An IP-Decoupling Approach to Host Mobility*

CHUN-CHIEH WANG, KUANG-HUI CHI+ AND CHIEN-CHAO TSENG++
Information and Communications Research Laboratories
Industrial Technology Research Institute
Hsinchu, 310 Taiwan
+Department of Electrical Engineering
National Yunlin University of Science and Technology
Yunlin, 640 Taiwan
++Department of Computer Science and Information Engineering
National Chiao Tung University
Hsinchu, 300 Taiwan
E-mail: cctseng@csie.nctu.edu.tw.

This paper presents the design, implementation, and performance evaluation of an approach to supporting host mobility in the wireless Internet. We observe that a single IP address conventionally serves dual purposes: a network-level routing directive and an endpoint identity of a certain session in transport or upper layers. Therefore, when a host affiliates with a new IP address due to movements, any pre-established active session thereupon suffers outage. Unlike prior schemes exploiting mobility agents, we propose to let a host use two IP addresses for network- and transport-layer identifications, respectively, and introduce a translation table that associates the two identities. Such an association results from querying or dynamic updates to the Domain Name System, or from direct message exchanges between two communication peers so as to optimize delivery paths. Simulation results show that our scheme outperforms counterpart Mobile IP and Mobile IPv6, in terms of communication delay, packet error rate, message space overhead, and potential packet loss during handoffs. Additionally, qualitative discussions conclude that our proposal lends itself to IPv4 and IPv6 interconnected networks, or multi-tier environment such as wireless Local Area Networks and mobile telecommunication overlay systems.

Keywords: mobile network, TCP/IP, mobile Internet protocol, domain name system, route optimization, ingress filtering, address translation

1. INTRODUCTION

Given the prolific advancement of portable communication devices and current trend towards ubiquitous computing, one can retain network connectivities over a wireless medium in the course of movements. While this facilitates a mobile user's convenient access to the Internet services anytime and anywhere, incorrect message delivery results. The problem stems from that an IP address conventionally serves both as an endpoint identity in transport layer and a network-level routing directive — each session, once established on a host, is decisively bound to the local IP address. Consequently

Received November 8, 2004; revised May 23, 2005; accepted June 27, 2005.
Communicated by Yu-Chee Tseng.

* Preliminary version appeared in International Computer Symposium 2002, Hualien, Taiwan, 2002.

* This work has been supported by the National Science Council of Taiwan, R.O.C., under grants NSC 95-3113-P-009-004, NSC 95-2219-E-009-009, and NSC 95-2221-E-224-016-MY2.

when the host affiliates with a new IP address due to changing attachment points to the system, any ongoing session is disrupted. It is nontrivial to contend with the problem since tight resource constraints on mobile nodes and low bandwidth of unreliable wireless links form aggravating denominators which make traditional TCP/IP communication paradigms untenable in this setting [2, 4, 5].

To streamline communication with mobile nodes, the research community has proposed to augment the Internet Protocol (IP), namely Mobile IP [25] or Mobile IPv6 [14], to hide varying host locations from higher layer services. In this light, most current schemes rely on additional entities (mobility agents) to avoid modifying protocol fabrics in existing Internet nodes to the extent possible [6, 24-27]. These schemes incorporate legacy considerations, yet at the potential expense of limited efficiency and on a premise that mobility support is available only in the presence of a local mobility agent. These prerequisites impede widespread applications.

As a remedy, we develop an efficient approach without exploiting any mobility agent. We designate a mobile host to use two IP addresses for network- and higher-level identifications, respectively. The former signifies the host IP address in present use for correct routing, whereas the latter a unique session on the end host. A translation table is introduced to associate the two addresses. Such an association is derived from querying or performing secure dynamic updates to the Domain Name System (DNS) [18, 19], or is learned from direct message exchanges between two communication parties. In this manner, correspondent nodes aware of the host current location can deliver packets straightway towards the destination.

Qualitative and quantitative comparisons with Mobile IP or Mobile IPv6 reveal that our scheme is promising in several dimensions (see sections 4 and 5). A salient, unique strength is that our decoupled addressing approach allows of independent usage of different-layered protocols, which is unattainable in Mobile IP or Mobile IPv6 networks. Our approach applies to Internet nodes using IPv6 as the underlying stratum while higher-level connection establishment is still IPv4-based. In other words, our approach benefits the legacy protocol stack; existing higher-layer protocols remain operative without knowledge of IPv6 in actual use. Therefore a replacement for IPv4 does not affect overlying protocols. This favors phased-in IPv6 and the evolution of network development.

The remainder of this paper is organized as follows. A terse description of previous research is provided in the next section. Section 3 elaborates on our observations and developments in full. Kernel functions implemented to interface with TCP/IP protocols are illuminated as well. Section 4 discusses our design in applicability aspects such as backward compatibility, security considerations, in IPv4 and IPv6 interconnected networks or in multi-tier environment. Section 5 exhibits performance results relative to Internet Engineering Task Force (IETF) proposals. Lastly section 6 concludes this study.

2. PREVIOUS WORK

Mobile IP is an IETF standard specified to support transparent routing of datagrams to mobile nodes in the Internet. Insights and emerging applications of the protocol can be found in [1, 12, 15, 21]. To summarize, each mobile node has a permanent home address.

On a local network, there is a router termed mobility agent serving for mobile hosts as an attachment point to the system. A mobility agent whose network prefix mis-matches that of a mobile node's home address is said to be foreign to the node, or home otherwise. Mobility agents advertise their presence regularly via control messages, enabling mobile hosts to decide where to attach to.

A mobile host, upon each switch over to a new network, registers a locally acquired care-of address with its home agent. With knowledge of a binding between the host home address and care-of address, the home agent intercepts and tunnels future datagrams for the registered host by means of IP encapsulation [22, 23]. This detour routing implies communication delay and a burden on the networking entities along the path. To tackle such inefficiency, leverages known as route optimization have been devised in the following lines [26]. As with base Mobile IP redirecting traffic to a mobile host, the home agent replies correspondent nodes with an appropriate Binding Update message in addition. Correspondent nodes thus cache the host care-of address so that own datagrams can hereinafter reach that communication peer through direct tunneling. Despite being effective, these extensions require protocol changes on a potentially overwhelming vast set of Internet nodes.

Base Mobile IP is also subject to ingress filtering [10] which mandates routers to forward only outward packets with source addresses belonging to the same local network. In this case, routing is no longer independent of packets' source address. Since a mobile node originates packets invariably using its home address as the source address, a topologically correct reverse tunnel is provided when the host visits a foreign network [20]. The tunnel spanning from the mobile host care-of address up to the home agent conveys datagrams issued by the host. The home agent next decapsulates and forwards the received datagrams onto the local link. Henceforth, normal delivery proceeds as if the packets were sent by the mobile host itself. This scheme, however, accounts for longer routing paths and nontrivial load at home agents.

Mobile IPv6 by Johnson and Perkins [7, 14] operates with a similar flavor but provisions marked improvements over Mobile IP. Among others, supports for route optimization and ingress filtering have been embedded as ingredient rather than optional part of the protocol. For this purpose, new IPv6 extension headers, Routing Header and Destination Options, are defined. The Routing Header is tagged by correspondent nodes to optimize their communication with a mobile host. In practice, given a packet to be sent, an IPv6 node indicates the mobile host care-of address cached *a priori* as the destination address and the host home address in the Routing Header as the final destination. The packet will then reach the mobile node directly and the intended protocol thereof. Moreover, for ingress filtering, a mobile host uses its current care-of address as packet's source, and adds a Destination Options header signifying its home address. Hence such packets will transit foreign network routers correctly.

Two remarks on IPv6 routing are drawn for a mobile node situated away from home. First, it suffices to examine datagrams' basic and extension headers for delivery to or from the mobile node. Further IP encapsulations are not needed as in Mobile IP. Second, packets in this case carry additional Routing or Destination Options headers, which defacto reflects message space overhead and delays routing to some degree.

