Is Latency the Real Enemy in Next Generation Networks?

David N. Cottingham, Pablo A. Vidales
Computer Laboratory, University of Cambridge,
Cambridge CB3 0FD, United Kingdom
{david.cottingham, pablo.vidales}@cl.cam.ac.uk

Abstract

This paper positions the idea that apart from the network vertical handover latency effects on the TCP/IP stack, there is another challenge that shadows ubiquitous networking. The TCP-connection adaptation time required when roaming between two disparate wireless technologies can be even longer than the total handover period. Thus, the impact of the adaptation time needs to be minimised and considered when dealing with seamless networking in heterogeneous environments. We present an experimental testbed that has been used to characterise the latency during vertical handover. Later, we introduce the concept of adaptation time *(t_a) and show the experimental value of t_a, obtained from the collected traces. Finally, we discuss the effects of t_a on the TCP/IP stack during heterogeneous handovers. We conclude the paper proposing some solutions to minimise the adaptation time.*

1. Introduction

In a world where always-on, wireless connectivity is becoming more prevalent, many devices are now emerging with multiple network interfaces. These allow such terminals to utilise different networks depending on their availability. For example, a mobile handset might normally use a GPRS or 3G connection to access Internet resources when on the move, but when in a field of wireless LAN (WLAN) coverage it could take advantage of the increased bandwidth. With an ever increasing number of different access technologies, such as 802.11b/g, WiMax [8], and wired LAN, devices must be capable of selecting and adapting to different networking interfaces whilst on the move.

The Mobile IP version 6 (MIPv6) protocol provides support for such attachment point migration. A mobile node may start a communication with a correspondent node on one interface, and subsequently continue that connection using a different base station on the same network, (horizontal handover), or by migrating the connection to a different physical interface (vertical handover). Such vertical handovers are of particular interest in forthcoming heterogeneous networks.

Perhaps the most significant observation relating to network heterogeneity is that different access technologies have very disparate characteristics. For example, in terms of bandwidth we may range from wired LAN at 100 Mb/s or 1 Gb/s, through WLAN networks of 56 Mb/s and 11 Mb/s, past 1 Mb/s UMTS links, and on to GPRS with perhaps only tens of Kb/s. Meanwhile the variation in the round trip time on the various links is enormous, with GPRS latencies of the order of 1000 ms, WLAN 10 ms, and 1 ms for LAN. In addition, packet loss characteristics differ in both magnitude and distribution. Such high variability will require intelligent handover policies and methods to ensure minimal user experience disruption. In the future, significant revenue will be derived from data traffic over cellular networks: to enable this we must provide quality of service guarantees. To do so it is imperative that we provide seamless handovers.

Inevitably, any handover incurs a penalty in terms of the time taken for the mobile node to register with the new base station or network (the *handover latency*). This handover period has been analysed in previous work on MIPv6 handovers [16], and various techniques have been developed to mitigate it. Work has included router advertisement caching [4], Fast Handovers for Mobile IPv6 (FMIPv6) [10] and Hierarchical Mobile IPv6 (HMIPv6) [15]. FMIPv6 aims to decrease the total latency to almost only the Layer 2 handover time. This approach has been shown to perform well in intra-technology (i.e. horizontal) handovers [6], [2]. The HMIPv6 approach is designed to reduce the degree of signalling required and to improve handover speed for mobile connections by managing local mobility in a more efficient way.

Other initiatives relating to vertical handovers in heterogeneous environments include the Moby Dick project [6, 11], using FMIPv6, which has been suc-

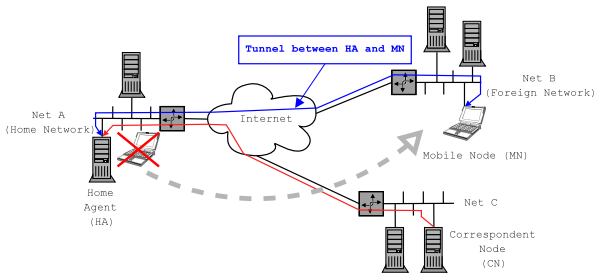

Figure 1: Basic operation of Mobile IPv6

ceeded by the Daidalos project [1]. The main differentiator from this work with ours is our usage of a real operator's live 3G network. The Nomad project [2] [7] (terminated June 2004), investigated roaming using MIPv4 (i.e. assuming the presence of foreign agents), compared to our analysis of MIPv6 for 4th generation networks.

Such work has enabled an open connection to continue with relatively minor interruption. However, this interruption is defined as the time for which no traffic can reach the mobile node from the correspondent node due to link switching delay and IP protocol considerations (see [9]). It does not consider whether the new connection is being utilised at all efficiently post-handover. In this paper we show that this *adaptation time* is significant compared to the total period required for detection, connection and registration, and point to ongoing research efforts to reduce its significance.

