed when the application had to swap from WIFI to GPRS as it had to wait for the GPRS connection to be established.

Moreover, the application that was developed for the client's side could be used without any configuration or experience needed from the rescue worker. In addition, it could present to the rescue worker many useful information, such as its current GPS coordinates, the communication option used and the number of packets sent from using each communication method. Finally, the client's application was able to present a Map to the user with its current location based on the information retrieved from the GPS module.

Concluding, the difficult task of developing an application for a mobile device with many hardware and software limitations, and equip it with the capability of sending GPS coordinates over different connectivity options was achieved successfully.

6.4.3 Server vs Requirements

As the main focus of this project was on the client's side the high level requirement for the server's application was to be able to just receive the GPS coordinates regardless of the connectivity option that the client used for their transmission. This high level requirement was split into the following detailed requirements :

- *It should be able to **listen for GPS coordinates regardless the connectivity option** that the client used to sent them*

□ The experiment that we conducted proved that the developed application for the server was able to listen for GPS coordinates regardless of the connectivity option used for their transmission.

- *It should be able to **identify the exact node** that sent a packet and the **exact connectivity option** used*

□ The application developed for the server was successfully identifying the exact connectivity option used for each received packet and also the node ID of the client that send it. Moreover, the application was able to analyze the content of each packet and present the suitable information to the Mountain Rescue coordinator.

- *It should be able to **identify packets received in the wrong order**.*

□ The application developed for the server was able to identify packets received in the wrong order. Although such an incident did not occur during the presented experiment, it was identified in a few of the numerous experiments that we did and was successfully confronted from the server. Interestingly, there are occasions when even SMS messages arrive in wrong order.

The application for the server side has successfully satisfied the above requirements and was also able to present, in a nice way, all the retrieved information to the Mountain Rescue coordinator. The coordinator, located at the headquarters of the team, is able to see a map that gets updated every time a message is received and could also be informed of the node ID of the client that send it. Moreover, it could be informed of the connectivity option used, the overall number of messages received, the time that a packet was sent and many more information regarding the status of the server and the last message received.

Judging from the conducted experiment it can be concluded that the devised system has successfully met its main aim, as GPS coordinates were transferred from a client roaming in an environment with heterogeneity in the provided connectivity options. Moreover, the each part of the devised system has also successfully met all its requirements. Although the developed system is just a prototype, and as such is evaluated, it provides a promising grounding for future work in this domain.

Chapter 7 : Conclusion

This chapter concludes the work that was developed for this MSc project. Section 7.1 suggests future work and improvements that could be done to the developed prototype system and Section 7.2 performs an overall evaluation and describes the milestones presented from this project.

7.1 Future Work and Improvements

The system that was developed for supporting the concept of location awareness in a Mountain Rescue domain is the outcome of an MSc project with a strict time limit. Future work could be done to improve the envisaged system and add functionalities to both the client's and the server's application. Moreover, the communication framework that was presented could also be thoroughly evaluated, extended and studied for improvements.

The author of this dissertation identified nine distinct theoretical and practical points that could be carried out for improving this project, based on the fact that the main focus of it was the application on the client side and its ability to swap and utilize different connectivity options. Namely, the main proposed improvements for future work are :

- ***Enabling communication among mobile clients***

□ The application developed for the client's side has the ability to send GPS coordinates only to the server of the Mountain Rescue Team located at Cockermouth. Regarding the concept of location awareness it could be a benefit if the packets that each client was sending could be received and used from other clients in the same region next to it. To be specific, if a client was sending its GPS coordinates to an IPv6 multicast address then all the other mobile nodes which have been subscribed to listen for packets addressed to this multicast address would be able to receive the packet and identify the exact location of a roaming node in a region. This process would inform a rescue worker about the location of the other members of the team during a mission.

- ***Confront the possibility of having no available connectivity option***

□ The application developed for the client's side currently does not provide any support if there is no available connectivity option, in the case of the client not being "covered" from the WIFI hotspot of the mobile router and not detecting signal from the GSM network in order to utilize GPRS or SMS. There are two suggestions that could be used to improvement this concept on the client's side. If there is communication among the mobile clients (described above) then a client could send its GPS coordinates to a "visible" client next to it which could utilize one of the defined connectivity options, in order to forward the initial packet to the server. Another suggestion for providing support to a client which has no available connectivity option for a specific period, could be to store its GPS coordinates on the specified intervals while it is roaming and send them in a fast pace when it regains connectivity with one connectivity option. Apparently, further studies should be done on this suggestion because when a client has no connectivity option for a long time it will store a lot of GPS coordinates and it would not be efficient for sending all of them if, for example, it detects that it can utilize the SMS connectivity option. In such an occasion, maybe the client could be able to send boarder coordinates which define a specific region that the rescue worker covered when the client could not utilize any connectivity option. Though, such an issue seems not to be apparent if the client regains connectivity with an IP based connectivity option.

- ***Further study of the networking protocol used***

□ The application developed for the client's side utilizes the TCP network protocol for the transmission of GPS coordinates over the defined IP-based connectivity options, based on its advantages described in Section 5.2.2. Although TCP was found reliable and efficient in terms of sending coordinates over a Mountain Rescue domain, it introduces complexity when the client tries to acquire a TCP socket in order to utilize it over one of the IP-based connectivity options. Since the transmitted data are not that critical[38], a study of how UDP behaves in a domain with no fixed network infrastructure could provide useful results and might eliminate the complexity on the client side when swapping from one connectivity option to another.

- ***Identifying and confronting routing inefficiencies on the application layer***

□ The unpredictable and constant mobility of the search parties and their leaf-nodes along with the physical characteristics of the Mountain Rescue domain introduce a lot of inefficiencies in terms of providing network coverage and routing packets to a destination. Although many mechanisms have been developed to confront these inefficiencies in the network and transport layer of the TCP/IP stack, their feasibility and practicality in such a domain is still in doubt. Future work could be done on the application layer to equip the developed application for the client's side to confront inefficiencies that the network could not inherently and efficiently tackle.

