MobileNAT: A New Technique for Mobility Across Heterogeneous Address Spaces

Milind Buddhikot, Adiseshu Hari
Lucent Bell Laboratories
{mbuddhikot, hari}@bell-labs.com

Kundan Singh
Columbia University
kns10@cs.columbia.edu

Scott Miller
Lucent Bell Laboratories
scm@lucent.com

Abstract

We propose a new network layer mobility architecture called MOBILENAT to efficiently support micro and macro-mobility in and across heterogeneous address spaces common in emerging public networks. The key ideas in this architecture are as follows: (1) Use of two IP addresses – an invariant virtual IP address for host identification at the application layer and an actual routable address at the network layer that changes due to mobility. Since physical address has routing significance only within a domain, it can be a private address and therefore, does not deplete the public IP address resource. (2) New DHCP enhancements to distribute the two addresses. (3) A new signaling element called Mobility Manager (MM) that uses Middlebox Communication (MIDCOM) framework to signal the changes in packet processing rules to the Network Address Translators (NATs) in the event of node mobility. Our proposal does not require any modifications to the access networks and can seamlessly co-exist with the existing Mobile IP mechanisms and therefore, can be used to provide seamless mobility across heterogeneous wireline and wireless networks. We report implementation details of a subset of our ideas in a testbed with Windows XP clients and Linux based NATs.

I. INTRODUCTION

Current trends indicate that the Internet address space will be strained further as the deployment of new wireless access networks (e.g., CDMA2000, 802.11 networks) enables diverse and potentially mobile end-systems such as telemetry devices, sensors, PDAs, laptops to be connected to the Internet. In the past, Network Address Translation (NAT) [25] devices have been widely used to combat the problem of address space depletion. These devices, typically deployed at the edge of a network, translate internal private address of every packet to external globally unique IP address and vice-versa. Though the next generation IPv6 standard defines a very large 128-bit address space and potentially eliminates the need for NATs, its widespread adoption and deployment is still unlikely in near future. In addition, NATs offer other benefits, such as private addressing in large public networks, frequent changes in addressing plans, use of heterogeneous (e.g. IPv4, IPv6) addressing schemes, and policy enforcement. Given this, NATs will continue to be an attractive solution in near future. In such a case, the network will consist of large number of domains, each with its own address space, that are delineated from the other domains and the public Internet by NAT devices. We call such domains *NAT-domains*.

Mobility is an important characteristic of wireless networks that enables location transparent access to network services. With the growth of public wireless networks, mobility across domains with heterogeneous address spaces will be needed. The mobility support at the network layer has been investigated extensively in recent years and several mobility solutions have been reported. We argue that these existing mechanisms have several deficiencies and present a new mechanism called *MobileNAT* that rectifies them and addresses the current trends. It supports both intra-domain micro mobility and inter-domain macro mobility and can co-exist with IETF standardized Mobile IP protocol [8].

The rest of the paper is organized as follows. Section II describes the basic network, mobility and session model we assume in our design. In Section III, we introduce the idea of using two addresses per host [19]. We then describe how address translation can be employed via the Anchor Node (AN) and Mobile Node (MN) to support intra and inter-domain mobility for sessions with hosts in the public Internet. Then we consider intra-domain and inter-domain sessions and highlight an interesting problem of virtual address aliasing and possible solutions. Section IV describes the use of DHCP and a new element called Mobility Manager (MM) to signal mobility events in the network. Section V presents the design and implementation overview of our MOBILENAT prototype using Windows XP client and Linux NAT box. In Section VI, we discuss possible extensions, limitations and other relevant aspects of the MOBILENAT scheme. Finally, Section VIII presents the conclusions.

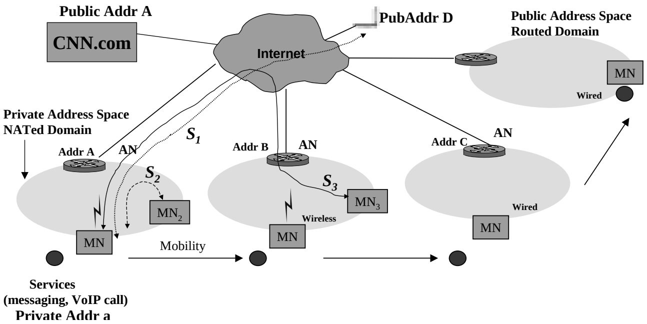


Fig. 1. *Basic network and mobility model*

II. Network and mobility model

In our basic model used to design MobileNAT (Fig. 1), we assume that each domain, called NAT-domain, consists of a homogeneous address space such as IPv4 or IPv6 and is delineated from the rest of the Internet by an Anchor Node (AN). Conceptually, an Anchor Node consists of two logically separate elements connected in series (Fig. 2): (1) Address Translator (AT) which performs Network Address Translation (NAT) [25] or Network Address Port Translation [25], and (2) a traditional router.

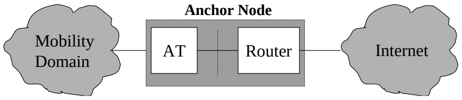


Fig. 2. *Structure of the anchor node*

In this paper, we consider only IPv4 address spaces. The addresses of the kind $192.168.*, 10.*, 172.16.*$ have been reserved as private addresses which need not be globally unique in the Internet [13], [23].

We believe that a large percentage of devices such as sensors, PDAs, laptops do not offer globally accessible services and therefore, do not need globally unique IP addresses all the time. For example, when accessing Internet from a public hotspot at a railway station, the user mostly accesses services from the email and web servers but does not offer long-lived services to other Internet users. Therefore, temporary private addresses within the NAT-domain will be common. All non-private addresses are globally unique and therefore, are public addresses. A NAT-domain may also allow co-existence of the private and public addressing.

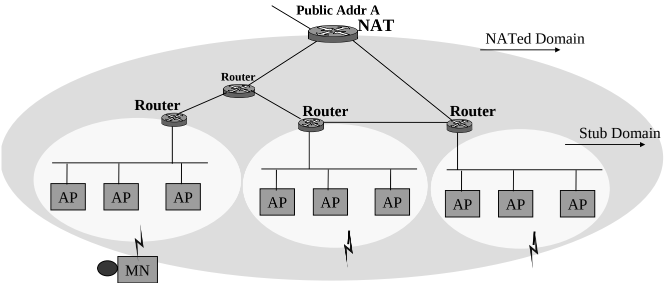


Fig. 3. *Details of a NAT-domain*

Different domains may use different access mechanisms such as 802.11 wireless LAN or 3G wide area wireless access. Each NAT-domain can be a layer-2 (e.g., switched Ethernet) or layer-3 (e.g., IP) routed network. A layer-3 domain consists of several layer-2 subnets interconnected by IP routers as shown in Fig. 3. A Mobile Node (MN) can experience three kinds of mobility: (1) layer-2 mobility within the same subnet, (2) layer-3 mobility across subnets, and (3) layer-3 mobility across adjacent domains. The MN may have several active TCP/IP (e.g., web, telnet) and UDP/IP (e.g., VOIP) sessions as shown in Fig. 1. For example, S_1 is an **Internet session** with a host in the public Internet, S_2 is an **intra-domain session** with a host in the same NAT-domain, and S_3 is an **inter-domain session** with a host in a different NAT-domain. As the MN roams in the above fashion, all the on-going sessions (S_1, S_2, S_3) should be preserved.

