A Mobility Scheme for Personal and Terminal Mobility

Bu-Sung Lee, Teck Meng Lim, Chai Kiat Yeo, Quang Vinh Le
Centre for Multimedia and Network Technology, School of Computer Engineering
Nanyang Technological University, Singapore
Email: {ebslee,limtm,asckyeo}@ntu.edu.sg

Abstract—An IP mobility support protocol that enables personal and terminal mobility for IP-based applications is put forward. This protocol does not require new network entities or support from network service providers. It comprises an innovative IP-to-IP address mapping module at the network layer and an user agent to interact with a directory service server and correspondent nodes. It does not require a permanent IP address and a home server. It does not use tunnelling on mobile nodes nor alter route path of IP packets. In this paper, we describe our implementation and present our experimental results. Experiments show that this protocol works for UDP and TCP connections without affecting the throughput of the mobile node on a wireless LAN. Related works are also discussed and quantitatively compared. As an example, this protocol provides seamless execution for applications like VoIP and video conferencing on mobile nodes that roam across wireless networks.

I. INTRODUCTION

The outcome of the advances in communication technologies is leading pervasive communications into an existing new paradigm with the goal of providing seamless communication everywhere at all times. Seamless communication is required for user who wants to remain connected to the Internet while on the move. Wireless technologies are the key network interfaces used in this scenario. However, to enable seamless communication in an environment with heterogeneous wireless network technologies poses a challenge to the research community. Movement of a user results in a change in the point of attachment to the Internet on the user’s wireless device. The device can be changing from an IEEE 802.11 wireless access point (AP) to another IEEE 802.11 AP (intra-handover) or from an IEEE 802.11 AP to an IEEE 802.16 AP (inter-handover). Changing the point of attachment to the Internet can cause a change in IP address for the device, which causes breakage in existing communication sessions. Thus, the goal of terminal mobility is to allow a wireless device to maintain seamless sessions as it moves across IP sub-networks or crosses IP networks which results in a change in IP address.

On the other hand, a user can pick up a wireless device anywhere and be uniquely identified by an identifier when communication sessions are established. This type of mobility is called personal mobility whereby a user is able to relocate and start communication sessions using any device.

Mobile IP (MIP) [1] is the proposed standard to enable terminal mobility. In MIP, each mobile node (MN) is assigned a permanent IP address so that it can be globally addressed in the Internet by any correspondent node (CN). In the basic operation of MIP, a home agent is used to redirect IP packets to MN in an IP-in-IP tunnel. MIP is able to support both TCP and UDP connections in a MN when this MN changes its IP address as it roams between wireless networks.

Protocols like session initialisation protocol (SIP), H.323 and skype allow personal mobility. Typically, SIP allows a single user to be located at different terminals and referenced by the same logical SIP uniform resource identifier (URI). However, once a connection is established, the movement of MN which causes a change in IP address will lead to a loss in the connection. A SIP reinvite request scheme [2] has been proposed to allow UDP-based SIP applications to continue seamlessly on a MN but this scheme does not support TCP connections without using MIP.

We propose a terminal-based IP mobility support protocol, terminal mobility support protocol (TMSP), that allows both personal and terminal mobility. This scheme only requires an upgrade to the operating system of terminal devices (e.g. MN and CN). Like MIP, it allows a MN to continue its TCP and UDP connections when this MN changes its IP address due to roaming. TMSP has the following advantages. As compared to MIP, it does not need a permanent IP for each MN, and does not require any HA. It ensures that packets are routed directly between CN and MN via the conventional IP routing.

In section II, we describe the workflow of TMSP. We describe our implementation of TMSP and report our experimental results in section III. Related works are described and discussed in section IV, followed by a quantitative performance comparison in section V.