2. Mobile IPv6: basic functionality

While a Mobile Node (MN) is connected to its Home Network (HN), i.e. the network where its Home Address (HoA) is located, no special mode of operation is needed and packets are forwarded (using normal IP routing) between the mobile node and any other node it is communicating with (the Correspondent Node).

When a MN is connected to a network other than its home network (i.e. it is visiting a foreign network) the MN acquires an IPv6 address belonging to the address space of the Foreign Network (FN) it is visiting, called the Care-of Address (CoA). The MN announces its CoA (by sending a Binding Update message, BU) to an special entity, called the Home Agent (HA) that is located in the MN's home network. Furthermore, this special router (HA) "represents" the mobile terminal when it is absent. A home agent usually serves all mobile nodes of a home network. The HA traces all the MN's movements, and keeps the mapping between home addresses and care-of addresses—using the BU sent by the nodes—in memory (in the binding cache).

The home agent intercepts the packets sent to the MN's home address while the MN is away from its home network and establishes a bidirectional tunnel with the MN's CoA, in order to redirect these packets to the MN's current point of attachment to the Internet. The MN also uses this tunnel to send its traffic to the Correspondent Node (CN) avoiding in this way any ingress filtering. This basic functionality of Mobile IPv6 is depicted in Figure 1.

Furthermore, MIPv6 also defines a Route Optimisation (RO) procedure to avoid the suboptimal routing problem caused by the use of bidirectional tunnelling through the HA. This procedure enables the MN to also send binding updates to the CNs. Packets sent by the MN then have the MN's CoA as source address, but also carry a special IPv6 *home address destination option*, containing the MN's home address, allowing a CN to use this address as the source address when delivering the received packets to its upper layers (i.e. mobility is transparent to the layers above IP). In the reverse direction, the CN sends the packets addressed to the MN's CoA, but also inserts a Type II routing header—see Section 15.9 of [9]—with the MN's HoA as a unique next hop. In this way, the MN can also manage the mobility in a transparent way with respect to those layers above IP.

From this scenario, we can see that Mobile IPv6 is suitable for providing support for roaming between networks and that it can be used from an Ethernet network to a wireless network, between homogeneous networks and, more relevant to this work, between diverse access technologies. Nevertheless, we note that Mobile IPv6 has been conceived to support macro-mobility, and it is less suitable for micro-mobility, in which, for example, a host moves between two cells of a wireless LAN. In the latter case, the mobility can be more efficiently implemented by using link layer mechanisms.

In this document, the use of Mobile IPv6 for macro-mobility in 4G communication systems is analysed, where mobile nodes will usually roam between disparate wireless technologies to obtain ubiquitous connectivity. We use the term *horizontal handover* to mean a handover between two networks of the same access technology, whilst a *vertical handover* is between two networks of distinct technologies. A *downward handover* takes place from a network with relatively ubiquitous coverage to another of lesser reach, but most probably greater quality of service (e.g. GPRS⇒WLAN), whilst an *upward handover* is the reverse.

[1] http://www.ist-daidalos.org/
[2] http://www.ist-nomad.net/

3. Experimental setup

To emulate a next generation (4G) integrated networking environment, our experimental testbed setup consists of a loosely-coupled, Mobile IPv6-based GPRS-WLAN-LAN testbed as shown in Figure 2. The cellular GPRS network infrastructure currently in use is Vodafone UK's production GPRS network. The WLAN access points (APs) are IEEE 802.11b APs. Our testbed has been operational since March 2003, its 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 the GGSN are co-located in a single CGSN (Combined GPRS Support Node). A well provisioned virtual private network (VPN) connects the Laboratory 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 4G integrated network, mobile nodes (e.g. laptops) connect to the local WLAN network and also simultaneously to GPRS via a Phone/PCCard modem. The GPRS cards in use are classified as "4+1", meaning that they are able to simultaneously listen to four downlink channels, whilst using a single uplink channel. Assuming the coding scheme is the commonly used CS-2, this corresponds to a maximum bandwidth of 13.4 Kbit/s per channel [13]. The Mobile Node's MIPv6 implementation is based on that developed by the MediaPoli project [14], chosen for its completeness and open source nature.

4. Latency characterisation

The period of connection interruption that takes place on a handover can be partitioned into multiple logical stages. These are as follows:

- *Detection Period (t_d).* The time taken by the mobile terminal to discover the available network(s), using link layer signalling or the network layer detection mechanism.

- *Configuration Interval (t_c).* The interval from the moment a mobile device receives a Router Advertisement, to the time it takes to update the routing table and assign its new Care-of Address based on the received network prefix, including the Duplication Address Detection delay (DAD). This interval depends on the terminal characteristics (e.g. memory, processing power, etc.).

- *Registration Time (t_r).* This elapses between the delivery of the Binding Update to the Home Agent and correspondent nodes, and the reception of the first packet at the new interface – with the new Care-of Address as the destination address.