III. Basic ideas in MobileNAT

In this section, we describe the details of MobileNAT and its components. We first present the main ideas and then describe how mobility events are handled.

A. Two IP addresses per host

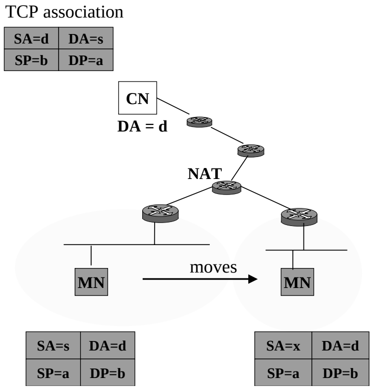


Fig. 4. Problem with IP address

Consider a mobile node (MN) (Fig. 4) with IP address $SA = s$ with an active TCP session of type S_1 to a corresponding node (CN) with IP address $DA = d$. The TCP layer at MN maintains the 5-tuple $[TCP, s, d, a, b]$ identifying this connection, where $SP = a$ and $DP = b$ are the source and destination port numbers. Similarly the 5-tuple at CN is [TCP, d,s,b,a]. When the MN moves to a new network, and gets a new IP address, say $SA = x$, the original TCP connection association at the server is no longer valid and therefore, the connection is lost. This results from the limitation of overloading the IP address with two functions: host identification by TCP layer, and network attachment information for routing [19], [6], [18].

This problem of session preservation on mobility can be solved at various layers: socket layer, transport layer and network layer. For instance, the TCP stack at the CN and MN can be modified to maintain the connection when the tuple changes. MobileNAT is a network layer mobility approach that works at the IP layer, and hence is transparent to the higher transport and application layers [8], [19]. To address the dual functionality problem of the IP address, we use two IP addresses for each MN (Fig. 4) [19]:
Virtual IP (A_v) address: A fixed address is used for host identification that does not change on mobility. This is exposed to the TCP/IP stack and higher layer applications.
Physical IP (A_p) address: This address identifies the *current* point of MN attachment to the Internet and is used for routing packets to the MN. Therefore, A_p has routing significance within a domain or in the entire internet. Clearly, this address must change as the MN moves.
Both A_p and A_v can be either private or public addresses as shown in Table I. Using two IP addresses per host may first seem wasteful. However, except case 4, in all the other cases, at least one of the addresses is private and therefore does not conflict in the public Internet.

TABLE I

Types of A_p and A_v addresses

	A_p	A_v
case 1	Private	Private
case 2	Private	Public
case 3	Public	Private
case 4	Public	Public

B. Intra-domain mobility for Internet-sessions

For simplicity, we first consider case 1 in Table I, where both A_p and A_v are private addresses. Fig. 5 shows a TCP connection from the MN with addresses (A_p, A_v) to the correspondent node, the CNN web site, with address CNN and port 80. The MN's

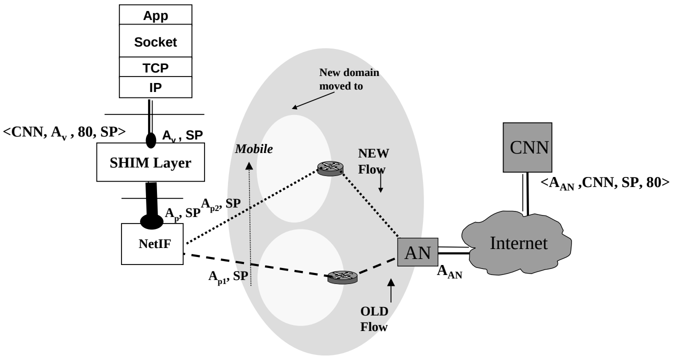


Fig. 5. Handling intra-domain mobility for Internet-sessions

TCP association is $(TCP, A_v, CNN, SP, 80)$. Since the address A_v is not used for packet routing, before MN transmits the packet to AN, it must *transform* it to make it routable on the Internet. One technique is to translate A_v in every packet to A_p. Specifically, for all the transmitted packets, the MN changes the source address SA – an operation called Source NAT (SNAT) and for received packets it changes the destination address – an operation called Destination NAT (DNAT). A thin software layer, called *shim-layer*, between the TCP/IP stack and the network interface driver in the client machine can maintain translation rules SNAT, DNAT:$A_v \mapsto A_p$. When the packet with source address $SA = A_p$ and destination address $DA = CNN$ reaches the anchor node, the address translator in AN maps the private address A_p to its external publically routable address A_{AN}. The CNN web server therefore maintains a 5-tuple association $(TCP, CNN, A_{AN}, 80, SP)$. The AN maintains a address mapping rule $A_p \mapsto A_{AN}$. In addition, the AN may perform translation on the TCP source port (SP). Note that for the traffic from the CNN server to the MN, similar translations are applied in the reverse direction at AN and MN. We can see that the address A_p is used for routing packets to MN in the NAT-domain and A_{AN} is used to proxy-receive packets from the Internet destined to the MN. The SNAT operation on MN's outgoing traffic is optional as the source address is not used for NAT-domain routing. This however, requires that the AN maintain a address mapping rule $A_v \mapsto A_{AN}$ and the subnet routers in the NAT-domain not do source address based ingress filtering.

As the MN moves, the address A_p changes within the NAT-domain but the 5-tuples maintained at the MN and CNN server do not change. This however, requires that the address mapping rules maintained at MN and AN must be changed. The name *MobileNAT* follows from the fact that NAT is performed in a mobile node and also, in the AN to support mobility. Since these NAT operations are transparent to the CN, no changes are required to CN.

The AN can use one of the two policies: (a) **Policy** P_1: If possible, expose MN's A_v external to the domain. (b) **Policy** P_2: Never expose MN's A_v. In case 1 discussed above, since A_v is private, policy P_1 is not possible and the default policy of AN is P_2.

In all the remaining cases, MN still maintains $A_v \mapsto A_p$ rule. Also, if AN enforces policy P_2, it maintains the $A_p \mapsto A_{AN}$ rule. The alternate scenario where AN enforces P_1 is considered below:

Case 2 [private A_p, public A_v :] A_v has global routing significance and all packets within the public Internet with this destination address can always reach the AN. Therefore, under policy P_1, AN maintains a rule $A_p \mapsto A_v$ for the traffic between MN and external CN, and exposes the MN's A_v address to the Internet.

Case 3 [public A_p, private A_v :] A_v has no global routing significance and therefore, AN must maintain a translation rule $A_p \mapsto A_{AN}$. Since, AN does not expose A_p to the Internet, this case does not deplete the public IP addresses.

Case 4 [public A_p, A_v :] Similar to case 2, since A_v is global, AN maintains a rule $A_p \mapsto A_v$. Although A_p has global significance, it is used only within the NAT-domain.

Table II summarizes the translation at AN for intra-domain mobility. We will later discuss the implications of using P_2 and exposing a public A_v transparently outside the domain on the inter-domain mobility.