- ***Better GPS support***

□ During the development and testing of the application for the client, the GPS coordinates obtained from the satellites were found to be interestingly accurate with an average error of ±5 meters. The GPS coordinates that were transmitted and recorded during our experiment (displayed in Figure 39) practically prove the accuracy of the obtained GPS coordinates when the GPS module has locked signal from satellites within its sight. In these occasions the notion of location awareness is sufficiently provided to the person monitoring the server side. However, roaming on the actual Mountain Rescue domain may introduce a significant error on the obtained GPS coordinates that the client transmit and therefore further studies should be done to confront this concept. One possible solution to complement and provide backup to the GPS location capability would be to make use of the Location Based Services [49] from the GSM module of the client, in order to identify its location and maybe

[38] *No one could object that sending GPS coordinates is far less critical than, for example, withdrawing money via e-banking where each packet should be reliably sent.*

compensate it with the one obtained from the GPS module. Another alternative solution for the client which might not be able to obtain GPS coordinates, would be to request GPS coordinates from a client roaming in the same region so that it could -at least- have a relative indication of its position. As becomes apparent, the later aforementioned alternative solution for improving the GPS support could be used if the clients are able to communicate with each other.

- ***Security***

□ The application that the client runs includes a tiny notion of security by using a special security code for each different client. An improvement on the designed system could be if the communication framework itself could improve the notion of security on the packets being sent. Although the concept of supporting a Mountain Rescue team does not call for security features such as confidentiality or non-repudiation, features to improve the authentication of the clients and the integrity of the transmitted data could be useful. Moreover, IPv6 has inherent security features which would definitively increase the security in the transmitted packets.

- ***Using acknowledgments in the application layer***

□ The current design of the system does not require any response from the server side regarding the packets that it receives from the client. Although the current model was found to be effective, as only a 2.7% of the transmitted packets were lost in our experiment, further studies should be done in the actual Mountain Rescue domain in order to prove if there is a demand for providing a bidirectional communication among the clients and the server.

- ***Server demands position of a client***

□ An interesting improvement of the project could be if the Mountain Rescue coordinator could obtain on demand the location of a specific client. For example, if the Mountain Rescue coordinator knows the exact node IDs of the devices that the leaders of each search party carry, he could request their location on demand, regardless of the specified interval that the devices use for sending GPS coordinates. This feature would enable the Mountain Rescue coordinator to make more informed decisions regarding his organizational commands during a mission.

- ***Improve the interval used among the connectivity options***

□ The current implementation of the client's application tries to send GPS coordinates every 15 seconds regardless of the connectivity option used. As there are no real statistical data of the connectivity options that are provided in the specified Mountain Rescue domain, this interval was defined as a reasonable time period that the client device should wait before sending new GPS coordinates. As becomes apparent further studies should be done on this issue to identify how efficient this interval is, regarding the updates that the Mountain Rescue coordinator would like to see and the cost to support a real mission with many clients. The applications developed for both the client and the server provide powerful log functionalities that can enable such a study in a real Mountain Rescue mission.

Apart from the main points described above there are a lot more improvements that could be done to extend the functionalities of the system and improve its efficiency. For example, the system does not include a mechanism for synchronizing the time used from the clients and the server and therefore the recorded timestamps on the server side cannot provide such grounding for further studies on the average of seconds that a packet needs to reach the server. Another improvement could be if the server application would be able to draw the route of the client instead of just pinpointing its last known position.

A numerous of more improvements could also be carried out but as becomes apparent, an MSc time limited project had the intention to implement a prototype for the proof of concept that the topic required.

7.2 Overall evaluation

The general aim of this project was to be able to support the concept of location awareness in the Mountain Rescue domain that the Cockermouth Mountain Rescue team operates in. It was decided that this aim could be achieved by equipping the mobile device that the rescue worker would carry with the capability of obtaining GPRS coordinates and transmitting them to the server located at the headquarters of the team.

The physical characteristics of the Mountain Rescue domain and the unpredictable movement of the rescue workers roaming in the domain during a mission, prevent the deployment of a fixed network infrastructure that could provide network services that could eventually transfer the GPS coordinates from the clients to the server. Therefore, it was required from the author of this dissertation to provide a communication framework which should include the viable connectivity options that seem to apply to this Mountain Rescue domain, regarding the specific scope of transmitting GPS coordinates. Consequently, an application was required to be developed on a mobile device (a PDA) which should be able to identify the availability of the defined connectivity options and utilize them based on their prioritization for transmitting the GPS coordinates. Finally, an application for the server was also required to be able to listen for the GPS coordinates that the clients sent regardless of the connectivity option used.

As presented throughout this dissertation, a communication framework was defined, identifying, assessing and prioritizing the connectivity options that could be used on this domain. The three defined connectivity options for the transmission of the GPS coordinates are ; using IPv4 or IPv6 via a WIFI network, using IPv4 via GPRS over the GSM network and using SMS messages over the GSM network. The developed application for the client side was successful in being able to identify their availability and utilize them for the transmission of the coordinates. Finally, the application developed for the server side was also successful in listening for incoming messages regardless of the mean that was used for their transmission.

The theoretical and practical evaluation that was conducted proved that the developed prototype for the proof of concept of this MSc dissertation was working according to the defined requirements. Furthermore, many additional features were also developed to aid the notion of location awareness by graphically representing the location of the client in both the server and the client side, although such features were not required from this project.

In conclusion, the envisaged and developed system for this project proved the required proof of concept of transmitting GPS coordinates from a mobile device over the best suited connectivity option to a server, and successfully met all the defined requirements.

