Performance Issues with Vertical Handovers – Experiences from GPRS Cellular and WLAN Hot-spots Integration

Rajiv Chakravorty†,Pablo Vidales‡,Kavitha Subramanian†, Ian Pratt†, Jon Crowcroft†
†Computer Laboratory & ‡Laboratory for Communications Engineering
University of Cambridge, JJ Thompson Avenue, Cambridge CB3 0FD, U.K.
Email: FirstName.LastName@cl.cam.ac.uk *e.g. Rajiv.Chakravorty@cl.cam.ac.uk*
http://www.cl.cam.ac.uk/coms/

Abstract

Interworking heterogeneous wireless access technologies is an important step towards building the next generation, all-IP wireless access infrastructure. In this paper, we present an experimental study of inter-network mobility between GPRS Cellular and 802.11b-based WLAN hot-spots, and analyse its impact on active transport TCP flows. Our experiments were conducted over a loosely-coupled, Mobile IPv6-based, GPRS-WLAN experimental testbed. Detailed analysis from packet traces of inter-network (vertical) handovers reveals a number of performance bottlenecks. In particular, the disparity in the round trip time and bandwidth offered by GPRS and WLAN networks, and presence of deep buffers in GPRS, can aggravate performance during vertical handovers. This paper summerizes the practical experiences and challenges of providing ***transparent mobility*** *in heterogeneous environments.*

Based on our observations, we propose a number of network-layer handover optimisation techniques, e.g. Fast Router Advertisements (RA), RA Caching, Binding Update (BU) simulcasting and layer-3 based soft handovers that improve performance during vertical handovers. The paper concludes with our experiences of migrating TCP connections, and the impact that this has on applications such as ftp and web.

1. Introduction

World over, mobile Internet access is showing strong growth fuelled by the increasing popularity of WiFi (802.11b-based WLANs) and world-wide deployment of wide-area wireless networks such as GPRS and 3G. Multi-mode mobile devices (e.g. GPRS-WLAN pcmcia cards) are also becoming affordable, and a growing number of devices such as laptops, PDAs and handhelds are equipped to connect to different networks.

Cellular networks and WLAN hot-spots are complementary wireless access technologies. While WLAN offers limited coverage and higher data access rates, cellular networks like GPRS or 3G provide geographically wide-area coverage but comparatively lower access bandwidth (refer Table 1). As a consequence, 802.11-based WLANs are being widely deployed as *WiFi hot-spots* e.g. in hotels, cafes, etc., whereas GSM/GPRS cellular networks provide "always-on" mobile services such as voice, short message services (SMS), e-mail and web browsing.

Network Characteristics	GPRS Cellular	802.11b-based WLANs
Coverage	Almost Ubiquitous	Local (50-100m *hot-spots*)
Bandwidths	Low (30-50kbps)	High (5-7Mbps)
Round Trip Times	High (500-3000ms)	Low (5-40ms)
Modulation	GMSK	CCK
Medium Access Control	slotted aloha TDMA (only uplink)	CSMA/CA (DCF)
Security	Relatively Secure	Weak (WEP-based)
Mobility	Link-layer (voice opt.)	Layer-3 (link assist.)
Roaming	Subscription-enabled	N.A. (optional)
Deployment Cost	Very High	Low
Cost of data services	High	Low

Table 1. GPRS Cellular and 802.11b WLANs

Wireless networks differ intrinsically in their physical-layer, medium access and link-layer mechanisms. Different mechanisms are used to meet different requirements of the wireless medium (local-area or wide-area). To cope with harsh-outdoor mobile environments, cellular links require use of sophisticated signal processing, interleaving, channel estimation techniques, FEC/link-layer ARQ, etc. The net effect of this is that cellular links typically suffer from high and variable round trip times, link outages and burst losses e.g. during deep fading and handovers. Consequently, end-user experience in cellular environment is quite different from the relatively stable 802.11b-based WLANs.

Overview

Network link-layer characteristics play an important role in wireless network integration. This is particularly true if networks exhibit vastly different characteristics. In table 1, we have highlighted some of the key characteristics of the GPRS cellular and 802.11b-based WLAN networks. A mobile user's access to either GPRS or WLAN networks will be typically based on policy and performance issues of the networks such as the access bandwidth, coverage, cost, security, etc.. Therefore transparent mobility in this environment should aim to achieve efficient handovers i.e. *inter-network (vertical)* handovers between GPRS and WLANs.

In this paper, we specifically focus on the performance of such vertical handovers. In particular, we are interested in examining the impact of vertical handovers on the performance of the TCP transport protocol. Our work aims to:

1. **Partition** the different components contributing to latency during vertical handovers between GPRS cellular and 802.11b-based WLAN networks,
2. **Identify** performance problems during vertical handovers and its impact on transport TCP when using the Mobile IPv6 protocol,
3. **Explore** the extent to which the Mobile IPv6 protocol can hide the differences of the disparate underlying link-layer technologies, and,
4. **Demonstrate** the efficacy of different optimization techniques at different layers (network and transport) to improve performance during vertical handovers.

As part of the Cambridge Open Mobile System (COMS) project[1], we are investigating different aspects of GPRS/3G cellular and 802.11b-based WLAN hot-spots integration. Our experiments are performed in a fully-integrated, Mobile IPv6-based WLAN–GPRS testbed, and we believe this to be possibly the first such work that has attempted to practically evaluate the impact of inter-network handovers on transport performance in a *realistic, Mobile IPv6-based, commercial cellular network testbed setting*.