- *Adaptation Time (t_a).* When dealing with vertical handovers at the transport level, we need to consider t_a in the total handover latency. This delay only occurs when the mobile host adapts the connection to the new technology at the transport layer, adjusting the TCP state machine parameters (e.g., congestion window size, timeout timers, etc.), due to differences in the link characteristics. Thus for the case of TCP transmissions, the transport-layer latency (T_t) is equivalent to the network-layer latency (T_n) plus the adaptation component (t_a), as shown in Figures 4 and 3.

The network-layer latency is given by:

$$T_n = t_d + t_c + t_r \tag{1}$$

where
t_d is a random variable with probability $p(t_d)$:

$$p(t_d) = \frac{1 - \int_0^{t_d} p_{t_{RA}}(t)}{\overline{t_{RA}}} \tag{2}$$

$$p_{t_{RA}}(t) = \begin{cases} \frac{1}{RAint_{MAX} - RAint_{MIN}} & \text{if } RAint_{MIN} \\ & \leq t \leq RAint_{MAX} \\ 0 & \text{otherwise} \end{cases}$$

$$t_c = t_{DAD_{LL}} + t_{DR} + t_{CoA} + t_{DAD_{NL}} \tag{3}$$

$$t_r = t_{RR} + t_{BU} \tag{4}$$

t_{RA} = Time period between consecutive Router Advertisements (RAs),
$RAint_{MAX}$ = Maximum RA interval (i.e. time between two consecutive Router Advertisements),
$RAint_{MIN}$ = Minimum RA interval (i.e. time between two consecutive Router Advertisements),
$t_{DAD_{LL}}$ = Time taken for Duplicate Address Detection for link local address,
t_{DR} = Time taken for Default Router configuration,
t_{CoA} = Time taken for configuring the new Care of Address (CoA),
$t_{DAD_{NL}}$ = Time taken for Duplicate Address Detection for CoA,

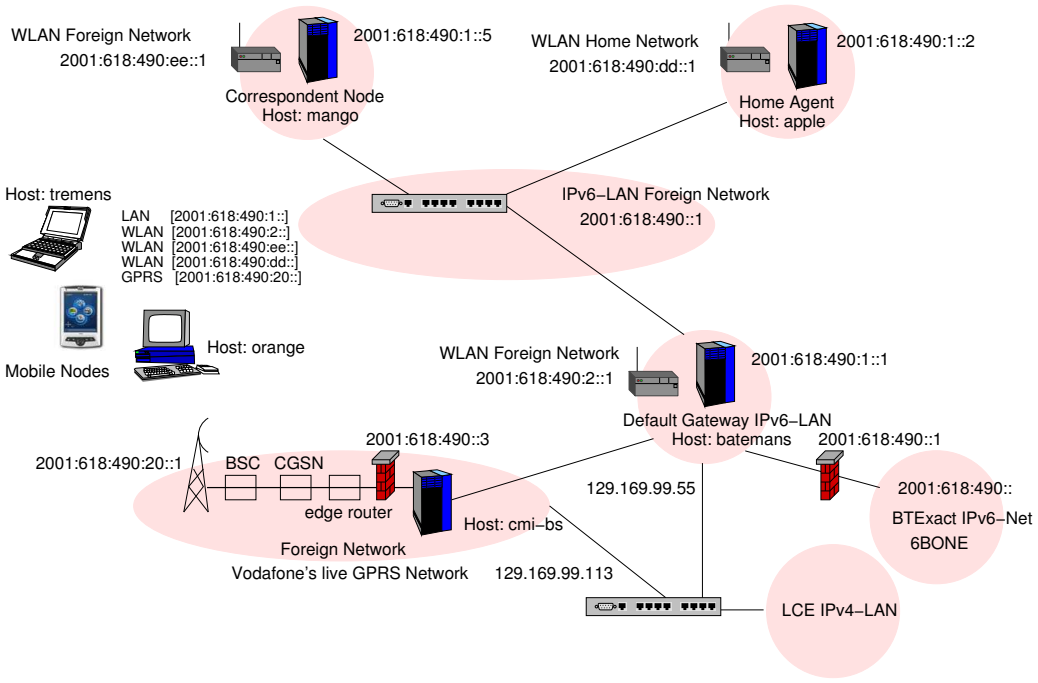

Figure 2: Experimental setup for 4G test bed.

t_{RR} = *Time taken for entire Return Routability procedure (= 1.5 RTTs in the best case scenario—i.e. no packet losses)*

t_{BU} = *Time taken for update of the binding in the Home Agent (HA). This can be done simultaneously while updating the CNs.*

The overall latency is found by summing the delays to discover the new network (t_d), to build the Binding Update message using the prefix from the new access router (t_c), and to register the recently formed CoA with the Home Agent and correspondent nodes (t_r), as shown in Equation 1. The network discovery delay depends on the movement detection mechanism, but if the generic L3 Neighbour Discovery based mechanism is used by the MN, this time depends mostly on the Router Advertisement frequency and also on the policy that the MN uses to consider a router unreachable. Generally, the MN may use the Advertisement Interval (if included in the Router Advertisements received by the MN) field as an indication of the frequency with which the current default router is sending these messages. Therefore, if during this time interval (which indicates the maximum time between two consecutive RAs) the MN does not receive any new RA from the current default router (i.e. at least one RA has been lost), the MN can use the event of losing a certain number of RAs as a possible L3 handover indication (in MIPL, for example, a figure of 2 lost RAs is used, triggering the MN to send Router Solicitation messages on all the available interfaces to discover new reachable routers).