References

[1] G. Bell. "***Laws Of Predictions***". Presentation. Available at : http://research.microsoft.com/acm97/gb/tsld010.htm

[2] G. Bell. "***Laws Of Predictions***". Presentation. Available at : https://research.microsoft.com/acm97/gb/sld047.htm

[3] Definition of ***GPS*** on Wikipedia. Available at : http://en.wikipedia.org/wiki/Global_Positioning_System

[4] **Network Mobility Group** of Lancaster University's Website. Available at : www.network-mobility.org

[5] **Cockermouth Mountain Rescue Team**'s Website. Available at : http://www.cockermouthmrt.org.uk

[6] B. McCarthy, C. Edwards and M. Dunmore. "***Applying Network Mobility Techniques to a Mountain Rescue Domain***".

[7] Easydevices.co.uk Corporation. "***Buy GPS Mobile Phone Products***". Available at: http://www.easydevices.co.uk/pcp/Mobile_Phone.html

[8] U2010 Project. "***Ubiquitous IP-Centric Government & Enterprise Next Generation Networks***". Website : http://www.u-2010.eu/index.php?id=11

[9] RUNES Project. "***Reconfigurable Ubiquitous Networked Embedded Systems***". Website : http://www.ist-runes.org/video_tutorials.html

[10] 6NET Project. "***Large-Scale International IPv6 Pilot Network***". Website : http://www.6net.org/

[11] Definition of **MANNET** on Wikipedia. Available at : http://en.wikipedia.org/wiki/Mobile_ad-hoc_network

[12] F. Peyret, D. Betaille and G. Hitzy. "***High-precision application of GPS in the field of real-time equipment positioning***". May 2000.

[13] B.McCarthy, C. Edwards and M. Dunmore. "***Network Transparency in a Mountain Rescue Domain***". To appear at TERENA Networking Conference 2007.

[14] B.McCarthy, C. Edwards and M. Dunmore. "***The Integration of Ad-hoc MANET) and Mobile Networking (NEMO): Principles to Support Rescue Team Communication***". ICMU Conference 2006 Proceedings (Pages 284 – 290), London, UK, October 2006.

[15] **Network Mobility IETF Charter**. Website :
http://www.ietf.org/html.charters/nemo-charter.html

[16] P. Thubert et al. "***Nemo Basic Support Protocol***". IETF RFC 3693, January 2005.

[17] B.McCarthy, C. Edwards and M. Dunmore. " ***Applying NEMO to a Mountain Rescue Domain***".

[18] Definition of ***IPv4*** on Wikipedia. Available at :
http://en.wikipedia.org/wiki/IPv4

[19] 6DISS Project. "***IPv6DISSemmination and Exploitation***". Website at :
http://www.6diss.org/e-learning/

[20] Definition of ***IPv6*** on Wikipedia. Available at :
http://en.wikipedia.org/wiki/Ipv6

[21] J. Davies. "***Webcast : Introduction to Internet Protocol Version 6***". Available at
http://support.microsoft.com/servicedesks/webcasts/seminar/shared/asp/view.asp?url=/servicedesks/webcasts/en/WC051806/manifest.xml

[22] S. Deering and R. Hinden. "***Internet Protocol, Version 6 (IPv6) Specification***. RFC2460, December 1998.

[23] Definition of ***IPSec*** on Wikipedia. Available at :
http://en.wikipedia.org/wiki/IPsec

[24] R. Atkinson. "***Security Architecture for the Internet Protocol***". RFC 1825, August 1995.

[25] Garmin Corporation. "***What is GPS?***". Available at :
http://www8.garmin.com/aboutGPS/

[26] Definition of ***DHCP*** on Wikipedia. Available at :
http://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

[27] NMEA 2000 Standards Committee. "***NMEA 2000 Standard***". Available at :
http://nmea.org/pub/2000/index.html

[28] Definition of ***WGS84*** on Wikipedia. Available at :
http://en.wikipedia.org/wiki/WGS84

[29] Department of Defense (DoD) and Department of Transportation (DoT). "***USNO NAVSTAR Global Positioning System***". Available at :
http://tycho.usno.navy.mil/gpsinfo.html

[30] Garmin Corporation. "***GPS GUIDE for beginners***". Available at :
http://www8.garmin.com/manuals/GPSGuideforBeginners_Manual.pdf

[31] Definition of ***Trilateraton*** on Wikipedia. Available at : http://en.wikipedia.org/wiki/Trilateration

[32] J. Person. "***Writing Your Own GPS Applications***". Available at : http://www.developerfusion.co.uk/show/4652/2/

[33] Definition of ***Differential Global Positioning System*** on Wikipedia. Available at http://en.wikipedia.org/wiki/DGPS

[34] M. Dunmore. "***An IPv6 Deployment Guide***". 6NET Consortium, September 2005.

[35] Microsoft Corporation. "***Introduction to IP Version 6***". Published: September 2003, Updated: January 2007.

[36] R. Hinden, S. Deering and E. Nordmark. "***IPv6 Global Unicast Address Format***". RFC 3587, August 2003.

[37] C. Huitema and B. Carpenter. "***Deprecating Site Local Addresses***". RFC 3879, September 2004.

[38] P. Srisuresh and K. Egevang. "***Traditional IP Network Address Translator (Traditional NAT)***". RFC 3020, January 2001.

[39] R. Droms. "***Dynamic Host Configuration Protocol***". RFC 2131, March 1997.

[40] V. Fuller and T. Li. "***Classless Inter-domain Routing (CIDR) : The Internet Address Assigment and Aggregation Plan***". RFC 4632, August 2006

[41] Microsoft Corporation. "***Understanding Mobile IPv6***". Published: April 2004, Updated: January 2007.

