Seamless Roaming between GPRS and WiFi Networks[1]

M. Adda and J. Saunders

University of Portsmouth, Portsmouth, UK
mo.adda@port.ac.uk
jim@jlsnet.co.uk

Abstract

This paper discusses the research and the development of an engineering solution, which would permit mobile nodes to roam between different types of wireless network seamlessly. The two wireless network protocols that are of interest in this project are 802.11b (WiFi), a wireless local area network, which is increasingly being used in homes and offices, and GPRS, a wireless wide area network which works as a data extension to the GSM mobile phone network. Although there are many solutions to this roaming problem, in this paper we discuss a new software based solution, which would decrease network-swapping errors, and obtain better performance. The new solution, NetSwap, is implemented in network simulation software, SN-2, and tested against results from previous hardware implementations with encouraging results.

Keywords Wireless Networks WiFi, GPRS, GSM, network Simulation, HiperLAN/2, Roaming, UMTS and Bluetooth.

[1] Support from INTEL Corporation (UK) and Orange (PCS) Ltd.

1- Introduction

The modern generation of subscribers do not want to be tied to a desk, study or office. Technologies such as mobile phones, PDA's and laptop computers have become an everyday tool in people lives enabling them to become more mobile. Mobile technology has advanced in recent years allowing people to communicate and access information via a number of methods. For instance, Internet, SMS, Chat/Messaging Services, E-mail and VoIP are some recently born method of communication. Technology trends such as these redefine the way in which people live, and create the need to connect mobile devices without wires; from this comes the birth of Wireless Networking. There are a number of different types of wireless network protocols in existence today WiFi, GPRS, UMTS, WiMAX, Bluetooth, and HiperLAN/2.

As these protocols are all fairly new technologies developed by a variety of organisations for different purposes, not all of them are compatible and can therefore cause problems when roaming between them. The two wireless network protocols that fall within the scope of this paper are WiFi and GPRS. Both the GPRS and WiFi wireless network types provide a degree of mobility. However, roaming between them is a difficult task. Referring to figure 1, consider the following scenario:

An office worker, who begins working on his laptop in an office connected to a corporate WiFi network, may also wish to work while travelling home at the end of the day. On leaving the office, he also leaves the coverage range of the corporate WiFi network; he is still covered within range of a GPRS network. The worker may have to wait a considerable time while the computer detects the loss of one network connection (WiFi) and re-route all traffic to the new connection (GPRS). Due to the time delay, any network applications may timeout, throwing connection errors. The same problem may also occur when the worker reaches home and wants to change to his home WiFi network.

The main problem, when roaming between different wireless network types, is that a mobile node must first detect the loss of a wireless connection and therefore its route to a destination. The mobile node then needs to find another available wireless connection, reset its routing tables and change protocol types, before it can then communicate across the new link. During this process the correspondent node does not know when or where the mobile node has moved and therefore does not know where to send its replies. This process takes time and may result in huge packet loss, during which any real-time network applications, such as streaming video and audio, file transfers and games may be likely to timeout.

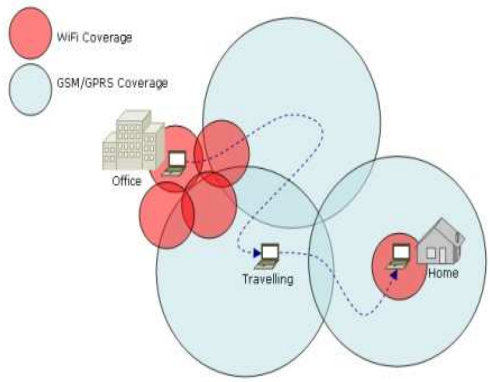

Figure 1 - Mobile Roaming Between WiFi and GPRS Networks

2-Current solutions

In Mobile IP networks a mobile node roams away from one network and is no longer reachable using its original IP address. Therefore, the home agent keeps track of its Mobile Node's locations in a table called a 'Mobility Binding Table'. When the mobile node is 'at home', it is connected to the same network as the home agent, and all network traffic is routed as normal. When a mobile node roams and is connected to a foreign network, it continues to converse with any previous correspondents (e.g. web servers). However, it maintains the source address of all packets, setting them to the address of the home network and not that of the foreign network in which the node now resides. Changing the source address of a packet does not affect the way it is routed. This is because it is not read by routers and therefore delivered as usual.