Paper Outline

The next section characterizes the latencies involved in two main steps of the vertical handover process, i.e. the handover decision and the handover execution. Section 3 describes a loosely-coupled, Mobile IPv6 based GPRS-WLAN testbed, while Section 4 focuses on the handover execution process which has a significant contribution in the overall handover latency, study its various components and overall impact on transport performance. In Section 5 we explore different network-layer handover optimization schemes, e.g. Fast Router Advertisements (RAs), RA Caching, Binding Update (BU) simulcasting, smart buffer management using a proxy in GPRS networks, and Layer-3 based *soft* handovers that can help minimize handover execution latencies to improve performance. Section 6 draws experiences from the testbed experiments, while the last section concludes our paper.

2. Characterizing Vertical Handovers

A handover process between hybrid wireless networks can be characterized in two main steps: (1) a handover *decision* process and, (2) a handover *execution* process. A handover *decision* is the process of deciding (by the mobile node, network or by both) *when* to perform a handover. After the decision to handover is taken, the handover *execution* process comes into play. Handover decision and detection steps can sometimes overlap, as there are scenarios when the decision process may require more additional probing of the network (e.g., duplicate address detection time in Mobile IPv6). However, the handover latency can be broken into the following three main components:

- Detection Period (t_d). It is the time taken by the mobile terminal to discover (e.g. using link-layer beacons) that it is under the coverage of a new wireless access network to the instant it receives a router advertisement (RA) from the new access router. When the mobile is under the coverage of the new network, it can detect this coverage using (1) trigger-based router solicitation or, (2) wait to receive a router advertisement from an access router in the visited network [6]. For simplicity of exposition in this paper, we additionally consider that t_d would include any duplicate address detection (DAD) time, if any.

- Address Configuration Interval (t_c). This is the interval from the time a mobile device receives a router advertisement, to the time it takes to update its routing table, and assign its interface with a new care-of-address (CoA) address. The new CoA is based on the prefix of the new (visited) access router available from the router advertisement.

- Network Registration Time (t_r). It is the time taken to send a binding update to the home agent as well as the correspondent node, to the time it takes to receive the first packet from the correspondent node. Note that MIPv6 does not specify waiting for a binding acknowledgment from a correspondent, as it is optional, hence, we only consider the case when a mobile node receives a packet from the correspondent.

1 (http://www.cl.cam.ac.uk/coms/)

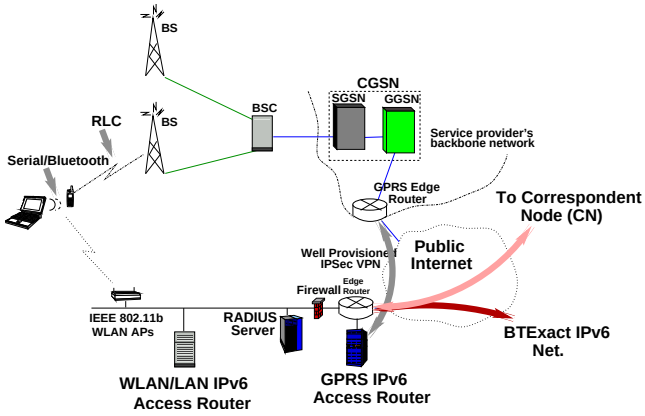

Figure 1. Loosely-coupled, Mobile IPv6-based GPRS-WLAN Experimental Testbed.

Thus, an *IP-level* (network layer) handover will consist of the network detection period, address configuration interval, and network registration time. This also suggests that optimizing IP-level vertical handover latency would essentially involve minimizing t_d and t_r, since t_c depends upon the computing capability of the mobile device.

3. Test Environment and Tools

Our experimental setup consists of a loosely-coupled, Mobile IPv6-based GPRS-WLAN testbed as shown in figure 1. The cellular GPRS network infrastructure currently in use is the Vodafone UK's production GPRS network. The WLAN access points (APs) are IEEE 802.11b APs.

The GPRS infrastructure comprises base stations (BSs) that are linked to the SGSN (Serving GPRS Support Node) which is then connected to a GGSN (Gateway GPRS Support node). In the current Vodafone configuration, both the SGSN and GGSN nodes are co-located in a single CGSN (Combined GPRS Support Node). A well provisioned virtual private network (VPN) connects the Lab network to that of the Vodafone's backbone via an IPSec tunnel over the public Internet. A separate "operator-type" RADIUS server is provisioned to authenticate GPRS mobile users/terminals and also assign IP addresses.

For access to the wireless testbed, mobile nodes (e.g., laptops) connect to the local WLAN network and also simultaneously to GPRS via a Phone or PCCard modem. The mobile node's MIPv6 implementation is based on that developed by the MediaPoli project [16], chosen for its completeness and open source nature.

We brokered a semi-permanent IPv6 subnet from BTExact's IPv6 Network, which connects us to the 6BONE. Using the address space, we are able to allocate static IPv6 addresses to all our IPv6 enabled mobile nodes. A router in the lab acts an IPv6/IPv4 tunnel end-point to the BTExact's IPv6 network. This router is also an IPv6 access router (Home Agent) for the lab's fixed-internal IPv6-enabled network and also for the internal WLANs (shown in figure 1). Routing has been configured such that all GPRS/WLAN user traffic going to and from mobile clients passes through the internal router, enabling us to perform traffic monitoring.

Since the GPRS cellular network currently operates only on IPv4, we use a SIT (Simple Internet Translation) to tunnel all IPv6 packets as IPv4 packets between the mobile node and a machine providing IPv6-enabled access router functionality on behalf of the GPRS network. Ideally, the GGSN in the GPRS network would provide this functionality directly, but using the tunnel incurs only minor overhead.

All the characterization tests for GPRS-WLAN vertical handover were analysed using a version of `tcptrace` program updated (`tcptrace+`)[2] to trace TCP connections for Mobile IPv6 handovers.

4. Experiments with Vertical Handovers

We have evaluated the impact network layer hard handovers have on TCP transport flows. We consider the case of a GPRS-WLAN network testbed based on the current Internet standards and discuss our practical experiences. More thorough description is available in the form of a companion technical report [4]. Table 2 provides a brief description of the type of vertical handovers that are discussed in this paper.