[42] C. Perkins. "***IP Mobility Support for IPv4***". RFC 3344, August 2002.

[43] Definition of ***GPRS*** on Wikipedia. Available at : http://en.wikipedia.org/wiki/GPRS

[44] Definition of **EDGE** on Wikipedia. Available at : http://en.wikipedia.org/wiki/Enhanced_Data_Rates_for_GSM_Evolution

[45] Definition of **UMTS** on Wikipedia. Available at : http://en.wikipedia.org/wiki/UMTS-TDD

[46] Ordnance Survey Corporation. "***GSM coverage in Great Britain***". Available at : http://www.ordnancesurvey.co.uk/oswebsite/business/sectors/wireless/gsmcoverage.html

[47] Orange Corporation. "***Orange – Coverage and International Calling – UK Coverage***". Available at : http://coverage.orange.co.uk/uk/UKCoverageSearch.htm

[48] BT Corporation. "***BT Coverage – UK***". Available at : http://217.32.166.65/mapviewer/BTMobileBusiness/btMobileHome.jsp

[49] Definition of ***LBS*** on Wikipedia. Available at : http://en.wikipedia.org/wiki/Location-based_service

[50] HP Corporation. "***Additional Product Information***". Available at : http://h20000.www2.hp.com/bc/docs/support/SupportManual/c00713094/c00713094.pdf

[51] eXpansys UK Ltd. "***HP IPAQ 6300 Series Review***". Available at: http://www.globalpositioningsystems.co.uk/hp-ipaq-pocket-pc-h6365-in-depth.html

[52] Mtekk Corporation. "***Take a closer look at the IPAQ 6300***". Available at : http://www.mtekk.com.au/tabid/54/articleType/ArticleView/articleId/357/default.aspx

[53] Pretec Corporation. "***Image of Pretec external GPS device***". Available at : http://ec1.images-amazon.com/images/I/41V6Q3B17HL._SS500_.jpg

[54] HP Corporation. "***Image of HP Ipaq 6915***". Available at : http://h10003.www1.hp.com/digmedialib/prodimg/lowres/c00589398.jpg

[55]D. Fitton. "***Java Support on Pocket PC***". Available at : http://www.comp.lancs.ac.uk/~fittond/ppcjava.html

[56] Sun Corporation. "***Java ME Technology***". Available at : http://java.sun.com/javame/technology/index.jsp

[57] Nokia Corporation."***Nokia tools and Nokia SDKs for developers***". Available at: http://www.forum.nokia.com/main/resources/tools_and_sdks/listings/index.html

[58] Sun Corporation. "***Java ME – Java Technology for the Wireless Industry (JTWI), JSR185***". Available at : http://java.sun.com/products/jtwi/

[59] S. Berka. "***Java for PocketPC PDA's***". Available at : http://www.berka.name/stan/jvm-ppc/java_for_pda.html

[60] Microsoft Corporation. "***.NET Compact Framework***". Available at : http://msdn2.microsoft.com/en-gb/netframework/Aa497273.aspx

[61] Microsoft Corporation. "***.NET Framework***". Available at : http://msdn2.microsoft.com/en-gb/netframework/default.aspx

[62] Advanced Wireless Planet. "***GPRS module with RS232***". Available at : http://www.gsm-modem.de/gprs-module.html

[63] Falcom Wireless Communications Corporation. "***Falcom C2D-SI-GPRS***". Available at : http://www.falcom.de/?id=821&cat=4&pr=41#

[64] Nokia Corporation. "***Nokia 12i GSM module. Let your machines talk***". Available at : http://www.d-d-s.nl/fotos-nokia/n12i_datasheet_a4_v2.pdf

[65] Lay Networks Corporation."***Comparative analysis – TCP – UDP***". Available at : http://www.laynetworks.com/Comparative%20analysis_TCP%20Vs%20UDP.htm

[66] E. Rodriquez. "***TCP vs UDP***". Available at : http://www.skullbox.net/tcpudp.php

[67] V. Gupta, S. V. Krishnamurthy and M. Faloutsos. "***Improving the performance of TCP in the presence of interacting UDP flows in ad hoc networks***". Department of Computer Science and Engineering, University of California.

[68] M. Mohammadpour and W. Shandwick. "***Microsoft Windows Mobile 5.0 for Pocket PCs Reviewers Guide***". January 2006. Available at : http://www.microsoft.com/presspass/newsroom/mobile/docs/Windows_Mobile_%20PPC_Rev_Guide.doc

[69] J. Wilson. "***What's New in the .NET Compact Framework 2.***0". November 2005. Available at : http://msdn2.microsoft.com/en-gb/library/aa446574.aspx

[70] OpenNETC Consulting, LLC. "***Compact Framework***". Available at : http://www.opennetcf.com/CompactFramework/tabid/85/Default.aspx

[71] Jon Person. "***GIS and GPS Components for Visual Studio .NET***". Website : www.gpsdotnet.com

[72] D. DePriest. "***NMEA data***". Available at : http://www.gpsinformation.org/dale/nmea.htm

[73] G. Baddeley. "***GPS – NMEA sentence information***". January 2001.Available at: http://aprs.gids.nl/nmea/#rmc

[74] G. Baddeley. "G***lenn Baddley GPS – NMEA sentence information***". 2007. Available at : http://home.mira.net/~gnb/gps/nmea.html

[75] Google Corporation. "***Google Maps API Concepts***". Available at : http://www.google.com/apis/maps/documentation/index.html

[76] Microsoft Corporation. "***Internet Explorer for Pocket PC – HTML and Object Model Reference***". Available at : http://www.microsoft.com/mobile/developer/technicalarticles/html.doc

[77] Mozilla Foundation. "***Minimo Webbrowser***". Available at : http://www.mozilla.org/projects/minimo/

