An End-System Approach to Mobility Management for 4G Networks and its Application to Thin-Client Computing

Leo Patanapongpibul[a] *leo.p@cantab.net*

Glenford Mapp[b] *g.mapp@mdx.ac.uk*

Andy Hopper[a] *ah12@cam.ac.uk*

[a]Digital Technology Group, Computer Laboratory, University of Cambridge, UK
[b]School of Computer Science, Middlesex University, UK

This paper describes work centred around providing greater autonomy for mobile nodes to roam in Mobile IPv6 wireless networks based on a new handoff mechanism. This technique, called the Client-based Handoff, enables mobile nodes to roam in foreign wireless networks without having to be controlled by the network infrastructure. The mechanism incorporates three algorithms: a router advertisement cache, the invocation of TCP mechanisms and techniques to handle subnetwork outages in order to reduce packet loss and handoff latency. An experimental Mobile IPv6 testbed was developed to evaluate the proposed mechanism and is described in this paper. The testbed supports both horizontal and vertical handoffs. Experimental results are also presented.

The outcome of this approach was used to support mobile thin-client computing using a Virtual Network Computer (VNC) environment. Relevant experiments were carried out and the results show a compelling improvement in throughput of up to 50% compared to a VNC environment without the supporting architecture.

I. Introduction

Mobile IP does not have a mechanism to manage handoffs in the same way as cellular networks such as GSM. Conventional handoffs in GSM are mobile-assisted, but the decision to perform a handoff is network-controlled. The network would carry out the measurement of the mobile node's signal strength and, if the mobile node is in a call, would prepare the context to be handed off to a nearby cell. When the mobile node is in standby mode, the network keeps track of the device through a paging mechanism. Being able to control the mobile node's movements and calls, provides cellular infrastructures with an easier method to authenticate, authorise and account (AAA) for the mobile node's network usage.

Mobile IP has been designed for IP networks – a free for all medium – where the management of the network is distributed, with end users having as much control over the traffic as do the core entities in the network. Due to the nature of IP networks and for heterogeneity between different networking technologies, it may not be practical for handoffs to be network-controlled in the same way as cellular networks. To impose restrictions on mobility and charging for network connections, an entity in the core network is necessary to authenticate, authorise and account for a mobile node's activity. Hence, the pricing and policy model for packet-switched IP networks will be different to circuit-switched cellular networks.

Thus, there is a trade off between heterogeneity and the IP network having to provide mobility management when considering the AAA aspect. The work presented in this article takes a different approach to the network-controlled model.

In a cellular network, mobile nodes wishing to roam away from its home network would need the operators to have prior peering agreements. A client having complete control over which network it wishes to join eliminates the need for such roaming agreements. The client can independently obtain a security association with the network it wishes to join without depending on its network operator having to acquire a peering agreement. This approach can therefore be described as client-based and is investigated in this paper.

This research project is mobility-centric in a wide area network environment; IPv6 is the natural basis for supporting mobility and the next incremental upgrade for the Internet. There are other advantages in using IPv6, but these are not within the scope of this article. The solution to the mobility issue is also backward compatible, but requires IPv4 networks to have Mobile IP enabled.

The focus is on the struggle for dominance in the telecommunication marketplace. Network operators and service providers are often monopolising the market causing the customers to have no control over which network they can use once they buy into a contract. This research project aims to free the customer from their commitment to a particular network by making changes solely in the client in order to allow them to gain control over their own mobile networking needs, not those of the network operator. The outcome of this effort can prove to be beneficial both for the customers and network operators. This is because, besides the customer having control over the mobility management aspect, the network operator can focus on providing better and more varied services to the customer. In this project, a thin-client system was chosen as an example application which a network operator can use to attract more customers.

The Client-based Handoff Mechanism provides a mobile device with an intelligent method to select the network it wishes to join and manages seamless handoff with Mobile IP.

In this paper, we first discuss suitable steps towards an end-system approach to handoff management in Section II. Then in Section III, we provide an overview of the Client-based Handoff Mechanism. Section IV describes our network testbed used to study the feasibility of the handoff mechanism. We then describe experiments to evaluate the mechanism in Section V. In Section VI, an application of our end system approach to mobility management is introduced and evaluated. The application is called the Mobile VNC. Finally, we discuss some related work in Section VII, and conclude this paper with a summary of important research findings and future work in Sections VIII and IX, respectively.

II. Handoff Management for Wireless IPv6 Networks

There are a number of design issues to consider in establishing a suitable end-system solution to handoff management for use in a unified wireless IPv6 internetwork. Mobile IPv6 is the IETF standard (RFC 3775) for supporting mobility in IPv6 networks. Because the protocol is a layer-3 solution to mobility, it has not been designed for any specific layer-1 or layer-2 technologies. The protocol, however, has included a general explanation on how to interface with lower layer technologies. The handoff management design must consider various handoff scenarios a mobile node could encounter while roaming in wireless IPv6 networks. Active TCP sessions are also another factor which affects the perceived user experience on the overall handoff latency. Methods to optimise the recovery of TCP connections after a disruption caused by a handoff should also be considered. To be able to solve these issues whilst taking an end-systems approach, we should look at ways to optimise mechanisms already available in existing protocols.

II.A. Handoff Facilitators

A number of Mobile IP and Neighbour Discovery protocol (RFC2461) mechanisms are key to assisting a mobile node's handoff from one point of attachment to another. The Mobile IP mechanism is movement detection while the Neighbour Discovery protocol mechanisms are Router Discovery, Address Autoconfiguration and Duplicate Address Detection. These mechanisms are discussed below.

II.A.1. Router Discovery

This mechanism is used to locate nearby routers and to determine network prefixes. The network prefix is important to address autoconfiguration (RFC2462), which is performed by hosts to configure their network interface with valid IP addresses to access nearby network resources. A pair of ICMPv6 messages are defined for Router Discovery:

- ***Router advertisement*** is periodically multicast by access routers to all IPv6 hosts. Each advertisement contains a limited life-time. If another advertisement is received by the mobile node within the life-time of an advertisement message, then the related access router is reachable. Otherwise, once the life-time expires, the access router is assumed to be unreachable. At this point, the mobile node commences searching for a new access router. The Neighbour Discovery protocol specifies the interval of the advertisement to be between 3 to 10 seconds. However, Mobile IPv6 recommends a more frequent interval of 0.03 to 0.1 seconds. The movement detection (discussed below) can rely solely on router advertisements (i.e., Layer-3 trigger) for handoffs but has a trade off between the frequency of the router advertisement and the handoff latency. The router advertisement is necessary to acquire a care-of address for network connectivity on the new link.

- ***Router solicitation*** is multicast by IPv6 hosts to all IPv6 routers in the network in search of a new access router to join. If an access router is reachable, the router responds with a router advertisement. Unlike router advertisements, router solicitations are not sent periodically. This message is normally used by a node whose IP address requires renewal. In the case of a mobile node, the sending of a router solicitation may be necessary to immediately resume network connectivity on a new link. The handoff latency is largely dependent on the time of this movement detection and the round-trip time between the mobile node and access router.

II.A.2. Duplicate address detection

Duplicate address detection is a method to determine whether a mobile node's address is valid. The procedure is an intrinsically secure method to rule out nodes that implement IP address spoofing. Upon the receipt of a new binding update from the mobile node by the home agent, all of the registered home and care-of addresses are checked for any duplication. If this check fails, the binding update is rejected and a binding acknowledgement is sent to the mobile node with the Status field set to *Duplicate Address Detection failed* (134). This detection process can facilitate or prevent a successful handoff and influences the handoff latency.

II.A.3. Address Autoconfiguration

IPv6 defines two types of address autoconfiguration mechanism: stateful (also known as DHCPv6) and stateless. The latter is a straight forward approach for a host to form an IP address on its network interface. It uses locally available information and information advertised by routers. It is the best and fastest possible method for the mobile node to form a new IP address since the information is piggybacked with the router advertisement.

II.A.4. Movement detection

Movement detection is a technique defined by Mobile IP for mobile nodes to detect a move into a new network. A new access router can be discovered using layer-2 or layer-3 information. The Mobile IPv6 RFC suggests a movement detection based on layer-2, but does not specify how to perform the detection. The handoff response, thus the latency, is dependent on layer-2 and/or layer-3 triggers.

The Client-based Handoff Mechanism makes use of the existing IPv6 and Mobile IP functionalities described above. This empowers the mobile node with network-independent mobility management. The mobile node can initiate and control handoffs without the need to depend on specialised network entities in the core network to support handoffs.

Due to the mobile node's mobility independence, vertical handoffs in an overlay wireless network environment is made simple and practical. Since link layer information is used as a determinate for handoff, additional network interfaces on a mobile node can increase the choice of network link.

In the subsequent sections, we describe the Client-based Handoff Mechanism in detail. Our heterogeneous Mobile IPv6 network testbed which was purposely set up to test our proposed mechanism will also be described.

III. Overview of the Client-based Handoff Mechanism

There are two ways to provide a suitable handoff mechanism for mobile nodes. The first is to make modifications or extensions to the entities in the network infrastructure. Routers or base stations can be changed so that they will only send router advertisements to the mobile node when a handoff is necessary as opposed to periodically sending router advertisements. However, this means the approximate location and signal strength of the mobile node need to be cached in nearby routers or base stations. Additional signaling may be required in order to enable such a system to operate correctly. Furthermore, changes to routers are difficult and disruptive in contrast to an end-system approach which is less intrusive. Modifications to the core network infrastructure has the advantage of offering a complete mobility management protocol for the entire network to reduce handoff delays, but has the disadvantage of introducing greater complexity.

The second way is to make modifications or extensions on the client-side, i.e. the mobile node. In this case, it is the client that decides when a handoff is appropriate. This necessarily implies some loss of control on the network domain's side. The advantage of this, however, is its apparent simplicity and scalability, which are the reasons why the handoff mechanism described in this section is based on this approach.

This section describes the mechanism in detail and how it tackles the following issues:

1. Controlling and forcing handoffs
2. Determining the best link
3. Handing off at the appropriate time
4. Resuming active TCP connections

The Client-based Handoff Mechanism is illustrated in Figure 1 as a module in the TCP/IP protocol stack.