The configuration time depends on the terminal characteristics, however, some methods have been proposed to reduce this delay such as avoiding the DAD mechanism to minimise disruptions when roaming between access routers [1]. The last component (t_r) is dictated by the round trip time (RTT) of the network used for the registration process, as shown in Equation 4.

5. Adaptation delay component

We have investigated t_a for various possible scenarios in our test bed, and include the results for the particular cases of LAN to GPRS and WLAN to GPRS handovers, as these are the types that will most commonly be encountered. This type of handover is likely to be the common case, as users move from comparatively small areas of high bandwidth coverage to more ubiquitous but lower capacity access networks.

We define t_a to be from the point at which the mobile node receives the first data packet on the new interface from the correspondent node, to the point where the new interface's throughput first reaches the value of the average throughput it achieves over the lifetime of

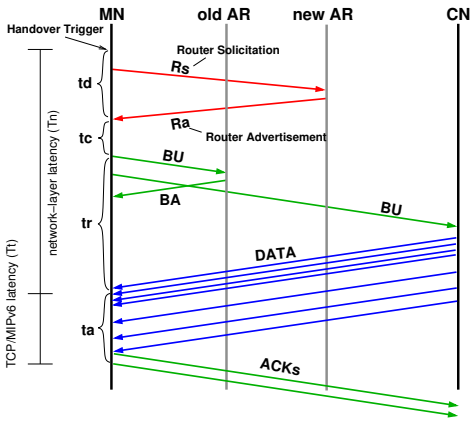

Figure 3: Latency partition between the Network & Transport layers.

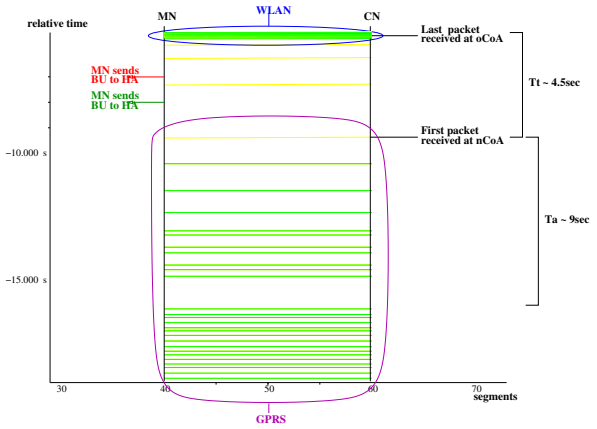

Figure 4: Adaptation delay during heterogeneous handovers.

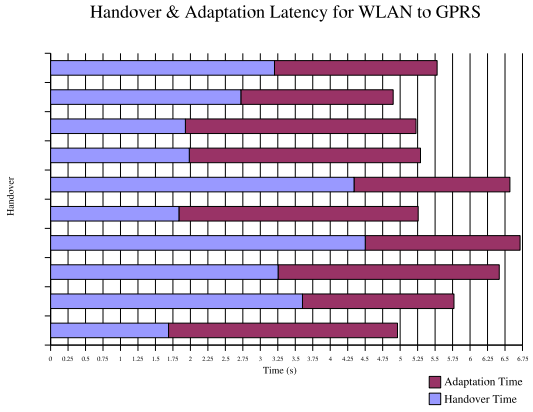

Figure 5: Total handover time for WLAN ⇒ GPRS, split into T_n and t_a.

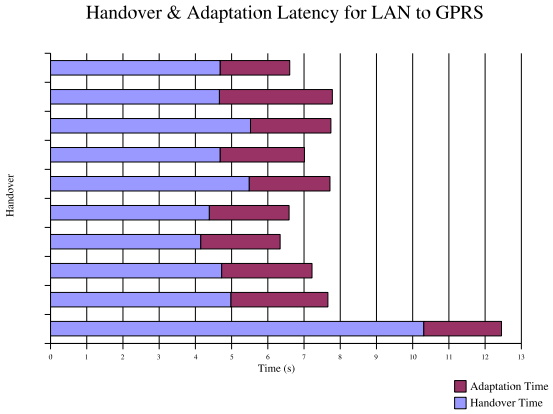

Figure 6: Total handover time for LAN ⇒ GPRS, split into T_n and t_a.

the connection. Values are calculated by measuring the quantity of data sent in each 0.1 second interval. This value was chosen as sufficiently large to filter out high frequency variation in the trace, whilst exhibiting the highly dynamic nature of the connection's throughput. The measurement tool analyses packet traces, and for each packet checks whether the elapsed time is greater than the interval. If so, a throughput value is calculated. When this technique is applied to GPRS data, which has high RTTs, we find that there are long pauses whilst the sender waits for the receiver to acknowledge the data. Consequently the next packet in the trace can be multiple intervals later, resulting in fewer points than might be expected on the throughput graph. We regard this as a more realistic view than counting data per strict time interval, as such a method would produce a trace that implied throughput fell to zero frequently, unless the interval was very large, in which case the trace would not be a useful indicator of the variation in throughput.

