Mobility Support for Thin-Client Systems

Leo Patanapongpibul
Laboratory for Communication Enginnering
University of Cambridge
Cambridge, United Kingdom
lbp22@eng.cam.ac.uk

Glenford Mapp
School of Computer Science
Middlesex University
United Kingdom
g.mapp@mdx.ac.uk

Abstract—**Mobile IP enables mobile computers to roam transparently in any network. However, the current proposed protocol specification does not support a suitable handoff mechanism to allow a mobile computer to change its point of attachment from one network to another. This paper describes a technique to support thin-client systems with our handoff mechanism while providing subnetwork outage support for a mobile host which makes use of Internet Protocol version 6 (IPv6) and Mobile IP without the need to introduce a new mobility management protocol or make changes to the network infrastructure. Results from handoff experiments show a dramatic reduction in the handoff latency and that mobility support for thin-clients is feasible.**

I. Introduction

The proliferation of mobile computers has created a need for transparent mobility. Internet Protocol version 4 (IPv4) is widely used in all networks but is a relatively old protocol originally designed for wired networks. With the advent of wireless computing, new problems have emerged which challenge the capabilities of IPv4.

Over the years, the research community has introduced new methods to overcome these problems and support mobile networking. Perkins [1] introduced Mobile IP for IPv4 to support mobile hosts roaming away from their home network domain, thereby allowing them to retain active network sessions without having to restart their network services.

In the first Mobile IPv4 proposals, there were problems with triangular routing, security and other wireless networking issues, including the need to add new components to the IPv4 network infrastructure. The IETF Mobile IP working group was created to solve these problems and refine the protocol. Mobile IP for IPv4 is now an Internet standard (RFC3344) whereas Mobile IP for IPv6 is on course to becoming a standard.

Since IPv6 was designed to replace IPv4, considerations for introducing new functionalities and improving on IPv4 were taken into account. IPv6 routers have built-in functions eliminating the need for a Foreign Agent. Triangular routing and tunneling required for Mobile IPv4 can now be avoided through IPv6 routing headers. Security problems are intrinsically solved with improved addressing architecture and scalability issues are overcome with its 128-bit address space.

Despite all the benefits from IPv6, Mobile IP still needs some refinements. One such refinement includes a handoff mechanism and provision for mobility management.

II. Motivation for Supporting Thin Clients in Wireless Networks

We avoid the problem with migrating processes, eliminating the need of Mobile Agents with the use of the Virtual Network Computing (VNC) system – an ultra thin-client computing approach. Other than the mobile extension to the Open Group's Network Computer Reference Profile, there has been little development to enable mobile thin-client computing. Mobile network computing tends to require a window system already installed on the client while application process runs remotely or gets downloaded to the client. Some of these are mature technologies such as the X Window System, CORBA and Mobile Agents which require an application platform to run on a “thick” client.

The IETF have laid the ground work for context transfer at the network layer. The Client-based Handoff Mechanism [2] and the technique used by the fast handoff method [3] can be extended to support signaling messages to support a thin-client system for mobile computing. A similar method can be used for the migration of Mobile Agents, but would require a higher overhead due to the variety and complexity of processes.

Portable stateful computing devices (e.g. laptops and PDAs) are widely available and come with a greater amount of processing power, storage capacity and longer battery life increasingly at a more affordable price. However, such devices are prone to damage or theft. Other inconveniences are such as the necessary effort to synchronise or backup portable data with a fixed computer for data retention.

Unlike stateful devices, stateless devices do not run application or system code on the appliance. In this paper, 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 fix 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 to changes in network conditions is necessary to minimise disruptions to user computing interaction.

III. Background to Thin-client Systems

There are five categories of thin-client systems:

- **Ultra-thin client systems** where both the application and windowing system are all executed on the server such as VNC.
- **Network window systems** notably the X window system [4] and LBX to support X over a low-bandwidth connection [5]. The windowing system is executed on the client whereas the application runs on the server.
- **Network computer systems**, such as Novell Netware[1], require applications and the windowing system to execute on the client. Other than an operating system, the client does not store any application locally and needs to download the application from a remote server.
- **Browser-based systems** use the web browser on the client to interface to an application server, typically used for middleware applications.
- **Remote control computing systems** allow only one user at any time to control a PC (server), where all the windowing system and applications are executed on the server, by sending screen updates to the client PC. For example, LapLink [2] and PC Anywhere [3].