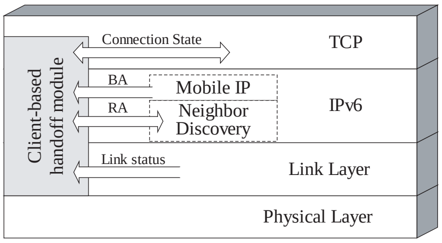

Figure 1: Relative position of the Client-based Handoff Mechanism in the TCP/IP protocol stack.

III.A. Controlling and Forcing Handoffs

The mobile node initiates a handoff every time it receives a Router Advertisement (RA) from any Home Agent (HA). The handoff mechanism provides the mobile node with the capability of filtering RAs to avoid the default processing of handoffs. Thus, handoffs can be forced when required.

III.B. Determining the best link

To enable the mobile node to select the best point of attachment (also valid for those with multiple network interfaces) an *RA cache* is introduced in the handoff module. This provides the mobile node with the capability of choosing the best link from the cache. An algorithm, based on prioritising RAs, was devised to assist with the best link decision. The two most important criteria used to determine the priority of the RAs stored in the cache are:

- the link signal strength[1], i.e. signal quality & SNR level; and
- the time since an RA entry was last updated.

Note that the link bandwidth is not a criterion for the algorithm. This may have an affect on latency-sensitive applications such as VoIP. The reason for excluding the criterion is because of the following two assumptions. Firstly, the IP Multimedia Subsystem (IMS) is one solution that is being adopted by cellular network operators to handle handoffs from circuit-switched voice to VoIP and vice versa. Secondly, cellular network operators are gradually upgrading to support higher data rates.

The other two less important criteria are:

- whether or not the access router is link-local; and
- the number of hops to the access router.

It may be argued that the number of hops to the AR is typically one. However, for network domains that implement, say, Hierarchical Mobile IP, the configuration of the network is a cascade of ARs (also known as HAs). Thus, it is possible for an RA to be received from a router further up the hierarchy or an adjacent router.

III.C. Handing off at the appropriate time

Although the application of the aforementioned criteria will generally yield a higher data throughput by handing off to the best link, there are cases where it is advantageous to trade off a potential increase in signal strength against maintaining an active data connection. In order to adopt the handoff algorithm accordingly, the handoff module takes into account the state of TCP connections. Hence, when a handoff is necessary, an open TCP socket will cause the threshold value of the signal strength criterion to be lowered and the handoff to be delayed. Because of this, disruptions to TCP connections can be avoided if the difference between the current link quality and the threshold level is minimal. Once the signal strength drops below the lower threshold value or there are no open TCP sockets, the RA Cache entry flagged with the highest priority is passed to the IP packet handler for processing.

The handoff module depends on a link status handler which monitors the link connectivity. This avoids the need to decrease the RA interval in the access routers in order to improve the detection speed of a link disconnection as suggested in the Mobile IP RFC.

III.D. Resuming active TCP connections

TCP is designed for the wired Internet where network nodes have access to a reliable network infrastructure. In contrast, TCP was not designed for mobile nodes whose network connectivity to the Internet via the wireless medium can be unpredictable and variable depending on their location.

Disruptions to TCP connections can be noticeable to the user because mobile nodes, especially thin-client devices, tend to be in the downlink state while roaming from one wireless link to another. Furthermore, disruptions can still be experienced even though the mobile node is moving around within the coverage boundaries of a wireless access point. Wireless network coverage can vary, therefore, the worse-case scenario has to be considered in finding a solution for minimising the disruption to TCP connections. The

[1] For Wireless LANs, the preamble of Ethernet packets contains the signal strength information.

worst possible case scenarios are unanticipated handoffs and intermittent link disconnections.

At a single packet drop, TCP can assume that the network is congested causing the transmitter to throttle the transmission by decreasing the congestion window to the minimum size. This works well for a wired network, however, in a wireless network, the link quality can vary especially when dealing with moving computers. A drop in packet does not necessarily mean congestion in the network, but rather a weak link connectivity.

There are two approaches to enhance the user experience of TCP applications for mobile computing. One approach is to modify or introduce a new TCP variant. Another approach is to avoid any internal modifications and provide a mechanism which can act as a catalyst to invoke existing TCP mechanisms. The work in this article takes the second approach where TCP mechanisms, such as fast retransmit, are utilised to improve the mobile computing user experience.

Taking advantage of the functionalities provided by TCP is essential for the handoff mechanism to be fully aware of activities in the higher layers of the protocol stack for handoff intelligence. Our handoff mechanism monitors the TCP connection states for all network connections including outgoing and incoming packet queues, and triggers a TCP fast retransmit or TCP persist mode at the correspondent node. When a handoff is imminent, the handoff mechanism lowers the signal quality threshold to delay the event and buffers the last TCP acknowledgement packet necessary for use after the handoff. The appropriate TCP mechanism is then invoked for the respective handoff scenarios discussed in Section IV.B.

III.E. Link Adaptation

There is the argument of different link speeds when handing off from, say, a faster network to a slower one. In such an event, our mechanism can determine the type of the new link based on the RAs in order to choose the best action to resume an active TCP data transfer. The technique to resume TCP transmission when handing over between different link speeds or type, i.e., a handoff between different domains or different network technologies would be to trigger a TCP persist mode at the sender. For handoffs between subnets where the link speed and type are equivalent, the technique for resuming TCP connections is to trigger a TCP fast retransmit and fast recovery at the sender. These are simple solutions to link adaptation for TCP traffic. However, other type of traffic such as UDP require a different method of link adaptation. This issue is beyond the scope of this paper. The discussion of link adaptation is an active research topic and may be explored as future work.

IV. Our Mobile IPv6 Testbed

With the collaboration of the University of Cambridge Computer Laboratory, our Mobile IPv6 testbed is extended to support a connection to Vodafone's GPRS network. The combined Wireless LAN (WLAN) and GPRS testbed is illustrated in Figure 7. A number of publications [1, 2, 3] resulted from work carried out on the combined testbed. In this section, we describe the testbed and the handoff process in detail.

There are two key motivating factors for the collaboration to setup the testbed. The first is to evaluate performance issues of Mobile IPv6 in a wireless overlay (heterogeneous) network environment. The second factor is the need to develop enhancements, where necessary, for seamless handoffs between different wireless networks.

A mobile node is configured with two network connections, one to our (DTG) WLAN testbed with an Orinoco WLAN PC card and the other to the GPRS network via a serial point-to-point link to a GPRS mobile phone. For thorough testing purposes, the latest GPRS phones and cards from a number of manufacturers are employed.

The base stations in the GPRS infrastructure are directly linked to the Serving GPRS Support Node (SGSN) which is then connected to a Gateway GPRS Support Node (GGSN). The current operator's configuration has the SGSN and GGSN co-located in a single Combined GPRS Support Node (CGSN) [4]. A virtual private network (VPN) connects the Laboratory network to Vodafone's network backbone via an IPSec tunnel over the Internet. A Remote Authentication Dial-In User Service (RADIUS, RFC2865) server, separate from the operator's server, is provisioned to authenticate GPRS mobile users/terminals and assign IP addresses.

Special arrangements with Vodafone and the two University of Cambridge departments - Computer Laboratory and Engineering Department - enable GPRS and WLAN data traffic to be routed through the combined testbed. Routing has been configured to force all GPRS and WLAN user data traffic going to and from the mobile nodes to pass through a IPv4/IPv6 Linux router. This router, illustrated in Figure 7, enables traffic monitoring.

The GPRS network does not support IPv6. This means all IPv6 packets destined for a mobile node visiting the GPRS network had to be tunneled to the mobile nodes as shown in Figure 7. The method to support IPv6 in the GPRS network is described in detail below. Note that all of the nodes in the testbed, including all correspondent nodes, support Mobile IP and route optimisation.

IV..1. IPv6 Data Communication in the GPRS network

The home agent of the mobile node is in the WLAN part of the testbed. When the mobile node switches from its WLAN interface to its GPRS connection, a tunnel is automatically established between an IPv4/IPv6 edge router and the mobile node. This router is responsible for sending router advertisements in the GPRS network. It is also reachable by the mobile node's home network since it is part of the IPv6 Internet. This means all binding updates from the mobile node in the GPRS network can be routed to the home agent. Binding updates are tunneled from the mobile node to the GPRS IPv4/IPv6 edge router and then routed normally to the home agent.

When the mobile node wishes to set up a data connection to a correspondent node, provided that a binding of the mobile node's care-of address and home address already exists at the home agent, a binding update is first tunneled to the GPRS edge router and routed normally to the correspondent node. Following a successful binding update, packets destined for the mobile node are routed to the GPRS edge router where they are then encapsulated (RFC2473) and tunneled to the mobile node.

The soft state tunnel set up to carry IPv6 traffic over the IPv4 Internet is called a Simple Internet Transition (SIT) tunnel. As mentioned above, when the mobile node is in the GPRS network, a SIT interface is activated on the IPv4/IPv6 router and the mobile node. However, the tunnel between the SIT interfaces cannot be established due to firewalls between the various network domain. The GPRS network is under the Computer Laboratory's network domain administration and the IPv6 Internet (6BONE) is only accessible through the Engineering Department. As a result, when the mobile node wishes to communicate to a correspondent node in the 6BONE, the encapsulated IPv6 packets need to propagate through the Computer Laboratory's and Engineering Department's firewalls. Thus, a "hole" has to be in place in each of the firewalls to allow the flow of IPv6 packets which are encapsulated in IPv4 packets.

IV.A. Types of Handoff

Handoffs are categorised into two groups: horizontal and vertical handoffs.

Horizontal handoff is the handoff between any two points of attachment of the same wireless network technology.

Vertical handoff is the handoff between any two different wireless network technologies. There are two subsets for this type of handoff. The first is an *upward handoff*. This occurs when a mobile node moves higher up in an overlay wireless network, e.g. from a micro-cell (WLAN) to a macro-cell (3G). The second subset is a *downward handoff*. This is when a mobile node moves down in an overlay, e.g. from a macro-cell to a micro-cell.