C. Alternative to address translation at MN

An alternative to using address translation at MN is *tunneling* (Fig. 6). In this mode, an IP-in-IP tunnel [24], [20] is used to forward the packets from the MN to AN. The outer IP header has source address A_p of the MN, whereas the inner IP header has

TABLE II

Summary of packet mapping rules at AN

	A_p	A_v	AN rules	
			Policy P_1	Policy P_2
case 1	Private	Private	$A_p \mapsto A_{AN}$	$A_p \mapsto A_{AN}$
case 2	Private	Public	$A_p \mapsto A_v$	$A_p \mapsto A_{AN}$
case 3	Public	Private	$A_p \mapsto A_{AN}$	$A_p \mapsto A_{AN}$
case 4	Public	Public	$A_p \mapsto A_v$	$A_p \mapsto A_{AN}$

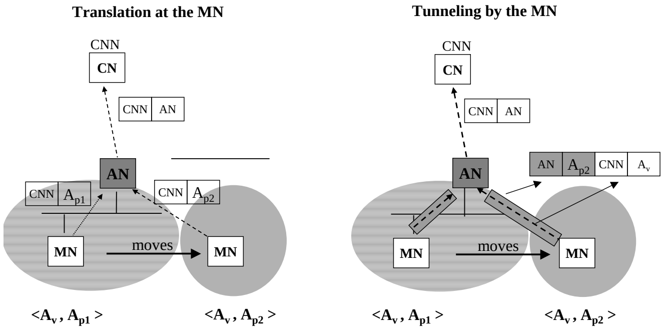


Fig. 6. Tunneling vs. translation

A_v as known by the CN. The AN strips off the outer header before forwarding the packet to the CN. The reverse traffic from the CN to MN does not necessarily need a tunnel from AN to MN. If tunneling is used in the reverse direction, the source address in the outer header can be that of the CN or the AN.

The advantage of tunneling is that it involves less processing overhead. The disadvantage is the additional header overhead, hence the increased packet size on bandwidth limited links such as wireless access links. For example, an Internet telephony application using 8 kb/s codec (e.g., G.729) with 20 ms packetization interval has 20 bytes payload, 12 bytes RTP header, 8 bytes UDP header and 20 bytes of IP header. Each IP packet in translate mode will be 60 bytes whereas that in tunnel mode will be 80 bytes (33% more). With translation, we need one actual IP and one virtual IP per mobile node. With tunnel mode, we can share the same virtual address across multiple nodes, provided a private node does not talk to another private node with the same virtual address. The choice of tunnel or translate mode can be made by the mobile node or can be a domain specific service provider policy. The chosen mode can be signaled at the time of acquisition of A_p, A_v addresses. If the domain already needs a NAT, then the processing overhead at AN in translation mode is not an issue since NAT co-located at AN will anyway do the translation. Also, the MN modifications are simple as most OS support standard IP-in-IP tunneling.

D. Inter-domain mobility for Internet-sessions

When the MN originates a new transport connection from within a NAT-domain, we call the AN of that domain as the Home-NAT for this connection. Fig. 7 shows a TCP session from an MN with addresses (A_{p1}, A_v) to the CNN web-server's HTTP port 80. The AN with external address AN_{home} is the Home-NAT for this session. When the MN moves to another domain containing AN with address AN_{vis}, it acquires a new actual IP address A_{p2}. We call the AN of this visited domain VD as the Visited-NAT.

Since the CN still thinks that it is connected to the old address AN_{home}, we need to signal the *Home-NAT* to forward the packets to the Visited-NAT's external address AN_{vis}. Also, the Visited-NAT is signaled to forward the packet on this session to the internal MN's actual address A_{p2}. Again, the choice of tunneling or translation between the Home-NAT and Visited-NAT is possible.

In the translation mode, the Home-NAT receives the packets from the CN, performs NAT using translation rule $(AN_{home}, SP_1) \mapsto (AN_{vis}, SP_2)$, and forwards the packet to the visited NAT. The Visited-NAT performs additional NAT using translation rule $(AN_{vis}, SP_2) \mapsto (A_p2, SP)$. Clearly, this case requires one external address AN_{vis} and AN_{home} per 65,536 TCP/UDP flows.

In the tunneled mode, the Home-NAT receives the packets from the CN node and tunnels them to the Visited-NAT in an IP-in-IP tunnel. The Visited-NAT removes the tunnel header, performs NAT using translation rule $(AN_{home}, SP_1) \mapsto (A_{p2}, SP)$ and routes the packet to MN. Since the NATs are expected to be on high-speed wireline network, the bandwidth overhead of tunneling is not

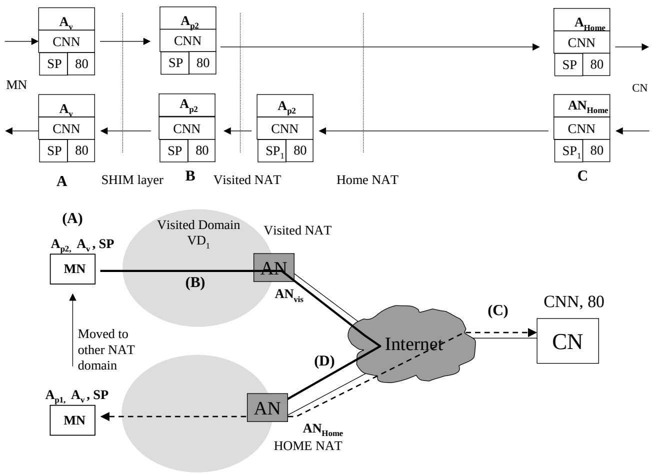


Fig. 7. Mobility across domains: preserving Internet-sessions

really an issue. Since all TCP/UDP flows from different MNs can be aggregated in a single IP-in-IP tunnel, a single IP address is sufficient, thus reducing the number of external IP addresses required.

Note that the traffic from the Visited-NAT to the CN can be either (a) direct, in which case the visited NAT fakes its source address as that of the Home-NAT, or (b) it can be always reverse proxied through the Home-NAT (in either tunnel or translate mode). Case (b) is useful if the router to which AN in the visited domain connects performs ingress filtering based on source address.

When the MN moves to another visited NAT-domain VD_2, if it uses a private A_v, the AN of VD_2 becomes the Home-NAT for the new connections originated in that domain. For the connections originated in domain VD_1 that are still active, the AN of VD_1 continues to be the Home-NAT. If a MN with long lived sessions moves across multiple domains, different connections may end up with different Home-NATs and Visited-NATs. A service provider may enforce a policy on how long a node can use its AN as Home-NAT after moving out of its domain. When the node returns back to the Home-NAT, the mappings for the existing connections must be updated. As long as the MN is using the old virtual IP of the Home-NAT domain, it must refresh this address with that domain.

If A_v is a public address exposed to the Internet (policy P_1), then the Home-NAT is like a statically assigned Home Agent (HA) in Mobile IP, and the Visited-NAT resembles the Foreign Agent (FA).

E. Intra-domain mobility for intra-domain sessions

To handle the intra-domain sessions between nodes in the same domain (S_2 in Fig. 1), the virtual address A_v is assigned routing significance within the NAT-domain such that all packets destined to A_v are routed to AN.

Fig. 8 shows an active session between an MN ($A_{p1} = 10.0.1.5$, $A_{v1} = 10.128.0.2$) and a fixed CN ($A_p = 10.0.4.9$)[1] Based on the destination address (DA), the MN recognizes CN to be intra-domain and does not perform the SNAT on packets to CN. These packets $(SA = A_v, DA = A)$ are routed by standard IP routing to the CN (line 2 in Fig. 8). The packets from CN to MN $(SA = A, DA = A_v)$ are routed to the AN by virtue of the fact that all packets destined to A_v are routable to AN. The AN performs the DNAT: $A_v \mapsto A_p$ on the packets to forward it correctly to MN (line 1 in Fig. 8). The shim-layer in MN performs additional DNAT before forwarding the packets to its transport layer.