This research is only concerned with *ultra-thin client systems* due to its need to support multiple users, to provide a centrally managed system and to impose the minimum amount of data storage, battery power consumption and size of the device on the mobile client. The simplicity of administrating an ultra-thin client system is the key driver to extend its use to a high mobility environment.

There are a number of ultra-thin client system vendors on the market. The most popular and fully functional systems are ORL/AT&T VNC, Citrix Systems Metaframe, Sun Microsystem Sunray, Microsoft Remote Desktop Protocol (commonly known to be used in the Windows Terminal Server), SCO Tarantella and Graphon RapidX. With the exception of VNC, all of these systems are proprietary products and have their strengths and pitfalls. Nieh *et al.* benchmarked the performance of some of these systems [6]. The results indicated that VNC and Sunray offer faster web browsing when a high network bandwidth is available. Citrix and the Microsoft Remote Desktop Protocol (RDP) provide an optimised and better encoding scheme which made them perform better than VNC and Sunray in networks running at lower bandwidths. For video playback, Sunray was the best performer. Because the testbed in this research is based on the Linux operating system, the most appropriate ultra-thin client system was VNC. As Nieh has shown, VNC performs well at high bandwidth, thus the work in this paper will show an architecture to reliably maintain a high speed connection for “thin” mobile devices.

IV. Device mobility of Stateless Thin Clients

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[4], used an indoor wireless ATM technology limiting its use inside a building. However, with the advent of wireless LAN and higher speed data access through cellular networks (e.g., 3G), the feasibility of thin client device mobility are 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 paper introduces an architecture which makes use of exiting IPv6 and newly proposed network signaling methods for supporting the roaming of mobile nodes in wireless IP networks (rather than ATM).

[1] Novell Inc., http://www.novell.com/products/netware6/

[2] Laplink Software Inc., http://www.laplink.com/

[3] Symantec Corporation, http://www.symantec.com/pcanywhere/

[4] The Videotile, 1996 http://www.uk.research.att.com/tile.html

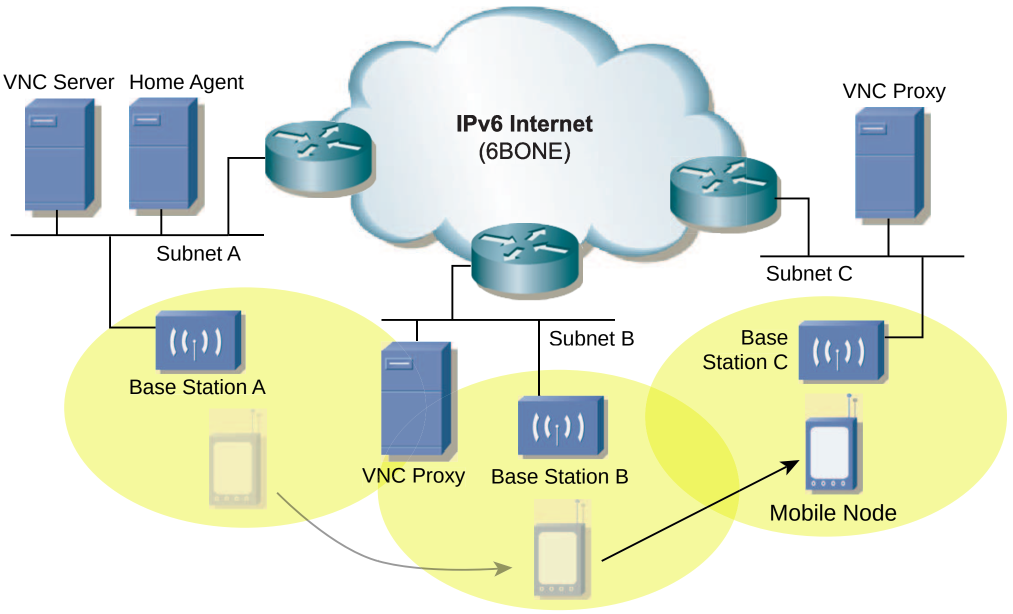

Fig. 1. Supporting mobility and VNC

V. The Mobile VNC Architecture

The Mobile VNC Architecture 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.

RealVNC Ltd. has provided us with the VNC system. The VNC server is installed in the mobile node's home domain; the VNC Proxy is installed in the Access Router (AR) of each network; and the VNC viewer is installed on the mobile node.

In this section, an overview of VNC is given followed by the key enabler for global mobility with thin-client devices running a VNC viewer. Section V-C explains the method used to guarantee the quality of service between the VNC Server and VNC Proxy. Finally, the signaling mechanism used to support the roaming user in the Mobile IPv6 testbed is described.