IV.B. Handoff Scenarios

There are two situations where handoff can be initiated:

Scenario 1 - Discontinuous Handoffs: The current mobile node's point of attachment becomes out of range (e.g., beyond a WLAN coverage or a disconnection from a LAN), preventing any data transmission or reception.

In these situations, the execution of a handoff is forced but without the knowledge of the next new point of attachment to which the mobile node can reconnect. Thus, this is called a *discontinuous handoff* since the mobile node is unable to anticipate a new link to the network. In this scenario, there is likely to be severe packet loss because it is uncertain when the next network attachment will occur, hence to prevent this, it is better to use the TCP persist timer at the correspondent node. The procedure is carried out in the following sequence. As described in Section III.D, prior to a disconnection from the network: the signal strength threshold is lowered; TCP acknowledgements are sent advertising a zero window; and the last TCP acknowledgement packet is buffered by the handoff mechanism. Once the mobile node is within reach of a link, the sending of a Router Solicitation (RS) is forced to quickly acquire an RA. After the Mobile IP registration process finishes, the buffered acknowledgement packet (non-zero window) is transmitted only once to the correspondent node in response to the previous zero window acknowledgement packet, hence, allowing the sender to resume the TCP transmission. A discontinuous handoff could also happen unexpectedly. In such an unlikely event, it is not possible to minimise the handoff latency because there is no time to trigger the TCP persist timer.

However, the fast retransmit technique used for continuous handoffs is applied in this situation without any performance degradation.

Scenario 2 - Continuous Handoffs: In a wireless network, the signal strength of the link between the mobile node and current base station reaches a predefined threshold and there are other base stations capable of providing better connectivity. In this situation, the mobile node can trigger a handoff to the link with a higher signal strength. This is called a *continuous handoff*.

After the Mobile IP registration completes, TCP connections are quickly resumed in the following way. As described in Section III.D, prior to the handoff: the signal strength threshold is lowered; and a TCP acknowledgement packet is stored in the handoff mechanism buffer. After the handoff, this packet is retransmitted more than three times to the correspondent node. This induces a fast retransmit and fast recovery causing the sender to ignore its retransmission timer and perform a retransmission of the missing segments. This method of resuming a TCP connection is highly dependent on the Mobile IP registration time and requires the correspondent node to send a Binding Acknowledgement (BA) after a Binding Update (BU). Note: the Mobile IPv6 specification states that the sending of a BA is optional. In this case, a BA is necessary for the mobile node to know when it is able to start sending the buffered TCP acknowledgement packet.

IV.C. Handoff Execution

Handoff execution assumes the mobile node is within the wireless coverage of the network it will handoff to. However, Mobile IPv6 specifies a movement detection method that generates a latency component. The specification, reflected in MIPL, defines layer-3 triggers for movement detection which can be slow in responding to link changes. In comparison to layer-2 triggers, the response time in detecting a link change is faster but this method of movement detection requires the tight coupling of the movement detection component with the physical layer. As discussed in the previous section, this has been done with WLAN but it is not possible with GPRS as yet due to the proprietary nature of the protocol stack code[2]. Even though the handoff execution is considered as a *posteriori* to a handoff decision, nevertheless, the detection time needs to be included in the overall handoff latency equation, to be consistent with the protocol specification.

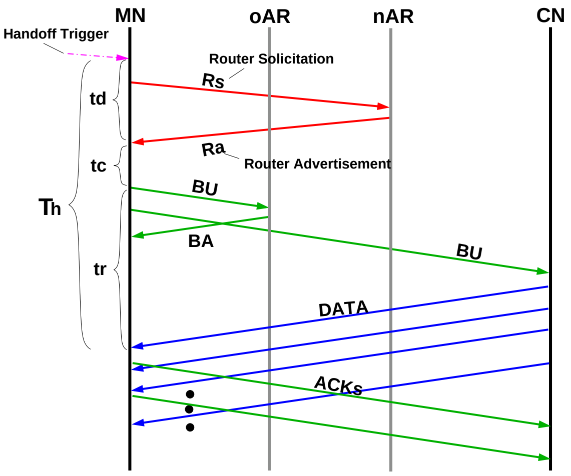

Figure 2: Partitioning the Handoff Latency. Note: oAR = old access router, nAR = new access router.

To analyse handoff execution in handoffs, the handoff latency is partitioned into its sub-components contributing to the overall handoff latency. The notations for the time latency components first seen in [5] is followed here to formulate an equation which defines the overall handoff latency for a mobile node. This is the amount of time to initiate a disconnection from the old network access point to the instant when the first packet is received from the new network access point.

From the observation of the Mobile IPv6 signaling shown in Figure 2, the handoff latency can be broken down into the following components:

- *Detection Time* (t_d). This is the time from the mobile node to discover that it is now under the coverage of a new wireless access network to the instant it receives a router advertisement from the new access router. This time can be reduced through the use of layer-2 (L2) triggers. However, the impact of this optimisation is different in each vertical handoff scenario. Note that depending upon continuous and discontinuous handoffs, the detection time may vary from zero up to a significant percentage of the overall latency.

  Thus, the detection time (t_d) is directly influenced by how frequently the new network advertises its presence by means of router advertisements. However, if a L2 trigger is used, the detection time is considered to be the time between the L2 trigger reception and the instant when the mobile node receives the router advertisement, in

[2] Radio Resource Control (RRC) component in the GSM/GPRS protocol stack is capable of periodically sending the signal strength information to the higher layers.

response to the generated request sent from the mobile node to the new access point following the reception of the L2 event. Note that in anticipated handoffs the detection time is independent from the router advertisement frequency of the new network.

- *Address Configuration Time (t_c).* This is the latency encountered when a mobile node receives a router advertisement and forms the new care-of address, updates its binding cache and routing table and configures its interface with the new IPv6 address. It is important to observe that this time includes L2 reconfiguration because forming the new care-of address implies selecting the active interface [MAC address].

- *Registration Time (t_r).* It is the time to register the new care-of address with the home agent and to update the correspondent nodes, perform a Duplicate Address Detection if necessary and receive the last binding acknowledgement (BA) packet either from correspondent nodes or home agent (whichever arrives later). This is commonly known as the Mobile IPv6 registration process.

- *Packet Forwarding Time (t_f).* This is the time after the reception of the last binding acknowledgement from either the home agent or the correspondent node to the time the mobile node receives the first data packet from the correspondent node. This component is insignificant, thus it is not illustrated in Figure 2.

The total handoff latency (T_h) is therefore given by:

$$T_h = t_d + t_c + t_r + t_f \tag{1}$$

The total latency is shown in Figure 2. Note that the parameters are constant and the magnitude of each component may vary significantly depending on the vertical handoff properties (upward, downward, continuous and discontinuous). However, other components may be optimized in order to reduce the overall latency, hence improving the mobile user experience.

For example, the coverage to the current access point may be lost before the mobile node manages to handoff to the new access point (occurs commonly in high mobility environments), thus a downward vertical handoff can still afford to delay a handoff decision as the mobile node remains under coverage of the network higher up in the overlay. Note that a handoff decision for the case of upward vertical handoff cannot be delayed, as coverage from networks lower down the overlay (e.g. WLANs) can be lost before it finally handoffs to networks higher up in the overlay. Hence not all vertical handoff decisions can be anticipated, but, rather, in some cases they can be delayed to achieve minimal latency.

V. Experiments

We divide the experiments into two categories. Firstly, we test the effectiveness of the Client-based Handoff Mechanism in performing horizontal handoffs. Secondly, we compare the handoff latency for horizontal and vertical handoffs with the Client-based Handoff Mechanism disabled. It is not necessary to investigate the improvement of the Client-based Handoff Mechanism in the latter experiment. This is because the Client-based Handoff Mechanism is independent of the network medium, i.e. physical layer.

In both experiments, we measure the handoff latency based on a TCP file download to a mobile node. In addition, the testbed is set up to operate under the following conditions:

- All access routers including the home agent are set to multicast router advertisements in accordance with the recommended values specified by the Neighbour Discovery protocol (RFC2461).

- For all cases, a vertical handoff assumes that the multi-mode mobile device has all its network interfaces (WLAN/GPRS) powered on simultaneously to reduce the initialization time. This does not necessarily mean all the interfaces are linked simultaneously to a network. Only one interface is connected to a network at any one time during a data transmission.

- Vertical handoffs are performed between visiting networks. Hence each of the binding messages sent between any of the two Mobile IP network entities (the mobile node, home agent or correspondent node) 1 traverse IP hop.

The test setup consists of a web server with Mobile IPv6 support in Network C and a mobile node roaming (away from its Home Network) in the WLAN (Network A) and GPRS networks as shown in Figure 7.

In the experiments, a 25Mb file transfer initiated by executing *wget* on the mobile node in one network domain is continued by forcing a handoff to the another network domain after a file transfer of more than 5Mb. The mobile node is then forced to handoff back to the

MIPv6 Enhancement	Handoff Latency (ms)				Average Ratio handoff latency/ download time
	Min	Mean	Max	Stdev	
None	2277	4209	4759	1041	165
Mechanism (Scenario 1)	112	633	895	241	29
Mechanism (Scenario 2)	191	238	258	24	11

Table 1: The handoff latency during a 25Mb file download from a web server performed over 10 runs.

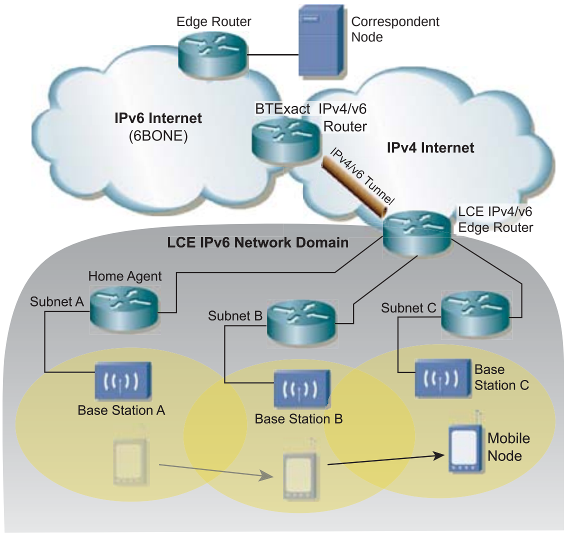

Figure 3: Our (DTG) WLAN Mobile IPv6 Testbed.

previous network domain after a file transfer of more than 15Mb. The forcing of a handoff after a sizeable download is decided arbitrarily to ensure TCP slow start does not affect the consistency of the results from the tests. The traffic is captured with `tcpdump` running on an intermediate router. `Tcptrace` is used to analyse the traces obtained from `tcpdump`. However, modifications are required on a version of `tcptrace` to enable support for the processing of Mobile IPv6 data packets.