When the MN moves to another subnet in the same domain, it acquires a new actual address $A_{p2} = 10.0.2.7$, the DNAT rules at the AN and MN are appropriately altered and the CN to MN session continues to work (lines 3, 4).

Alternatively, the CN can have a virtual address $A_{v,CN}$ that is registered in the local DNS, so that the MN does not need to distinguish between the public Internet and intra-domain CN. The packets destined for CN with $DA = A_{v,CN}$ are always routed to AN. The AN performs DNAT: $A_{v,CN} \mapsto A_{p,CN}$ and SNAT: $A_{p,MN} \mapsto A_{v,MN}$ before forwarding packets. This method also preserves the session when both nodes are mobile. However, the traffic to a non-mobile node with a virtual destination address is

[1] CN may be an intra-domain web server with a private address accessible only within the NAT-domain and not from the Internet.

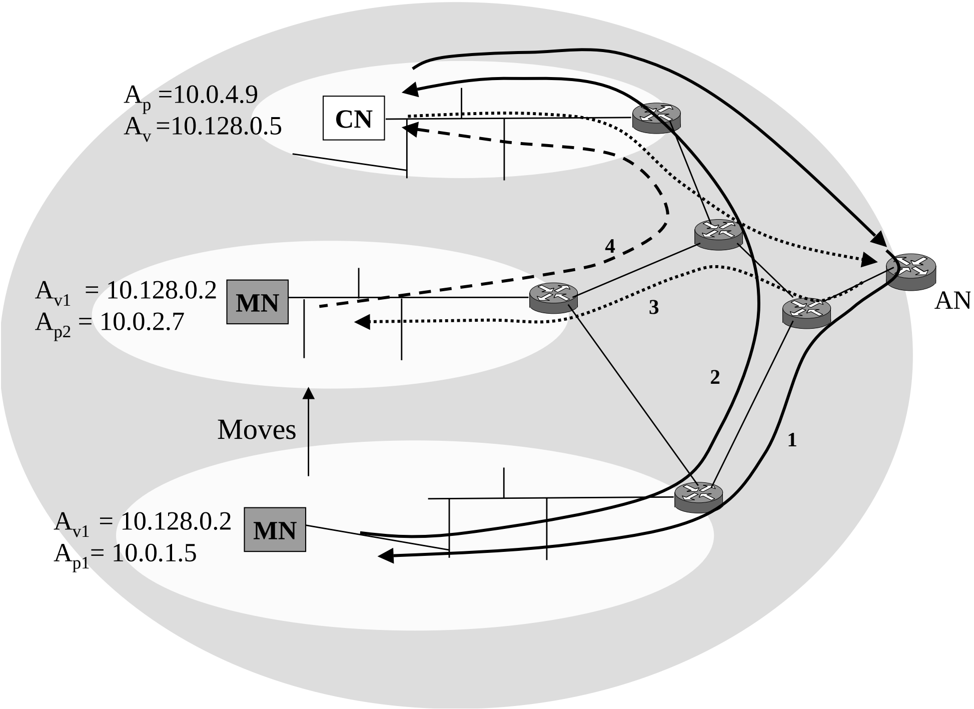


Fig. 8. Intra-domain session when MN moves to a new subnet within the domain

inefficiently routed via AN.

If MN moves out of the NAT-domain, the CN will be unreachable and the session S_2 cannot be preserved. We believe that this represents the correct semantics for the intra-domain sessions. If the CN's services are on a publically routable address, then the AN can proxy the packets correctly.

F. Preserving inter-domain sessions

An inter-domain session (S_3 in Fig. 1) is feasible in two cases: (1) *CN has a public address*: In this case, the NAT-domain must have internal routing that uses public addresses, not all of which are exposed to the Internet. Specifically, AN should expose only the addresses of those CN nodes that should be reachable from the public Internet. (2) If the NAT-domain uses a private address space and the CN therefore, has a private address, the AN of the domain must serve as a proxy for the CN and advertise its public address as the address for the CN. Therefore, the inter-domain sessions are similar to the Internet-sessions considered in Section III-B and III-D.

G. Virtual address aliasing

When the MN moves to the new NAT-domain, it attempts to renew its A_v and obtain new $A_{p,new}$. However, since private address allocation in different NAT-domains can be independent, in the worst case, another node in the domain may already be using A_v. We call this scenario as *virtual address aliasing*. If the new AN uses $A_v \mapsto A_p$ SNAT or DNAT rules for some other node's A_p, such aliasing results in ambiguity in data forwarding. Note that although A_v may not be unique, (A_p, A_v) pair is always unique in a domain. So the MN can obtain a new $A_{v,new}$ and $A_{p,new}$ during address renewal if it discovers A_v conflict. It then tunnels its packets to the AN using $[SA = A_{p,new}, DA = A_N]$ tunnel header to encapsulate the $[Originalpacket]$, which has $(SA = A_v, DA = A_{CN})$. The AN decapsulates the packets, and uses $(A_{p,new}, A_v)$ to differentiate them from existing (A_p, A_v) pair.

The MN continues to use the old virtual address A_v for the old sessions, and uses the new $A_{v,new}$ as the preferred virtual address for the new sessions. Unless the MN closes all the old sessions, it can not release A_v. Until then, it can not establish a new session to another conflicting node in the network with the same IP address. However it can still establish sessions to other non-conflicting nodes using $A_{v,new}$. If the MN's OS is not capable of assigning two virtual IP addresses to the same interface, the shim-layer tries to expose another virtual adaptor with the new virtual address. If this also fails, then it gives a choice to the user to (1) continue with the old sessions, and not establish new sessions, or (2) close all the existing sessions and start afresh. When the shim layer detects that all the sessions are closed it automatically removes the old virtual IP mappings and uses the new IP for all purposes. To avoid this connection tracking overhead, prompting the user is preferred over automatic handling.

An alternative approach to avoid aliasing allocates non-overlapping range of virtual addresses among different NAT-domains. This is useful only for a single service provider network with multiple NAT-domains.