4.1. Testbed Operation

The testbed was operated under the following conditions:

1. Network discovery for the mobile node was performed based on router advertisements. In this case, the mobile node waits for the first router advertisement (R_a) to arrive from the access router of the visited network.
2. Unless stated otherwise, all access routers including the home agent are set to multicast router advertisements in accordance to the recommended values specified by the neighbour discovery protocol [22].
3. For all cases considered in these tests, the multimode mobile device has all of its network interfaces (LAN/WLAN/GPRS) powered on simultaneously to avoid the interface initialization time.

2 Source Code publicly available: http://www.cl.cam.ac.uk/users/rc277/soft.html

Handover Type	Description of the inter-network (vertical) handover
Hard	Break the old IP point of attachment and then Migrate to the new one ('Break-before-Make') – Section 4 and 5
Soft	Migrate to the new IP point of attachment and then break the old one ('Make-before-break') – Section 5.0.4
Anticipated	Handovers executed based on movement (coverage status) of the new network (anticipated using link triggers) – Section 5.0.2
Unanticipated	Coverage of the new network known a *priori*, but execution could be delayed by an application or user. – Section 5.0.2

Table 2. Description of Handover types discussed in this paper.

However, this does not necessarily mean all interfaces are linked to their respective networks.

4. All hosts run Linux 2.4.16. We use a Motorola T260 GPRS mobile, which is a "3+1" (maximum 39.6kbits/s downlink datarate) handset.

For the tests conducted, handovers were forced from WLAN to GPRS and vice versa. For testing handovers, file downloads were initiated by the multi-mode mobile device over WLAN from an internal web-server (acting as a correspondent node). During downloads, we force a vertical handover to GPRS and back again to WLAN.

In the testbed, we allow all traffic to pass through an intermediate router, and simultaneously monitor the traffic (using `tcpdump`) to/from the web-server and the mobile device during all active data sessions. As mentioned earlier, the internal router is also the IPv6 access router for the WLAN, and there is a separate GPRS access router (logically co-located to the GGSN), that acts as an access router for the GPRS network. These routers were set to advertise router advertisements randomly between 3 to 10 seconds as recommended by RFC 2461 [22].

4.2. Vertical Handover Evaluation

Using the GPRS-WLAN testbed, we have investigated the extent to which we can migrate TCP connections during vertical handovers.

We initiated a file transfer of about 25MB size from the mobile node on a WLAN and forced a handover to GPRS and vice versa. We collected `tcpdump` traces of the handover in an internal router as well as at the mobile client.

For one such GPRS↔WLAN handover trace shown in figure 2, we find that it takes around 4s to handover from WLAN→GPRS, and in this case the TCP data session backs-off at the server end, re-transmitting 3 times before an ACK piggybacked along with the binding update is available from GPRS. The handover from GPRS→WLAN takes about 7s. Thus, there are two components that contribute to the total vertical handover latency:

- *IP-level (network) handover latency* ($t_d + t_c + t_r$) – the total time to detect and migrate the IP points of attachments, and,
- *residual TCP back-off time* (t_{tcp}) – the interval for which a TCP flow remains (exponentially) backed-off even after the IP-level handover.

Ideally, handover latency should be composed mainly of the detection time and time for the migration of IP point of attachment. However, the impact of the vertical handover also causes the source TCP to timeout and exponentially back-off to retransmit again. If the IP point of attachment has already migrated before the source TCP retransmits, the mobile node should receive this packet from the new access network. Notice that in figure 2, the lower right-hand close-up plot for GPRS→WLAN handover shows no such TCP retransmissions from the source for a substantial duration. This is explained by the buffering offered by current GPRS networks.

Most GPRS networks provide a substantial amount of buffering (for every GPRS mobile device) in their GGSN nodes, observed at up to 200KB [5][3]. Because of such deep buffers in GPRS networks, long-lived TCP sessions will progressively increase their congestion window until they exceed this threshold, experience loss and then recover using fast re-transmit (halving their congestion window). However, the buffering is rather more than the bandwidth-delay product of the GPRS downlink typically $\tilde{1}$0KB. The resultant packet queuing that happens at the GPRS GGSN leads to the source RTO (TCP's Retransmission TimeOut value) becoming inflated. Thus, as we can see from figure 2 the source experiences a substantial overall handover latency. The total handover process is compounded by the unnecessary backoff in the transport TCP flows, which can further impact performance of applications.

In fact, we see that the first packet after the handover is available to the mobile host only after the web-server has timed-out to retransmit, and that it eventually retransmits all in-flight packets that were actually lost during the handover process over GPRS. The amount of data buffering is high for any long-lived TCP session (or number of active TCP sessions e.g. web flows). Therefore, the extent of packet loss will also be proportional to the buffering in the GPRS GGSN at the time of the handover.

While buffering exists even in other networks (e.g. WLAN), the situation is particularly exacerbated in