Further details of the two experiments are described in the following subsections.

V..1. Experiment: Effectiveness of the Client-based Handoff Mechanism

The impact of handoffs, with various parts of the mechanism enabled, on a TCP download to the mobile node is investigated. Firstly, the Client-based Handoff Mechanism without the TCP enhancements is tested under discontinuous (Scenario 1) and continuous (Scenario 2) handoff scenarios as discussed in Section IV.B. Then, the TCP enhancements built as part of the Client-based Handoff Mechanism are tested under the same conditions.

The testbed used in this study is illustrated in Figure 3. In the experiments, the mobile node performs handoffs between the two foreign WLAN networks: subnet B and subnet C. The mobile node's home network, base station A, is disabled after the mobile node executes a handoff to subnet B to prevent the mobile node from returning to its home network.

In the first test, the TCP enhancements are disabled to test the effectiveness of the RA Cache in reducing the handoff latency for TCP flows. Testing Scenario 1 requires the respective base station B or C to halt Router Advertisement transmissions and link beacons (by terminating their wireless interface, but still powered up) to mimic a subnetwork outage (the outage time is kept to a maximum arbitrary duration of 3 minutes) when a handoff is forced at the mobile node. Testing of Scenario 2 requires the mobile node to be within range of both base stations B and C when handoffs are forced at the mobile node. Note that a number of TCP download trials were conducted to conclude that at least 10 TCP downloads are sufficient for each test.

The results in Table 1 shows the significant reduction in the handoff latency with the Client-based Handoff Mechanism. Notice that the handoff latency without the use of the Client-based Handoff Mechanism is not divided into continuous and discontinuous handoff scenarios. This is due to the indifferent handoff latency results obtained from both scenarios. As mentioned in an earlier section, Mobile IP depends on layer-3 triggers and therefore the conditions of the two scenarios have no effect on the overall handoff latency. It is also necessary to note that the handoff latency with the mechanism under Scenario 1 does not include the subnetwork outage period. Thus, the values obtained are the effective handoff latency times.

The handoff latency and download time ratio shows a reduction of up to 15 times with the mechanism enabled. The handoff results for Scenario 1 show a significantly lower latency compared to the results without the mechanism. This suggests this handoff method

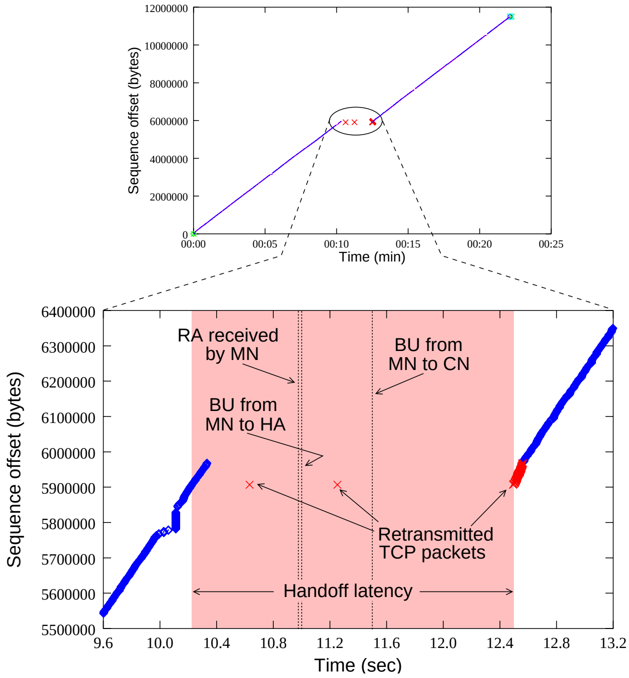

Figure 4: TCP Sequence plot of a handoff without the Client-based Handoff Mechanism.

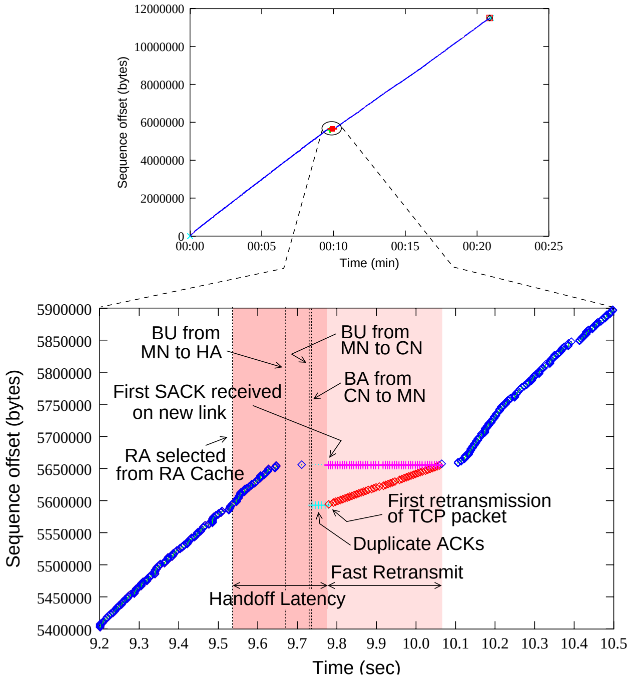

Figure 5: TCP Sequence plot of a continuous handoff (Scenario 2) with the complete Client-based Handoff Mechanism.

could also be used for continuous handoffs. The advantage of this method is the TCP state at the sender is kept constant before and after a handoff whereas the TCP state of the sender changes for the reconnection method which utilises fast retransmit and fast recovery (Scenario 2). In both techniques, slow start is avoided optimising the download duration

V.A. Analysis of a handoff

Taking one of the mean handoff latency logs for each of the horizontal handoffs performed with and without the Client-based Handoff Mechanism, we dissect and analyse the handoff latency components for one handoff. The dissection is illustrated as a TCP sequence graph as shown in Figure 4, 6 and 5 using `Tcptrace`, which we modified [6] to support Mobile IPv6.

The diagram in Figure 4 clearly shows the handoff latency is severely affected by two components: the Mobile IP registration time and the TCP connection reestablishment time. The Mobile IP registration delay is due to its dependence on router advertisements for movement detection. During this period, the correspondent node has gone into TCP congestion avoidance mode. The TCP reconnection time thus increases the handoff latency because the mobile node has to wait for the next retransmission. A slow start takes place once the TCP connection is reestablished which is not apparent in the plot because of the relatively high network bandwidth.

Figure 5 shows the TCP sequence plot for a continuous handoff utilising the complete mechanism. After receiving a binding acknowledgement from the correspondent node, the last outgoing ACK is sent to the correspondent node as duplicate ACKs to ensure a fast retransmission. This avoids a slow start altogether. In cases of lower bandwidth networks, the RTO will be proportionally higher allowing enough time for the Mobile IP registration time to complete and, thus, avoiding congestion control altogether. This has shown to be true from the majority of the experimental results.

The TCP sequence plot for a discontinuous handoff is shown in Figure 6. During handoff, the correspondent node remains in TCP persist mode. Zero window probes (ZWPs) are sent at exponentially backed-off time intervals to get the mobile node to respond with a non-zero window ACK to resume the download. Upon the completion of the Mobile IP registration process (indicated by the reception of a binding acknowledgement from the correspondent node), the last outgoing ACK in the buffer is retransmitted to the correspondent node without the need to wait for the next ZWP.

Note that the time delay between the sending of binding updates to the mobile node's home agent and correspondent node varies from 0.1 to 0.4 seconds in

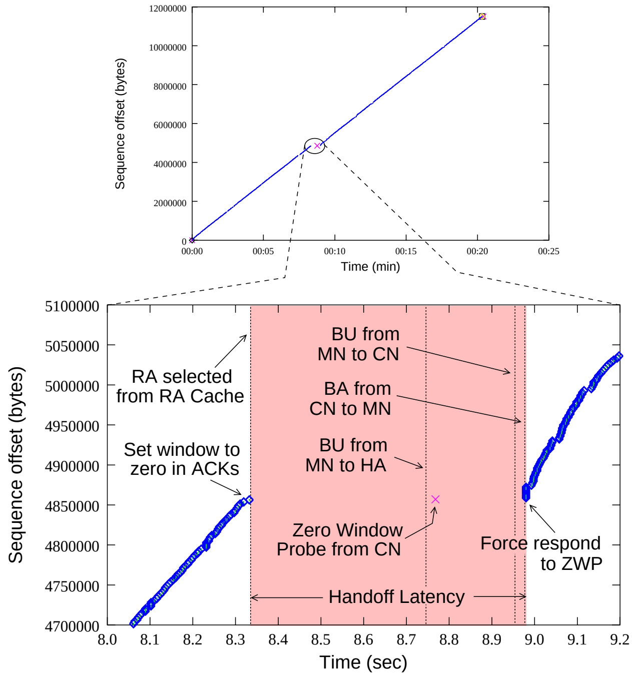

Figure 6: TCP Sequence plot of an discontinuous handoff (Scenario 1) with the complete Client-based Handoff Mechanism.

the TCP sequence plots. From the `tcpdump` [6] logs, the high delay is caused by the mobile node sending binding updates to the link-local home agents. This delay occasionally affects the mechanism for continuous handoffs because of the retransmission timer expiration at the correspondent node.

V.A.1. *Experiment: Vertical Handoff Behaviour*

For this experiment, handoffs are forced from WLAN to GPRS and vice versa. To test the performance of the handoff, file downloads are carried out between the web server and the multi-mode mobile node. The mobile node roams between the WLAN and GPRS foreign networks as shown in Figure 7.