Other than Mobile IP, a study in [13] employs IP Loose Source Route option instead of encapsulation techniques for datagram delivery. This proposal necessitates the involvement of mobility agents in routing process. Nonetheless, under the circumstance that most Internet hosts are unable to perform correct route reversal, packets may therefore not be expedited from correspondent nodes back to mobile hosts as the option specified. Another issue is concerned with performance, that is, packets carrying this type of option might receive poor service from general routers.

An approach by Snoeren and Balakrishnan adopts dynamic updates to the Domain Name System to track host whereabouts [28]. Existing transport connections are retained by introducing a MIGRATE_WAIT state into the Transmission Control Protocol (TCP) state transition diagram and a novel set of Migrate options in the SYN segment. This scheme is said to be *end-to-end* in the sense that established sessions are enabled to negotiate seamlessly a change in endpoint IP addresses without using any intermediaries like mobility agents. Notably this approach requires no modifications to IP substrate but tailors transport protocols and applications at end hosts.

Gupta and Reddy present a redirection protocol for implementing transparent client access to network services [11]. This proposal is suited to several dimensions such as mobile environment. In this respect, a mobile host instructs its home agent to notify correspondent nodes where to reach the host. Correspondent nodes next substitute future packets' destination addresses properly to the current location of the mobile node. This effectively amounts to redirecting IP packet flow from traffic origin sites without utilizing tunneling. On the other hand, the mobile host sends outgoing datagrams with its care-of address as the source. Datagrams are translated by the recipient node before being handed to higher layers as if they came from the home address. This scheme might lead to repeated interactions between mobile hosts, home agents and correspondent nodes, causing signaling overhead.

3. THE PROPOSED APPROACH

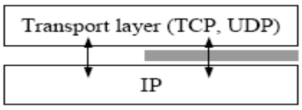
In view of above deficiencies in Mobile IP-based schemes, we propose an end-to-end approach to supporting host mobility instead of using any mobility agents to mediate in communication activities. For better reasoning, this section describes our approach in the context of IPv4 network environment. The design tenet presented here applies equally to an IPv6 setting as well.

3.1 Overview

Conventional TCP/IP communication progresses under a central assumption that a host IP address is used not only for routing purpose, but also for identifying certain session thereon. In other words, a transport- or higher-layer connection is statically coupled with the local IP address in current use. When the host moves to another network, the coupling becomes invalidated because of underlying IP address discrepancy. Consequently, all sessions established in light of the coupling will time out with a termination.

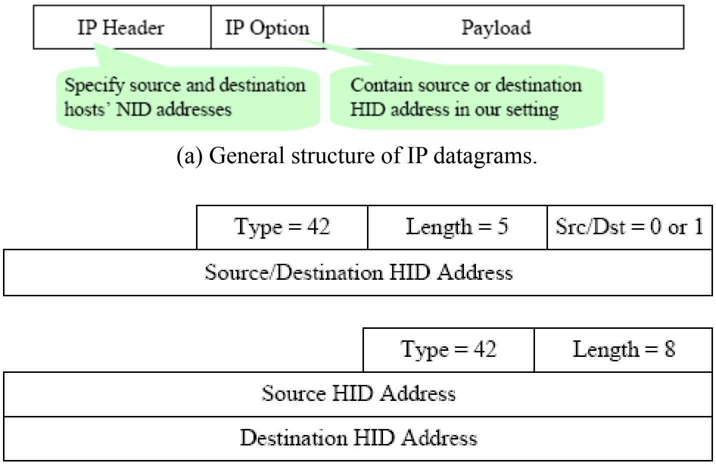
The basic idea behind our proposal, referred to as IP-decoupling approach, is to let a host use two IP addresses for network- and higher-level identifications, respectively. The former, termed network-identifier (NID) address in this text, indicates the current location of the host, whereas the latter, host-identifier (HID) address, determines a session endpoint in transport layer or above. Notice that HID remains unchanged wherever the host migrates, as opposed to NID which alters upon host movements. Additionally we use a table to associate the two addresses for translating packet flow between network and upper layers. The addressing associations derive from responses to DNS queries, secure updates to the name system, or are deduced from message exchanges between two communication parties. Either communicating side is thereby enabled to correlate an appropriate NID address with the session's HID address in question. To optimize route, the NID address is utilized for direct packet delivery towards the destination node without tunneling.

Our approach is embodied in a software module interfacing with transport and network layers. The diagram below shows the relative position of our development (grayed part) in terms of the overall TCP/IP protocol suite.



3.2 HID Option

We use a fixed HID address to locate one endpoint of a session that connects two Internet hosts. Since the HID in use may differ from the underlying NID address over host migrations, we propose a new IP option for outgoing packets to convey additional information so that each recipient node can resolve which endpoint the packet actually comes from or addresses to. In practice, the IP Option part of Fig. 1 (a) accommodates the information, whose format is outlined in Fig. 1 (b). Note that integer 42 characterizes this type of option. When the Length field records 5, the option carries the source endpoint's HID address if the Src/Dst field is 0, or the destination endpoint's HID address


IP Header
IP Option
Payload
Specify source and destination hosts' NID addresses
Contain source or destination HID address in our setting
Type = 42
Length = 5
Src/Dst = 0 or 1
Source/Destination HID Address
Type = 42
Length = 8
Source HID Address
Destination HID Address

(a) General structure of IP datagrams.

(b) Formats of the proposed HID option.

Fig. 1. Generic structure of an IP datagram with HID option.

otherwise. When the Length field is 8, both the source and destination Endpoints' HID addresses are present.

3.3 NID and HID Addresses Assignments

We stipulate that a mobile node initiates new sessions assigning its current NID address as the HID address, which will last the session's lifetime. Such assignments are exemplified in Fig. 2, where the mobile node is presumed to use NID address ip_m originally. The mobile host activates two sessions toward its communication peer, at events 1 and 4b, respectively. The second session is set up (event 4b) while the previous one is in progress (event 4a.) Remarkably a host may allocate distinct HIDs for connection establishments along with migrations.

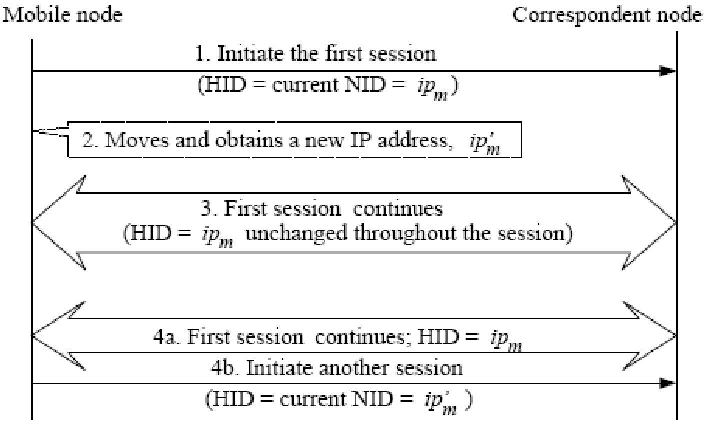
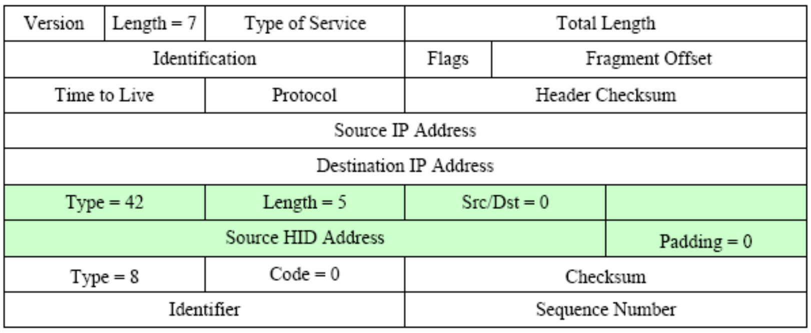

Mobile node
Correspondent node
1. Initiate the first session
(HID = current NID = ip_m)
2. Moves and obtains a new IP address, ip'_m
3. First session continues
(HID = ip_m unchanged throughout the session)
4a. First session continues; HID = ip_m
4b. Initiate another session
(HID = current NID = ip'_m)

Fig. 2. Example of HID address assignments.