II. THE WORKFLOW OF TMSP

TMSP is a terminal-based IP mobility support protocol. It does not need a permanent IP address assigned to each MN. It provides IP mobility for each data connection in a MN using an IP-to-IP address mapping method. This method ensures IP address transparency for the protocol stack above the network layer. It uses a user agent (UA) to manage the change in IP address. It relies on a dynamic directory service mechanism, like dynamic domain name server [3] and SIP redirect server (SIP-RS), to provide an URI to IP address resolution. The role
of UA is to inform correspondent nodes (CN) and update the directory service whenever MN changes its IP address. User can use any device and starts IP roaming using a SIP URI 1. When a device (i.e., MN) connects to the Internet, it will register its current IP address and user’s URI with a SIP-RS as specified in the user’s URI.

Fig. 1 shows a message flow diagram when an application in CN starts a connection with a peer application in MN. A CN can locate the MN using MN’s URI. MN and CN do not need to be registered with the same SIP-RS. The application at CN attempts to get the IP address of MN by calling the gethostbyname() function using MN’s URI. This gethostbyname() function can be modified to filter SIP URI requests and perform a SIP invite request to the MN’s SIP-RS. MN’s SIP-RS resolves the location of MN and returns the IP address of MN using a SIP temporarily moved message. The current IP address of MN is then returned to the application at CN. After that, a connection is made using the connect() function. In the kernel space, our my-connect-hook() function is called before the default kernel-space-connect() function to perform a SIP invite request to MN. The UA at CN sends a new SIP invite request directly to the UA at MN, who in return, replies with a SIP okay. Thereafter, the kernel-space-connect() function is called whereby connection establishment takes place without further alteration.

During the connection establishment, states of the connection are kept in a mapping reference table (MRT) in kernel space. Once the connection is completed, the MRT at each node will contain the Reference Source IP address (RSA) which is the current IP address of the node, Reference Destination IP address (RDA) which is the IP address of its correspondent node, and Current Destination IP address (CDA) for each port. Initially, the CDA is the same as the RDA. The CDA is changed whenever MN reports a movement.

Fig. 2 shows a MN re-inviting CN and re-registering with its SIP-RS after moving to another network where it is given a new IP address - “B”. The re-invitation updates the CDA of MN in CN’s MRT. This enables packets from CN to MN to be routed to this new IP address of MN. Fig. 3 shows the source and destination IP addresses of two packets before and after they are passed through the network layer of the protocol stack at MN and CN. To ensure that the packet will be routed to the CDA of MN and to ensure IP address transparency during changes in the IP addresses, an IP-to-IP address mapping scheme is employed. This scheme works at the network layer of the protocol stack.

As shown in fig. 3, the application at CN is unaware that the MN has changed its IP address to B and it continues to use the MN’s RDA A which is assigned when the connection is first established. When the packet reaches the network layer, the IP-to-IP address mapping function replaces the destination IP address of the packet to the CDA found in the MRT as referenced by the port number. It also updates the packet’s checksum fields for the transport layer and the network layer. This packet then travels to MN. At the network layer of MN,
the IP-to-IP address mapping function replaces the destination IP address of the packet with the RSA in the MRT based on the port number. After that, the packet is sent to the higher layer.

On the other hand, the application at MN is also unaware that MN has changed IP address and continues to use the IP of MN that is assigned when the connection is established. When the packet reaches the network layer, the IP-to-IP address mapping function replaces the source IP address of the packet to the CDA of the MN. It also updates the packet’s checksum fields for the transport layer and the network layer. This is necessary to prevent network ingress filtering [4]. This packet travels to CN. At the network layer of CN, the IP-to-IP address mapping function replaces the source IP address of the packet with the RDA in the MRT based on the port number. After that, the packet is sent to the higher layer.

By using the IP-to-IP address mapping function, the applications does not need to be aware of IP address changes on MN. Thus, IP address transparency is achieved by ensuring that a packet is restored to the original IP addresses used by the application at the time the connection is established.