The values obtained for each of the intervals are averaged to give the mean throughput of the connection over the entire test. We discount the final interval point due to its throughput being distorted by the test being ended. Due to the highly variable nature of GPRS links it is not realistic to attempt to define a threshold throughput value for all tests: instead we do so on a trace by trace basis. Using simple linear interpolation, we find the time at which the throughput exceeds the mean of the trace. This is the adaptation time.

Using the test bed described in Section 3., we monitored the throughput of a TCP connection from the correspondent node to the mobile node, for 20 handovers

from WLAN to GPRS and another 10 for LAN to GPRS. An example of the averaged throughput of a connection directly after the network handover is complete is shown in Figure 7.

We analyse the case of a TCP handover because studies have shown that 85% of the traffic in the Internet is generated by TCP connections [12]. It is therefore essential that research be carried out into ensuring that such connections can be seamlessly migrated on vertical handovers.

For each handover, we measured T_n and t_a, obtaining the aggregated results shown in Table 1. Figures 5 and 6 show that the adaptation time is a significant fraction of the total handover time. Clearly this increased connection interruption time will have a significant effect on any applications that require even a mediocre quality of service.

We have also investigated downward handovers from GPRS to WLAN and GPRS to LAN. The results in both cases indicate that the adaptation time is negligible, and therefore we do not include the data here. However, we note that TCP performance is affected by reordering and duplicated packets. This will be discussed in Section 6.3..

6. Impact of adaptation component on the TCP/IP stack

In this section we examine three cases: upward handovers (WLAN⇒GPRS and LAN⇒GPRS), downward handovers (in particular GPRS⇒WLAN), and finally consider the effects that using a soft handover mechanism has on the TCP stack.

6.1. Upward handovers

From the data presented in Section 5. it is clear that the adaptation time is significant compared to the handover time for a TCP connection. There are multiple factors that contribute to this being the case. In the following discussion we will do not consider the initial packet loss that is inherent in a hard handover (i.e. when the old interface is disabled before the new one is enabled, thereby losing packets that are still in flight). Instead, we examine the situation after those lost packets have been retransmitted.

WLAN ⇒ GPRS	Mean	%	σ	Min.	Max.
Handover time (T_n)	2848	49.3	1127	1103	4782
Adaptation time (t_a)	2786	50.7	628	2142	4022
Total latency	5635	100	881	3802	7285
LAN ⇒ GPRS	Mean	%	σ	Min.	Max.
Handover time (T_n)	5356	68.5	1791	4146	10303
Adaptation time (t_a)	2355	31.5	337	1924	3119
Total latency	7711	100	1751	6339	12452

Table 1: Latency partition for vertical handovers using MIPL. The tables show the average value, in milliseconds, for WLAN to GPRS and LAN to GPRS respectively.

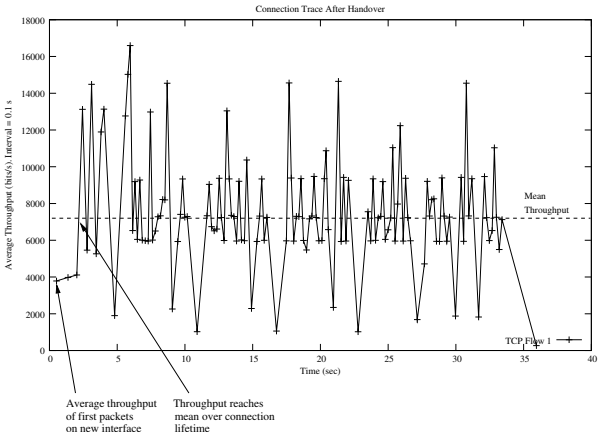

Figure 7: Connection throughput on GPRS link after handover from WLAN interface.

TCP calculates Round Trip Time (RTT) values by subtracting the transmission time of certain packets from the reception time of their corresponding acknowledgements. An exponentially weighted moving average is taken to smooth any perturbations. When performing a vertical handover from a low latency to a high latency access network, the sender's TCP estimate of the RTT is excessively low. Timeouts occur whilst acknowledgements are still in flight on the new high latency interface. This results in the sender perceiving packet loss, and retransmitting already received data.

As the acknowledgements are still in flight, the sender has no way to "clock the link". Hence, fast retransmit is not possible, and the congestion window (`cwin`) must fall to 1 Maximum Segment Size (MSS). Subsequently, when the acknowledgements are eventually received, the congestion window will be increased by TCP's slow start phase, up to a maximum of the old value of `cwin`/2. However, this previous value is that for the high bandwidth WLAN or LAN connection, which is far greater than a typical value for a GPRS link. Therefore it is likely that the slow start algorithm will overshoot, causing actual packet loss (note that this is did not take place in the trace shown in Figures 7 or 8).