We assume that a mobile node is able to secure an IP address, from some local Dynamic Host Configuration Protocol (DHCP) server [8], for instance, whenever switching to a new network. In that case, the mobile host also notifies correspondent nodes of its new NID address. Aspired from Binding Update messages in Mobile IPv6, we approach these notifications using an ICMP Echo Request plus a designated HID option (Fig. 3). In this way, each packet is sent using the source and destination hosts' current NID addresses. Datagrams traverse an optimized route yet free from being ingress filtered.

3.4 Locating a Mobile Node

As in [28], this study takes advantage of the widely-deployed Domain Name System (DNS) to locate a mobile node. Because most Internet applications resolve host names to an IP address at the beginning of a transaction or a connection, this mechanism is viable in our architecture. As illustrated in Fig. 4, when a mobile host acquires a new IP address upon movements, it invokes a secure update procedure [9] to the root name server (step 1). Therefore, in response to any DNS query by a prospective correspondent node,



Version	Length = 7	Type of Service	Total Length
Identification		Flags	Fragment Offset
Time to Live	Protocol	Header Checksum	
Source IP Address			
Destination IP Address			
Type = 42	Length = 5	Src/Dst = 0	
Source HID Address			Padding = 0
Type = 8	Code = 0	Checksum	
Identifier		Sequence Number	

Fig. 3. ICMP Echo Request with HID option.

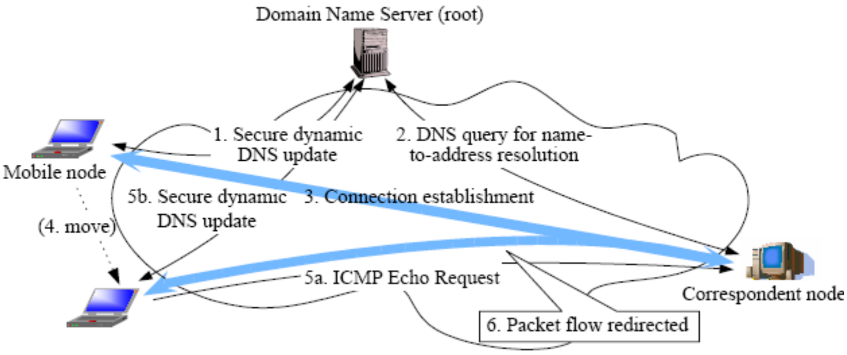

Domain Name Server (root)
Mobile node
1. Secure dynamic DNS update
2. DNS query for name-to-address resolution
5b. Secure dynamic DNS update
3. Connection establishment
(4. move)
5a. ICMP Echo Request
Correspondent node
6. Packet flow redirected

Fig. 4. Using DNS queries to locate an internet host. Upon changing the IP address, a mobile host performs a secure DNS update and informs correspondent nodes of its new location.

the server returns the latest NID address of the mobile host (step 2). We set the Time-To-Live field of the DNS A-records for all mobile nodes to *zero*, to prevent stale name-to-address mappings from being cached elsewhere. This precludes possible connection failures due to resolving mobile nodes to obsolete addresses.

After the host current IP address is available, normal connection to the intended host follows (step 3). Suppose that the host moves again some time later and obtains a new IP address (step 4), the mobile host issues an ICMP Echo Request plus the afore-mentioned HID option towards its correspondent nodes (step 5a). Such an ICMP message is to notify correspondent nodes about the mobile host new NID address so as to attract subsequent packets to the new location. Also, another DNS update is sent to the root DNS server to renew the host name-to-address mapping (step 5b).

3.5 Address Translation

The sequel describes strategies for an Internet host to structure a translation table that maps an HID to a proper NID. Packet translations take place transparently prior to sending or receiving datagrams by upper level protocols through IP machinery. This makes the IP addresses known to overlying protocols operate as invariant HIDs for packets address assignments.

3.5.1 Host-oriented addressing

Initially void, a host-oriented translation table consists of 3-tupled entries, each of the form

<*HID*, *NID*, *Timeout*>.

The first two fields record the HID and NID in present use by a communication party, along with the remaining time to prune off the whole entry. The Timeout field is refreshed whenever the entry is accessed. We next illuminate how to maintain the table by means of Fig. 5.

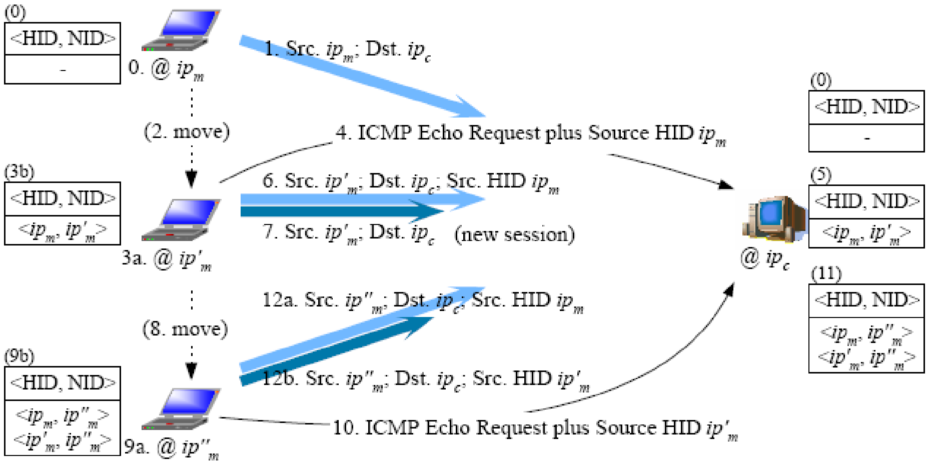

(0)
<HID, NID>
-
0. @ ip_m
1. Src. ip_m; Dst. ip_c
(0)
<HID, NID>
-
(2. move)
4. ICMP Echo Request plus Source HID ip_m
(3b)
<HID, NID>
<ip_m, ip'_m>
6. Src. ip'_m; Dst. ip_c; Src. HID ip_m
7. Src. ip'_m; Dst. ip_c (new session)
3a. @ ip'_m
(5)
<HID, NID>
<ip_m, ip'_m>
@ ip_c
(11)
<HID, NID>
<ip_m, ip''_m>
<ip'_m, ip''_m>
12a. Src. ip''_m; Dst. ip_c; Src. HID ip_m
(8. move)
(9b)
<HID, NID>
<ip_m, ip''_m>
<ip'_m, ip''_m>
12b. Src. ip''_m; Dst. ip_c; Src. HID ip'_m
10. ICMP Echo Request plus Source HID ip'_m
9a. @ ip''_m

Fig. 5. Illustrating host-oriented translations.

As far as a host is concerned, sessions originating from or terminated at the local IP address are carried out as normal communication activities; no entry is stored for the host in its local translation table. When the host moves later, *e.g.*, from addresses ip_m to ip'_m, the table expands as follows:

- All the NID fields matching ip_m are changed to ip'_m.
- Another new entry is generated with HID and NID fields set to ip_m and ip'_m, respectively, unless an identical entry has already existed.

Subsequent to above operations exemplified in steps 3b and 9b of Fig. 5, the mobile host issues an immediate ICMP Echo Request plus an HID option (Fig. 3) that instructs correspondent nodes to update their own translation tables accordingly.

For each outbound packet, the sending host examines the Packet's source and destination addresses, say ip_m and ip_c, respectively, as indices to look up HIDs in its translation table. The corresponding NID of the ip_m-matched entry is utilized in place of the packet's source address. Such a replacement imposes an additional IP option indicating the Source HID address ip_m onto the packet. (See steps 6, 12a and 12b of Fig. 5 for an illustration.) On the other hand, the mapped NID of the ip_c-valued entry, if any, substitutes the packet's destination address instead, with an extra Destination HID address ip_c specified. Then routing process takes over. The recipient node will tell, from the attached HID option, correct traffic source or intended endpoints, and restore the packet as though it had arrived intact without address translations.