A. Overview of VNC

The idea originated from the Videotile – a display device with an LCD screen, a pen and an ATM network connection – where the user interface is a video, streamed as tiles on to this remote display with only those parts of the screen that has changed.

Since then, ORL has taken the Network Computer (NC) and the Teleporting System [7], which uses the X Window System, ideas further with VNC – an ultra-thin client.

The virtual network computing (VNC) system is a method to execute applications and the window system remotely on a server from a client. There are two components to VNC: a VNC server daemon runs on a server and a VNC viewer runs [upon execution by the user] on the client machine to connect to the server. The protocol used between the server and client is called the VNC protocol which can operate over TCP/IP. VNC has been built for Microsoft Windows, UNIX, Linux, etc. It is open source, allowing anyone to make additions, modifications or port the system to any operating system platform.

The display on the server is encoded in a serial fashion before it is shipped to the viewer for display.

Because the framework of this research has been built to show the application of the next generation Internet, VNC has been extended to support IPv6. IPv4 is not a suitable protocol for supporting mobility without drastic changes to the network infrastructure.

B. Key Enabler for Global Mobility

To support a roaming wireless thin device, network bandwidth is a key limiting factor. The wireless signal can vary depending upon the location. Therefore, in this work, it is assumed there is network coverage throughout the globe. The next most important limiting factor is the reliability of the network connectivity.

A VNC proxy is introduced to resolve the network latency issue. The advantages of introducing such an entity into the network infrastructure are:

- Reduces the number of TCP retransmissions and screen updates between the server and client.

- Secures the network without having to bypass firewalls.
- Transparent accounting and billing
- Link speed and bandwidth between the server and proxy can be guaranteed with QoS.

There are a number of TCP-based proxies such as I-TCP [8] and MTCP [9] which split the connection between the wired and wireless part. The end-to-end semantics is not maintained in such cases which introduces many security and privacy issues. With an application such as a VNC Proxy, the end-to-end semantics is conserved.

C. Quality of Service between the VNC Server and VNC Proxy

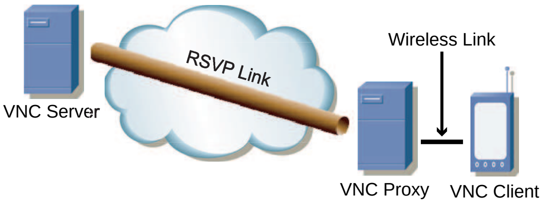

Fig. 2. Guaranteeing link reliability over the IPv6 Internet

Guaranteeing the quality of service for a flow can be achieved by one of the following two methods: through information in the IP header or by a control protocol such as the Reservation Protocol (RSVP).

IPv4 Type of Service field in the header offers a highly coarse granular *Differentiated Services* (DiffServ) quality of service as oppose to IPv6 which implements the equivalent Traffic Class field with an additional Flow Label field offering a finer granular DiffServ.

The VNC server would have to supply a value for the Traffic Class via the service interface to the IPv6 service. This is necessary for forwarding routers to identify and distinguish the priorities of various IPv6 packets. Similarly to IPv4, this field is still an ongoing development to provide a number of DiffServ for IP packets rather than using a control protocol to guarantee quality of service.

Further special handling by the intermediate IPv6 routers can be requested by the VNC server with the Flow Label field, however, this field is still experimental and can change as the Internet makes the transition from IPv4 to IPv6. Currently, in the 6BONE, the field is ignored by most routers. The source would set the field to a default value of zero. The Flow Label helps intermediary routers to identify the type of packet into pre-determined flows, hence substantially reducing computational overhead. The disadvantage of DiffServ is quality of service is not guaranteed.

RSVP is a common control protocol to provide quality of service for *Integrated Services* (IntServ) where applications can choose among multiple and controlled levels of delivery service for their data packets. This requires intermediate nodes in the network to support the control messages, thus guaranteeing quality of service. IntServ does not scale as well as DiffServ due to the additional support required by the forwarding routers.

Because the future of DiffServ support in IPv4 and IPv6 is not yet clear, in this work, RSVP is the most suitable method to guarantee the quality of service for the Mobile VNC architecture.

D. Signaling Mechanism: transferring VNC sessions between VNC Proxies

The Context Transfer Protocol (CTP) [10] proposed within the *Seamoby* IETF working group charter was 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. However, there is a counter argument which may be more appropriate for a network-controlled approach since a VNC Proxy would need to be present in every network or subnetwork.