III. IMPLEMENTATION AND RESULTS

We implemented TMSP on Linux Kernel 2.6 using a kernel module to perform IP-to-IP address mapping and a null character device driver for storing the MRT so that it can be accessed from both the kernel and user spaces. We use two netfilter hooks: (1) NF_HOOK_LOCAL_OUT and (2) NF_HOOK_LOCAL_IN to modify outgoing and incoming packets in the protocol stack, respectively. Briefly, the source port of the outgoing packet is used to obtain the CDA in the MRT, which in turn is used to replace the destination IP address of this packet. Likewise, the destination port of the incoming packet is used to obtain the RSA in the MRT, which in turn is used to replace the source IP address of this packet. It is important to recalculate the checksum fields of each packet in the network layer and transport layer that are being modified in both the netfilter hooks. The routing cache must be cleared when the MN changes its network attachment point, i.e., whenever MN has obtained a new IP address. This is necessary as the kernel caches routes for TCP connections which can forward TCP packets to the previous gateway that may not exists in the newly joined network.

The UA is implemented as a daemon process that listens at a fixed port (15000) for all incoming SIP messages from any CN. Using the SIP URI in the SIP invite request, UA updates the CDA of the MRT entry that matches this SIP URI.

To illustrate TMSP’s IP mobiliy support, we perform a forced handover by changing the “ESSID” of the wireless interface, re-assigning IP address and altering the default route of the iptables on the MN. This force handover filters the delay incurred by network detection and IP address acquisition time during handover. A recommendation for detecting network attachment can be found in [5]. On each forced handover, MN sends SIP invite request to all CDA in its MRT.

In addition, we block the kernel functions: tcp_sendmsg() and udp_sendmsg() when the MN is obtaining a new IP address. This is necessary as the network interface is temporarily down during this process and calls to the above functions will break the existing connections.

We have successfully tested TMSP with video streaming and video conferencing applications. To test TMSP, a testbed that comprises two wireless APs, a SIP-RO, a CN and a MN is set up as shown in fig. 4. This test-bed uses a 1.86 GHz centrino processor as MN and two 2.4 GHz desktop computers running as IEEE 802.11b APs using atheors chipset wireless LAN (WLAN) PCMCIA cards as wireless HUBS.

The handover latency comprises three timings. They are movement detection time, new IP address acquisition time and SIP re-invite request processing time. The first two timings depend on whether the handover is performed using a single wireless interface or multiple wireless interfaces. The time required to detect movement into another network in MN and to obtain an IPv4 configuration can be significant in the total handover latency when a MN moves between points of attachment. There is a standard proposed [5] to introduce a set of steps in order to decrease the handover latency in moving between points of attachment. There is also a standard protocol proposed to improve the delay in new IP addresses acquisition for DHCP using a rapid commit option [6] whereby the usual 4-message exchange is reduced to a 2-message exchange. Apart from these, there are other handover techniques, optimisation and tools proposed in the literature of Mobile IPv6 (MIPv6) [7], like fast handover [8] which introduces a protocol to improve handover latency due to MIPv6 procedures, hierarchical MIPv6 [9] which reduces the signalling delay incurred by informing HA and/or CN of any changes in IP address of MN using the hierarchical nature of IP addressing within the visited network, as well as IPv6 stateless address autoconfiguration for creation of local and global IPv6 addresses in a stateless and automatic manner [10].

We recorded a 125 ms delay when a forced handover is performed by MN, from which 96 ms are used for exchanging
Fig. 5. This figure shows the number of packets received in every 100 ms interval at MN for a 1 Mbps UDP connection from CN. MN switches between WLAN A and WLAN B in every 120 s from the start of the experiment.

Fig. 6. This figure shows the number of packets received in every 100 ms interval at MN for a TCP connection from CN. MN switches between WLAN A and WLAN B on a 6 Mbps channel.

SIP invite requests from MN to CN. In the following sections, we report the operation of the TMSP through experiments.