Let *pk.src* and *pk.dst* denote the source and destination addresses, respectively, of a packet *pk*. Further, let *r* be an entry of type *row* that comprises the translation table $\mathcal{R}$. We use *r.HID* and *r.NID* to represent the HID and NID fields of *r*, respectively. Our proposal can then be summarized in Fig. 6. Observe that the Source HID Address option may be tagged with an ICMP Echo Request or piggybacked by users' data messages.

```
var new: row;
   $\mathcal{R}$: set of row init $\varnothing$;    (* translation table *)
Upon changing network addresses from $ip_m$ to $ip'_m$:
      forall $r \in \mathcal{R}$, $r.NID = ip_m$ do $r.NID := ip'_m$;
      $new.HID, new.NID := ip_m, ip'_m$;
      $\mathcal{R} := \mathcal{R} \cup \{new\}$;
      Send an ICMP Echo Request plus Source HID Address $ip_m$ to all correspondent nodes;
For an outgoing packet pk:
      if $r \in \mathcal{R}$: $r.HID = pk.src \neq r.NID$ then
         begin tag pk with Source HID Address pk.src;
                 $pk.src := r.NID$
         end
      if $r \in \mathcal{R}$: $r.HID = pk.dst \neq r.NID$ then
         begin tag pk with Destination HID Address pk.dst;
                 $pk.dst := r.NID$
         end
For an incoming packet pk:
      if Source HID Address, say $ip_m$, is present then
         begin if $\nexists r \in \mathcal{R}$: $r.HID = ip_m$ then (* pk is ICMP message *)
                    begin forall $r \in \mathcal{R}$, $r.NID = ip_m$ do $r.NID := pk.src$;
                            $new.HID, new.NID := ip_m, pk.src$;
                            $\mathcal{R} := \mathcal{R} \cup \{new\}$
                    end
                    $pk.src := ip_m$
         end
      Replace pk.dst with Destination HID Address, if any
```

Fig. 6. Host-oriented addressing algorithm.

3.5.2 Session-oriented addressing

Each entry of a session-oriented translation table takes the form

$<HID_s, NID_s, pn_s, HID_d, NID_d, pn_d, Timeout>$,

comprising both the HID and NID addresses together with a port number (*pn*) used by two communication parties; subscripts *s* and *d* denote source and destination sites, respectively, of the given connection. This translation mechanism is to facilitate TCP or User Datagram Protocol (UDP) message exchanges which exploit port information. For reasoning, let us next consider Fig. 7, where a mobile host with original address ip_m activates two telnet connections to some node at address ip_c.

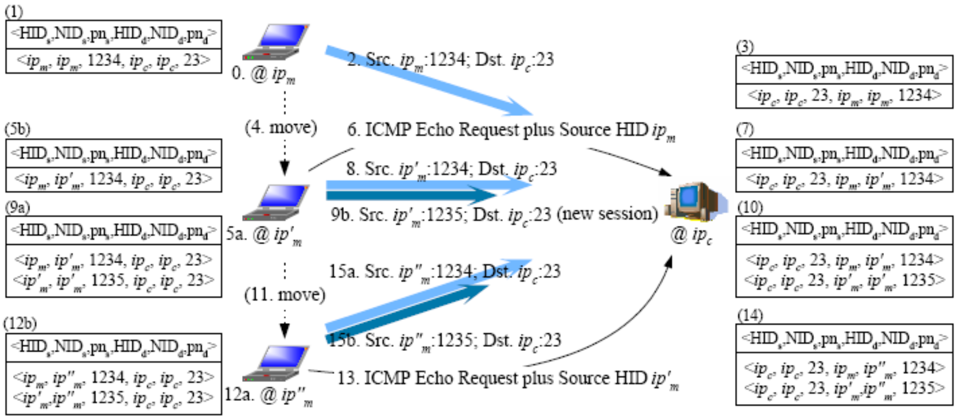

(1)
<HID_s,NID_s,pn_s,HID_d,NID_d,pn_d>
<ip_m, ip_m, 1234, ip_c, ip_c, 23>
0. @ ip_m
2. Src. ip_m:1234; Dst. ip_c:23
(3)
<HID_s,NID_s,pn_s,HID_d,NID_d,pn_d>
<ip_c, ip_c, 23, ip_m, ip_m, 1234>
(4. move)
(5b)
<HID_s,NID_s,pn_s,HID_d,NID_d,pn_d>
<ip_m, ip'_m, 1234, ip_c, ip_c, 23>
6. ICMP Echo Request plus Source HID ip_m
(7)
<HID_s,NID_s,pn_s,HID_d,NID_d,pn_d>
<ip_c, ip_c, 23, ip_m, ip'_m, 1234>
8. Src. ip'_m:1234; Dst. ip_c:23
(9a)
<HID_s,NID_s,pn_s,HID_d,NID_d,pn_d>
<ip_m, ip'_m, 1234, ip_c, ip_c, 23>
<ip'_m, ip'_m, 1235, ip_c, ip_c, 23>
5a. @ ip'_m
9b. Src. ip'_m:1235; Dst. ip_c:23 (new session)
@ ip_c
(10)
<HID_s,NID_s,pn_s,HID_d,NID_d,pn_d>
<ip_c, ip_c, 23, ip_m, ip'_m, 1234>
<ip_c, ip_c, 23, ip'_m, ip'_m, 1235>
(11. move)
15a. Src. ip"_m:1234; Dst. ip_c:23
(12b)
<HID_s,NID_s,pn_s,HID_d,NID_d,pn_d>
<ip_m, ip"_m, 1234, ip_c, ip_c, 23>
<ip'_m,ip"_m, 1235, ip_c, ip_c, 23>
15b. Src. ip"_m:1235; Dst. ip_c:23
(14)
<HID_s,NID_s,pn_s,HID_d,NID_d,pn_d>
<ip_c, ip_c, 23, ip_m, ip"_m, 1234>
<ip_c, ip_c, 23, ip'_m,ip"_m, 1235>
12a. @ ip"_m
13. ICMP Echo Request plus Source HID ip'_m

Fig. 7. A scenario of session-oriented translations for telnet.

On detecting a new session, we add an associated entry in the translation tables of both sides. This is fulfilled by the initiator host first depositing <ip_m, ip_m, 1234, ip_c, ip_c, 23> in its maintained table (step 1). The source (destination) HID and NID are set to a common ip_m until the host (destination node) changes its IP address. The host then directs data toward its correspondent node after having inspected each packet for proper NID modifications (step 2). In this case, however, packet flow proceeds without address alterations. Analogous to foregoing step 1, the recipient node will record the new session but reverses the order of endpoints (step 3).

When acquiring a new address ip'_m, the mobile host conducts actions:

- All the NID fields of the local translation table recording ip_m are reset to ip'_m (step 5b or 12b for instance).
- Send an ICMP Echo Request with Source HID ip_m to correspondent nodes, so as to update their translation tables accordingly (step 6 or 13).