H. *Moving between a NAT-domain and the public Internet*

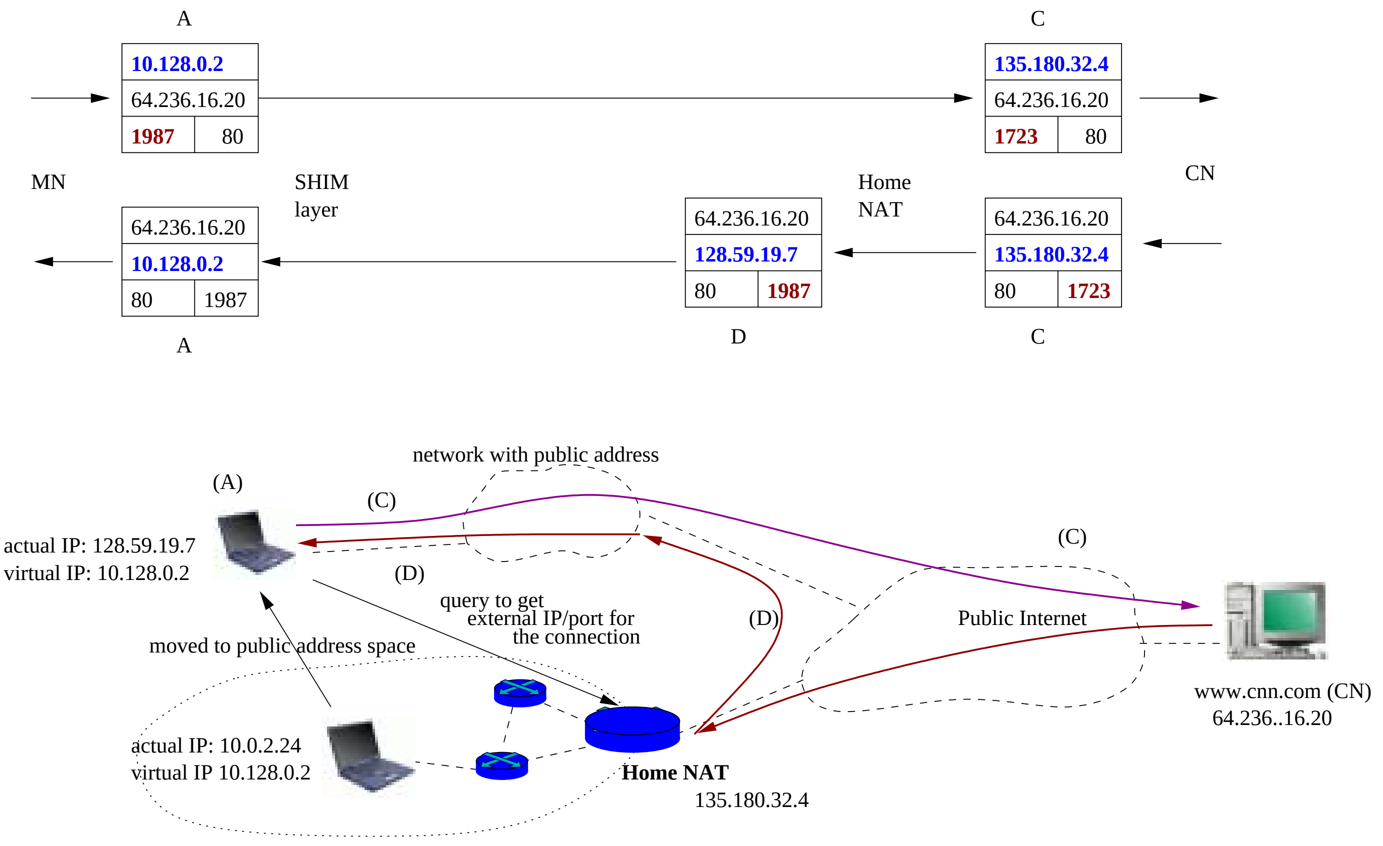


Fig. 9. MN from private address space to public address space

Fig. 9 shows a node moving from a private NAT domain to a public address space. The tunnel mode between the AN and MN is same as Mobile IP with the home agent at the Home-NAT, and the co-located foreign agent at the MN. Since there is no Visited-NAT, the shim-layer needs to do the translation and signal the Home-NAT's to update translation tables.

A node, already in a public address space, presumably using Mobile IP with a global home address, when moves to the NAT domain, can use the existing IP-in-UDP tunnel approach [16]. This scenario is not applicable to MOBILENAT. However, if the node's home agent is co-located with the AN, then it may choose to switch to MOBILENAT mode from the previous Mobile IP mode.

IV. NETWORK ARCHITECTURE

Fig. 10 illustrates the network architecture that implements the MOBILENAT ideas. It consists of the NAT-domains each of which contains three main MOBILENAT components: (1) an anchor node AN, (2) collection of a DHCP server and relays and (3) a new signaling entity called Mobility Manager (MM).

The AN may be implemented as a (1) traditional edge router with NAT support, or (2) a separate NAT device connected to a traditional router.

We propose using DHCP to acquire the two per-host IP addresses (A_p, A_v) and to signal the mobility events. We assume that the NAT-domain consists a layer-3 routed domain segmented into subnets interconnected by routers. Each subnet has a DHCP

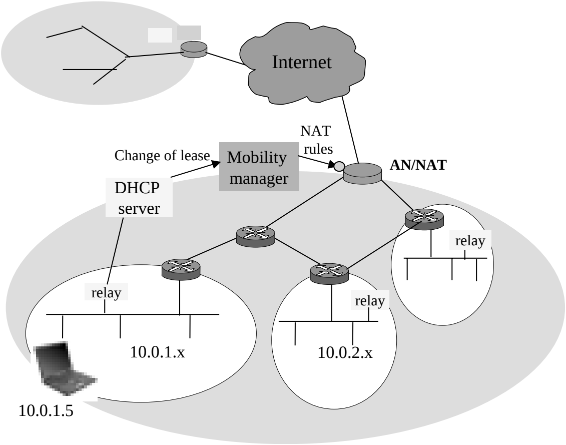


Fig. 10. MOBILENAT network architecture

relay either co-located with the router or separate, which forwards the DHCP requests to the per-domain DHCP server. The server maintains the address allocation maps and leases for the domain. If the entire domain is a layer-2 switched or bridged domain, a single server co-located with the AN is sufficient.

When the MN boots or moves to a new subnet, it sends DHCP messages to discover the domain's DHCP server and to request A_p and A_v using our new DHCP extensions. Analogous to current mode of DHCP operation, lease on A_p, A_v addresses must be periodically renewed. The lease periods for A_v and A_p can be either the same or different. In the case they are different, the A_v lease can be for long durations and A_v must be included in messages for renewal of A_p. In the event, lease on A_v cannot be renewed, the client knows *virtual address aliasing* likely occurred. Also, note that the DHCP scope definition for A_p is much the same way scope is defined in current DHCP server setups – typically one scope per TCP/IP network segment (subnet). Since A_v lease is valid across the entire domain, the scope of A_v is not subnet specific. The client device must be capable of detecting subnet moves. For example, a 802.11 client can listen to the Access Point (AP) beacons and decipher the change of subnet from the advertised ESSID.

The third entity in our architecture, Mobility Manager (MM), signals mobility events to the AN. A single MM may control several ANs in one or more domains and may be co-located with the DHCP server. The MM talks to ANs using the Middlebox communication protocol framework [26], [28] over secure communication channels such as IPSec [15]. The MN can obtain the MM address using DHCP configuration, much like obtaining default gateway and DNS, WINS server addresses. When the A_p, A_v corresponding to an MN changes, the DHCP server or the MN conveys the mobility event to the MM which in turns uses its MIDCOM functions to signal the changes in the mapping rules (Table II) to the AN. The details of the messages exchanged for this are not described in this paper.

One can use a separate lightweight protocol between the MN and MM instead of defining our DHCP extensions. The MN can learn the MM address via DHCP and register with it. The MM may authenticate the MN using AAA back-end infrastructure similar to the Mobile-IP registration procedure.

The DHCP messages exchanged between the MN and the DHCP server should be authenticated [7], and should use a timestamp in the signature to prevent replay attacks. This verifies to the DHCP server that it is talking to the correct client, and to the client that it is talking to the correct server.

The DHCP server, MM and NAT in our architecture are mutually trusted entities and secure communication must exist among them. Similarly, the MMs in different domains should use appropriate secure communication.

V. PROTOTYPE IMPLEMENTATION

This section describes the design and implementation overview of our MOBILENAT prototype using Windows XP client and Linux NAT. We implemented and demonstrated connection migration of an active TCP telnet session and a RealVideo session with external public node when the MN in the NAT domain moves from one subnet to another.