Figure 8 shows a trace for handover from a LAN to a GPRS interface. This clearly shows the initial relatively low throughput corresponding to the small value of `cwin`.

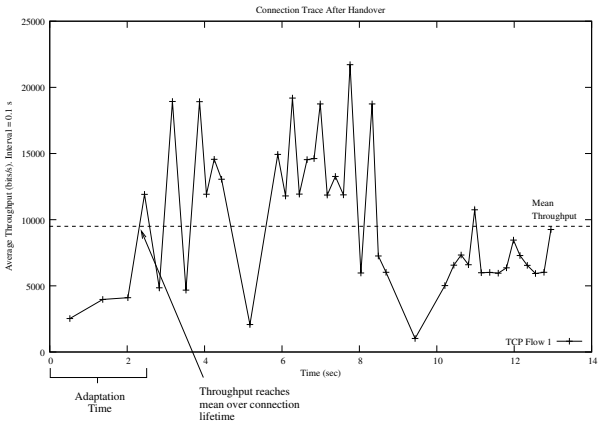

Figure 8: Connection throughput on GPRS link after handover from LAN interface.

Once the slow start phase has ended, TCP resumes its linear increase phase, incrementing the congestion window by one MSS per RTT, which on a LAN connection would result in a relatively smooth throughput trace. However, GPRS has such variable RTTs that the throughput of the connection oscillates significantly.

Figure 6 shows that the fraction of the total handover time due to t_a is consistently less for the LAN $\Rightarrow$ GPRS scenario (on average 31.5% of the handover is t_a, with a σ of 6.1), than for WLAN $\Rightarrow$ GPRS (50.7%, σ of 13.9). Comparing the values of t_a between the two types of handover in Table 1, we see that they are approximately similar, as would be expected, given that the congestion window falls to 1 MSS in both cases (though note that the RTT estimates would be more disparate in the LAN case, compared to WLAN). However, the values of T_n differ significantly. On examining the packet traces, it emerges that the extra time that the LAN handover takes elapses between the binding acknowledgement being sent from the HA to the MN, and the route optimisation process being initiated (i.e. a few packets are tunneled via the HA in the meantime). An ongoing research problem is to discover the reason for this extra time.

6.2. Downward handovers

When performing a vertical handover from a higher latency access network to one of a lower latency, such as GPRS$\Rightarrow$WLAN, TCP's RTT estimate is evidently overconservative. This implies that the if the handover itself requires a period of time less than the timeout value, the sender will not assume that there has been packet loss, and hence `cwin` will not be decreased. In practice, with hard handovers, packets in flight to the old interface are lost, and therefore selective acknowledgements received from the new interface. The sender is able to use such acknowledgements (also known as *duplicate* acknowledgements), to detect the loss before a timeout occurs, and hence can make use of TCP's fast retransmit algorithm to resend the missing packet(s). This then allows the sender to decrease their `cwin` to only half of the previous value (rather than to only one MSS), ensuring that the rise time to a suitable value for the new interface is not as prolonged. Hence t_a for such handovers is negligible, even when not using a soft handover mechanism.

However, this does not mean that such downward handovers are without any issues. Of significant interest is the re-ordering of packets, due to the differing latencies of the two networks. New data may begin to be received on the lower latency interface whilst packets that precede it in the sequence number order may still be in flight on the higher latency interface. The effects of this are briefly outlined in Section 6.3..

6.3. Soft handovers

Figure 9 shows an example trace collected using a modified version of MIPL, showing the benefits and drawbacks of soft handovers in vertical scenarios; this figure shows the WLAN-GPRS-WLAN case, sending TCP traffic between the CN and the MN. A soft handover takes place when the old interface is still able to receive packets that are currently in flight, even when the new interface is fully operational. As evident in the plot (centre), performing soft handovers leads to dramatic reductions in handover latency (there is no packet loss during the handover, although packet reordering can occur).

The MN is connected to the WLAN (upper left corner) and initiates a handover to the cellular network. However, the source (CN) continues sending packets destined to the old interface for approximately 1ms (until SND.UNA+SND.WND-SND.NXT reduces to zero). The mobile terminal receives the on-the-air packets through the old interface and responds sending ACKs using the new interface, which is, in this scenario, much slower than the previous one. The CN times out and starts retransmitting these packets, whereas the MN sends SACKs to the source because it is receiving duplicated packets (see lower left corner plot). We can observe that when the MN performs an anticipated upward handover (WLAN-GPRS), the disparity in the link characteristics affects the TCP connection, retransmissions occur, and the benefits of soft handovers are less dramatic.