3 However, the UK's Vodafone GPRS network has recently been reconfigured to reduce the allocation to about 30KB

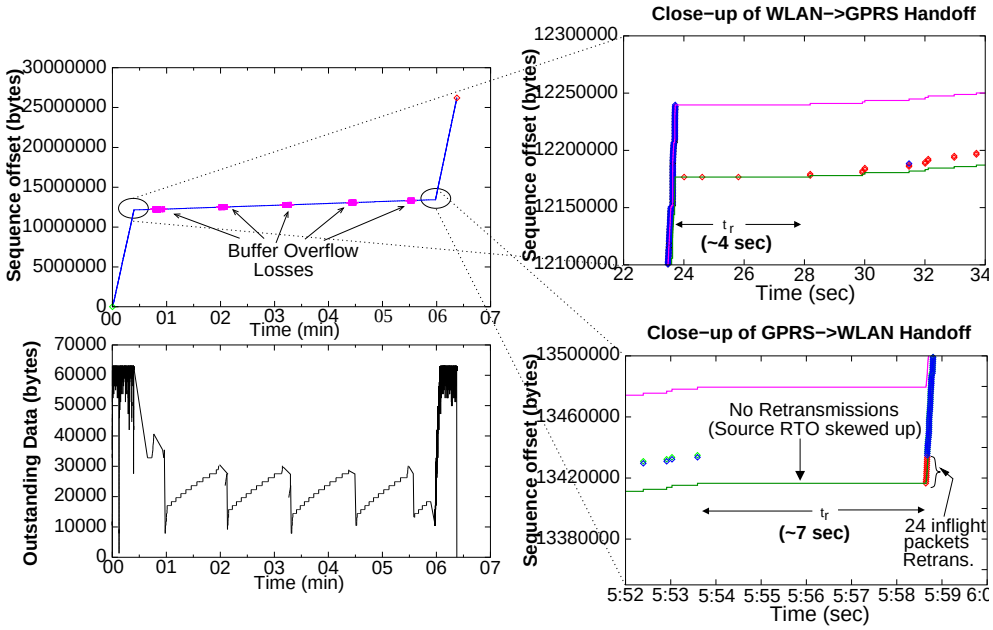

Figure 2. Impact of Mobile IP vertical handovers on TCP performance. Right half (top) shows the close-up of WLAN→GPRS handover, while right (bottom) shows GPRS→WLAN handover.

GPRS due to the presence of deep buffers and specifically the low-bandwidth nature of such links. The high buffering in GPRS aggravates matters for TCP flows during GPRS→WLAN handovers due to RTT and RTO inflation. However, once the source times out to successfully retransmit, it can then rapidly increase its congestion window soon after it starts receiving packets (ACKs) from the WLAN, so as to quickly normalize its RTO values. The bottom-right close-up plots of figure 2 shows how the sequence trace shoots-up soon after the handover to WLAN.

4.3. Latency Partition

Analysis of the GPRS↔WLAN handover in figure 2 shows the handover latency of a single trace. In this section, we statistically partition the overall vertical handover latency from more than 20 such handover runs.

As discussed earlier, total handover latency is sum of the detection time (t_d), configuration time (t_c), and registration time (t_r). In our case, t_d depends on the router advertisement frequency. For these tests, we set the router advertisement frequency to vary randomly between 1s (minimum) to 3s (maximum).

Table 3 gives the handover latency partition. For the WLAN→GPRS case, we find a mean t_d of around 808ms, while GPRS→WLAN gives a t_d of about 2241ms. These values for t_d, can show high variability due to the frequency of the router advertisements. On the other hand, registration time (t_r) is a function of the network link-layer characteristics; hence, these values should be typically higher when executing upward vertical handovers (e.g. for WLAN→GPRS handovers) as binding updates are sent over the high latency GPRS link.

However, table 3 show almost similar registration times (t_r) for the WLAN→GPRS as well as the LAN→GPRS case, when compared to the GPRS→WLAN. Ideally, since WLAN offer links that have low RTTs, the registration times for GPRS→WLAN handover should have been lower. Because of the high buffering in GPRS GGSN, the source perceives inflated RTTs, and hence, an inflated RTO. Consequently, in many cases, the registration process could complete only after the source eventually retransmitted with a high value of the RTO. This leads to high vertical handover latency for GPRS→WLAN.

Additionally, we notice that the standard deviation in the total handover latency for GPRS→WLAN are higher than those for WLAN→GPRS. This variation in the handover latency for GPRS→WLAN is due to the variability as seen in the amount of excess buffering at the GPRS GGSN during handovers as discussed earlier. The impact of this is high variability in the RTTs perceived by the source (here the web-server), consequently leading to significant variations in its RTO calculation, and hence, in this case variable handover latencies.

The configuration time (t_c), which depends upon the host

WLAN↔GPRS (the split in ms)	WLAN→GPRS Min	Mean	Max	Std. Dev.	GPRS→WLAN Min	Mean	Max	Std. Dev.
t_d	200	808	1148	304	739	2241	3803	919
t_c	0.853	0.87	0.890	0.109	0.380	1.062	1.186	0.233
$\hat{t}_r$	2339	2997	3649	395	2585	4654	7639	1611
T_h	3323	3806	4438	310	5322	6896	8833	1118

Table 3. Handover Latency Partition (in ms). ($\hat{t}_r = t_r + t_{tcp}$)

computing capability and state of the interface (for e.g. up, down, suspend) were measured at considerably lower relative values. Other factors also contribute to the handover latency, for example, tunneling IPv6 packets over an IPv4 network adds to the some overhead.

5. Improving Handover Performance

As shown in the previous section, overall vertical handover latency is the sum of IP-level handover latency and any residual TCP back-off time. In this section, we focus on number of network-layer (IP-level) optimizations that can improve vertical handover performance. We discuss use of fast router advertisements (RA), RA caching, BU simulcasting, and the use of a Layer-3 based *soft* handover approach to improve performance.

5.0.1. Fast Router Advertisements (RAs) Fast RAs improve handover performance by minimizing detection time during handovers. *Note that instead of using fast RAs, one can also explicitly solicit RA during the handover. However, the cost of RA solicitation over GPRS links is high (due to the high RTTs) and as we shall see in this section, increasing RA frequency is a better option.*

The neighbour discovery protocol RFC 2461 [22] specifies a random router advertisement interval between 3s and 10s, this interval is large given the impact detection interval can have on overall handover latency. By reducing the router advertisement interval, we can improve the detection time and overall handover latency.