For each outbound packet, the sending host finds an entry in its translation table whose HID_s, pn_s, HID_d, and pn_d jointly define the packet's endpoints. The NID_s and NID_d of the matched entry are then employed as the packet's source and destination addresses, respectively. On reception through network delivery, the correspondent node compares the packet's endpoints against quadruple NID_d, pn_d, NID_s, and pn_s fields of the local translation table. The HID_d and HID_s of the matched entry are used to restore the packet's source and destination addresses, respectively. Given the notations of section 3.5.1, we formulate our leverage in Fig. 8.

```
Upon changing network addresses from $ip_m$ to $ip'_m$:
    forall $r \in \mathcal{R}$, $r.NID_s = ip_m$ do $r.NID_s := ip'_m$;
    Send an ICMP Echo Request plus Source HID Address $ip_m$ to all correspondent nodes;
For an outgoing packet $pk$:
    if $r \in \mathcal{R} : (r.HID_s = pk.src) \wedge (r.pn_s = pk.pn_s) \wedge (r.HID_d = pk.dst) \wedge (r.pn_d = pk.pn_d)$ then
        $pk.src, pk.dst := r.NID_s, r.NID_d$;
    else begin $new.HID_s, new.NID_s := pk.src$;
                $new.HID_d, new.NID_d := pk.dst$;
                $new.pn_s, new.pn_d := pk.pn_s, pk.pn_d$;
                $\mathcal{R} := \mathcal{R} \cup \{new\}$
    end
For an incoming packet $pk$:
    if Source HID Address, say $ip_m$, is present then
        forall $r \in \mathcal{R}$, $r.NID_d = ip_m$ do $r.NID_d := pk.src$
    else if $r \in \mathcal{R} : (r.NID_d = pk.src) \wedge (r.pn_d = pk.pn_s) \wedge (r.NID_s = pk.dst) \wedge (r.pn_s = pk.pn_d)$ then
            $pk.src, pk.dst := r.HID_d, r.HID_s$;
    else begin $new.HID_s, new.NID_s := pk.dst$;
                $new.HID_d, new.NID_d := pk.src$;
                $new.pn_s, new.pn_d := pk.pn_d, pk.pn_s$;
                $\mathcal{R} := \mathcal{R} \cup \{new\}$
    end
```

Fig. 8. Session-oriented addressing algorithm.

Recall that packet's source and destination addresses in conjunction with port numbers suffice to identify a TCP or UDP connection. Since full endpoints information is employed to look up the mapped identities in the translation table, HID options hereafter need not be attached with users' data messages.

3.5.3 Remarks

The prescribed treatment saves HID options, thereby message space, from being conveyed in data packets. Hence it may be tempting to adopt session-oriented translations for TCP or UDP messages and host-oriented translations for other types of traffic. Such synergies support internetworking host mobility as well.

Unlike the session-oriented operation tenet, the host-oriented scheme deals with inbound packets by restoring addresses in light of each carried HID option, rather than by referring to the translation table. Since multiple HIDs can be mapped to a common NID but not vice versa, mapping from NID to HID possibly leads to resolution conflicts. This behooves us to perform host-oriented address translations for incoming packets by exploiting other technique than table lookup.

It is probable that a mobile host re-visits a network using the same, previously acquired IP address. If this is the case, there will exist some entry in the host translation table with identical HID and NID values, *i.e.*, address translations consulting the entry become redundant. For efficiency, we can refine our host-oriented scheme to bypass such ineffective translations. Alternatively, a coincident entry can be purged from the table as if the host never migrated out of the local network.

3.6 Fraud Control

Each predefined ICMP message causing updates to translation tables requires authentications, or one may masquerade as a communication party by signaling another peer to take over data sessions.[1] For security, each mobile node generates its own public and secret encryption keys. The public key, to be shared between two communication parties, is sent along with a connection start-up, as depicted by step 1 of Fig. 9. In case the correspondent peer receives later an ICMP Echo Request for traffic redirection, a challenge-response procedure follows (steps 5 and 6.) Then redirecting packet flow is allowed only if twofold verifications at step 7 pass successfully. Otherwise, data stream proceeds from the correspondent node to the last authenticated location of the mobile host.

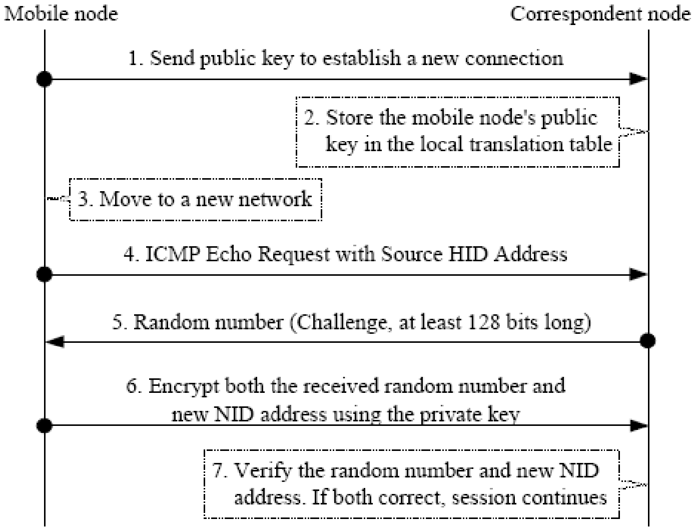

Mobile node
Correspondent node
1. Send public key to establish a new connection
2. Store the mobile node's public key in the local translation table
3. Move to a new network
4. ICMP Echo Request with Source HID Address
5. Random number (Challenge, at least 128 bits long)
6. Encrypt both the received random number and new NID address using the private key
7. Verify the random number and new NID address. If both correct, session continues

Fig. 9. Message flow of the fraud control mechanism.

3.7 Implementation

Our proposal has been implemented in FreeBSD 4.4.1-Release. We incorporated host- and session-oriented translation mechanisms into the TCP/IP protocol suite. For non-TCP/UDP packets, we introduce new functions ip_input_new() and ip_output_new() into the operating system kernel. Function ip_input_new(), a variant to ip_input(), invokes the modified ip_dooptions() which is responsible for processing HID options. Function ip_output_new(), a variant to ip_output(), replaces an HID address with the corresponding NID address discovered on the host-oriented translation table and inserts an HID option in the packet header. For incoming TCP or UDP messages, tcp_input() and udp_input() replace both source and destination addresses with the HID addresses recorded on the session-oriented translation table. Regarding outgoing TCP or UDP messages, the HID

[1] This is referred to as *connection hijacking* in some literature.

addresses are changed to the NID addresses before calling function ip_output(). Note that the host- and session-oriented translation tables should be accessible to IP layer modules, since any receipt of ICMP Echo Requests with HID options will cause updates to the two tables.

4. DISCUSSIONS

The presented approach exhibits utilities in several aspects. First and foremost, datagram delivery paths are optimized in that mobility agents do not engage in routing procedure any longer, nor is additional tunneling required. Communication delay and network routers workload can be thereby reduced substantially.

4.1 Heterogeneous System

Our scheme is apt for deployment in a multi-tier system like wireless Local Area Networks and mobile telecommunication overlay architecture, where a mobile node with more than one radio access capabilities can roam around. An expository description of such architectures is given in [3]. As shown in Fig. 10, observe that variations in NID addresses due to switching between networking devices do not affect the HID address in the upper layer. Therefore, data sessions originated from or terminated at the HID address will remain without disruption.

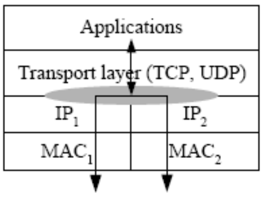


Fig. 10. Dual-mode entity using our proposal (grayed part).

Our scheme can also be easily tailored to fit into an IPv6 setting. Modifications involve a shift to the IPv6 address size of 128 bits and using ICMPv6 Echo Request to trigger updates to translation tables. Methodology of translating packet headers and translation table maintenance in IPv4 contexts still applies.

Another pragmatic consideration is to provide roaming support for IPv4 and IPv6 interconnected networks. So far, there are no standard approaches to integrating Mobile IP and Mobile IPv6 [29]. With our scheme, nevertheless, a dual-stack mobile node aware of both IPv4 and IPv6 can migrate freely within such an interconnected environment. This is because our proposal operates where either protocol is running; an HID address, once determined, will behave as a static connection endpoint, irrespective of changing whatever attachment points to IPv4 or IPv6 networks. Therefore our decoupled addressing scheme enables mobility support across different network-layer protocol to be realized in a unified way.

4.2 Inter-Operability Concerns

Consider two nodes, only one of which implements the proposed mechanism. It can be seen that an in-between connection establishment is entirely feasible, because the two parties assign their current IP addresses to endpoints, and transport protocols thereon use the two addresses passed from the network layer to calculate the checksum. The node ignorant of our scheme validates the received datagrams' checksum correctly since the connection originator, though mobile, appears momentarily stationary like an ordinary Internet host. On the other hand, the mobile connection originator utilizes a fixed HID throughout (recall section 3.3) to verify the checksum accordingly. This session will remain effective until the mobile node moves elsewhere.