In the testbed, all traffic is set up such that it passes through an intermediate router which simultaneously monitors the traffic (using `tcpdump` [6]). Traffic is sourced from the web server to the mobile node during all active data sessions. As mentioned earlier, the internal router is also the IPv6 access router for the WLAN with a separate GPRS access router (logically co-located with the GGSN), which acts as an access router for the GPRS network.

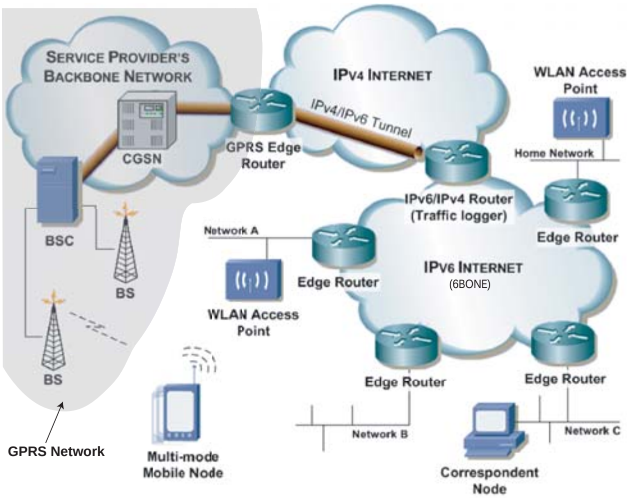

Figure 7: The Wired LAN, Wireless LAN and GPRS network with Mobile IPv6 support in an overlay configuration.

V.B. Limitations

There are a number of limitations which arises as a result of developing the handoff mechanism and testing it on a live testbed.

Firstly, the effectiveness of the Client-based Handoff Mechanism in performing vertical handoffs could not be investigated due to the closed source and complex nature of the GPRS protocol stack. As mentioned in Section III, the handoff mechanism design is based on utilising link layer triggers to invoke a handoff. This introduces a challenge in performing vertical handoffs from the WLAN part of the testbed to Vodafone's GPRS network. There is no open standard which will allow third party software to directly access the signal quality information from the mobile station's Radio Resource sublayer in the GPRS protocol stack. The only method to access the signal quality information of a mobile station would be via the application interface, i.e. AT commands, such as those defined in the open standard 3GPP TS 27.007. This approach can significantly increase the handoff latency. Therefore, our second experiment is conducted by forcing vertical handoffs. As a result, the RA Cache and TCP enhancements in the Client-based Handoff Mechanism cannot be utilised to minimise the mobile registration period.

Nevertheless, there is an advantage of testing the effectiveness of the Client-based Handoff Mechanism in performing horizontal handoffs. In a heterogeneous network, we rely on Vodafone's GPRS network. Because this is a public network, there are external variables such as the channel availability which can

WLAN↔GPRS Handoff	WLAN→GPRS				GPRS→WLAN			
(the split in ms)	Min	Mean	Max	Stdev	Min	Mean	Max	Stdev
Detection Time (t_d)	200	808	1148	304	739	2241	3803	919
Configuration Time (t_c)	0.853	0.870	0.890	0.009	0.380	1.062	1.186	0.233
Registration Time (t_r)	2339	2997	3649	395	2585	4654	7639	1611
Total Handoff Latency (T_h)	3323	3806	4438	310	5322	6896	8833	1118

Table 2: Handoff latency partition (in ms) for WLAN↔GPRS taken over 10 runs

severely impact the overall handoff latency.

Secondly, the TCP enhancement in the Client-based Handoff Mechanism assumes the available bandwidth of the new path between the mobile node and correspondent node to be of the same order-of-magnitude as the previous path. For horizontal handoffs, this would not be a problem since the RTT will be similar. However, in the case of vertical handoffs where the RTT can substantially differ from one network technology to another, triggering a fast retransmit may prove ineffective to reduce the handoff latency. A solution to this problem is to use the Binding Update Bi-Casting technique [1].

Thirdly, the current design and implementation of the Client-based Handoff Mechanism are limited to asymmetric applications (e.g., thin-client computing) where the bulk of the data is received by the mobile node because it would be ineffective to trigger, say, the TCP persist timer during a handoff at the correspondent node since data is being sent from the mobile node. Despite this limitation, it is sufficient to demonstrate the use of the handoff mechanism with a mobile computing application discussed in Section VI. The mechanism can be extended to support symmetric applications in future (see Section IX).

Finally, the experimental results are based on one mobile node. If more than one mobile node in the same cell needed to handoff to an adjacent cell, there could be a congestion of TCP packets in the base station of the new cell due to the simultaneous transmission of duplicate ACKs by the Client-based Handoff Mechanism. Furthermore, the handoff of multiple mobile nodes each having active TCP sessions could cause the synchronisation of TCP connections. It is not possible to reliably investigate these scalability issues due to the unstable nature of the Mobile IPv6 implementation used in the testbed. These issues are left for future work.

V.C. Results and Discussion

Table 2 shows the breakdown of the handoff latency components for vertical handoffs. The detection time (t_d) and registration time (t_r) are the two main components which greatly influence the overall handoff latency. This clearly shows the motivation for the Client-based Handoff Mechanism to minimise the effect of these latency components on the handoff performance.

Comparing the results of horizontal (Table 1, Client-based Handoff Mechanism disabled) and vertical (Table 2) handoffs, there appears to be an insignificant difference in the total handoff latency between WLAN↔WLAN and WLAN→GPRS. The horizontal handoff latency range from 2.277 seconds to 4.759 seconds with a mean time of 4.209 seconds. In comparison, the WLAN→GPRS handoff latency range from 3.323 seconds to 4.438 seconds with a mean time of 3.806 seconds. Both of these handoff latency ranges are similar and the difference in the mean values is evident from the standard deviation of the results.

The more significant result, however, is the GPRS→WLAN handoff latency which range from 5.322 seconds to 8.833 seconds with a mean time of 6.896. This handoff latency is greater by approximately 2 seconds. The reasons for the higher latency is due to the network characteristic of GPRS: the larger packet buffers in the GGSN nodes and the stark contrast in network bandwidth. The maximum data rates for WLAN and GPRS are 11Mbps symmetric and 48Kbps asymmetric (downlink rate is defined by GPRS PC Card specification), respectively.

VI. An Application of End-System Approach to Mobility Management: Mobile VNC

In this section, we envisage and realise an application which makes heterogeneous networking, i.e. borderless computing, possible and useful. One such application is *thin-client* computing.

Contrary to the trend of a "thick" mobile device, e.g. laptops, providing better support for distributed applications or stand-alone applications, a stateless mobile device is advocated and an architecture to make such devices feasible for wireless computing is proposed.

Previously it has not been possible to provide a good user experience for such applications of mobile computing in a public network infrastructure, however, it is increasingly becoming a reality with the roll out of higher data rate services such as 3G and WLAN hotspots and the functions offered by the Client-based Handoff Mechanism.

Unlike stateful devices, stateless devices do not run application or system code on the appliance. In this article, it is defined as the execution of the windowing system and applications entirely on a server through thin clients. Thin-client systems are a proven technology which is well suited for fixed broadband network connections. Upon a disconnection in the link, or poor network coverage, the user response rate becomes problematic. Therefore, a completely stateless client may not be ideal for an environment where network coverage can be unpredictable. A truly portable stateless device will only be ideal in an enclosure, such as a building or an aeroplane. A method to adapt and cope with changes in network conditions is necessary to minimise disruptions to human computer interaction (HCI).

There are several types of thin-client applications [7]. This research is only concerned with *ultra-thin client systems* due to its centralised nature and its ability to support multiple users. In addition, the system is ideally suited for mobile devices because it imposes no user data storage, extremely low power consumption and the end product is lightweight. The simplicity of administrating an ultra-thin client system is the key driver to extend its use to a high mobility environment.

Thin-client systems offer user mobility by means of providing user access to their desktop virtually anywhere in the world as long as there is a relatively high speed network connection. However, device mobility of thin-clients has not been explored in a global environment. The Videotile[3], used an indoor wireless ATM technology limiting its use inside a building. However, with the advent of WLAN and higher speed data access through cellular networks (e.g., 3G), the feasibility of thin-client device mobility is becoming ever more realistic. With the lower power consumption on the battery of the mobile device, server power computing, close to zero administration, greater application robustness and no risk to loss of data through theft or damage there are more advantages to move to thin-clients. Such a system would be highly appropriate for corporate employees where information is naturally accessible through a centralised infrastructure providing greater security.

This section introduces an architecture which leverages the mobility management solution in this paper for supporting the roaming of mobile thin-client devices in wireless IP networks (rather than ATM).

VI.A. The Mobile VNC Architecture

The concept of *Mobile VNC* is introduced in this section. This term is defined as a system which enables server-based computing whilst the user is roaming with a tetherless and stateless thin-client device running a permanent VNC Viewer.

Supporting roaming thin-client devices involves a number of entities: a VNC Server, a VNC Proxy, a VNC Viewer and a signaling mechanism to transfer a VNC session between VNC proxies. A VNC Proxy is introduced to resolve the network latency issue. The advantages of introducing such an entity into the network infrastructure are:

- The local cache reduces the number of TCP retransmissions and screen updates between the server and client;
- Minimise HCI disruptions due to micro-mobility handoffs
- The enforcement of network security as firewall are not by-passed;
- Transparent accounting and billing; and
- Link speeds and bandwidth between the server and proxy can be guaranteed with Quality of Service (QoS) mechanisms such as *Integrated Services* (IntServ) and *Differentiated Services* (DiffServ) since the proxy server will be part of the fixed infrastructure.

A VNC Proxy is autonomous and is managed by the local network operator. If a network does not have a VNC Proxy in place, then the VNC client is able to communicate directly with the VNC Server.

Two other important components are necessary to create the architecture: guaranteeing the QoS between the VNC Server and the VNC Proxy; and the signaling mechanism to move a roaming client's VNC session between VNC Server and VNC Proxies.