Mobile IP is a good solution to wireless roaming, but it has its downfalls. Firstly, there is need for more specialised hardware. Secondly, when a mobile node travels across a number of different wireless networks, it will build up a 'daisy chain' number of foreign agents. Traffic may have to be routed across this chain of different networks before it gets to the mobile node. This will add a large degree of latency and limit the bandwidth availability to the slowest link in the daisy chain [1].

With Occasionally Connected Computing (OCC), applications could be written a little more intelligently when a network connection cannot be found. A few applications offer you the option to "Work Offline/Online". For example, Microsoft Outlook will work offline, allowing a user to write e-mails and send them. The e-mail will actually be sent the next time the user connects to the Internet and Outlook goes online. However, this is not done automatically. The user will have to tell Outlook to go online as and when needed, otherwise a whole array of networking errors may be encountered. If an application were designed to automatically connect and disconnect to the Internet ad-hoc through whatever connection was available, be it Ethernet Cable, WiFi or GPRS, it would reduce network/Internet traffic. This reduction would occur because there would be no unused bandwidth taken up during idle periods.
Two companies are looking at an Application Layer (OSI Layer 7) idea, Intel in a project called OCC (Occasionally Connected Computing) [2], and IBM on a project called MQ Anyplace [3]. Both companies are creating network software, which operates regardless of how or when it is connected to a network or the Internet. However, this works well for applications that operate on batch style processing, such as e-mail, file transfer or newsgroups, but would be less suitable for activities such as streaming media which require a constant real-time connection.

Intel, Nokia and a number of other organisations are developing SoC (System on Chip), a single piece of silicon, which includes multiple DSP's (Digital Signal Processors) along with high-speed reconfigurable logic processors, in order to execute multiple wireless protocols. SoC is still in development, and research is being done in gradual stages. At the time of this writing, testing is being carried out on WiFi (802.11) and HiperLAN/2, which share similar protocols and are therefore easier to switch between. The eventual aim is to be able to swap between further protocols including GPRS and UMTS. The ability to roam between different wireless network types is not the only advantage of SoC, Mobile devices can be cheaper to build, and develop because they require less silicon. They can also physically be smaller and consume less power [4].

Software Defined Radio (SDR) is similar in many ways to SoC in that it still relies on the power of DSP technology, and is a development, which allows further radio flexibility. By simply modifying or replacing software programs or firmware, it can completely change the radio's functionality, changing communications protocols and frequency bands. Such flexibility allows for easy upgrade to new modes and protocols without the need to totally replace hardware. There is a great difference between a radio, which uses software internally for some of its functions (e.g. the software in a GPRS radio which is responsible for dialling number) and a SDR radio that can be completely redefined through modification of its software [5]. The concept of SDR is rapidly gaining commercial popularity, not only for the GPRS and mobile phone industry, but also for wireless computer networks such as WiFi. Intel's brand name for SDR is called 'Radio Free' and their aim is "to allow products with embedded silicon to connect to multiple networks, regardless of their respective protocols or requirements."[6].

Both SDR and SoC are still under development. Both would require new hardware to replace current wireless network devices. Mobile users who do not wish to replace their existing expensive hardware would need a software solution to work on top of their current set ups.