In certain circumstances, certain network applications that perform packets' address conversions midway internally need minor adaptations to exploit our scheme to the full. For instance, a Network Address Translation (NAT) server should update private and public IP address mappings upon intercepting an ICMP Echo Request with HID options originated from its managed network.

4.3 Analogy to Mobile IP-Based Schemes

For a better exposition of the proposed approach, several perspectives are highlighted. Note first that whatever schemes support host mobility requires some form of location updates. Location update messages are essentially an imperative in a mobile environment. Although our architecture does not subsume any home agents, our approach employs the DNS server for location tracking as opposed to registrations with home agents in Mobile IP. We neither eliminate the cost of location updates, nor impose unacceptably extra signaling overhead in this regard (as compared with Mobile IP.)

For route optimization, we introduce an address translation table to offer direct message exchanges between communication parties. Our translation table appears to function like the binding cache on a correspondent node in Mobile IP (or Mobile IPv6). However, an important distinction lies in that the translation table is maintained for processing inbound and outbound packets, whereas the binding cache for outbound datagrams only. In addition, our scheme does not deliver packets by means of tunneling. This mitigates the load on networks and routers along delivery paths through the Internet.

4.4 Remarks

In our architecture, we rely on the DNS server for address resolution. If a mobile host does not ever change its IP address (*i.e.*, the host behaves as a fixed node), the host remains reachable throughout. However, when the host moves and switches its address, correspondent nodes can no longer locate the mobile host without using its domain name. Despite this potential weakness, we advocate leveraging the use of DNS for location tracking for the following reasons:

- The Internet employs domain names and DNS servers to keep the availability of hosts because associated IP addresses might change from time to time. It is also very common that Internet service subscribers do not own fixed IP addresses from their ISPs

(Internet Service Provides) but require dynamic allocation upon each login. In other words, DNS lookup for hosts are common occurrences in practice. Therefore, in line with such conventions, we choose to update the name-to-address mappings for mobile hosts to the widely-deployed DNS server.

- IPv6 is gaining prevalence. In the forthcoming IPv6, each host shall be associated with a 128-bit address. We argue that a long address is hardly mnemonic and people tend to first contact DNS for host address resolution. This trend implies that communication without using domain name is becoming less likely in the near future.

5. PERFORMANCE EVALUATION

This section demonstrates quantitative comparisons of the proposed approach with Mobile IP and Mobile IPv6 under a parameter setting listed below.

Bandwidth over a local wireless link	11 Mbps
Bandwidth over the Internet	100 Mbps
Bit error rate in a wireless cell	10^{-5}
Bit error rate in wired internetworks	10^{-11}
Routing latency	0.5 ms
HID option size in IPv4	7 or 10 bytes
Size of extra encapsulated IPv4 header	20 bytes
HID option size in IPv6	20 bytes
Size of Mobile IPv6 Destination Option	20 bytes

Communication environment under discussion is simulated using an 8 × 8 wraparound mesh,[2] where each cell represents the coverage area of a wireless subnetwork. A mobile host, home agent, and correspondent node are positioned arbitrarily on the grid at the outset. Then, after an exponentially distributed period of sojourn time, the host moves randomly to one of four neighbor cells. Packets sent by the host within a cell is Poisson distributed with parameter λ (10 packets/second) in number, while geometric distributed with parameter 0.5 in size.

Since our approach can be applied to IPv4 or IPv6 network environment, figures in the sequel compare our approach with counterpart Mobile IP and Mobile IPv6, respectively. Mobile IP under discussion does not provide route optimization (referred to as base Mobile IP). Performance metrics of interest include communication delay, packet error rate, overhead of packet size and operation time, and packet loss during handoffs. The horizontal axis of all figures below marks what percentage of overall data traffic TCP or UDP messages account for.

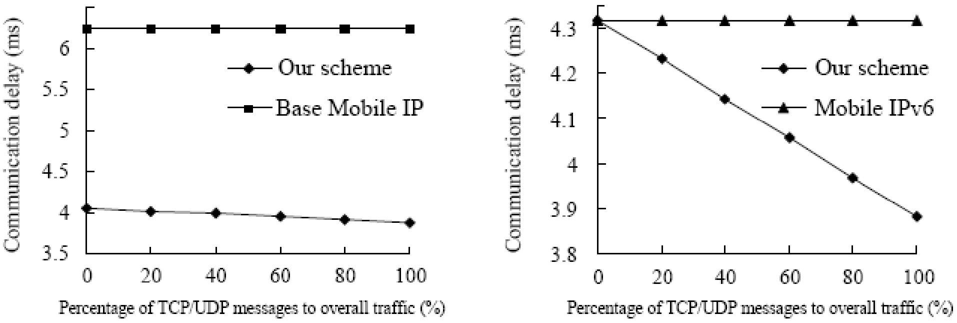
5.1 Communication Delay

Communication delay refers to the average elapsed time of delivering a packet from a mobile host to its correspondent node. This measure can be approximated as

[2] Similar models can be found in [16, 17].

$$\text{Communication delay} = \frac{\text{packet size}}{\text{network bandwidth}} + (\text{routing latency}) \times (\text{hop count}). \quad (1)$$

Here routing latency represents the processing time of an intermediate router; hop count indicates how many routers a packet traverses in delivery path.



(a) In IPv4 contexts.

(b) In IPv6 contexts.

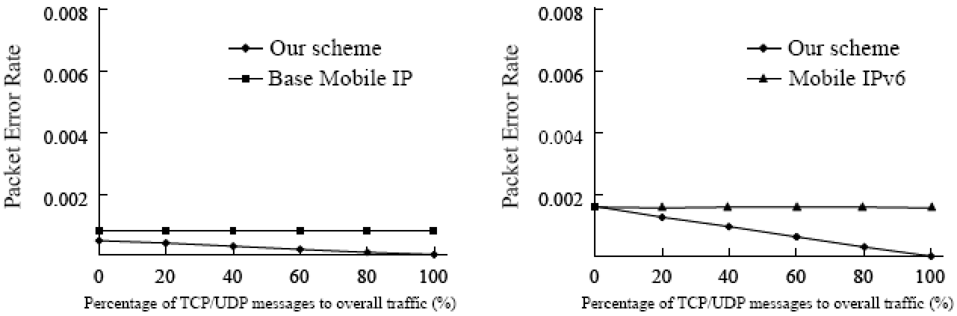
Fig. 11. Communication delay versus the amount of TCP/UDP traffic.

Fig. 11 (a) shows that in terms of communication delay, our scheme outperforms Mobile IP by around 37%. This performance gain is ascribed to two main factors. First, Mobile IP tunneling encapsulates datagrams, causing more message space overhead than ours. Message space overhead means parts of a normal packet excluding the actual header and data payload. Therefore, a tunneled packet results in a message space overhead of one additional packet header, namely 20 bytes, whereas our scheme introduces HID options *potentially* at a space overhead of 7 or 10 bytes. Another reason for our outperformance is that communication path in Mobile IP may form non-optimal triangle route. Transmission time thus increases as passing through transit routers.

From Fig. 11 (b), it can be seen that our approach outperforms Mobile IPv6 by up to 10 . Mobile IPv6 takes optimal routes, as does our scheme, implying that the term (routing latency) × (hop count) of Eq. (1) should be identical in both schemes. However, message space overheads differ. Recall that our scheme does not attach any HID options to TCP or UDP messages (section 3.5.2.) This is not the case for Mobile IPv6 carrying Destination Option headers. Hence, our message space overhead reduces as there is more and more TCP or UDP traffic in the network. This contributes to less add-on to packets. Shorter packets facilitate network transfer, reducing overall communication delay.

5.2 Packet Error Rate

The second performance index concerns at what rate datagrams cannot arrive intact at intended destinations. We define this metrics, packet error rate, as a ratio of error-free packets received by the destination node to total packets sent by the mobile host during a time interval. Given prescribed respective bit error rates for wireless cells (10^{-5}) and wired internetworks (10^{-11}), Fig. 12 (a) compares our packet error rate against that of base Mobile IP scheme. Our scheme can achieve better data transfer thanks to shorter



(a) In IPv4 contexts. (b) In IPv6 contexts.