[78] Microsoft Corporation. "***Windows CE Networking Team Weblog***". Available at: http://blogs.msdn.com/cenet/archive/2006/06/06/620360.aspx

[79] Microsoft Corporation. "***Connection Manager***". 2007. Available at : http://msdn2.microsoft.com/en-us/library/bb416435.aspx

[80] P. Foot. "***Windows Mobile 6.0 and Connection manager***". Micorosft's Forum. Available at : http://forums.microsoft.com/MSDN/ShowPost.aspx?PostID=2088878&SiteID=1&mode=1

[81] HP Corporation. "***Additional Product Information – Ipaq 6915***". Available at : http://h20000.www2.hp.com/bc/docs/support/SupportManual/c00713094/c00713094.pdf

[82] S. Mayr. "***Sending short messages (SMS) via GSM mobile phones***". Available at : http://www.scampers.org/steve/sms/

[83] Definition of ***WAP*** on Wikipedia. Available at : http://en.wikipedia.org/wiki/Wireless_Application_Protocol

[84] Definition of ***MMS*** on Wikipedia. Available at : http://en.wikipedia.org/wiki/Multimedia_Messaging_Service

[85] SecureW2 Corporation. "***SecureW2 Client***". Available at: http://www.securew2.com/

Appendix : Research Proposal

Abstract

In recent years the concept of network mobility is increasingly being investigated in order to gain valuable knowledge for a number of scenarios. The mobile networks that are formed to facilitate the needs of a Mountain Rescue Team are undoubtedly part of this domain and raise the issue of location awareness. The knowledge of location of each of the members of a Mountain Rescue Team could be valuable with the aim of organizing their mission efficiently and accurately. This report presents a proposal for a project in this area with particular emphasis on the communication model and the exact wireless communication technology that should be used to assist the issue of location awareness. In particular, this project will investigate and implement the transmission of the GPS coordinates from a device that a rescue worker will hold, to a centralized server located at the Mountain Rescue Team's headquarters via a number of wireless options.

1. Introduction

Network mobility is a relatively new concept that is being researched and developed during the last few years. Wireless technology has become a ubiquitous part of our society but is currently facilitating with success individual nodes that move around a certain area. The concept of network mobility includes the support of whole networks (with leaf nodes) that are moving in an unpredictable fashion. This network mobility becomes apparent in various domains but has a significant importance in critical operations such as a mountain rescue mission.

The members of the Network Mobility Research Group at Lancaster University [1] are examining the network mobility concept in conjunction with the requirements of the Cockermouth mountain rescue team [2]. The developed mobile network, referred onward as Mountain Rescue Network, identifies a complex mobile model that tries to provide data networking capabilities to mountain rescue workers [3]. In the case of an emergency incident, the Mountain Rescue Team is divided in independent search parties that are composed of a cluster of rescue workers and all-terrain rescue vehicles [3]. These parties are scattered around the valley areas of the Lake District where an incident has occurred and try to spot and aid people or animals. In order to improve the efficiency and accuracy of the work of the members of these search parties, they should carry a small and lightweight device which should transmit each member's GPS coordinates to a centralised server located in the Mountain Rescue Team's headquarters.

This report presents a proposal for a project on the concept of location awareness in a mountain rescue domain. Particular emphasis is going to be given on the way that the GPS coordinates should be transmitted from and to the devices that the rescue workers carry and also to their headquarters. There is a certain necessity to provide real-time updates to the chief member of a team and to the personnel located in the headquarters that are remotely administering a rescue mission. The exact lightweight device, its operating system, its wireless communication capabilities, the type of the communication technology and the devised communication model that will facilitate them are some of the concepts that are going to be thoroughly researched on this project.

The remainder of this report is structured as follows. Section 2 describes the background of the problem domain focusing on the network mobility concept and the characteristics of the Mountain Rescue Team which define some basic requirements of this project. Moreover, Section 2 constraints the notion of this domain presenting the main focus of this project. Section 3 identifies the primary and the secondary objectives along with the methodology that should be followed in order to accomplish these aims. Section 4 identifies the actual tasks that form this MSc project and a schedule of how they should be organized. Finally, Section 5 describes the resources that are at our disposal in order to successfully carry out this project.

2. Background – Analysis of the Problem Domain

This section describes in brief the characteristics of the Mountain Rescue Team (section 2.1), some basic requirements of this team and the network mobility concept which arise communication issues when a search party roams. Moreover, section 2.2 defines the exact problem of the Mountain Rescue domain that this MSc project will focus on.

2.1. Mountain Rescue Team

The Cockermouth Mountain Rescue Team that is collaborating with the members of the Network Mobility Group of the Lancaster University is composed of about 40 members that are split into different search parties in the case of an incident. These parties usually consist of a small number of rescue workers and generally move independently to cover a geographical area according to the incident. These search parties could be envisaged as mobile networks and the rescue workers of each party as the leaf nodes of the network. The members of a search party can carry a simple WiFi enabled device (such as a PDA) and communicate with each other “covered” by a short range wireless hotspot (blue circle, Figure 1). The wireless coverage of the hotspot is projected in the area where a search party roams from a more bulky device (Mobile Router – MR, Figure 1) that a special member of the party is carrying in a rucksack [3]. The communication among these devices usually takes place using the 802.11b/g WLAN standard. The connectivity of these mobile networks is supported by a directed long coverage wireless hotspot (gray circle, Figure 1) using a wireless “technology” such as 802.16, that is located in one of the all-terrain vehicles that is parked in the closest approachable area to an incident [3].

The communication of these mobile networks could be supported by the NEMO Basic Support protocol [4] which presents a basic milestone in the concept of network mobility. This basic protocol guarantees the continuous communication of the nodes of a single mobile network that is roaming across different access network [5]. Along with this session continuity, the NEMO BS protocol takes advantage of IPv6 guaranteeing that every node is reachable with a unique IP address and provides a communication in a way that the mobility of the network is transparent to the nodes inside it [4].