A. Client side components

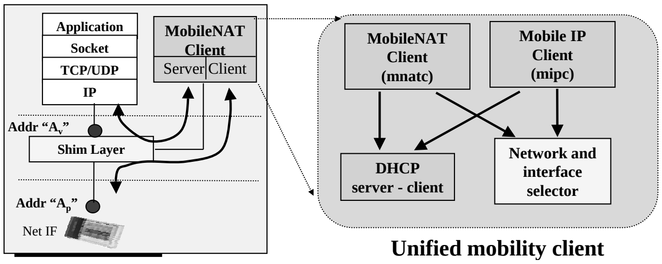


Fig. 11. MOBILENAT client on Windows XP

The client implementation has two parts: (1) the *shim-layer* is located between the TCP/IP stack and the network interface and acts as an intermediate driver, and (2) the application level entity called mnatc processes the DHCP client messages and interacts with the shim-layer.

A.1 Driver functions

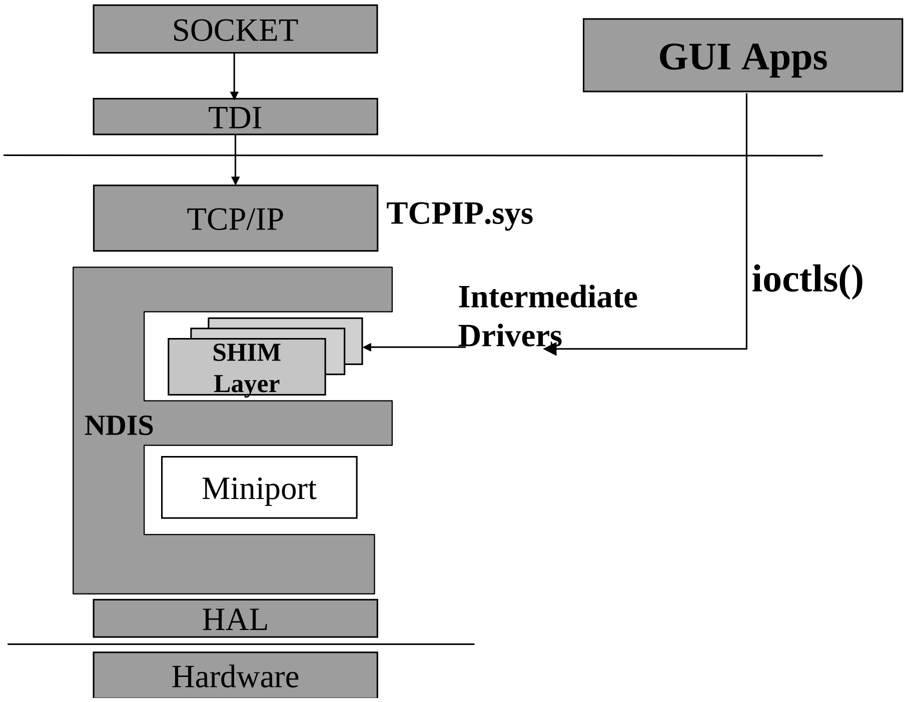


Fig. 12. Windows intermediate driver architecture

Microsoft Windows OS uses Network Driver Interface Specification (NDIS) to enable communication between a Network Interface Card (NIC) and a protocol (such as TCP/IP or NetBIOS). Fig. 12 illustrates various kernel-mode drivers it supports. A *miniport* driver directly manages a NIC. An intermediate driver interfaces between the upper-level protocol drivers and the miniport driver. The protocol driver provides interface to the higher level socket library and applications.

We implemented our MOBILENAT shim-layer as an intermediate driver by extending the *Passthru driver* available in the Windows Driver Development Kit (DDK). The Passthru driver does not modify any packet, but provides hooks to monitor the packets. Our driver allows configuring the MN using a new ioctl command in one of the four different modes: default *Passthru mode*, *Mobile-IP node in home network*, *Mobile-IP node in foreign network*, *Mobile-NAT node in translate mode*. This allows implementing both Mobile IP and Mobile-NAT in the same module.

In Windows, the network protocols such as TCP/IP, ARP, DHCP and ICMP are all implemented in a single monolithic black-box driver, TCPIP.sys. Since we do not have control over the driver's internal functions, we need to duplicate some of the DHCP, ARP and check-sum computation functions in the shim-layer to handle IP address mapping between virtual and actual addresses.

A.2 MobileNAT client

The MOBILENAT client, mnatc.exe, implements a DHCP client and a stripped down DHCP server, and must be running on the mobile node for the MOBILENAT functionality. It registers with the shim-layer driver to receive all incoming and outgoing packets of interest such as DHCP. When the TCPIP.sys driver sends a DHCP request, the shim-layer diverts it to mnatc, which then initiates a new DHCP request asking for virtual address along with other configuration parameters. We defined new DHCP options to request and receive virtual address A_v. The client extracts the appropriate fields such as the A_p, A_v, DNS name server, Mobility Manager address, gateway IP and subnet mask from the response, and sets these values in the shim-layer driver using our new ioctl commands. The shim-layer uses ARP to resolve the IP addresses of the gateway or other subnet nodes, to their MAC addresses for sending the packet. The mnatc client then invokes the DHCP server functions to satisfy the original DHCP request from TCPIP.sys. In this response, it specifies the default gateway as 10.0.0.1 and the subnet mask as 255.0.0.0. Other parameters, like DNS server IP address and domain name are as received from the actual DHCP server. It also indicates that the (virtual) IP lease will never expire and takes care of renewing the actual IP lease from the external DHCP server, transparent to the TCP/IP layer. Since the media disconnect and connect indications are blocked in the shim-layer, TCPIP.sys does not know about them and does not initiate any new DHCP request when the cable is re-connected.

The DHCP client and server functions can be decoupled from the MOBILENAT client to build an integrated unified mobility client [4] supporting both Mobile IP and MOBILENAT (Fig. 11). The co-located Mobile IP mode needs DHCP. If the node has multiple interfaces (e.g., 3G and 802.11b) and is within the reach of multiple access points, the selection of the appropriate active interface and network must account for measured signal strength and user assigned priorities based on pricing and preference. Our interface selector periodically polls the intermediate driver to collect radio signal strength and perform statistical averaging [4].

B. Network components

In our prototype, the DHCP server and mobility manager (MM) are integrated together and are co-located in AN. The server allocates the IP addresses and also configures the NAT using the Linux netfilter modules. We defined a new setsockopt option to change the IP address of the existing connections in the Linux connection tracking module.

B.1 Background on Linux NAT

We use the netfilter, iptables, ip_conntrack and ip_nat modules in the Red-hat Linux kernel 2.4.18 for NAT functions. The netfilter module provides a generic framework for any filtering and translation of IP packets on a router or host. It provides processing hooks at various stages in the routing path. In particular, for routers, the pre-routing stage can have DNAT and the post-routing stage can have SNAT. Moreover, standard NAPT functionality of mapping internal private IP and port to external public IP and port is implemented using the MASQUERADE table in the post routing stage for packet going to the external interface.

These NAT tables can be specified using the iptables command, which is a replacement for the old ipchains. The first packet in a session is processed by the ip_nat module, that looks up into the various tables, translates the IP address, and creates a connection tracking mapping. For example, if we set the MASQUERADE table using iptables to allocate IP address (e.g., in range 135.180.32.1-135.180.32.7) and port (in range 8000-10000) for outgoing connections, and make an outgoing TCP connection to a public host, the ip_nat module picks up an available external IP and port from the specified range for the connection and creates the connection tracking entry as an internal data-structure in the ip_conntrack module. All subsequent packets in the session are applied the same address translation of the selected address and port. When the reply comes from the public host, the connection tracking entry makes sure that the packet goes to the correct IP and port of the internal private node.