3-Proposed solution

In this section, we describe the mobility solution formulated by the author after previous work experience in the wireless networking area and studies of the problem. For the rest of the paper we refer to this solution by '**NetSwap**' that allows seamless swapping between wireless networks. The '**NetSwap**' solution has parallels with the conventional postal system. Take the following scenario:

A student living in Portsmouth decides to first post all his outgoing mail to his parents living in Eastbourne by putting all post within a second envelope addressed to the parents. On receiving any letters, the parents then take the original envelope containing the true receivers address out of the second envelope, and put it back in the post box. It would then appear to the receiver that the letter originated from Eastbourne due to the postal mark printed on the envelope. As the receiver replies, they send all post back to Eastbourne as this is where they perceive it came from. The parents then forward any replies onto the student in Portsmouth by writing "Please Redirect to…" with the student's current address. The letters, which the student posts onto their parents contained within a second envelope, is encapsulated. The concept of encapsulation is one of the central ideas of the ‘**NetSwap**’ solution. There are two main components in the NetSwap system: -

3.1 The NetSwap Driver

The NetSwap Driver is a piece of software, figure 2, which will represent a ‘virtual network interface’ in a Mobile Node. The driver sits on top of all other network device drivers. Its purpose is to fool any Internet application into thinking that the mobile node has a constantly connected network device. The NetSwap driver will then act as a ‘proxy’, utilising the other different network devices (e.g. GPRS and WiFi) depending on their availability. There is a standard for communicating with most new network drivers called NDIS (Network Driver Interface Standard). NDIS is fast becoming an integral part of all Windows operating systems and all Windows compliant Network adapters are shipped with NDIS compatible drivers.

Figure 2 - NDIS - NetSwap Driver [7]

The NetSwap Driver is responsible for monitoring all connections. It routes the traffic from the application to one physical card at a time, depending on availability and priority order. Priority may be based on cost, routing metric, bandwidth, or it can be user defined. For example, WiFi may be of higher priority than GPRS due to it being a quicker connection. The main job of the NetSwap Driver is to fool any applications into using the ‘virtual network interface’ before it uses any other network interface. One method of doing this would be to force the metric values for all other interfaces in the system up, thereby giving the virtual interface the lowest metric. For example, the driver could give itself a metric of 10 and move Ethernet to 11, Wireless to 31 and GPRS to 51. Therefore, Internet application sending network traffic will go through the NetSwap Driver with the new lower metric. The NetSwap driver must then force all packets to be sent through a special router (see figure 3 – NetSwap Gateway) by encapsulating them in a second IP header. It should then communicate with the special router by letting it know which interfaces are currently available on the mobile node. This communication would be done at regular intervals while switching between interfaces.

3.2 The NetSwap Gateway

The NetSwap Gateway is a router sitting on the Internet, which acts as a central point to relay all traffic through. As discussed earlier, the mobility problem occurs when a mobile node swaps between two networks and the correspondent server is unaware of where the mobile node has moved to, rendering it unable to reply. The NetSwap Gateway must have a static IP address as it is used as a ‘known point’ and should not change. Its job is to forward all traffic to and from the mobile node’s current address. There is an ‘always on’ connection as far as any applications running on both the mobile node and the correspondent are concerned, and the fact that the traffic goes through the NetSwap gateway should be transparent to both ends.

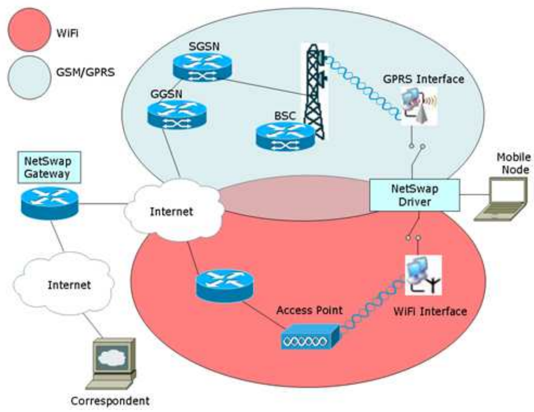

Figure 3 - Switching Between Network Interfaces

4- Simulation and results

Although the system had been built and tested, we have used a network simulation tool such as NS-2 to recreate a live network system, calculating packet transfers across a network between two or more nodes and also simulating loss, queuing, delay, and processing time. The purpose of this quick simulation, shown in figure 4, is to try and observe whether the “roaming” problem also occurs when swapping between network routes in the NS-2 simulator.