A key characteristic of the network that each search party forms, is that although the whole network is moving and its communication should be effectively supported, its nodes are in fact relatively static in respect with one another. This characteristic in conjunction with the physical unpredictable mobility of the search party introduces a variety of scenarios which define new requirements that cannot be confronted with a single “mobility solution” [5]. For example, a network may move out of the coverage of a hotspot leading either to a total lack of connectivity for the nodes of the network or the enforcement of another type of connectivity, such as the nested case of the MR1’s and MR2’s networks that is illustrated in Figure 1.

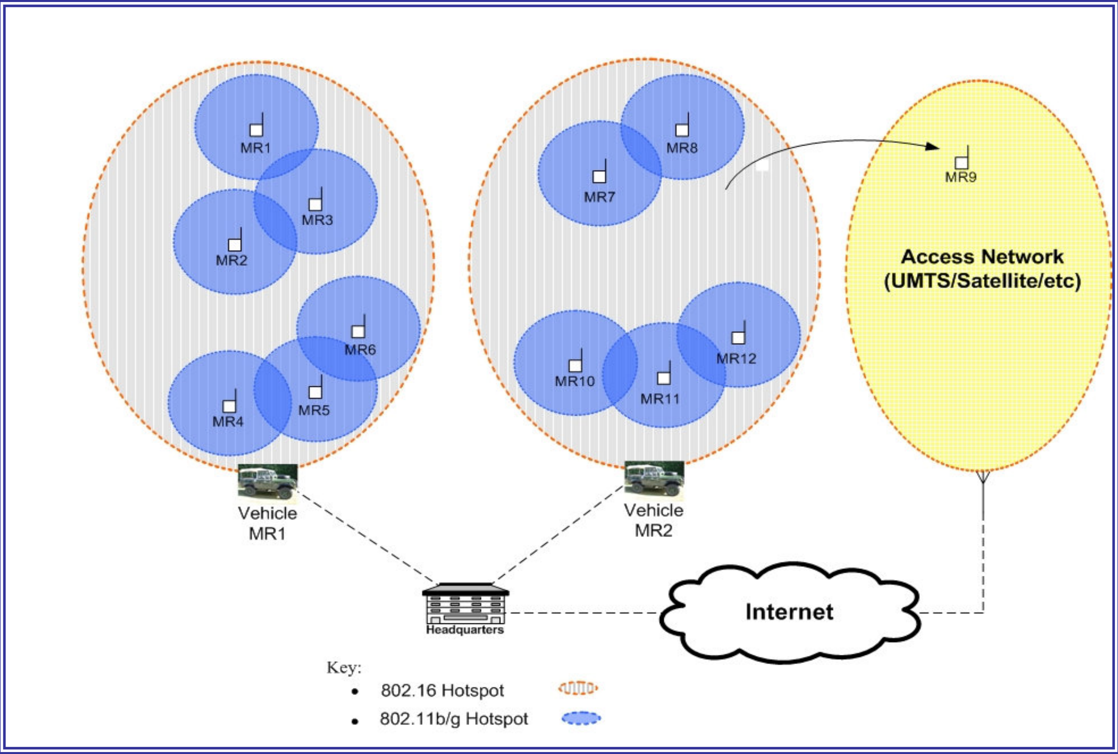


Figure 41 : Example of a Mountain Rescue Network’s infrastructure (adapted from [8])

The general purpose of the development of the Mountain Rescue Network is to enable the rescue workers to efficiently share data that could aid their mission. For example, the location of each member of a search party or the sharing of a photo of a difficult approachable area can be of great value to the Mountain Rescue Team. Moreover, the exact location of each member can help the personnel located at the headquarters to organize a mission more accurately and efficiently. Thus, the application that would be developed for this project should be able to identify the available communication option and use it accordingly in order to transmit the GPS coordinates of a node.

At this point it should also be noted that the nodes of the Mountain Rescue Network should be organized in an ad-hoc fashion and should be generally auto-configured. Moreover, the Mountain Rescue Domain introduces a total lack of fixed network infrastructure and a generally unpredictable movement of its teams and their members. Finally, the rescue workers are and should be confronted as individuals without any expertise on configuring the devices or the network and in fact, the whole network communication should take place in a completely transparent fashion without bothering the workers during a critical mission.

2.2. Defining the focus of the project

The Mountain Rescue Network presents the ideal scenario for researching the network mobility concept with an interesting emphasis on the location awareness of each end/leaf node. As described in the previous section, a basic and important starting point for facilitating the members of the Rescue Team is to be able to project real-time location information in the devices that they carry. Moreover, the administrative personnel that are located in the headquarters should obtain real-time location information on the server aiding the management of the rescue mission.

Therefore, the primary focus of this project is to provide a solution that will enable the location awareness of a rescue worker in the Mountain Rescue Domain. The author of this project intends to focus on the communication options that are available to facilitate the concept of location awareness and the implementation of the application that these devices should have installed. In order to achieve this effectively a communication model should be devised according to the requirements of the Mountain Rescue Team and the application.

Three significant communication options seem to apply to this domain in order to provide the transmission of the GPS coordinates effectively. The first one is to transmit the coordinates via IPv6 over an 802.11 network, the second one is via IPv6 over a GPRS/GSM network and the third one is to transmit them via SMS over the GSM network. This project intends also to define the exact type and frequency of the data that should be transmitted according to the available communication option.

3. Proposed Research

This section breaks down the problem domain (defined in Section 2.1) and the focus of the project (defined in Section 2.2) into primary and secondary aims (Section 3.1) and the methodology that should be followed in order to accomplish these aims (Section 3.2).