We defined a new setsockopt option, SO_REPLACE_NAT_IP, to change the connection tracking data-structure in the ip_conntrack and ip_nat modules when the node moves and acquires new address, so that the active sessions are preserved. The connection tracking module maintains a hash-table of all the connections in each direction. The hash table maintains the tuple of protocol, source IP, destination IP, source port and destination port. The forward direction structure is linked to the reverse direction. The connection structure also has protocol specific information (port number) and information about other modules like NAT and FTP helper). The hash key is computed using the source/destination IP addresses, ports and protocol. When the actual IP changes, the hash for the connection changes, hence its corresponding entry needs to be relocated from the old slot in the hash-table to the new slot.

The same is true for the internal hash tables in the NAT and FTP modules. In the current implementation, the connection tracking module invokes a function from the NAT module when the IP address changes.

Another limitation of the Linux NAT is that SNAT can be applied only in post-routing and DNAT only in pre-routing stages but not vice-versa. This means, for intra-domain sessions, we must convert the destination (DNAT) from virtual to actual IP before routing decision is made, and then after the routing decision is made, convert the source (SNAT) from actual to virtual. This causes problem with application protocols such as FTP, RTSP and SIP that send IP address in the signaling message payload. Consider a node with private address A_i initiating an FTP session to an external host A_h. Suppose the NAT has allocated the external IP of A_e and port $P_{e,1}$ for this connection. When the node sends an FTP command to download a file to its IP A_i and port P_i, the FTP module in NAT traps the message and changes the IP and port to A_e and $P_{e,2}$, respectively. Now the server will send the data packets to this new port $P_{e,2}$ of NAT, which will in turn forward them to A_i/P_i. If the actual IP changes from A_i to A_{new} the FTP module will not know about this change and the old association for the data traffic will break. Even with MOBILENAT the FTP module breaks, since the association is maintained with respect to the actual IP rather than the fixed virtual IP.

Linux NAT does not allow SNAT in pre-routing stage so the actual source IP can not be changed to the virtual source IP before the FTP module is invoked. An alternative approach is to use two different NATs. An internal NAT changes the virtual to actual and vice-versa, and the external NAT applies the standard masquerade table for the virtual IP to external IP (and port) for the connection. However, this involves more processing as the translation needs to be done twice. Alternatively, we can enhance the FTP and other helper modules to also provide setsockopt to change the IP from old to new.

B.2 DHCP server and Mobility Manager

We enhanced the udhcp [3] DHCP server to support MOBILENAT. The modifications for virtual IP address affects only the MOBILENAT hosts, and the server can still be used to allocate IP addresses to non-MOBILENAT hosts.

The server maintains a range of virtual IP addresses and a set of ranges for actual IP addresses for different subnets. The virtual IP address is returned in the new DHCP shim-layer address option of the response, only if the DHCP request had that same option element present. The actual IP address is allocated based on the subnet of the relay agent.

Since the DHCP server is co-located in the AN, for the sake of expediency, we integrated the Mobility Manager functions in the DHCP server. However, the same functions can be implemented in a separate active task. In our prototype, once virtual and actual IP are allocated for a host, the DHCP server updates the SNAT and DNAT tables in NAT to map the actual source IP to virtual source IP and the virtual destination IP to the actual destination IP for the packets coming in and going out on internal interface, respectively. Note that SNAT and DNAT are used only for intra-domain sessions and not applied for the connection from internal private node to external public node. If the DHCP server detects that the actual IP was changed for a given virtual IP for a node, then it also updates the connection tracking module with setsockopt command described earlier.

In our testbed, we have measured intra-domain mobility handoff latency of the order of few tens of milliseconds.

VI. DISCUSSION

In this section, we discuss possible extensions, limitations and other aspects of MOBILENAT.

***Eliminating special client software*:** One possible criticism of MOBILENAT scheme is that it requires changes to the client software - specifically, a shim-layer driver and a MOBILENAT client. However, this is not different from several other mobility schemes, the most prominent being Mobile IP. From our prior experience in building Mobile IP client software for Windows to support per-subnet Foreign Agent (FA) and co-located FA modes [4], we believe the complexity of MOBILENAT software is quite comparable.

An alternate approach that can eliminate the need for client software is possible (Figure 13): in this mode, the address translation functions from the shim-layer in the client MN are moved to a per-subnet anchor node AN such as the default subnet gateway, router or FA. The IP address of the host never changes despite change of subnets and domains, which leaves host transport connections uninterrupted. The subnet AN proxies the packets to and from the MN after appropriate translation. Clearly, the packets are forwarded between MN and AN only by layer-2 forwarding mechanism such as label switching, Ethernet switching or shared wired, wireless LANs. We illustrate this further with an example in Figure 13 where an MN maintains a fixed address A_v as it

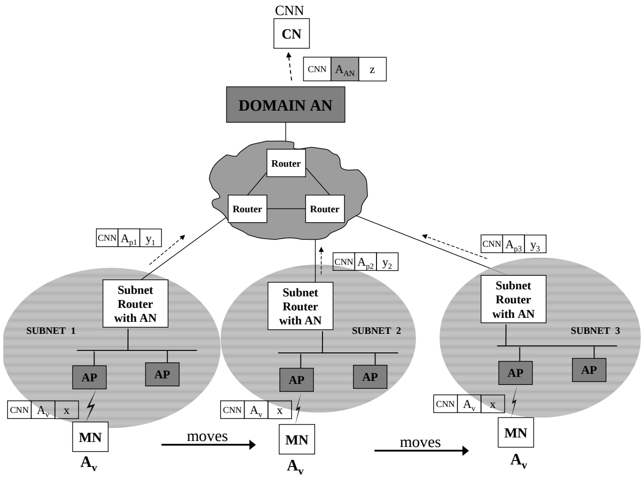


Fig. 13. Case when MN does not need special software

roams across 802.11 WLAN APs in different subnets. The MN initiates TCP flow to CNN website when it is in subnet 1. In this case, the packets transmitted by the MN have $(SA = A_v, DA = CNN, SP = x)$. The subnet router performs NAT to change these fields to $(SA = A_{p1}, DA = CNN, SP = y_1)$. The domain AN performs the last NAT to change the packet fields to $(SA = A_{AN}, , DA = CNN, SP = z)$ and sends the packet to the CNN website. When MN moves to second subnet, it continues to send out packets with fields $(SA = A_v, DA = CNN, SP = x)$, but the subnet router translates them to $(SA = A_{p2}, DA = CNN, SP = y_2)$.

The alternative approach above requires deploying AN in every subnet. We believe change in client to be easier and incremental than change in the network infrastructure, given that installing a new software or driver is just a few mouse click away for the Microsoft Windows users.

***Fast hand-off and DHCP latency*:** DHCP introduces additional latency that is not suitable for fast hand-off. We can modify the DHCP implementation so as to remove the timeouts on mobility, however that implies a non-standard DHCP implementation. Alternatively we can propose a new signaling protocol for mobility between the mobile node and the mobility manager similar to Mobile IP registration mechanisms.