For the other case (right side), the MN is connected to the GPRS network and starts a handover to the WLAN. Some packets are on-the-air, until the CN realises that

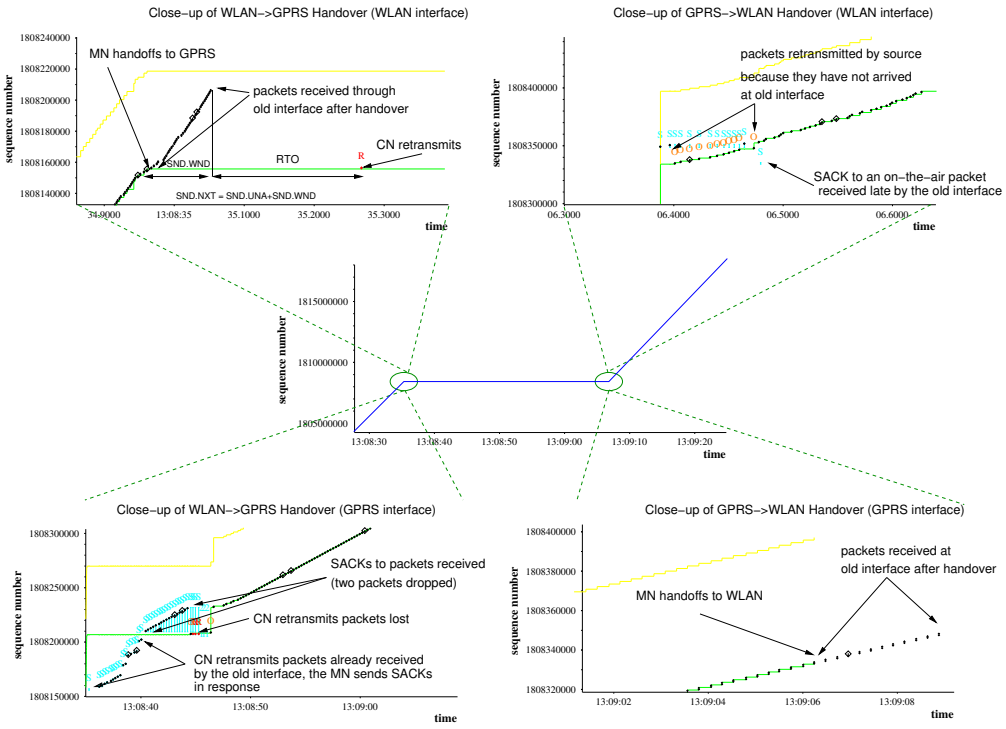

Figure 9: Impact of t_a during heterogeneous soft handovers.

the MN has moved – in this case, the registration process delay is very small, compared to the time that it takes for the packets to arrive at the old interface (lower right plot). The source starts retransmitting the on-the-air packets because these have not arrived to the MN due to the RTTs in the GPRS network. Finally, the MN sends some SACKs as it receives delayed on-the-air packets on the old interface, which have been already retransmitted by the source. We can see again that because of the huge differences between networks, the source retransmits packets that are already at the MN, wasting bandwidth and reducing the benefits of soft handovers. We are exploring TCP modifications to reduce the impact of link disparities in soft handovers.

The deployment of this kind of handover mechanism is crucial to offer real-time services, such as VoIP—which is one of the most promising services in 4G networks—and video streaming. Thus, handover mechanisms that retain on-the-air packets are critical for seamless roaming. In addition, mechanisms to prevent packet reordering are also essential for real-time applications that are two way (such as conversations or video-conferences), which cannot use a significant degree of buffering.

7. Potential solutions

In order to reduce the time taken for TCP connection adaptation, it will be necessary to modify how TCP calculates both RTTs and the congestion window when a connection is handed off to a new interface. Our current research is outlined below.

One possible option is to allow a "guessed" value for the new interface's RTT to be injected into TCP's estimation of the RTT, before the connection is handed off. This would not prejudice packets travelling on the old interface[3], and would ensure that on the new, higher latency interface, retransmissions due to initial timeouts were not necessary. The value to be injected could be the last RTT seen on that particular type of interface, possibly with a small fraction added on for safety. We regard this as a less crude alternative to simply storing a single "standard" value for the RTT for, say, a GPRS interface.

In order to prevent the congestion window for a connection being decreased excessively, we propose using a scheme similar to that of `cwin` clamping outlined in [3]. This involves a transparent proxy which rewrites pack-

[3]Though were there to be a packet loss, the timeout would be longer, and hence throughput would be marginally lessened. With fast retransmit this is not expected to be significant.

ets' TCP headers to have a new value for the congestion window. The proxy is aware of all TCP connections from all hosts using the GPRS link, and can therefore predict congestion and set the value of `cwin` accordingly.

The greater handover time for LAN ⇒ GPRS takes place between the binding update acknowledgement from the HA being received, to the route optimisation procedure being initiated. We are currently investigating why this is the case, and will subsequently attempt to implement methods to reduce the delay.