It is interesting to note that the latest IETF draft on Mobile IPv6 takes this issue into account, and specifies a much shorter interval between 30ms (`MinRtrAdvInterval`) to 70ms (`MinRtrAdvInterval`) for access routers using MIPv6 [6]. However, increasing RA interval should also take into account the resultant overhead caused, as there will now be a trade-off involved; increasing RA frequency can result in substantial overhead, especially over 'long-thin' links such as GPRS.

In order to evaluate the impact of RA frequency on the handover detection time, we modified the Linux IPv6 RA deamon (`radvd+` [20]) to support different RA interval values including one specified by the latest Mobile IPv6 draft [6].

Table 4 shows the effect of varying RA interval on mean handover detection time (t_d) from over 15 handover runs. In these tests, we started file downloads and then forced handovers between GPRS↔WLAN, keeping the RA frequency in one link constant, while varying the other and vice versa. We collected and analysed `tcpdump` traces at all network interfaces on the client-side as well as the internal router. What one would expect is that when the RA interval is reduced, the detection time spent waiting for the RA to arrive would also reduce. Based on the average values, we find that as we increase the RA frequency in WLAN the mean detection time also reduces.

The case is, however, somewhat different for GPRS. As we increase the RA frequency in GPRS, the mean detection time does not show substantial improvement when compared to WLANs. Though the best case values are still encouraging, deeper investigation from the `tcpdump` traces for many cases show RAs being delayed, and then quickly arriving in bunches along with the other TCP data packets. This phenomenon is not unusual, as packets can oftentimes experience highly variable delays and 'clumping', as shown from GPRS link characterization work [5]. Highly variable delays experienced in the GPRS link are due to the link-layer (e.g. ARQ in the RLC [5]).

Also interesting to note is that any increase in the RA frequency also increases the overhead over the GPRS link. Therefore there is trade-off involved; any improvement achieved in handover (detection) latency leads to worsening network overhead. RA overhead caused by significantly increasing the RA frequency can lead to substantial overhead in GPRS – an RA interval set at 30-70ms (as per the latest MIPv6 draft [6]) will lead to about 25-50% overhead in terms of actual bandwidth (considering a max. downlink datarate of 39.6kbps for a '3+1' GPRS phone) for 'long-thin' links such as GPRS.

Based on the results from these experiments, we feel that although increasing RA frequency does help improve detection time in WLANs, it is not the best option for networks such as GPRS. Not only is there a costly trade-off involved due of the additional RA overhead, but also the use of fast RAs is not necessarily a 'guarantee' in reduction of the handover detection time. Based on such trade-offs, RA intervals are perhaps somewhere between 0.5-1s in GPRS IPv6 access routers. This is about half the observed average GPRS RTT, and also ensures that the resultant RA overhead is neg-

RA Interval (`MinRtrAdvInterval`-`MaxRtrAdvInterval`)[6]	WLAN→GPRS t_d (ms) Mean(best)	WLAN→GPRS GPRS RA Overhead (39.6kbits/s downlink)	GPRS→WLAN t_d (ms) Mean (best)	GPRS→WLAN WLAN RA Overhead (8Mbits/s downlink)
RA1: 300ms-400ms	551(146)	4.75% - 6.33%	234(39)	0.0157% - 0.0210%
RA2: 200ms-300ms	360(187)	6.33% - 9.5%	242(69)	0.0210% - 0.0317%
RA3: 100ms-200ms	324(44)	9.5% - 19%	174(95)	0.0317% - 0.0633%
RA4: 40ms-70ms	217(41)	27.5% - 47.5%	86(44)	0.0917% - 0.1583%

Table 4. Impact of Fast RAs on handover detection time.

ligible.

5.0.2. Client-based RA Caching Client-based RA caching aims to eliminate detection time during vertical handovers. In the Fast RA scheme, waiting for RAs to arrive means that a certain amount of time will be expended before a mobile host can detect and receive the RA, and then configure its interface with a new CoA. However, it is possible to further improve handovers by eliminating the detection time altogether.

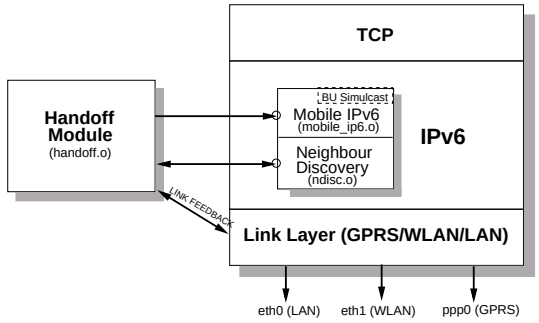

Figure 3. Mobile Client-based Module for RA caching.

One useful technique to eliminate t_d is to cache router advertisements. Using RA caching, we can eliminate detection time in handovers altogether. For example, during unanticipated handovers (refer table 2), the decision to handover typically depends on the application or user. For instance while moving from GPRS→WLAN, an application or user may want to complete an ongoing session over GPRS, and postpone the handover to WLAN. In such cases, handovers will not be initiated immediately upon reception of link-layer (L2) triggers from WLAN (i.e. anticipated handovers), but instead wait for a handover decision to be triggered by the application or the user. However, any RAs that are received during this period can still be cached so that when the decision to handover is taken, the detection time for RA lookup during handover execution is eliminated improving handover performance.

Note that for anticipated handovers in wireless overlay networks, RA caching has limited benefits. The benefit of RA caching is available only for the *upward* vertical handover case. As the network higher in the overlay (e.g. GPRS) is more omnipresent, RAs from GPRS can be cached a *priori*, and need not wait (or even sought) during upward vertical handovers leading to complete elimination of the detection interval.