Fig. 12. Packet error rate versus the amount of TCP/UDP traffic.

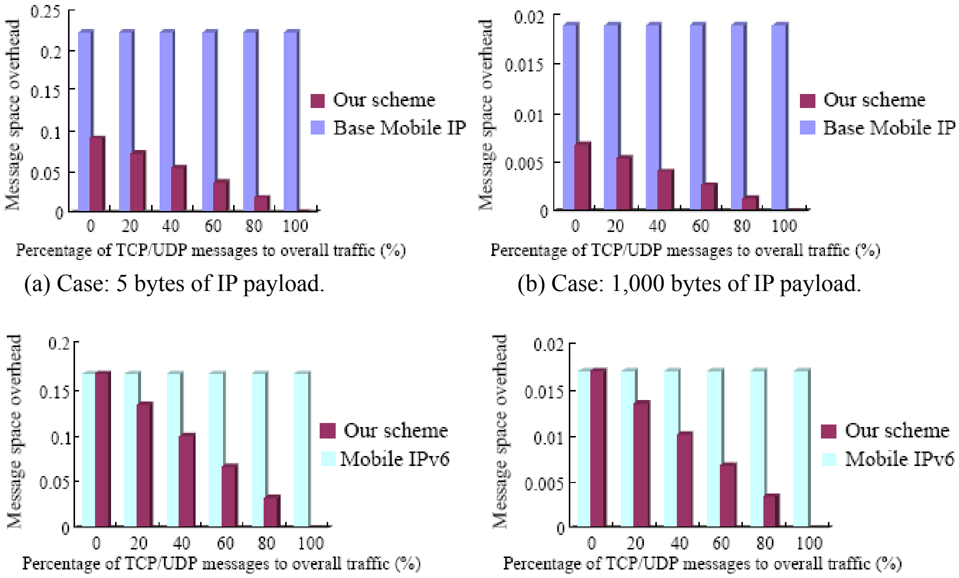
packet size, since our scheme attaches HID options of 7 or 10 bytes, in contrast with Mobile IP imposing additional encapsulation headers of 20 bytes. Another cause is routing detour in Mobile IP, because lengthy conveyance of a larger datagram is more likely to undergo corruption in transit.

Fig. 12 (b) indicates also that our approach outperforms Mobile IPv6 in terms of packet error rate. Further, such performance gain becomes more pronounced along with the increase in TCP or UDP messages. As stated earlier, plain TCP or UDP messages without carrying extension headers or HID options suffice in our architecture. On the contrary, Mobile IPv6 uses Routing Headers or Destination Options for correct message delivery. Hence, when the network traffic is mostly TCP- or UDP-oriented, packets in our architecture are comparatively compact in size and susceptible to efficient network transmission. Since TCP and UDP dominate today's Internet, our proposal is arguably well suited to wireless internetworking environment in respect of communication delay and packet error rate.

5.3 Overhead Considerations

This subsection considers the overhead of applying our approach as opposed to counterpart schemes. First, our proposal demands message space to accommodate HID options that are tagged with other types of packets than TCP or UDP. In contrast, Mobile IP encapsulates packets for tunneling, whereas Mobile IPv6 includes extension headers in each datagram, both representing another form of message space overhead. To formulate relative impacts, we define a normalized expression of this overhead to be m/(base IP header length + m + IP payload length), where m denotes the number of octets in a packet for mobility support purposes. For instance, m in our architecture is 7 or 10, but 20 in Mobile IP.

Fig. 13 shows that the proposed approach incurs least message space overhead. Figs. 13 (a) and (b) outline the results collected from an IPv4 networking scenario. In Figs. 13 (c) and (d), we assume 19 bytes of HID options carried by an IPv6 packet — 1 byte for Next Header, 1 for Length, 1 for Src/Dst, and 16 for the IPv6 HID Address fields, respectively (see Fig. 1).



(a) Case: 5 bytes of IP payload.

(b) Case: 1,000 bytes of IP payload.

(c) Case 5: bytes of IPv6 payload.

(d) Case: 1,000 bytes of IPv6 payload.

Fig. 13. Message space overhead versus the amount of TCP/UDP traffic.

As far as our proposal is concerned, another overhead stems from extra operations in the TCP/IP protocol stack that map HID to NID addresses or vice versa in translation tables. It can be seen that such operations essentially equate table lookups and pose insignificant time complexity if the table is manipulated in a binary tree fashion. In that case, letting n denote the total number of entries present, both search and insert operations cost $O(\log n)$ time. Furthermore, in practice, most operating systems allow for some certain maximum number of concurrent connections in action. Thus the overhead of additional interactions with TCP/IP protocols will not unduly impair our approach.

5.4 Packet Loss during Handoffs

Each handoff of a mobile host introduces a blackout period during which the system learns to re-locate the host. Datagrams meant for such a migrant node cannot reach the destination, causing traffic stream disruptions. This adverse effect, however, does not lead to connection breakages but degraded network throughput, since lost packets will be retransmitted by higher-level, reliable transport protocols.

Packet loss experienced by a mobile host can be formulated as a product of how long the blackout period lasts and the rate at which datagrams address to that host. The blackout interval in our architecture begins when the host departs from currently attached network and terminates when correspondent nodes receive ICMP Echo Requests from the host indicating its switch over to a new network. This amounts nearly to a round trip time, τ_1, between the mobile host and correspondent nodes. (Another latency of acquiring an address from some local DHCP server is neglected here because the transit delay of messages across internetworks appears dominant.) In contrast, the blackout period for

Mobile IP reflects a round-trip time, τ_2, between the mobile host and its home agent. It can be seen that our scheme suffers less packet loss than Mobile IP does if $\tau_1 < \tau_2$, because the packet arrival rate addressing a mobile host is nearly identical in the two subject approaches. Otherwise, lost packets in our scheme outnumber those in Mobile IP. The difference of packet loss relative to τ_2 per handoff can be expressed as

$$\frac{|\tau_1 - \tau_2|}{\tau_2}.$$

Note that τ_1 and τ_2 are closely related to user mobility dynamics and geographical dispersion of the mobile host, correspondent nodes, and the home agent.

When the mobility rate is low for infrequent handoff, packet loss will become less suffering. Nevertheless, comparisons of our scheme with Mobile IPv6 in this dimension can be reasoned similarly.

6. CONCLUSION AND FUTURE WORK

This paper presented a design and an implementation to support host mobility in wireless internetworking environment. We dealt with a central problem that communication session setups on a mobile host are no longer statically bound to the local IP address. In practice, we let a mobile host use two IP addresses for network- and higher-layer identifications, respectively, and devise a mapping table to associate the two identities. Packets to or from a session endpoint undergo with reference to the table proper address translations before delivery. Network applications are thus enabled to send or receive their traffic under the illusion of a typical TCP/IP protocol suite beneath. Without exploiting any mobility agents, our scheme operates in an end-to-end fashion; packets traverse optimized routes toward destinations, while free from being ingress filtered.

We contrived two data structures for a mapping table, over which three translation techniques were developed. The first is for general-purpose use, the next particularly for TCP or UDP messages carrying port numbers, whilst the third amalgamates prior two strategies. Addresses associations result from querying the Domain Name System or are gleaned implicitly from direct message exchanges between two communication parties. In addition, whenever changing to a new IP address, a mobile host initiates secure updates to its root name server in order to prevent name-to-address records from being cached elsewhere. This avoids possible connection faults because of resolving the host to an outdated address. Furthermore, the mobile host issues a control message, ICMP Echo Request carrying a predefined HID option, to notify correspondent nodes about its new location. Upon reception, each correspondent node will update its own translation table accordingly.

Simulation results show that our scheme remarkably outperforms Mobile IP and Mobile IPv6 in terms of communication delay, packet error rate, and excess datagram size. Performance aspects in operational overhead within the TCP/IP protocol stack and potential packet loss due to handoffs are also addressed. Other than quantitative comparisons, qualitative discussions reveal that our proposal is apropos to heterogeneous networking applications.