CTP involves the following network entities: the mobile node, the old access router (oAR) and the new access router (nAR). Note, the old access router is an entity that offered connectivity to the mobile node prior to a handoff and may have ceased or will cease to offer connectivity to the mobile node after a handoff. The new access router is the access router that offers connectivity to the mobile node after a handoff. To be in accordance with the CTP Internet Draft, all of these entities support IPSec ESP [11] with the replay protection mechanisms to provide per packet authentication, integrity protection and confidentiality.

The Client-based Handoff Mechanism [2] is extended to support the the *Context Transfer Trigger* required to initiate the transfer of context between access routers, in this case, the context is the VNC session of the mobile node active in a VNC Proxy. Because of this, it is naturally a mobile-controlled method since the request started at the mobile node.

Figure 3 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 to the nAR. This message contains the nAR and oAR IP addresses, the old care-

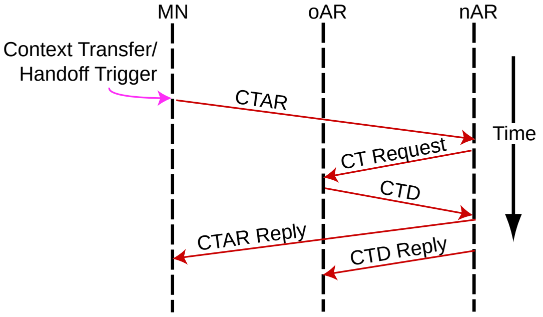

Fig. 3. Mobile-controlled: Context transfer protocol signaling message fbw initiated by the mobile node (MN).

of address of the mobile node, the new care-of address of the mobile node automatically configured from 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.

The authorisation token is verified by the oAR with a security algorithm. The token is computed with the HMAC-SHA-1 [12], [13] algorithm with the following inputs: the mobile node's previous care-of address, the Feature Profile Types (FPT) objects and the Replay field. FPTs are registered number space that allows a node to identify the type of context and the context parameters present in the protocol messages. The replay field is obtained through IPSec. The final authorisation token is the leading 32 bits obtained from truncating the results of the HMAC-SHA-1 security algorithm.

Once the token is verfied by the oAR, a Context Transfer Data (CTD) message is sent to the nAR. The message contains the mobile node's previous care-of address, the mobile node's new care-of address and the feature context of the VNC session.

The CTP Internet Draft suggests that a CTD Reply message can be returned to the oAR to inform it the processing progress of the received context. This option was not implemented in this work.

VI. Experiments

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

- 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*[5]. The sample video clip is a 25 frames per second MPEG-2 video.
- The VNC viewer was constrained to connect to the VNC server for a limited time of 5 minutes.
- The implementation of the Context Transfer Protocol does not include the security aspects.

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

1) *Experiment 1:* The first experiment involved selecting a suitable type of VNC encoding to be used in the testing of the overall system. The version of VNC used in the tests offers the following encoding: raw, hextile and ZRLE.

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 requires 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 fix client. The client was forced to perform 6 handoffs per minute to help determine the best VNC encoding scheme for a non-stationary user.

2) *Experiment 2:* The second experiment looked into the mobility aspect of stateless thin-client computing. The system was implemented as in Figure 1.

IPv6 offers a fine granular DiffServ behaviour with the Flow Label field in the header. This field is for intermediary routers to identify the type of packet into pre-determined flows. However, the intermediary routers in our testbed do not guarantee quality of service since these flow labels have to be agreed and set by the service providers' routers.

RSVP was used between VNC server and VNC proxy to guarantee the network bandwidth over the IPv6 Internet. The logical diagram of how the RSVP link would fit into the testbed is shown in Figure 2. 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 performed a number of handoffs per minute. The condition of the tests was the

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

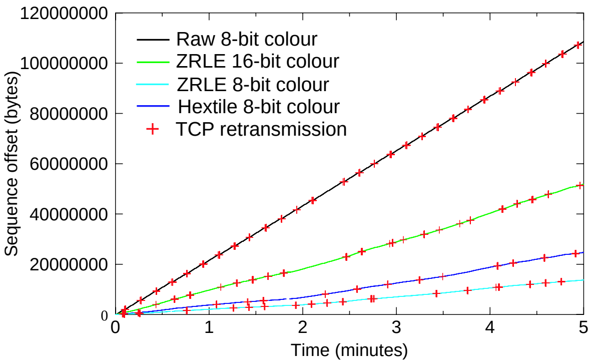

Fig. 4. TCP sequence number plots for the various types of encoding offered by VNC. The mobile client running the VNC viewer was tested under a handoff frequency of 6 handoffs per minute.

mobile node had to be constantly under the wireless network coverage of at least two points of attachment, i.e. base station A and B as illustrated in Figure 1. The first set of tests was to see the effectiveness of the VNC proxy without the Client-based Handoff Mechanism. The second set of tests was to use the mechanism while testing the improvement in screen updates with the VNC proxy.