8. Analysis Tools

To obtain the data in a suitable form for this paper, several simple tools were constructed using the Perl scripting language. These took as input the textual versions of the `tcpdump` [4] output from the testbed, and generated `gnuplot` [5] graphs of average throughput over time, from the point at which handover took place.

The resulting plot files were then analysed by an interpolation script, which calculated the time at which the throughput exceeded the average value over the connection lifetime. Values from multiple traces were written to a summary file, from which the mean, standard deviation, maximum and minimum could easily be calculated.

All the scripts are available from the authors' web page [5].

9. Conclusions

In this paper we have described how current research on handovers in heterogeneous networks focuses on decreasing the time for the new interface to begin to be used. We have exhibited data from a live test bed which indicates the adaptation time, t_a, for using the new interface is significant compared with the handover time, T_n, in the case of the new interface being of a higher latency than the old. In the case of the new interface being of a lower latency, we have found the adaptation time to be negligible, but note that other effects come into play. We have also characterised what takes place in terms of TCP's algorithms during this adaptation time, in each of the cases of upward, downward and soft vertical handovers, demonstrating the issues arising in each. Finally, we have briefly outlined further research in the area of decreasing the adaptation time for heterogeneous handovers.

[4] `http://www.tcpdump.org/`
[5] `http://www.gnuplot.info/`

Acknowledgements

The authors would like to thank Prof. Andy Hopper for his ongoing support of this work, and Jonathan Davies for his aid in solving problems when the analysis tools were being written. Pablo Vidales held a scholarship from the Mexican Government through the National Council of Science and Technology (CONACyT). He is now working at Deutsche Telekom Laboratories, Technischen Universität Berlin, Germany. Contact Pablo.Vidales@telekom.de.

References

[1] M. Bagnulo, I. Soto, A. Garcia-Martinez, and A. Azcorra. Avoiding DAD for Improving Real-Time Communication in MIPv6 Environments. In *Proceedings of the Joint International Workshops on Interactive Distributed Multimedia Systems and Protocols for Multimedia Systems*, pages 73–79. Springer-Verlag, 2002.

[2] C. J. Bernardos and I. Soto and J. I. Moreno and T. Melia and M. Liebsch and R. Schmitz. Mobile Networks Experimental evaluation of a handover optimization solution for multimedia applications in a mobile IPv6 network. *European Transactions on Telecommunications*, 2004.

[3] R. Chakravorty, J. Cartwright, and I. Pratt. Practical experience with TCP over GPRS. In *Proc. IEEE GLOBECOMM*, November 2002.

[4] R. Chakravorty, P. Vidales, K. Subramanian, I. Pratt, and J. Crowcroft. Performance issues with vertical handovers – experiences from GPRS cellular and WLAN hot-spots integration. In *Proc. IEEE PerCom*, March 2004.

[5] D. Cottingham. MIPv6 Dump File Analysis Scripts. http://www.cl.cam.ac.uk/users/dnc25/.

[6] A. Cuevas, P. Serrano, C. J. Bernardos, J. I. Moreno, J. Jaehnert, K. Hyung-Woo, J. Z. D. Gomes, P. Gonçalves, and R. Aguiar. Field Evaluation of a 4G True-IP network. In *IST Mobile Summit*, 2004. Lyon, France.

[7] N. A. Fikouras, A. Udugama, C. Görg, W. Zirwas, and J. M. Eichinger. Experimental Evaluation of Load Balancing for Mobile Internet Real-Time Communications. In *Proceedings of the Sixth International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Yokosuka-Kanagawa, Japan, October 2003.

[8] Intel. Understanding Wi-Fi and WiMAX as metro-access solutions. Technical report, Intel Corp., 2004.

[9] D. Johnson, C. Perkins, and J. Arkko. Mobility support in IPv6. Technical Report RFC 3775, IETF, June 2004.

[10] R. Koodli. Fast Handovers for Mobile IPv6, October 2004. draft-ietf-mipshop-fast-mipv6-03.txt (work in progress).

[11] V. Marques, R. Aguiar, C. Garcia, J. Moreno, C. Beaujean, E. Melin, and M. Liebsch. An IP-based QoS architecture for 4G operator scenarios. IEEE Wireless Communications Magazine, June 2003.

[12] S. McCreary and K. Claffy. Trends in wide area IP traffic patterns – a view from AMES Internet Exchange. Tech. rep., CAIDA, 2000.

[13] M. Meyer. TCP performance over GPRS. In *Proc. IEEE WCNC*, pages 1242–1252, 1999.

[14] MIPL. Mobile IP for Linux (MIPL). developed by the HUT Laboratory for Theoretical Computer Science – GO/Core project.

[15] H. Soliman. Hierarchical Mobile IPv6 mobility management (HMIPv6), October 2004. draft-ietf-mipshop-hmipv6-03.txt (work in progress).

[16] P. Vidales, J. Baliosian, J. Serrat, G. Mapp, F. Stajano, and A. Hopper. Autonomic system for mobility support in 4G networks (submitted for publication). *IEEE Journal on Selected Areas in Communications*, November 2005.