The QoS between the VNC Server and VNC Proxy can be guaranteed through DiffServ or IntServ. The diagram in Figure 11 illustrates how this may be done

[3] The Videotile, 1996
http://www.cl.cam.ac.uk/Research/DTG/attarchive/tile.html

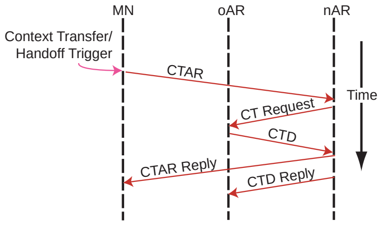

Figure 8: Mobile-controlled: Context transfer protocol signaling message sequence diagram initiated by the mobile node (MN).

using the Reservation Protocol (RSVP) which is a control protocol to provide QoS for IntServ.

The Context Transfer Protocol (CXTP) (RFC4067) proposed within the *Seamoby* IETF working group charter is used to deal with transferring information (or context) of a mobile node's VNC session between VNC Proxies. The signaling initiation can either be network-controlled or mobile-controlled. In this research, the focus is to offer the mobile node full control of its own mobility. The Client-based Handoff Mechanism is extended to support the *Context Transfer Trigger* required to initiate the transfer of context between access routers from the mobile node, in this case, the context is the VNC session of the mobile node active in a VNC Proxy.

Figure 8 illustrates the signaling involved in the context transfer. An L2 trigger in the Client-based Handoff Mechanism simultaneously invokes the Context Transfer Trigger.

This initiates the sending of a Context Transfer Activate Request (CTAR) message which is repeatedly sent at a specified time interval from the VNC Viewer manager on the client to the VNC Proxy manager to guarantee a context transfer.

This initiates the sending of a Context Transfer Activate Request (CTAR) message to the nAR. This message contains the nAR and oAR IP addresses, the old care-of address of the mobile node, the new care-of address of the mobile node automatically configured using a Router Advertisement message from the nAR, a request for the mobile node's VNC session to be transferred and a token generated by the mobile node to authorise the context transfer from the oAR to the nAR.

Once the nAR receives the CTAR message, it sends a Context Transfer Request (CT Request) message to the oAR. This contains the mobile node's previous care-of address, a request for the mobile node's VNC session to be transferred and the token generated by the mobile node authorising the context transfer.

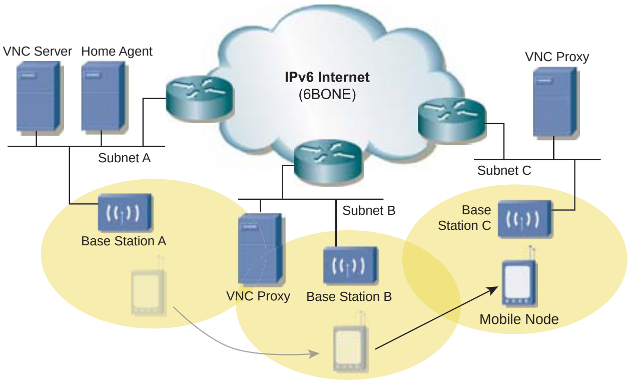

Figure 9: Supporting VNC mobility

The last 64-bit of the IPv6 address is set to the same identifier value for all VNC Proxies in the network shown in Figure 9. Therefore, the VNC Proxy IP address is a combination of the 64-bit prefix already present in the Router Advertisement and a universal 64-bit identifier for VNC Proxies in all networks instead of a unique 64-bit interface identifier as specified by RFC2374. Thus, the CTAR message sent from the mobile node VNC Viewer manager will always reach the VNC Proxy manager in the new network in which the mobile node has joined.

The CTAR message contains the port number of the old VNC Proxy. The number doubles as a method to uniquely identify the mobile node's VNC session. When the new VNC Proxy manager receives a unique CTAR message, it then sends a CT Request to the old VNC Proxy manager. Any subsequent duplicate CTAR messages are ignored. The old VNC Proxy manager responds to the new VNC Proxy manager with a CTD message containing the QoS information and the VNC Server name.

Note that the mobile node also connects to a VNC Proxy while in its home network. Hence, the VNC Server name does not need to be known by the mobile node.

Below is an example entry of a VNC session the VNC Proxy manager would be proxying:

$$\{\textit{VNC Server IP address}\}\text{::5901} \leftrightarrow \{\textit{VNC Proxy IP address}\}\text{::5999}$$

or

$$\{\textit{VNC Server name}\}\text{::5901} \leftrightarrow \{\textit{VNC Proxy name}\}\text{::5999}$$

The VNC Viewer manager in the mobile node would then connect to the VNC Proxy in the following way:

vncviewer {*VNC Proxy IP address*}::5999

or
vncviewer {*VNC Proxy name*}::5999

Once a CTD message is received from the old VNC Proxy manager, the new VNC Proxy manager invokes the VNC Proxy to open a session to the VNC Server. This allows screen updates and user input/output to be buffered and forwarded between the Server and Viewer. RSVP is initiated to perform a resource reservation, based on the QoS information in the CTD message, for the link between the VNC Proxy and the VNC Server (a receiver-orientated resource reservation, RFC2205). The new VNC Proxy issues a new VNC port number for the VNC Viewer to connect to this open session.

CXTP is extended to allow the new VNC Proxy manager to return a *CTAR Reply* message, which contains the new VNC Proxy port number issued for the VNC Viewer manager to connect the VNC Viewer to the VNC Proxy. Once the client connects to the VNC Proxy, a CTD Reply message is sent from the new VNC Proxy manager to the old VNC Proxy manager to terminate and tear down the relevant VNC session and RSVP path, respectively. If the old VNC Proxy does not receive this message, the session will expire after a set time. During an active VNC session between the VNC Server and VNC Proxy, RSVP sends periodic refresh messages to maintain the state along the reserved path.

VI.B. Experiments and Experiences

In all of the experiments, the following conditions were set for consistency in the final result.

- The VNC software was provided by RealVNC Ltd. The VNC Server was installed in the mobile node's home domain; the VNC Proxy was installed in the Access Router (AR) of each network; and the VNC Viewer was installed on the mobile node.
- A VNC session was initiated at the server so a VNC Viewer can connect to the session without any delay.
- `tcpdump` was used to log all traffic activities between the VNC Server and connecting VNC Viewer.
- Upon the execution of the VNC Viewer on the mobile device, a video clip was played using *mplayer*[4]. The sample video clip was a 25 frames per second MPEG-2 video.

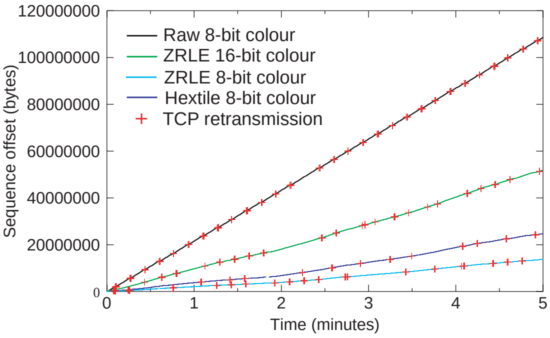

Figure 10: TCP sequence number plots for the various types of encoding offered by VNC. The mobile client running the VNC Viewer was tested under a handoff frequency of 6 handoffs per minute.

- The VNC Viewer was constrained to connect to the VNC Server for a limited time of 5 minutes which was sufficient to conduct the tests.
- The latency involved in setting up the security association and QoS paths were not considered.

Three experiments were carried out to investigate the effectiveness of the Client-based Handoff Mechanism and the VNC Proxy.

VI.B.1. Experiment 1

The first experiment involved the selection of a suitable type of VNC encoding used in the testing of the overall system. The version of VNC used in the tests offered the following encoding: raw, hextile and ZRLE (Zlib Run length Encoding).

VNC Server can send screen updates to the VNC Viewer in 8-bit and 16-bit colour. 8-bit colour was selected for all of the encoding schemes, and 16-bit colour was selected only for the encoding scheme which required the least amount of screen updates to be sent to the client. The VNC encoding scheme with the least number of screen updates was used for the remainder of the experiments.

The experiment did not simply involve a static session to a fixed client. The client was forced to perform 6 handoffs per minute to help determine the best VNC encoding scheme for a non-stationary user.

Figure 10 shows the result of the first experiment. Notice the raw encoding scheme has many times more TCP packet transmissions than the other encoding schemes due to the requirement for a greater screen update frequency. This scheme is highly unsuitable for mobile users when taking into account the average packet loss shown in Table 3. The encoding scheme

[4] mplayer, http://www.mplayerhq.hu/

with the least screen updates and packet loss is clearly ZRLE using 8-bit colour. This encoding scheme is used for the remaining experiments. Increasing the colour to 16-bit causes a higher number of packet loss as compared to the Hextile encoding making the higher colour option undesirable for the mobile client. The higher colour option will be advantageous for displaying video clips, otherwise, normal office applications do not require such a high colour depth.

VNC Encoding Scheme	Packet Loss
Raw	913
Hextile	123
ZRLE (8-bit colour)	118
ZRLE (16-bit colour)	256

Table 3: Average packet loss from 10 runs of a client running a 5-minute VNC session with video playback performing 6 handoffs per minute.

VI.B.2. Experiment 2

The second experiment looked into the mobility aspect of stateless mobile thin-client computing. The system was implemented as in Figure 9.

The intermediary routers in the testbed did not guarantee quality of service since this issue in IPv6 had not yet been agreed and set by the service providers' routers.

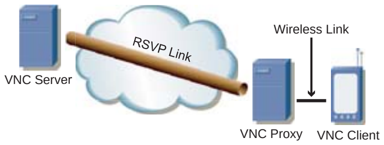

Figure 11: Guaranteeing link reliability over the IPv6 Internet

The logical diagram of how QoS could be guaranteed is shown in Figure 11. The diagram shows how a RSVP link would fit into the testbed. RSVP could not be implemented on routers in the IPv6 Internet (6BONE), thus the link was emulated by provisioning a direct 100Mbps Ethernet link between the VNC Server and VNC Proxy.