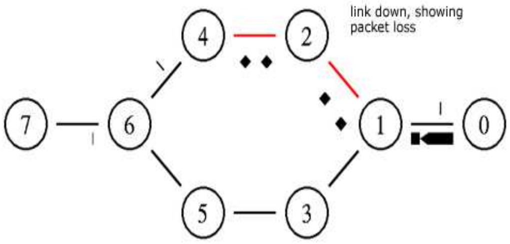

Figure 4 – Output of first NS-2 FTP simulation

The following graph, obtained from cute FTP application, shows the crossover between WiFi and GPRS. Initially, both connections are up and the FTP session is running across WiFi. The WiFi connection goes down at 1.00 seconds. After polling and determining that the WiFi connection is down, the NetSwap driver sends an update ICMP packet to the gateway. All new FTP traffic gets sent through the GPRS link and the gateway redirects any replies back through GPRS. Due to the time delay as the NetSwap driver polls its connections, and updates the gateway, there is a little more packet loss. However, this is still not enough to terminate the FTP session, which is again recovered by TCP/IP.

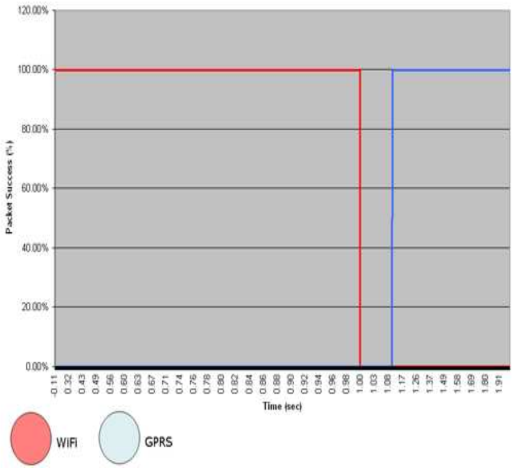

Figure 5 – NS-2 NetSwap implementation Results, with connection loss

The following tests with Streaming Audio using Microsoft Media Player produced some rather varied and unexpected results. Due to the fact that Media Player uses a buffer to read its audio stream ahead of what is actually being played, it was thought that this might offer a bit of leeway as interfaces were swapped between. The first test was to observe the effects, when both WiFi and GPRS interfaces are online and the WiFi interface loses connection while streaming media.

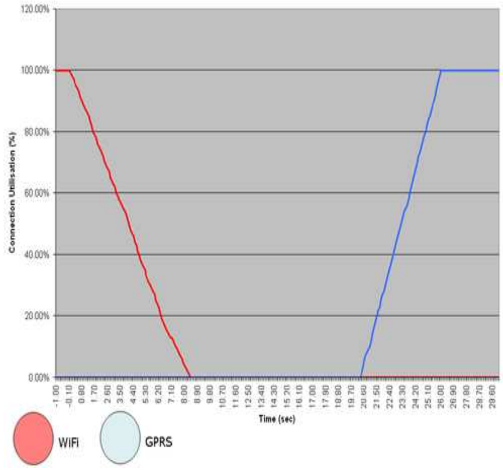

Fig 6– Streaming Media, WiFi > GPRS

These results were more as expected. Initially, Media Player was streaming media across WiFi due to its lower metric value. As the WiFi connection goes down, the buffer starts to run out. After the point where the buffer is empty, there is a delay while the routing tables are reset and a new route is established using GPRS. The buffer is then reloaded and streaming media resumed. This whole process takes approximately 26 seconds. Had the Media Player buffer been longer, there is a possibility that, as the interfaces changed; it may have appeared to swap a little more seamlessly. This is due to the fact that, during the time where the buffer is being utilised after an interface goes down, the routing tables could be reset. If this process is done in time, before the extended buffer runs out, streaming can resume without a break. The next graph shown the effect a WiFi reconnection has on Media Player stream that is currently streaming over GPRS.