3.1. Aims

As discussed in Section 2.1 the objective of this project is to provide location awareness information of each node of a Mountain Rescue Network.

Therefore, this basic aim is decomposed into the two following basic primary aims (PA) :

- **PA.1 :** We would like to **develop a communication model** that will define the type and the frequency of the data that should be transmitted from and to the devices that the rescue workers will carry according to the available communication method. Moreover, this model should take into account and define the server's response to the transmitted data.
- **PA.2 :** We would like to **develop an application** that the devices should have installed in order to identify and efficiently use the available communication option for the transmission of the GPS coordinates.

In conjunction with the above described primary aims some secondary aims can also be defined as complementary ideas that could lead to the enhancement of the outcome of the project. The proposed secondary aims are :

- **SA.1 :** To be able to confront synchronization and redundancy issues for the Mountain Rescue Network.
- **SA.2 :** To research other wireless technologies that may satisfy the requirements in a better way.
- **SA.3 :** To generalize the communication model in a way that could be used and extended to facilitate other applications in this domain
- **SA.4 :** To assess the viability, practicality and dependability of the developed outcome regarding the Cockermouth Mountain Rescue Team. We would also like to assess both the hardware and the software of the outcome of the project in the terms of performance and efficiency.
- **SA.5 :** To confront the security of the developed solution focusing on the reliability, availability, authenticity and integrity of the devices and the data.

3.2. Methodology / Approach

This project intends to deal with both the hardware and the software that could facilitate the Mountain Rescue scenario. However, the author will emphasize more on the development of the application and the communication model that will be used to transmit the GPS coordinates via the available wireless communication methods. It is our intention to investigate and define, rather than to construct, the most suitable off-the-self solution which could be used effectively and be carried from the rescue workers. An important consideration would be to form the data in a way that could be used from the already developed software that is installed in the headquarters' server. Moreover, the response of the server when it looses connectivity with a network or some specific nodes or when it regains that connectivity should also be examined according to the available time.

In addition, a thorough examination should be undertaken to define which of the available communication options is better in a certain context. The primary communication method is to transfer the GPS coordinates via IPv6 over the WiFi network that is projected in the area where a search party roams. However, when a node looses this wireless communication it should be able to send the coordinates via alternative methods such as using IPv4 over the GPRS/GSM network. Finally, a less efficient and most expensive solution might be to transmit the GPS coordinates via SMS using the GSM network. Although these three communication options currently seem suitable for the Mountain Rescue Domain a thorough examination should be undertaken in order to identify the approach that will be used and find out if there is any other more suitable option that could be used more efficiently for accommodating the location awareness concept.

During the development of the project consideration should also take place into the thorough examination of which of the secondary objectives can be satisfied. The prioritization of the secondary objectives and the definition of which of them should be satisfied can be a vital methodology. Throughout the development phase of the project, serious concern should also be placed upon the level of satisfying the primary and secondary objectives. A spiral iteration procedure might take place to consider and maybe modify the objectives according to the needs by assessing their importance during the early stages of the development of this project.

Finally, the testing phase of the application and the communication model will be done using the testbed that the Network Mobility Group has at its disposal. It is via the testing procedures that the current outcome will be assessed during the development of the project. The final outcome of the project could be assessed in conjunction with the members of the Cockermouth Mountain Rescue Team in a non-critical mission. The most important criterion on evaluating the developed solution is the efficiency of the communication model and the transmission of the data back to the headquarters in order to administer the mission of the team effectively.

4. Programme of Work

This section identifies the actual programme of work for the project defining the tasks that should be undertaken in order to satisfy the aims of this project (Section 4.1). In addition, section 4.2 presents a schedule of how the tasks should be organized and accomplished in the available time.

4.1. Tasks

The tasks that should be undertaken are generally defined by the objectives of the projects and the actual procedure for the successful completion of the dissertation. This section breaks down the aims described in Section 3.1 and splits them into the necessary tasks that should be completed.

The primary aims can be accomplished through the following tasks (T) :

- **PA.1 ↔ T.1 :** Discussion of the Rescue Team’s requirements with the members of the Network Mobility Lancaster’s Group. Definition of new requirements according to the needs of the location awareness application that will be developed.

- **PA.1 ↔ T.2 :** Research for the suitable device that can be used and carried from the rescue workers. Definition and documentation of the technical capabilities of the device. Definition and documentation of the decisions taken for the operating system that the device will use, the programming language, the API and the communication technology that is suitable for the development of the application.
- **PA.1 ↔ T.3 :** Research and comparison on the available communication options that could be used. Definition of which communication options can be used and the criteria of this decision.
- **PA.1 ↔ T.4 :** Definition and documentation of the type of data that will be transferred and the frequency of the transmission regarding the available communication method.
- **PA.1 ↔ T.5 :** Definition and documentation of how the lack of connectivity will be confronted from the end nodes and the server. Consideration about any redundancy options and the way that the end device and the server will obtain updates when connectivity is regained.
- **PA.1 ↔ T.6 :** Consideration of the data and the way that the device should project them to the rescue worker who carries it.
- **PA.2 ↔ T.7 :** Implementation of the transmission of the GPS coordinates according to the available connectivity method.
- **PA.2 ↔ T.8 :** Implementation of the communication\tunneling of the developed application with the GUI that a rescue worker observes on his\her device.
- **PA.2 ↔ T.9 :** Implementation\tweak (if needed) of the server's software to be able to use and present the GPS coordinates that it receives on a real-time manner.
- **PA.2 ↔ T.10 :** Testing and improving the implemented application.
- **T.11 :** Assessment of the defined communication model and the developed application in strict correlation with the requirements of the project.
- **T.12 :** General evaluation of the developed application and suggestion for further improvements. Identification of the milestones that it presents to aid the development of another application in the Mountain Rescue scenario. Can the presented location awareness solution be generalized and used in similar case scenarios?
- **T.13 :** Writing the actual report of the dissertation.