We have implemented a client-based handover module in Linux 2.4 for RA caching that completely eliminates the inter-network handover detection time. The source code of the soft handover/RA caching module is publicly[4] available [20]. Figure 3 shows the handover module, which caches RAs from different networks. In this implementation, the handover module is hooked to the neighbour discovery module (`ndisc.o`) of the IPv6 stack. RAs from different networks are first received by neighbour discovery module of the IPv6 stack, which are then passed to handover module. The handover module then checks if an RA from the same network is already cached, and makes an update if it has expired. For testing, the handover module uses a periodic (user configurable) timer, that performs automatic handovers between two different networks (by calling the `ndisc_router_discovery` function of the IPv6 stack). In the current implementation, the handover module also uses MIPL's Mobile IPv6 module during handovers.

Scheme	t_d (in ms)
Fixed RA: 300-400ms	551ms ($\pm$33)
With RA Caching	1.21ms ($\pm$0.51)

Table 5. Optimization with and without RA caching.

We have evaluated the performance of our handover module for RA caching. In these tests, we allowed RAs to be cached in the WLAN as well as the GPRS network, and then forced handovers between GPRS and WLAN periodically, by setting the timer in the handover module, while simultaneously downloading a file from the server.

Table 5 shows the mean detection time with and without the handover module. We find that the amount of time

4 Source Code: http://www.cl.cam.ac.uk/users/rc277/soft.html

it takes to detect an RA from the handover module cache to be negligible (effectively zero but for processing the RA from the cache), typically of the order of few milliseconds, when compared to the mean detection time when not using the handover module. Thus, RA caching can lead to complete elimination of the handover detection time.

5.0.3. Client-Assisted Simulcast of Binding updates In IP-level handovers, registration time (t_r) required to update a network is typically limited by the RTTs to the HA and the CN, whichever is higher, assuming update process is not sequential (this is true for most Mobile IPv6 distributions). One technique that can further optimize this latency, is to ensure that BUs are also sent along the faster of the two networks during a handover. For example, in case of a WLAN→GPRS handover, the BUs to the HA and the CN are sent using the GPRS link. Unfortunately, sending BUs over GPRS entails high RTT due to the high latency of the GPRS link. The registration process in this case entails one GPRS link-RTT, which is a disadvantage in terms of performance. However, we could still improve performance further by simulcasting BUs over links that are faster, to speed up the registration process. Simulcasting not only optimizes the registration time, but also makes the binding update process more reliable.

By simulcasting updates over both the links (GPRS as well as WLAN), one can achieve much faster registration. Mobile IPv6 [6] offers the opportunity to simulcast BUs for fast registration. In this case, the first BU to CN or HA is sent as usual: the source address in the BU is the new interface address. For simulcasted BUs, the source address has to be modified, to be replaced by the old network interface address, along with the new interface address as a destination option (as `alternate-COA`) [6]. With this mechanism, the CN is able to create a binding entry between the new interface address and home address of the mobile node.

Scheme used	t_r (in secs) (WLAN→GPRS case)
Without BU Simulcast	2.99s (±0.395)
With BU Simulcast	1.36s (±0.231)

Table 6. Optimizing t_r with BU Simulcast.

We implemented BU simulcast over MIPL's Mobile IPv6 source code. Implementation required modifications to the MIPL source code, to allow it to simulcast on every upward vertical (WLAN→GPRS) handover. In table 6, we show the mean registration times with and without BU simulcast for over 10 handover runs. We find that BU simulcast is able to achieve better performance during WLAN→GPRS handovers, being able to perform fast registration using the WLAN link.

Note, however, that use of BU simulcasting can have implications on the overall security of the solution, as it involves certain security vs. performance trade-offs. The current Mobile IPv6 specification introduces the "return routability procedure" [6], which is necessary to verify the authenticity of the mobile node for establishing a new binding entry with the correspondent node.

5.0.4. Soft Handovers with RA Caching The handovers discussed thus far have been *hard* handovers; we take 'down' (stop listening) from one interface and then (almost simultaneously) 'up' (start listening) from the other. As a result, packets that were already in-flight or those destined (and those that already made it) to the previous network interface are, unfortunately, discarded. These packets have to be retransmitted by the source, which leads to reduced performance during handovers. However, handovers can be made *soft* to improve inter-network handover performance. Traditionally, soft handovers have been successfully exploited for link-layer handovers in cellular networks [15]. We introduce soft handovers at layer 3 (the network layer).

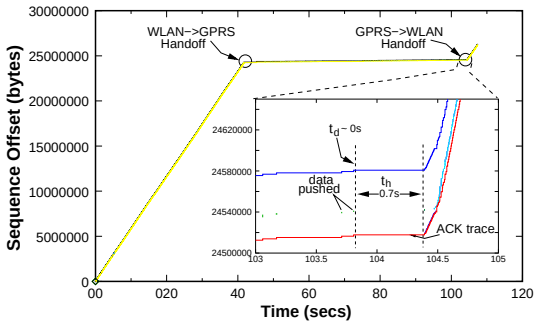

Figure 4. Soft Handover Performance with RA Caching. Total time for GPRS→WLAN handover takes about 0.7sec including that of IP level handover. Packets that arrive after the IP-level handover generate (dup)ACKs for out-of-order packets arriving from WLAN, consequently forcing the source TCP to go into fast-retransmit mode. Due to considerable differences between WLAN and GPRS link characteristics, incidence of out-of-order packets (24 outstanding packets in this case) is particularly high.

To achieve this, we had to modify our handover module (`handover.o` in figure 3) to support soft handovers, such

that after every handover, it allows all inflight IP-packets destined to the previous interface to be read, and be given to the application. Thus, it keeps receiving packets from the previous network interface, while at the same time allows for complete migration (registration) of IP points of attachment, before starting to send packets from the new interface.