The mobile node was forced to perform a number of handoffs per minute. The condition of the tests was that the mobile node had to be within the wireless network coverage of at least two points of attachment, i.e. base station A and B as illustrated in Figure 9. The first set of tests was to observe the effectiveness of the VNC Proxy without the Client-based Handoff Mechanism. The second set of tests was based on the first set of tests but with the handoff mechanism enabled.

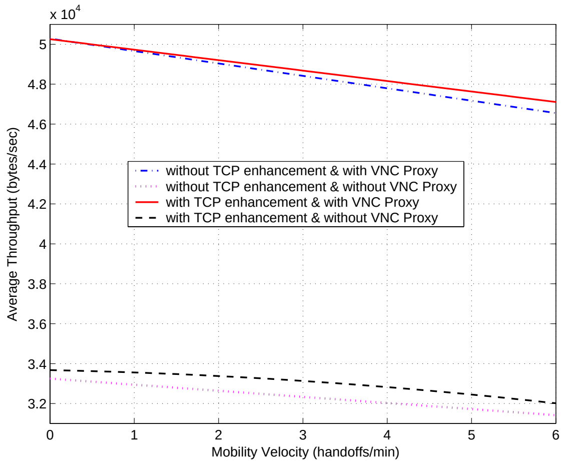

Figure 12: The improvement of the Client-based Handoff Mechanism and the VNC proxy on the average throughput of a 5-minute VNC session. In all of the experiments, the same video clip was played full-screened over the duration of the session.

In the second experiment, the VNC Proxy shows a clear improvement on the connection evident from the average throughput graph in Figure 12. The improvement on the throughput averaged at 47.0%.

The higher average throughput due to the Client-based Handoff Mechanism shown in Figure 12 labelled as a *TCP enhancement* clearly improve the TCP connection for cases where there is no VNC Proxy present.

VI.B.3. Experiment 3

Finally, the third experiment tested the effectiveness of the Client-based Handoff Mechanism in the event of subnetwork outages. The same system (see Figure 9) was used in this experiment. However, the mobile node was made to roam under the wireless network coverage of only one base station at any one time. The wireless coverage gap between base station B and base station C was set to 3 seconds, meaning that, while the mobile node was in this gap, it was disconnected from the network, hence a subnetwork outage.

Results from experiment 3 are illustrated in Figure 13 and 14. A higher number of screen updates were achievable with the help of the VNC Proxy as illustrated in Figure 14 evident by the higher throughput for cases where the VNC Proxy was used. Despite the

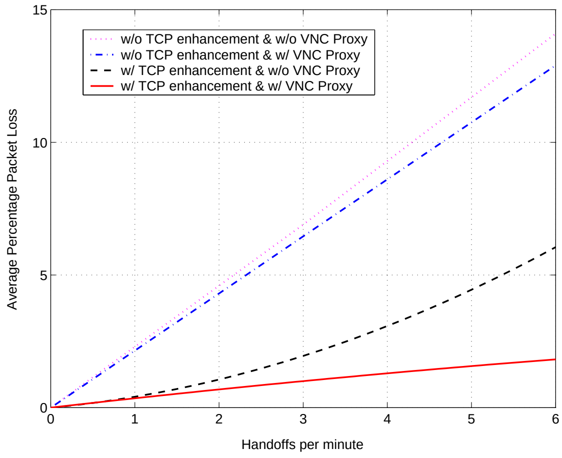

Figure 13: Average percentage packet loss over a 5-minute period VNC session with 3 seconds subnetwork outages between each handoffs.

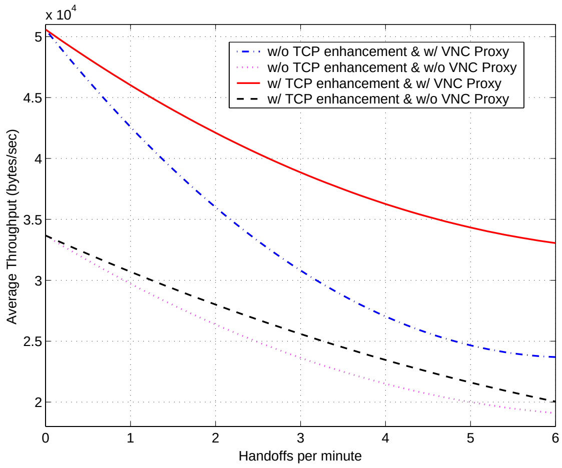

Figure 14: Average throughput over a 5-minute period VNC session with 3 seconds subnetwork outages between each handoffs.

higher number of screen updates, the average percentage packet loss due to the handoffs plotted in Figure 13 is low for VNC sessions assisted with a proxy as compared to sessions without a proxy. The packet loss in Figure 13 increases with a higher number of handoffs per minute which is associated with a higher user mobility.

VI.C. Results Summary

In experiments 2 and 3, due to the use of an experimental testbed and the WLAN device driver limitation to Ad-hoc mode, to achieve a reliable and consistent result, there had to be at least 8 seconds between each handoff. Unlike the Managed mode, the Ad-hoc mode restricts all devices to use the same frequency channel. This causes interference between the various base stations in the testbed. The Managed mode could not be configured due to the function restriction set by the WLAN vendor. Thus, a maximum of 6 handoffs per minute was attainable in the experiments.

A packet loss during a handoff equates to the loss of a VNC Client user input, VNC Server/Proxy *FrameBufferUpdate* packet or VNC Client acknowledgement packet. The loss of a user input would mean that the user will need to duplicate the input action. For the two latter types of packet loss, the VNC Client will need to trigger a FrameBufferUpdate request to force a display update immediately after a handoff. This feature could improve the user experience, but has not been adopted in our architecture.

VII. Related Work

In this section, some Mobile IP enhancements are surveyed, limiting the scope to handoff management since it is the key problem area which can effect real-time applications. Some comparison work [8] has been done in the past by Cheshire and Baker with a focus on routing issues when using Mobile IP. The location management issue is already addressed by the Mobile IP binding process which keeps track of the mobile host's location. The power consumption issues are beyond the scope of our work.

Solutions to better support the Mobile IP handoff process involve a combination of one or more protocol layers, e.g. link and network layers, or link and transport layers. A survey of these approaches is described below.

VII.A. Handoff Management

Wu *et al.* proposed an Intelligent Handoff Architecture [9] which is based on a WLAN environment. The authors modified the basic WLAN handoff algorithm to support handoff with Mobile IP when roaming. The technique uses the received signal strength (RSSI) and frame error rate (FER) as the deciding factor for handing off. It suggests changes to the HA to support ICMP notification to buffer packets for the MH when it is about to handoff. However, this can introduce scalability problems if a HA has to support many MHs although the paper argues the HA requires only a small space to buffer the packet temporarily. The paper briefly mentions a neighbour list which is stored in each mobility agent providing the MH global information related to the network topology. However, it does not make it clear how this list is created.

The architecture does not indicate how it will work for wireless network technologies other than WLAN. No implementation description or results have been given in this article.

Yokota *et al.* proposes a link-layer assisted method [10] to improve handoff performance in Mobile IPv4 networks. It does not introduce modifications to Mobile IP like that of Wu *et al.* but introduces a new network entity called a *MAC Bridge* and modifications to the WLAN Access Point (AP) instead. The AP and MAC Bridge functions appear to be similar to that of a HA and FA where the MAC address of the Mobile Node needs to be established before proceeding with the Mobile IP mechanism. This association is required to minimise packet loss during the Mobile IP registration period. The technique may be redundant since packets from the CN can still be received during the registration process with the new FA. However, in the case of AP failure or hard handoffs, the technique can prove to be valuable.

Balakrishnan *et al.* proposed Snoop [11] which is aimed at improving TCP applications based on the link layer information. It assumes the wireless link is lossy, thus introduces a way to sniff in-flight packets using a base station as a proxy. This proxy sends ACKs and suppresses duplicate ACKs to avoid the sender using congestion control when there are packet losses over the wireless link. These specialised base stations are required to have a large memory and processing power to store network packets while handoffs take place.

Omae *et al.* proposed and showed simulated results of a method to improve handoff performance in Mobile IPv6 networks with the use of a buffer implemented at the mobile node [12]. UDP and TCP packets are buffered in the mobile node in order to minimise packet loss in the event of a handoff. This technique can affect real-time traffic. In [13] a technique to improve handoff for real-time traffic however, employs a two-path handoff technique which uses properties of IPv6 and the Integrated Services (IntServ) QoS architecture.

Freeze-TCP by Goff *et al.* [14] and the more recent Internet Draft by Eggert *et al.* [15] are techniques to reduce handoff disruptions to TCP applications. The former solution makes use of the TCP persist mode method to stop the transmission of packets. The later solution uses the fast retransmit and fast recovery TCP mechanism to quickly resume a TCP connection. However, both techniques are not optimised for various types of handoff scenarios.

VII.B. Subnetwork Outage (Disconnection)

The techniques discussed in the previous sections solve handoff issues but neglect cases when there are subnetwork outages [16]. Nonetheless, there is a technique called M-TCP [17] which take this factor into consideration. It proposes the use of delayed ACKs sent on behalf of a receiver by means of a proxy to place the sender in persist mode to avoid losing packets during handoffs.

Proposals before the efforts recently initiated by the IETF working groups do not consider disruptions to TCP sessions based on the link type and network layer signaling (i.e. Mobile IP, IPv6) and therefore may not necessarily work and could in fact introduce unnecessary delays or overheads. Although the discussed solutions maintain end-to-end semantics, other solutions do not abide by this rule. MTCP [18] and I-TCP [19] break this rule by splitting the wireless and wired part and place, what is effectively, a proxy which acts on behalf of a mobile host to improve TCP performance.

VII.C. Efforts within the IETF

Two documents have been proposed within the *mobileip* charters (mip4 and mip6) to reduce the handoff latency of Mobile IP. These are Low Latency Mobile IPv4 Handoffs [20] and Fast Handovers for Mobile IPv6 (RFC 4068).