A. Effect of crossing a network with an active connection

This section details an experiment to illustrate that TMSP is able to resume an active connection when a MN travels from its current network to a neighbouring network. In this experiment, a UDP session using CBR traffic source is established from CN to MN to flood the wireless channel when MN is attached to WLAN A. At every 120 s-interval, MN switches between WLAN B and WLAN A using a script which performs a forced handover. Fig. 5 plots the number of packets received by a UDP connection at MN. This figure shows that MN with TMSP installed only encounters a very short duration (in hundreds of milliseconds) of data loss after it moves to another WLAN. The small breakage time is due to the loss in connection with the previous WLAN. The connection resumes right after MN has successfully configured a new IP address in WLAN B and informed CN of this new IP address. On the other hand, without TMSP, the UDP connection breaks right after MN moves to WLAN B at time 120 s. To illustrate that TMSP works on TCP, an experiment is performed whereby CN maintains a TCP connection with MN who moves from WLAN A to WLAN B and then back to WLAN A. Fig. 6 plots the number of packets received in every 100 ms interval. It shows that the TCP session continues seamlessly during the two handovers.

B. Effect of TMSP on the throughput of WLAN

We conduct an experiment to identify if our TMSP implementation affects the throughput of a WiFi interface. Experiments are run to compare the saturation throughput of a MN using a WiFi interface (IEEE 802.11g). MN floods the air channel using a CBR traffic at 30 Mbps. Table I tabulates the average throughput achieved by MN using various packet sizes. The throughput achieved is approximately the same regardless of whether TMSP is installed or not installed in the MN. Thus, this shows that TMSP does not affect the achievable throughput of the WLAN in this scenario. In addition, we also verified this measurement quantitatively using an analytical model [11] by G. Bianchi as shown in the table.

### IV. RELATED WORK AND DISCUSSION

A. Mobile IPv4

MIP specifies protocol enhancements that allow transparent routing of packet to MNs in the Internet. Each MN is assigned a HA which resides in the home network of MN. In addition, each MN is also assigned a home IP address (HoA), i.e., permanent IP address, regardless of its current network attachment point to the Internet. While in a foreign network, MN is associated with a care-of-address (CoA) issued by the foreign network’s AP. MIP provides for registering this CoA with the HA of MN through a message sequence called binding update. Every packet destined to a MN reaches its HA since the destination IP address of all packets are set to the HoA of this MN. The HA then redirects the packets for MN through an IP-in-IP tunnel to MN’s CoA. After arriving

---

**Table I**

<table>
<thead>
<tr>
<th>Packet Size</th>
<th>Analytical Throughput (Mbps)</th>
<th>TMSP Installed Throughput (Mbps)</th>
<th>No TMSP Throughput (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>600</td>
<td>16.3</td>
<td>14.74</td>
<td>14.87</td>
</tr>
<tr>
<td>700</td>
<td>18.11</td>
<td>16.65</td>
<td>16.68</td>
</tr>
<tr>
<td>800</td>
<td>19.75</td>
<td>19.18</td>
<td>19</td>
</tr>
<tr>
<td>900</td>
<td>21.25</td>
<td>21.02</td>
<td>20.98</td>
</tr>
<tr>
<td>1000</td>
<td>22.62</td>
<td>22.21</td>
<td>22.29</td>
</tr>
<tr>
<td>1100</td>
<td>23.88</td>
<td>23.46</td>
<td>23.32</td>
</tr>
<tr>
<td>1200</td>
<td>25.04</td>
<td>24.84</td>
<td>24.72</td>
</tr>
<tr>
<td>1300</td>
<td>26.12</td>
<td>25.8</td>
<td>25.81</td>
</tr>
<tr>
<td>1400</td>
<td>27.12</td>
<td>26.69</td>
<td>26.79</td>
</tr>
<tr>
<td>1450</td>
<td>27.59</td>
<td>27.1</td>
<td>27.21</td>
</tr>
<tr>
<td>1460</td>
<td>27.69</td>
<td>27.34</td>
<td>27.33</td>
</tr>
<tr>
<td>1470</td>
<td>27.78</td>
<td>27.54</td>
<td>27.37</td>
</tr>
</tbody>
</table>