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 1) 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 while the mobile node was in this gap, it is disconnected from the network, hence a subnetwork outage.

VII. Results and Discussion

Figure 4 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 also considering the average packet loss shown in Table I. The encoding scheme with the least screen updates and packet loss is clearly ZRLE with 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.

In the second experiment, the VNC Proxy showed a clear improvement on the connection evident from the av-

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

TABLE I

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

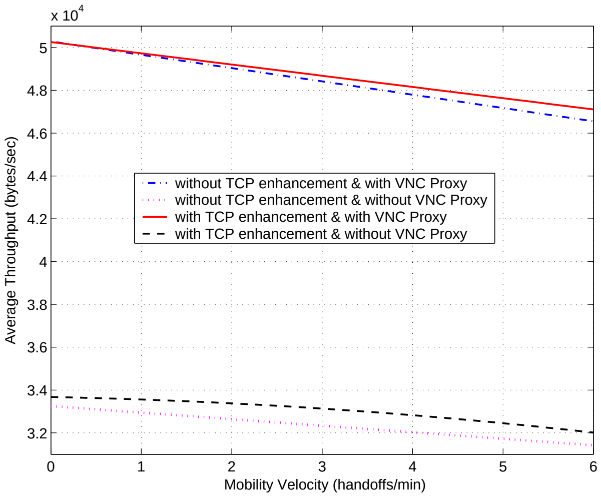

Fig. 5. The improvement of the client-based handoff mechanism and the VNC proxy on the average throughput of a 5-minute VNC session. In all of the experiments, the same video clip was played full-screened over the duration of the session.

erage throughput graph in Figure 5. The improvement on the throughput averaged at 47.0%.

The higher average throughput due to the Client-based Handoff Mechanism shown in Figure 5 labelled as a *TCP enhancement* clearly improved the TCP connection for cases where there is no VNC Proxy present.

Results from experiment 3 are illustrated in Figure 6 and 7. A higher number of screen updates were achievable with the help of the VNC Proxy as illustrated in Figure 7 evident by the higher throughput for cases where the VNC Proxy was used. Despite the higher number of screen updates, the average percentage packet loss due to the handoffs plotted in Figure 6 is low for VNC sessions assisted with a proxy as compared to sessions without a proxy. The packet loss in Figure 6 increase with a higher number of handoffs per minute which is associated with a higher user mobility.

In experiments 2 and 3, due to the use of an experimental testbed and the wireless LAN device driver limitation to Ad-hoc mode, to achieve a reliable and consistent

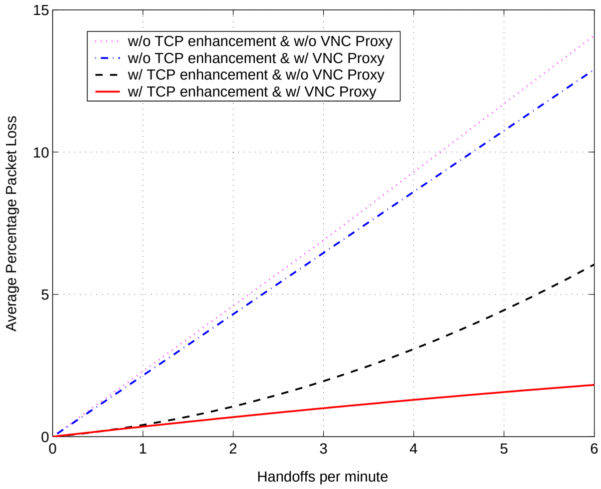

Fig. 6. Average percentage packet loss over a 5-minute period VNC session with 3 seconds subnetwork outages between each handoffs.