We have used this module to evaluate the performance of soft vertical handovers. We initiated a file transfer and allowed the soft handover module to periodically handover between GPRS and WLAN. Figure 4 shows one such trace of a handover using the soft handover module that also performs RA caching. The close-up plot shows that RA caching is able to completely eliminate the detection time, while use of soft handover would allow reading all incoming TCP packets from the old interface (GPRS), and ACK all of them from the new interface (WLAN). This process keeps the TCP self-clocking going, and the source transmitting packets even after the handover. As evident from the plot, this results in dramatic improvement in TCP performance with only 0.7s required for the overall handover.

A point to note here is that the source TCP enters fast-restransmit mode due to the dupACKs that are generated by the mobile client after a handover to WLAN. This leads the source TCP to become less aggressive (congestion window halved). A less aggressive TCP source is still preferable since the state of the new network is unknown. If the new access network is congested, an aggressive TCP source can further aggravate matters. In constrast, less aggressive TCP source can probe the new network conditions and adjust its data rate without causing much packet loss.

Due to the nature of the coverage offered in wireless overlay networks, soft handovers are typically applicable for the case of downward vertical handovers (e.g. GPRS→WLAN). Since GPRS coverage is virtually pervasive, packets from the GPRS interface can still be read even after a handover to WLAN, ensuring that inflight packets in GPRS are not lost. For a mobile user moving away from the diminishing coverage of WLANs, efficacy of soft handovers when applied to upward vertical handover remains questionable in high mobility environments. As in this case, it is never easy to determine if inflight packets arriving from WLAN could be saved. We are continuing to explore this issue further.

6. Experiences and Challenges Ahead

Our experiences with vertical handover optimization schemes are briefly summerized in table 7. These experiences suggest that a unified network-layer solution based on Mobile IP(v6) would need further support in order to sufficiently hide the impact of vertical handovers on performance. This is particularly true for streaming media applications, since disruptions during vertical handovers can cause perceptual degradation in the overall service quality to mobile users. Therefore, exploiting multi-layer (including cross-layer) optimizations here is important for benefitting performance.

Network-layer handover optimization schemes as suggested in this paper can aid the Mobile IP(v6) protocol. Schemes such as Fast RAs, RA Caching and Binding Update simulcast are some of the techniques that can be used to achieve better handover performance. Additionally, use of network layer soft handover techniques (of aggregating bandwidths during handovers) by exploiting network diversity in an overlay environment yields important benefits during vertical handovers.

An interesting aspect of our research is that it allows deeper investigation into many performance critical issues related to vertical handovers, and the resulting impact on transport as well as application performance due to the underlying network heterogeneity. While we have attempted to address some of the more important issues, there are others yet to be addressed.

Our research also poses other questions:

- How sensitive are the results to a loosely-coupled experimental testbed set-up?
- How well does Mobile IPv6 fare when compared to other techniques (for e.g. SIP - Session Initiation Protocol)?
- Can the soft handover approach applied at the network layer be used for high mobility environments? What quantitative benefits can we achieve using a similar approach for streaming media flows?
- Do schemes that benefit TCP also imply commensurate benefits for HTTP? How does that translate quantitatively for web performance?
- How can we best adapt applications e.g. web browsing, streaming media performance in the presence of vertical handovers?
- Can we define a standard architecture (preferably decentralized) that can allow universal roaming, charging and security in such environments?

Our ongoing research explores many such issues further.

7. Related Work

UC Berkeley's BARWAN project evaluated issues related to vertical handovers between Metricom Richochet and WaveLAN [14, 17]. Vertical handovers in BARWAN make use of a multicast address in the mobile host (the care-of-address) to receive advertisements from potential access points in an overlay. Furthermore, fast beaconing and packet/header doublecasting is used to optimize such handovers [14]. In [8], H. J. Wang *et al.* present a

Handover Optimization applied	Summary of the Experimental Evaluation
Fast RAs	Reduces 'detection time' during handovers. Not significant reduction in RA interval over GPRS possible – involves performance vs. network overhead trade-off.
Client-based RA Caching	Eliminates handover detection time. Exploits network diversity to 'pro-actively cache' RAs from the network, before finally migrating the IP point of attachment.
Client-assisted BU Simulcast	Reduces 'registration time' during *upward* (e.g. WLAN→GPRS) vertical handovers. Involves performance – security trade-off.
Soft handovers with RA Caching	Major improvement in handover performance possible. But causes dupACKs to be generated by the client during *downward* vertical handover, forcing the source into Fast Retransmit mode.

Table 7. Summary of Network-layer Handover Optimizations

policy-enabled handover system and show handover latencies around 9s and 26s with Metricom and GSM Cellular network, respectively. Their approach makes use of reverse tunneling to the home agent to avoid packets being dropped at the firewall.

In a recent study, M. Buddhikhot *et al.* in [12, 13] discuss two architectures: a tightly-coupled and a loosely-coupled integration architecture between 3G and WLAN. They show the design and implementation of a gateway, called IOTA, that combines several useful features to loosely integrate CDMA2000 and IEEE 802.11b-based networks. In [11], Magalhaes and Kravets provide a transport layer (TCP) solution for bandwidth aggregation using multiple WLAN interfaces, where the goal is to sum the bandwidth from these interfaces and offer it as a single large pipe to the end user. Similarly, MOPED architecture [3] offers higher capacity by adapting the home agent in Mobile IP to support aggregation of multiple links at the network and transport layers. SCTP (IETF RFC 2960) [21], PTCP [9] and [10] also specify bandwidth aggregation at the transport layer. These solutions are quite similar in spirit to our layer 3 based soft handover approach discussed in the paper.

Mobility remains a hot topic even in the IETF. Apart from Mobile IP/Mobile IPv6, they have two main protocols to manage mobility - Heirarchical MIP/MIPv6 [7] and a Fast Handover Protocol [18]. The main driver in these proposals, which are quite similar to ours, is to minimize the handover latency. An approach to improve performance here can also make use of a scheme similar to that used in micro-mobility protocols such as Cellular IP [2] or HAWAII [19]. Micro-mobility solutions are broadly aimed at improving mobility at the *subnet* level of a network domain. In [1], A. Campbell *et al.* provide a nice survey of such micro-mobility protocols.