---

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.
at the end of the tunnel, each packet is then delivered to MN. In the reverse direction, packets sent by MN are delivered to CN directly without passing through MN’s HA. This indirect route of packets is known as triangular routing. MIP provides two ways of acquiring a CoA for MN. A “foreign agent CoA” is a CoA provided by a foreign agent (FA) through its agent advertisement messages. The CoA is the IP address of this FA. In this scenario, FA is the end-point of the tunnel and, upon receiving tunneled packets, decapsulates them and delivers the inner packets to the MN. A "co-located CoA" is a CoA acquired by MN as a local IP address through some external means (e.g. DHCP). When using a co-located CoA, MN serves as end-point of the tunnel and it performs decapsulation of packets tunnelled to it. Thus, using the co-located CoA allows a MN to function without a FA.

Although MIP provides a solution for terminal mobility, it has several limitations which include triangle routing and encapsulation overhead. It also requires a permanent IP address for each MN, HA and FA. Since HA serves as a point of packets redirection for all its MNs, the network links connecting the HA to the network can be easily overloaded as packets destined to the same subnet field converge to the same HA. MIP uses IP-in-IP encapsulation [12] on packets for tunnelling. This increases the size of the packets.

On the other hand, TMSP does not have this overhead as it does not require tunnelling. TMSP also does not require any permanent IP address for each MN. It does not need network entities like HA and FA. The IP addresses when a connection is established are used as the reference addresses for the connection for TMSP’s IP-to-IP address mapping function in the network layer of the protocol stack to provide IP address transparency to higher layers. In TMSP, all packets from any node is routed directly to MN. Thus, inefficiency caused by triangular routing is nonexistent.

B. Mobile IPv6

IPv6 provides enhanced features for MIPv6. In terms of the operation, it is supposed to have enough IP addresses for all MNs. MIPv6 uses a bidirectional tunnel between MN and HA. Thus, packets sent by MN will be tunneled to HA where they are decapsulated and sent to CN. It also adds mobility extension headers to reduce the effect of tunnelling for its route optimised mode whereby packets are routed directly between CN and MN without the need for redirection by HA once a binding update from MN to CN is completed after a connection is established.

TMSP can also be implemented in IPv6. It will operate as described without the need to introduce any mobility extension header, and there is no need to rely on a HA to redirect packets before a binding update is sent from MN to CN as the SIP-RS directly responds with MN’s CoA upon CN’s request.

C. Mobility Support using SIP

Application Layer Mobility using SIP has been discussed by Schulzrinne et al. [2]. Their proposal works only for SIP applications. In the proposal, a SIP invite request is used to re-invite CN when a MN changes its IP address. This methodology only supports UDP connections and the authors propose using MIP for SIP applications that use TCP connections. TMSP is a network layer IP mobility scheme that is targeted to support IP-based applications. Our implementation uses SIP-RS as an IP directory service via an UA, which creates sessions with CNs to maintain the IP address changes. It does not affect the operation of any SIP applications.

V. Numerical Comparison

We evaluate the advantage of TMSP over MIP quantitatively by comparing the gain in network overhead generated by a traffic CBR source. A CBR source generates a total of $T = R \times (S_L) \times t$ bits for a duration of $t$, where $R$ is the rate (packets/second) of the traffic source and $L$ is the size of a packet.

For MIP, the amount of traffic generated onto the network from CN to MN is given by,

$$T^{MIP} = 8RtLH_{CN-HA}^{MIP} + 8Rt(L + L^{IP-in-IP})H_{HA-MN}^{MIP},$$

where $L^{IP-in-IP}$ is the size of the outer IP header used to encapsulate the datagram, $H_{CN-HA}^{MIP}$ is the number of hops from CN to the HA of MN and $H_{HA-MN}^{MIP}$ is the number of hops from the HA of MN to MN. In MIP, the HA of MN is a fixed redirection point for exchange of every packet between CN and MN. For TMSP, the amount of traffic generated onto the network from CN to MN is given by,

$$T^{TMSP} = 8RtLH^{TMSP},$$

where $H^{TMSP}$ is the number of hops between CN to MN.