As a potential weakness, our approach cannot rule out the possibility that the location update procedure by a mobile host (due to movements) is progressing, while a new correspondent node contacts the DNS server to resolve the host current address. In this event, the resolved address may be stale such that the correspondent node cannot correctly direct messages to the mobile host. This leads to connection-establishment failure or packet loss. Such adverse effects could be recovered by higher-level protocols querying the DNS server some time later again and retransmitting packets. We believe that such adverse effects are rare as well, since it is unlikely for a DNS query to coincide with an in-progress dynamic DNS update because the update is a lightweight procedure and can be completed shortly.

To conclude this study, we stress three directions for future work. The first is concerned with security. Section 3.6 described a fraud control mechanism that could be entrenched as part of the proposed framework. Aside from such authentication considerations, there are issues, namely authorization, non-repudiation, and encryption key distribution that merit more thorough investigations. Second, a smooth handoff procedure facilitating communication activities will add strength to the proposed scheme. Our development relies on upper layer protocols to recover lost datagrams (section 5.4), which, however, does not impair this scheme but might cost nontrivial network resource. Given substantial research on smooth handoff nowadays, any efficient technique can be applicable to our architecture. Last but not least, it is expected that future wireless Internet comprises multi-tier overlay networks providing mobile users with richer multimedia services. As outlined in section 4.1, our scheme forms a generic, convergent base supporting users to roam amongst heterogeneous networks. Further elaborations are definitely required to streamline our development into underlying network operations.

REFERENCES

1. 3GPP, “Combined GSM and mobile IP mobility handling in UMTS IP CN,” Technical Report No. 3G TR 23.923, 3rd Generation Partnership Project, 2000.
2. A. Acharya, “Structuring distributed algorithms and services for networks with mobile hosts,” Ph.D. Thesis, Dept. of Computer Science, Rutgers University, NJ, 1996.
3. K. Ahmavaara, H. Haverinen, and R. Pichna, “Interworking architecture between 3GPP and WLAN systems,” *IEEE Communications Magazine*, Vol. 41, 2003, pp. 74-81.
4. H. Balakrishnan, S. Seshan, and R. H. Katz, “Improving reliable transport and handoff performance in cellular wireless networks,” *ACM Wireless Networks*, Vol. 1, 1995, pp. 469-481.
5. R. Caceres and L. Iftode, “Improving the performance of reliable transport protocols in mobile computing environments,” *IEEE Journal on Selected Areas in Communications*, Vol. 13, 1995, pp. 850-857.
6. R. Caceres and V. N. Padamanabhan, “Fast and scalable wireless handoffs in support of mobile Internet audio,” *ACM/Baltzer Mobile Networks and Applications*, Vol. 3, 1999, pp. 350-363.
7. S. Deering and R. Hinden, “Internet protocol version 6 (IPv6),” RFC 2460, IETF Network Working Group, 1998.

8. R. Droms, *"*Dynamic host configuration protocol," RFC 1541, IETF NetworkWorking Group, 1997.
9. D. Eastlake, "Secure domain name system dynamic update," RFC 2137, IETF Network Working Group, 1997.
10. P. Ferguson and D. Senie, "Network ingress filtering: defeating denial of service attacks which employ IP source address spoofing," RFC 2137, IETF NetworkWorking Group, 1998.
11. S. Gupta and A. L. N. Reddy, "A client oriented, IP level redirection mechanism," in *Proceedings of the 18th Annual Joint Conference on IEEE Computer and Communications Societies* (*IEEE INFOCOM '99*), 1999, pp. 1461-1469.
12. U. Gustafson and J. Forslow, "Network design with Mobile IP," in *Proceedings of INET*, 2001, http://www.isoc.org/isoc/conferences/inet/01/CD proceedings/T40/inet T40.htm.
13. D. B. Johnson, "Mobile host internetworking using IP loose source routing," Technical Report No. CMU-CS-93-128, Dept. of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1993.
14. D. B. Johnson, C. E. Perkins, and J. Arkko, "Mobility support in IPv6," RFC 3775, IETF Network Working Group, 2004.
15. G. Karagiannis and G. Heijenk, "Mobile IP," Technical Report No. TR-CTIT 99-21, University of Twente, the Netherland, 1999, http://ing.ctit.utwente.nl/WU4/Documents/mobipa.pdf.
16. Y. B. Lin, "Modeling techniques for large-scale PCS networks," *IEEE Communications Magazine*, Vol. 35, 1997, pp. 102-107.
17. Y. B. Lin and V. W. K. Mak, "Eliminating the boundary effects of a large-scale personal communication service network simulation," *ACM Transactions on Modeling and Computer Simulation*, Vol. 4, 1994, pp. 165-190.
18. P. Mockapetris, "Domain names − concepts and facilities," RFC 1034, IETF Network Working Group, 1987.
19. P. Mockapetris, "Domain names − implementation and specification," RFC 1035, IETF Network Working Group, 1987.
20. G. Montenegro, "Reverse tunneling for mobile IP," revised, RFC 3024, IETF Network Working Group, 2001.
21. C. E. Perkins, "Mobile IP tutorial," http://computer.org/internet/v2n1/perkins.htm.
22. C. E. Perkins, "IP encapsulation within IP," RFC 2003, IETF Network Working Group, 1996.
23. C. E. Perkins, "Minimal encapsulation within IP," RFC 2004, IETF Network Working Group, 1996.
24. C. E. Perkins, "Mobile-IP local registration with hierarchical foreign agents," Internet Draft <draft-perkins-mobileip-hierfa-00.txt>, IETF Network Working Group, 1996.
25. C. E. Perkins, "IP mobility support for IPv4," RFC 3220, IETF Network Working Group, 2002. (See also C. E. Perkins, (ed.,) *Mobile IP Design Principles and Practices*, Addison-Wesley Longman, 1998).
26. C. E. Perkins and D. B. Johnson, "Route optimization in mobile IP," Internet Draft <draft-ietf-mobileip-optim-11.txt>, IETF Network Working Group, 2001.
27. C. E. Perkins and K. Y. Wang, "Optimized smooth handoffs in mobile IP," in *Pro-*

ceedings of the 4th IEEE Symposium on Computers and Communications, 1999, pp. 292-299.

28. A. C. Snoeren and H. Balakrishnan, "An end-to-end approach to host mobility," in *Proceedings of the ACM MOBICOM*, 2000, pp. 155-166.
29. S. L. Tsao and J. C. Liu, "Mobility support for IPv4 and IPv6 interconnected networks based on dual-stack model," Internet Draft <draft-tsao-mobileip-dualstack-model-02.txt>, IETF Network Working Group, 2000.

Chun-Chieh Wang (王俊傑) is currently a Software Engineer at the Department of Mobile Internet, Information and Communications Research Laboratories, Industrial Technology Research Institute, R.O.C. He received his B.S. and M.S. degrees in Computer Science and Information Engineering from National Chiao Tung University in 2000 and 2002, respectively. His current research interests include wireless Internet protocols and wireless Internet applications.

Kuang-Hui Chi (紀光輝) is currently an Assistant Professor at the Department of Electrical Engineering, National Yunlin University of Science and Technology, Taiwan, R.O.C. He received his B.S. degree in Computer Science and Engineering, Tatung University in 1991. He earned M.S. (1993) and Ph.D. (2001) degrees in Computer Science and Information Engineering, both from National Chiao Tung University. From 2001 to 2003, he was with Computer and Communications Research Laboratories, Industrial Technology Research Institute, R.O.C. His current research interests lie in beyond 3G networks. He is a member of the IEEE.

Chien-Chao Tseng (曾建超) is currently a Professor in the Department of Computer Science and Information Engineering at National Chiao Tung University, Hsinchu, Taiwan, R.O.C. He received his B.S. degree in Industrial Engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1981, and M.S. and Ph.D. degrees in Computer Science from the Southern Methodist University, Dallas, Texas, U.S.A., in 1986 and 1989, respectively. His research interests include wireless Internet infrastructure and protocols, and wireless Internet applications.