8. Conclusions and Future Work

In this paper we presented our practical findings and the means of enabling transparent mobility in heterogeneous environments. By conducting vertical handovers between GPRS cellular and 802.11b-based WLAN, we have analysed packet traces to determine the latencies of the steps in the handover process, and have examined the effects handover can have on active TCP flows. We also proposed and evaluated network-layer optimizations techniques to improve performance during vertical handovers. We highlight the main findings of our study as follows.

- The disparity in the link-layer characteristics (especially round trip times and bandwidths) between GPRS cellular and 802.11b WLAN link contributes significantly to the poor vertical handover performance. Appropriate handover optimizations are necessary to improve performance.
- The presence of deep buffers in GPRS can aggravate performance of certain applications (e.g. web flows, streaming media). For transport TCP flows, it may artificially inflate the source RTT (and RTO) leading to reduced performance during handovers.
- In terms of performance, soft handovers with Router Advertisement caching provide the most significant benefit. However, the use of soft handovers can lead to out-of-order delivery of data packets (e.g. during GPRS→WLAN handover).
- Experiences with the testbed demonstrate that by applying appropriate network-layer optimizations, we can effectively migrate active transport TCP flows with reduced disruption during vertical handovers.

Our work was conducted over cellular GPRS networks with high latency wireless links, significant jitter, and low data rates. However, the results are also quite applicable to other networks that have similar properties e.g. CDMA 1xRTT/2000.

Besides open issues discussed earlier, we are also looking at ways of taking this work further from other angles. Specifically, power consumption in a mobile device is often closely linked to the amount of data being transported

over active interface(s). This in turn can impact the way in which is a handover is performed – hard or soft handovers. Additionally, economic or pricing models applied by network operators operating cellular networks and WiFi hot-spots provides an important challenge. Our ongoing research considers sophisticated policy-based models for mobile devices to make such performance and cost optimizations possible.

Acknowledgements

A short abstract summary of this work was presented as a poster in ACM Mobicom 2003. We gratefully acknowledge the contribution of Leo Patanapongpibul in setting up the Mobile IPv6 experimental testbed and tcptrace+ source code.

References

[1] A. T. Campbell and J. Gomez-Castellanos. Ip micro-mobility protocols. *ACM Mobile Computing and Communications Review*, 2(1), 2001.

[2] A. T. Campbell, et al. Design, implementation and evaluation of cellular ip. *IEEE Personal Communications Magazine*, 7(4), 2000.

[3] C. Carter and R. Kravets. User devices cooperating to support resource aggregation. In *The 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002)*, 2002.

[4] R. Chakravorty and et al. On inter-network handover performance using mobile ipv6, technical report. http://www.cl.cam.ac.uk/coms/publications.htm. July 2003.

[5] R. Chakravorty and I. Pratt. Performance issues with general packet radio service. *Journal of Communications and Networks*, 4(2), Dec. 2002.

[6] D. Johnson, C. Perkins and J. Arrko. Mobility support in ipv6. In *IETF draft (draft-ietf-mobileip-ipv6-24.txt)*, 2003.

[7] H. Soliman, et al. Hierachical mipv6 mobility management. In *IETF draft (draft-ietf-mobileip-hmipv6-05.txt)*, 2001.

[8] H. Wang and R. H. Katz. Policy-enabled handoffs across heterogenous wireless networks. In *In Proceedings of 2nd IEEE Workshop on Mobile Computing and Applications (WMCSA 1999)*, 1999.

[9] H-Y. Hsieh and R. Sivakumar. A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts. In *Proceedings of the ACM Mobicom*, 2002.

[10] K. Chebrolu and R. Ramesh. Communication using multiple wireless interfaces. In *Proceedings of IEEE WCNC*, 2002.

[11] L. Magalhaes and R. Kravets. Transport level mechanisms for bandwidth aggregation on mobile hosts. In *Proceedings of the 9th International Conference on Network Protocols (ICNP 2001)*, 2001.

[12] M. Buddhikot, et al. Design and implmentation of a wlan/cdma2000 interworking architecture. *IEEE Communications Magazine*, 41(11), Nov. 2003.

[13] M. Buddhikot, et al. Integration of 802.11 and third-generation wireless data networks. In *Proceedings of the IEEE INFOCOM*, 2003.

[14] M. Stemm. Vertical handoffs in wireless overlay networks. *ACM Mobile Networks and Applications (MONET)*, 3(4), 1998.

[15] N. D. Tripathi, J. H. Reed and H. F. Vanlandingham. Handoff in cellular systems. *IEEE Personal Communications Magazine*, Dec. 1998.

[16] C. E. Perkins. Mobile ip. In *IEEE Communications Magazine*, May 1996.

[17] R. H. Katz and E. C. Brewer. The case for wireless overlay networks. In *SPIE Multimedia and Networking Conference (MMNC'96)*, 1996.

[18] R. Koodli et al. Fast handovers for mobile ipv6. In *IETF draft (draft-ietf-mobileip-fast-mipv6-05.txt)*.

[19] R. Ramjee, et al. Hawaii: A domain-based approach for supporting mobility in wide-area wireless networks. In *Proceedings of the ICNP*, 1999.

[20] radvd+, tcptrace+, Vertical Handoff Module. http://www.cl.cam.ac.uk/coms/software.htm.

[21] R. Stewart and et al. Stream control transmission protocol. In *IETF Request For Comments (RFC 2960)*, 2000.

[22] T. Narten, E. Nordmark and W. Simpson. Neighbor discovery for ip version 6 (ipv6). In *Request for Comments (RFC 2461)*, 1998.