In this comparison, we assume that IPv4 is used. Thus, the $L^{IP-in-IP} = 20$ bytes. We use a 2 min VoIP session that generates a flow of 89 180-byte packets per second, yielding a rate of 128 kbps. Fig. 7 shows the gain from (1) and (2)) for using TMSP over MIP in terms of the total traffic generated during the VoIP session, where CN to
MN is 5 hops away in the case of TMSP and a variation of 5 to 20 hops away in the case of MIP such that the CN is always 1 hop away from HA of MN. The parameters $H^{TMSP}$ is set to 5, $H^{MIP}_{CN-HA}$ is set to 1, and $H^{MIP}_{HA-MN}$ is a variable from 4 to 19. From the figure, the first plot point shows that there is a gain using TMSP even when the number of hops are the same (i.e., $H^{TMSP} = H^{MIP}_{CN-HA} + H^{MIP}_{HA-MN}$). This is because every packets is encapsulated with an outer IP header at the HA before it is redirected to MN. This gain increases further as the number of hops between MN and its HA increases from 4 to 19 hops. To give a further comparison, we plotted the gain (see fig. 8) in the same scenario as above but with a maximum route of 10 hops distributed in different proportion between $H^{MIP}_{CN-HA}$ and $H^{MIP}_{HA-MN}$. The x-axis denotes the distribution of the number of hops between $H^{MIP}_{CN-HA}$ over $H^{MIP}_{HA-MN}$. Each plot indicates the gain for using TMSP when the direct route between CN and MN has 3, 5, 7 and 10 hops. The figure shows that the gain increases as the number of hops due to routing increases. Gain increases are also registered with increasing number of hops between MN and its HA. In fact, this gain also increases with the duration of the VoIP session and increasing number of traffic sessions. Thus, the use of direct route offered by TMSP eradicates the inefficiency caused by packet redirection and the additional use of outer IP header to encapsulate every packet.

Before packets can be routed directly between CN and MN, the route optimised mode in MIPv6 still relies on HA’s redirection of packets before MN completes a binding update process in which it informs CN of its CoA. So, the initial N packets will suffer from packet redirection. A home address option carried by the destination option extension header is used in a packet sent by MN which is away from HA, to inform the CN of the MN’s HoA. A type 2 routing header is used by CN to allow packet to be routed directly to the CoA of MN. This header also contains the HoA of MN who will in turn replace the destination address of the IP datagram with its HoA. Both headers require an overhead of 48 bits to encapsulate the HoA of the MN which is 128 bits. Thus, without considering the traffic incurred by the initial N packets and assuming that the number of hops between CN and MN are the same, each packet is 44 bytes larger in MIPv6 as compared to TMSP.

VI. Conclusion

TMSP enables IP mobility through the use of an IP-to-IP address mapping function at the network layer of the network protocol stack, and a user agent to manage changes in IP address. It uses a signalling scheme that can rely on any form of directory service that enables dynamic updating. In our implementation, we use the SIP-RS to assist CN in locating MN’s CoA using MN’s SIP URI. In TMSP, packets are routed directly between CN and MN without any need for packet redirection, tunnelling and additional IP header. It does not need each MN to hold a permanent IP address as every connection is established based on the current IP address used and maintained duly as the MN changes its IP address. From our experiments, we show that TMSP works on both TCP and UDP protocols and it does not affect the throughput of the MN in a WLAN environment. We have also discussed and compared TMSP with the related works in terms of operational efficiency as well as quantitatively. PDAs and WiFi/WiMAX/UTMS phones can use TMSP as an enabler for seamless execution of VoIP or video conferencing applications without new infrastructure or hardware support from Internet service providers. TMSP is also extensible to support network mobility [13] for mobile routers that use flat ad-hoc routing [14].

REFERENCES