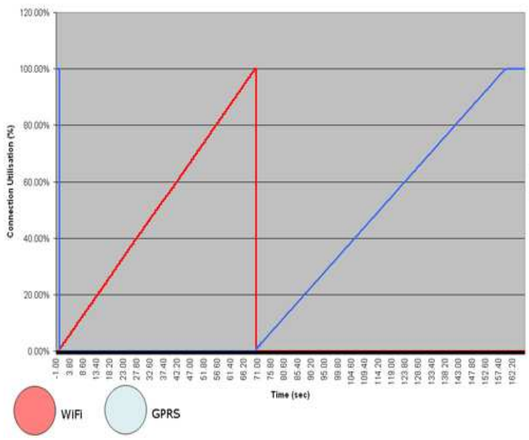

Figure 7 – Streaming Media, GPRS > WiFi > GPRS

The results are quite curious as they show that, at the point where WiFi is re-established, the buffer on GPRS is instantly cut and reloaded on WiFi. At the point where the WiFi buffer is fully loaded, it is again instantly cut and, strangely, reloads back on GPRS. The end result is that both WiFi and GPRS interfaces are online, but Media Player is still streaming its media across GPRS. Due to this, a long period of time (157 seconds) passes before streaming media is resumed, albeit on the more expensive, slower metric, GPRS. Following on from the previous test with both WiFi and GPRS interfaces online and media streaming over GPRS, a GPRS connection loss is now simulated. Showing the best Media Player results so far, the GPRS buffer runs out and the WiFi buffer reloads almost instantly. There is a total streaming media loss of approximately 15 seconds.

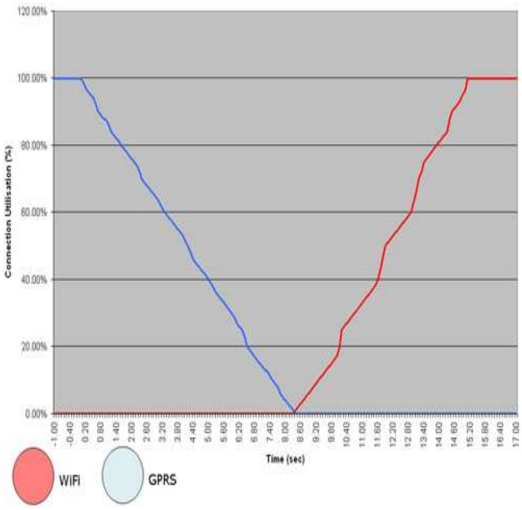

Figure 8 – Streaming Media, GPRS > WiFi

5- Conclusion

After implementation and testing of the NetSwap solution within the NS-2 simulator, some encouraging results were produced. These results proved that the NetSwap solution is able to cut down the packet loss and crossover time as a mobile node roams between different types of wireless network. As the NetSwap solution is not protocol dependent it is not restricted to GPRS and WiFi protocols. The solution can be used with any type of IP based network interface, including wired connections such as Ethernet. The NetSwap design is a software based solution, which means the driver could be implemented on any type of mobile node (including Laptops, PDA's or Mobile Phones) and the gateway can be implemented on any type of router (e.g. Cisco router or Linux box). It could be provided as a feature set to add to routers.

References

[1] Gigaport 2002, How Mobile IP Works, http://www.gigaport.nl/netwerk/access/ta/mip/en_mobileip.html , 2003

[2] Occasionally Connected Computing http://cedar.intel.com/cgi-bin/ids.dll/content/content.jsp?cntKey=GenericEditorial::mobile_occasionalconnect &cntType=IDS_EDITORIAL&catCode=CVS, 2003

[3] MQ Anywhere, http://www-3.ibm.com/software/integration/wmqe/ , 2003

[4] SOC - System-On-Chip, Retrieved from http://soc.ece.ubc.ca/, April 2004

[5] A Software-Defined Radio for the Masses, http://www.arrl.org/tis/info/pdf/020708qex013.pdf, Part 1, Page 13, Accessed 2003

[6] Radio Free Intel: Research and Development at Intel Retrieved from http://www.intel.com/labs/radio/, 2004

[7] What is NDIS?, SisTech - NDISOne, Retrieved from http://www.ndisone.com/whatisndis.html, 2003