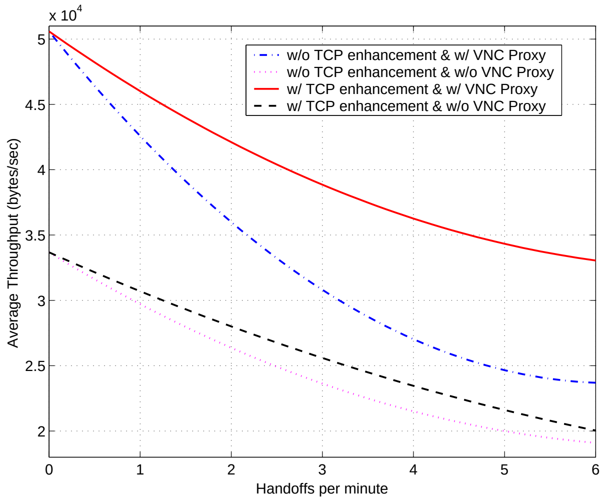

Fig. 7. Average throughput over a 5-minute period VNC session with 3 seconds subnetwork outages between each handoffs.

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 wireless LAN vendor. Thus, a maximum of 6 handoffs per minute was attainable in the experiments.

VIII. Conclusion

The Client-based Handoff Mechanism was used as a simple solution to provide a controlled handoff technique and a reduction in the handoff latency for supporting thin-client devices in IPv6 networks with Mobile IP support. The solution offers a decision making mechanism, known as “triggers,” for handoffs and a method to reduce the mobile host dependability on the router advertisement period and router solicitation. The Client-based Handoff Mechanism introduced the concept of a RA cache has been proven to reduce the handoff latency in our testbed. With the addition of the TCP reconnection methods in the mechanism, the handoff latency was reduced further. Periods of network outages can be determined by the mechanism and were tackled by forcing the sender into TCP persist mode. Even with a minimum outage time (close to zero seconds), the handoff latency was reduced substantially compared to handoffs without the mechanism.

With the VNC Proxy and guaranteeing QoS between the VNC server and VNC proxy, we achieved a higher number of screen updates on the VNC viewer. This clearly reflects the responsiveness to user interactions on thin-client devices. Thus, making ultra-thin client systems more usable for mobile networking.

IX. Future Work

With the help of the Client-based Handoff Mechanism, the ultra thin-client system described in this paper is made possible with benefits such as the reduction in the administrative overhead required to manage roaming hosts and, simultaneously, minimising disruptions to their network connections.

This work can be applied for heterogeneous IP network environments since no changes are required to entities in the network infrastructure. A GPRS-WLAN-LAN Mobile IPv6 network testbed has been set up in our lab in conjunction with the Computer Laboratory, University of Cambridge which makes use of the Client-based Handoff Mechanism [14]. Performance tests and measurements are being carried out to show the value of the work proposed in this paper for such environments.

X. Acknowledgements

The authors would like to thank Dr James Weatherall for his assistance with VNC. Further gratitude goes to Prof Andy Hopper for his guidance. This project is funded by the UK Engineering and Physical Sciences Research Council and AT&T Laboratories - Cambridge.

References

[1] C.E. Perkins, “IP Mobility Support,” IETF Request for comments, RFC 2002, October 1996.

[2] Leo Patanapongpibul and Glenford Mapp, “A Client-based Handoff Mechanism for Mobile IPv6 Wireless Networks,” in *Proceedings of the Eighth IEEE International Symposium on Computers and Communication (ISCC)*, June 2003, pp. 563–568.

[3] R. Koodli, “Fast Handovers for Mobile IPv6,” IETF Internet-Draft, draft-ietf-mobileip-fast-mipv6-06.txt, March 2003.

[4] R. Scheifler, J. Gettys, and R. Newman, *The X Window System: C Library and Protocol Reference*, DEC Press, 1988.

[5] J. Fulton and C. Kantarjiev, "An Update on Low Bandwidth X (LBX)," *The X Resource*, vol. 1, no. 5, pp. 251–266, January 1993.

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

[7] T. Richardson, G. Mapp, F. Bennett, and A. Hopper, "Teleporting in an X Window System Environment," *IEEE Personal Communications Magazine*, vol. 1, no. 3, pp. 6–12, Third Quarter 1994.

[8] 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.

[9] 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.

[10] J. Loughney, M. Nakhjiri, C. Perkins, and R. Koodli, "Context Transfer Protocol," IETF Internet-Draft, June 2003.

[11] S. Kent and R. Atkinson, "IP Encapsulating Security Payload (ESP)," IETF Request for comments, RFC 2406, November 1998.

[12] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," IETF Request for comments, RFC 2104, February 1997.

[13] P. Cheng and R. Glenn, "Test Cases for HMAC-MD5 and HMAC-SHA-1," IETF Request for comments, RFC 2202, September 1997.

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