4.2. Schedule

This section presents a schedule for the development of the proposed project. Figure 2 illustrates the Gantt chart presenting the duration of the tasks (defined in Section 4.1) in order to accomplish the aims of this project.

According to the Gantt chart all the tasks will be accomplished around the mid August and thus twenty extra days will be available for the improvement of the project and for the case that something has not gone according to the schedule. The duration column that is being presented in the Gantt chart does not include the days of the weekends because these are going to be used to iteratively look into each week's work and summarize it along with the improvement of the report according to the week's tasks

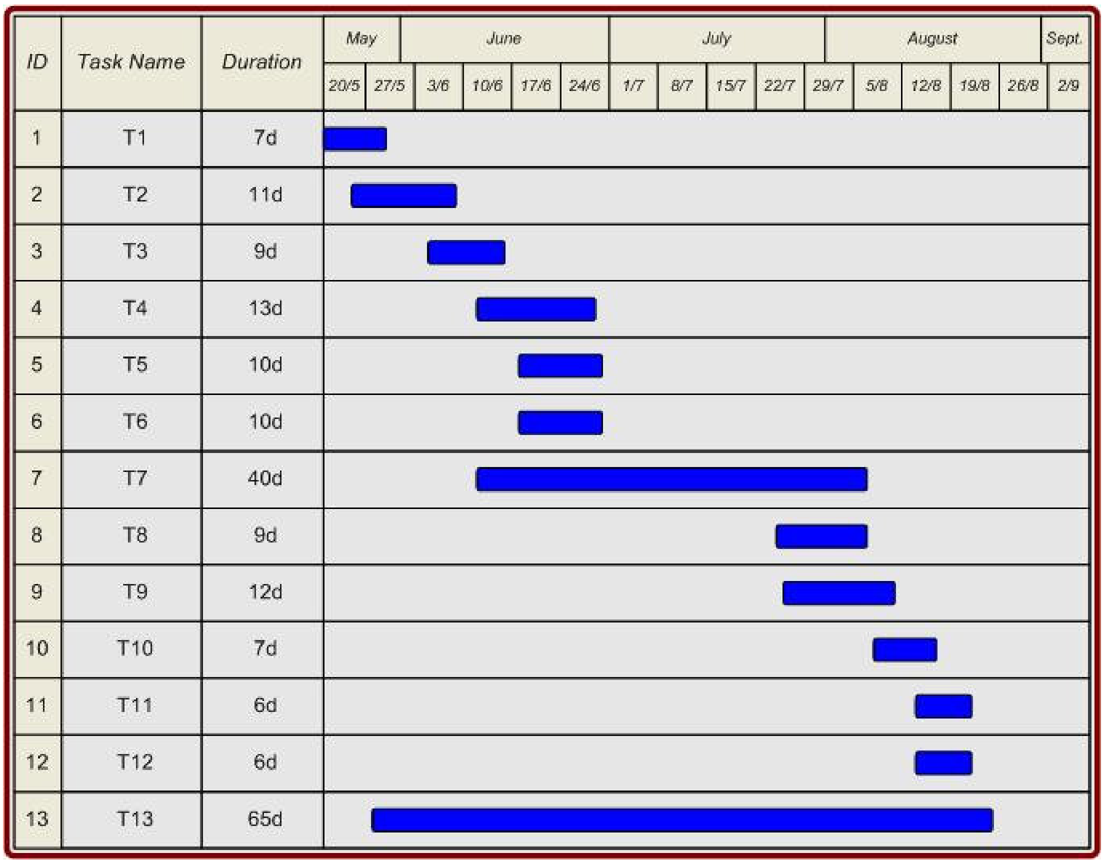


Figure 42 : The proposed Gantt chart for the development of the project

5. Resources

The Computing Department is able to provide an HP IPAQ 6300 PDA device for some initial experimentation. Moreover, it is capable of purchasing any off-the-self device according to any other project needs. Furthermore, an already developed testbed exists for simulations if needed and the server's software is at the disposal of the Network Mobility Group and can be used to simulate and test the developed application. Finally, a real-life testing case can be committed to evaluate the outcome of the project in collaboration with the Cockermouth Mountain Rescue Team in the valley areas around the Lake District.

References

[1] Network Mobility Group of Lancaster University's Website. Available at : www.network-mobility.org (accessed on 02/05/07)

[2] Cockermouth Mountain Rescue Team's Website. Available at : http://www.cockermouthmrt.org.uk (accessed on 02/05/07)

[3] B. McCarthy, C. Edwards and M. Dunmore. "***Applying Network Mobility Techniques to a Mountain Rescue Domain***".

[4] P. Thubert et al. "***Nemo Basic Support Protocol***". IETF Request For Comments 3693, January 2005.

[5] B.McCarthy, C. Edwards and M. Dunmore. "***The Integration of Ad-hoc MANET) and Mobile Networking (NEMO): Principles to Support Rescue Team Communication***". ICMU Conference 2006 Proceedings (Pages 284 - 290), London, UK, October 2006.

[6] H. Ohnishi E. Paik C. Ng and P. Thubert. "***Taxonomy of route optimization models in the nemo context***". IETF Draft "draft-thubert-nemo-rotaxonomy-03.txt", October 2004.

[7] B.McCarthy, C. Edwards and M. Dunmore. "***Advanced in MANEMO : Definition of the Problem Domain and the Design of a NEMO-Centric Approach"***

[8] B.McCarthy, C. Edwards and M. Dunmore. "***Network Transparency in a Mountain Rescue Domain***". To appear at TERENA Networking Conference 2007.