Recently, Williams and Pagtzis proposed a scheme called Localised Mobility Management (LMM) [21, 22] to minimise the Mobile IP signaling traffic to the Home Agent and/or Correspondent Node(s) for intra-domain mobility. They argue that signalling messages could take more than one hundred milliseconds when the mobile node is at some geographical and topological distance away from the CN and HA. This increases handoff latency, hence, packet loss at the old Access Router for the MN. The scheme introduces a LMM agent at the local subnet level to allow the MN to continue receiving traffic on the new subnet without any change in the HA or CN binding. However, this reintroduces triangular routing to Mobile IPv6. Thus, the scheme may not scale since LMM agents are required to minimise the length of the triangle leg it introduced to reduce the handoff latency. The LMM scheme is yet to be regarded as a proven technique.

Another IETF working group called *PILC* looks into defining how the IP Protocol Suite works with different types of link layers. A recent Internet Draft [23] attempts to characterise links and set out best-practice suggestions for Internet subnetwork designers. One recommendation made by the document to avoid discarding packets during a subnetwork outage is for an interface to be defined to the IP and higher layers allowing it to refuse sending packets when there is an outage, and for the interface to automatically ask IP for new packets once the link has been restored. If this is not feasible, it is recommended that the link layer retains one or more of the packets which could not be transmitted during the outage period, and retransmits these packets on reconnection.

A recent work in the IETF, based on the efforts by PILC, includes the development of the *TRIGTRAN* framework [24]. It proposes a mechanism to alert the transport layer about changes in individual links along the network path from source to destination. With this framework, hosts may request notification when trigger events such as Connectivity Interrupted, Connectivity Restored and Packets Discarded by Subnet occur.

The IETF PILC working group is in its early stages and therefore are lacking experimental support. There is no concrete implementation to validate their proposal and it is hoped these efforts would be widely supported in the near future.

VIII. Conclusions

The impact of the research work described in this article is twofold. Firstly, the Client-based Handoff Mechanism is a simple solution to provide a controlled handoff technique and a reduction in the handoff latency for IPv6 networks with Mobile IP support, i.e. beyond 3G networks. It reduces the mobile node's dependence on Mobile IP mechanisms: the router advertisement interval and the router solicitation behaviour. The concept of an RA cache and externally triggering the TCP mechanism have been shown to substantially improve the mobile computing experience as demonstrated in our testbed.

Secondly, the thin-client application shows how to take full advantage of the Client-based Handoff Mechanism and the next generation Internet to overcome the unpredictability and unreliability of IP networking. A Mobile VNC Architecture is proposed and evaluated to support mobile thin-client computing in wireless IP networks as opposed to the requirement for wireless ATM networks (i.e., the Videotile). Results show a high and relatively consistent throughput when a mobile thin-client is running a VNC session. In the event of subnetwork outages, packet loss and the effect of TCP slow start are entirely avoided.

IX. Future Work

The current prototype of the Client-based Handoff Mechanism evaluated in the testbed is well suited to significantly limiting packet loss for a mobile node's download (asymmetric data transfer) stream when handing off. This can be extended to support symmetric data transfers. For example, the data rate for the upload stream can be maintained by upsetting the TCP persist timer in the mobile node.

Furthermore, as an extension to our solution, the user can have the option of overriding the handoff mechanism to select their preferred network.

Vidales, Patanapongpibul and Chakravorty [2] began some early implementation work which makes use of the Client-based Handoff Mechanism in a heterogeneous network environment.

Mobile-controlled handoff has its benefits for heterogeneous networking. Nevertheless, there are several advantages for network-assisted handoffs. One advantage is that the network has the status information of the base stations adjacent to the current base station to which the mobile host is attached. Another advantage is the ability to optimise the radio resource across the entire network by ensuring the mobile host has a channel to the network. These advantages can be leveraged by the mobile host to further enhance the mobile computing experience. For heterogeneous networking, it is ideal that the status of the network and radio resource are freely available. A new protocol designed to be compatible with various wireless network technologies would be one solution for the mobile host to receive network status information. Such a protocol can be integrated into our handoff algorithm to achieve network-assisted handoffs.

Peering agreements is an issue that will be debated well into the future and will become an increasingly challenging problem when mobile devices are able to roam to any type of network. This discussion is beyond the scope of this article but is an area for further investigation.

Finally, in addition to supporting mobile thin-client computing in the wide area network, a mobile node could run its own operating system with the Coda network file system [25] for data storage. Suitably adapted versions of the Coda file server and Coda proxy file server could be used as substitutes for the VNC server and VNC proxy, respectively.

X. Acknowledgement

The authors would like to thank Pablo Vidales, Rajiv Chakravorty, Andrew Campbell and Jon Crowcroft for their support. We would also like to thank the Computer Laboratory and the Engineering Department for facilitating the roll out of the Mobile IPv6 testbed. Thanks also to Vodafone for their experimental 3G network in Cambridge, UK. EPSRC and AT&T Labs - Cambridge made this project possible by providing funding for equipment and personnel.

References

[1] P. Vidales, L.B. Patanapongpibul, G. Mapp, and A. Hopper, "Experiences with Heterogeneous Wireless Networks - Unveiling the Challenges," in *the Proceedings of the 2nd International Working Conference on Performance Modeling and Evaluation of Heterogeneous Networks (HET-NETs'04)*, July 2004.

[2] P. Vidales, L.B. Patanapongpibul, and R. Chakravorty, "Ubiquitous Networking in Heterogeneous Environments," in *the Proceedings of the 8th International Workshop on Mobile Multimedia Communications (MoMuc)*, October 2003.

[3] R. Chakravorty, P. Vidales, K. Subramanian, I. Pratt, and J. Crowcroft, "Practical experiences with wireless networks integration using mobile ipv6," Poster and Extended Abstract in the Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), September 2003.

[4] "An Introduction to the Vodafone GPRS Environment and Supported Services," Tech. Rep. Issue 1.1/1200, Vodafone Ltd., December 2000.

[5] Mark Stemm and Randy H. Katz, "Vertical Handoffs in Wireless Overlay Networks," *Mobile Networks and Applications*, vol. 3, no. 4, pp. 335–350, 1999.

[6] "tcptrace(http://www.tcptrace.org), tcptrace+(http://www-lce.eng.cam.ac.uk/∼lbp22/opensource.html), tcpdump(http://www.tcpdump.org), iperf(http://dast.nlanr.net/projects/iperf/)," .

[7] J. Nieh, S. J. Yang, and N. Novik, "Measuring Thin-Client Performance Using Slow-Motion Benchmarking," *ACM Transactions on Computer Systems*, vol. 21, no. 1, pp. 87–115, February 2003.

[8] Stuart Cheshire and Mary Baker, "Internet Mobility 4x4," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, August 1996, pp. 318–329.

[9] Nen-Fu Huang Jon Chiung-Shien Wu, Chieh-Wen Chieng and Gin-Kou Ma, "Intelligent Handoff for Mobile Wireless Internet," *Mobile Networks and Applications*, vol. 6, no. 1, pp. 67–79, 2001.

[10] Toru Hasegawa Hidetoshi Yokota, Akira Idoue and Toshihiko Kata, "Link Layer Assisted Mobile IP Fast Handoff Method over Wireless LAN Networks," in *Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, Sept. 2002, pp. 131–139.

[11] H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," *ACM Wireless Networks*, vol. 1, no. 4, pp. 469–481, 1995.

[12] Koji Omae, Takehiro Ikeda, Masahiro Inoue, Ichiro Okajima, and Narumi Umeda, "Mobile Node Extension Employing Buffering Function to Improve Handoff Performance," in *Proceedings of the Fifth International Symposium on Wireless Personal Multimedia Communications*, 2002, vol. 1, pp. 62–66.

[13] Janise McNair, Ian F. Akyildiz, and Michael D. Bender, "Handoffs for Real-Time Traffic in Mobile IP Version 6 Networks," in *IEEE Global Telecommunications Conference (GLOBECOM)*, November 2001, vol. 6, pp. 3463–3467.

[14] T. Goff, J. Moronski, D. S. Phatak, and V. Gupta, "Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments," in *Proceedings of INFOCOM*, 2000, pp. 1537–1545.

[15] L. Eggert, S. Schuetz, and S. Schmid, "TCP Extensions for Immediate Retransmissions," IETF Internet-Draft, June 2001, draft-eggert-tcpm-tcp-retransmit-now-02.txt, work in progress.

[16] G. H. Forman and J. Zahorjan, "The Challenges of Mobile Computing," *Computer*, vol. 27, no. 4, pp. 38–47, April 1994.

[17] K. Brown and S. Singh, “M-TCP: TCP for mobile cellular networks,” *ACM Computer Communication Review*, vol. 27, no. 5, pp. 19–43, 1997.

[18] R. Yavatkar and N. Bhagawat, “Improving End-to-End Performance of TCP over Mobile Internetworks,” in *IEEE Workshop on Mobile Computing Systems and Applications*, December 1994.

[19] A. V. Bakre and B. R. Badrinath, “I-TCP: Indirect TCP for mobile hosts,” in *Proceedings of the 15th International Conference on Distributed Computing Systems*, June 1995, pp. 136–143.

[20] Karim El Malki, “Low Latency Handoffs in Mobile IPv4,” IETF Internet-Draft, December 2003, draft-ietf-mobileip-lowlatency-handoffs-v4-05.txt, work in progress.

[21] T. Pagtzis, C. Williams, P. Kirstein, C. Perkins, and A. Yegin, “Requirements for Localised IP Mobility Management,” in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, Louisiana, New Orleans, March 2003, pp. 1979–1986.

[22] C. Williams, “Localized mobility management requirements,” IETF Internet-Draft, March 2003, draft-ietf-mobileip-lmm-requirements-03.txt, work in progress.

[23] P. Karn, “Advice for Internet Subnetwork Designers. Performance Implications of Link Characteristics (PILC),” IETF Internet-Draft, Feb. 2003, draft-ietf-pilc-link-design-13.txt, work in progress.

[24] S. Dawkins, C. E. Williams, and A. E. Yegin, “Framework and Requirements for TRIGTRAN,” IETF Internet-Draft, Feb. 2003, draft-dawkins-trigtran-framework-00.txt, work in progress.

[25] M. Satyanarayanan, “Mobile Information Access,” *IEEE Personal Communications*, vol. 3, no. 1, pp. 26–33, February 1996.