***Route optimization for intra-domain sessions*:** All packets to the virtual address of MN go to the NAT device before getting routed to the actual address of the MN. If the MN is talking to another MN in the same NAT domain, then we can optimize the routing path by sending packets directly between the two nodes. However the MN needs to detect that the other side is also a MOBILENAT client in the same domain and indicate to it the actual IP address such that the packets can be sent by the other side directly to the actual IP address of this MN. When this MN moves, the other side will detect the ICMP host unreachable error and will start sending the packets to the virtual IP (to the NAT) again. There are two options to convey the actual IP: define a signaling between the two MNs (similar to [27]) or define a signaling from the NAT/MM to the MN. Both approaches should allow secure (authenticated and authorized) signaling.

***Co-existence with Mobile IP and Hawaii*:** If a mobile node uses Mobile IP instead of MOBILENAT it can still roam in the NAT domain. The only difference is that it does not have a virtual IP address. Other IP based micro-mobility mechanisms (like Hawaii [22] or hierarchical Mobile IP) can also co-exist with the MOBILENAT scheme. In fact in an incremental deployment scenario, one can use Mobile IP for macro-mobility and MOBILENAT for intra-domain mobility.

***Co-existence with non-mobile IP clients*:** A non-mobile client using simple IP can also exist in the NAT domain.

***Using other existing protocols*:** Parts of the existing protocols such as IDMP [5], RSIP, IPv6 are similar to some of the schemes in MOBILENAT proposal. For example, address translation from IDMP and destination option from IPv6 can be explored and reused instead of inventing a new signaling option.

***IPv6 in NAT domain*:** With IPv6 NAT domain, we do not need to use private IP addresses as there will be plenty of IPv6 public addresses. This involves the case when the MN has virtual and actual public IP addresses. However, the NAT device is still needed to translate between this IPv6 domain and the external IPv4 public Internet.

***Single unified mobility client*:** A unified mobility client implements mobile IP, simple IP and MOBILENAT mechanisms and dynamically selects the best suited and available mechanism for mobility and network access. For instance, if a foreign agent is not found but a mobility manager is present then it uses the MOBILENAT scheme, whereas if both foreign agent and mobility manager are present but the user wants to host a service (e.g., a web server) then it may use the Mobile IP scheme with its well known public home address.

***Mobility to 3G network*:** The MobileNAT mechanism can be easily used for supporting mobility in 3G networks. For example, in case of 3GPP2 networks, the NAT, mobility manager and DHCP server can be implemented in the Packet Data Serving Node (PDSN) that interfaces radio access network (RAN) to a packet switched network[1]. Similarly, a Gateway GPRS Support Node (GGSN) in a 3GPP (UMTS) network can implement these functions [2].

***Paging using IP multicast*:** Existing IP multicast based paging mechanisms [21] can be reused in MOBILENAT. Whether this is sufficient or we need more specific paging scheme is for further study.

***Multiple NAT for load balancing*:** It might be desirable to share the address translation load among multiple distributed NAT devices in a single domain. Secondly, a network can have more than one NAT devices along the path from MN to CN. The details of packet processing and message flows in this scenario are excluded from this paper due to space constraints.

***Services by MN need DNS updates or public IP address*:** If a mobile user wants to host a public service (e.g., web server or media server) it should be accessible from outside the NAT domain. This is a generic problem with NAT. The MOBILENAT system can be extended to allow leasing a public virtual IP address to the internal node from the external public address pool available with the NAT device. Alternatively, dynamic DNS updates can be used to update the host name to IP address mapping for the mobile node.

***Security Considerations*:** The security issues need to be addressed in the following places:

- MOBILENAT should work with IP security (IPSec). The AH (authentication header) mode is not possible with network address translator devices in the network, however the ESP (encapsulated security payload) should be allowed. IPsec is used in VPNs (virtual private networks).
- MOBILENAT should work with transport layer security (TLS) like secure socket layer (SSL).
- The inter-domain mobility requires packet flow between two different NATs in two different domains, i.e., the Home-NAT forwards the packets to the Visited-NAT. The system should also allow reverse tunneling from Visited-NAT to Home-NAT so that the Visited-NAT does not have to spoof the source IP address.
- The mobile nodes subscribed to some independent service provider with appropriate AAA infrastructure should be allowed to roam in the MOBILENAT domain.

VII. RELATED WORK

The topic of supporting mobility on the Internet is a well-researched topic and several macro and micro-mobility schemes have been reported in the literature [8], [27], [10], [5]

A. Mobile IP (MIP) and its variants

IETF has standardized Mobile IP (MIP) [8] as a network layer mobility scheme. In the MIP architecture, the MN has a fixed home address (HA). When the MN is in its home network, it uses its home address. When the MN moves to a different foreign network, it discovers a foreign agent (FA) in the local network, and registers with its home agent (HA) in the home network. The packets from MN to CN are sent directly with the source IP as the home address. The packets from CN to MN reach the HA. The HA encapsulates the IP packet inside another IP packet and forwards it to the FA using IP-in-IP tunnel. The FA decapsulates the packet and forwards the actual inner packet to the MN. Since MN and FA are in the same broadcast network, the destination IP remains as home address of the MN, and the forwarding is done at layer-2. The outer tunnel from HA to FA has the source IP of the HA and destination IP of the FA. The same IP address of the FA can be used by any number of MN for tunneling. In the absence of a FA in the visited network, the MN may use a co-located foreign agent. However, in this case it will need one local IP address apart from the global fixed home address.

The triangular routing is prominent in Mobile IP and extensions have been proposed, e.g., reverse tunneling, location register (MIP LR) and route optimization (MIP RO), to avoid the triangular routing.

***IPv6 mobility*:** The new IPv6 endpoints can be built such that they respect the destination IP option. This is similar to the route optimization, where the CN can discover the actual IP address of MN, and send the packets directly to the network in which MN is currently located. Since IPv6 is not yet deployed, this solution is not feasible.

***Mobile IP with NAT*:** Since FA and MN inside a private address space are not visible to the HA, a UDP-in-IP tunnel is proposed [16], instead of IP-in-IP from the HA to the FA, to traverse through NAT and NAPT devices. The HA knows that the FA is behind NAT if it discovers that the IP address (private) of the FA is different from where the packet is actually received (external address of NAT). The NAT mapping is established from the node in the private address space (FA or MN) to the HA in a signaling message, and the reverse mapping is used by the tunnel from HA to FA.

B. Micro mobility protocols

Mobile IP alone is not suitable for micro-mobility and fast-hand off. Every time the MN moves, it must inform the home agent of the new location. If the HA is far-off, informing the HA every time the MN moves introduces undesirable delay in hand-off. A number of micro-mobility protocols have been proposed, to provide a hierarchical signaling such that the global HA is not informed if the MN moves within one domain. The local domain gateway handles the micro-mobility. These protocols assume Mobile IP for global mobility.

***Cellular IP (CIP)*:** One big problem with Cellular IP is that it uses proprietary non-IP protocol within the domain [29] and can not inter-operate with other IP endpoints in the domain. This is suitable for a single service provider environment but not for a heterogeneous mix of endpoints.

***Hawaii*:** Hawaii [22] micro-mobility protocol allows an MN to carry a fixed IP address within the domain and therefore, requires the intra-domain routers to maintain per-host routing information. However, this entails changes to routing infrastructure, specifically, implementation of specialized path setup schemes that update forwarding table entries in the domain routers. Hawaii scheme therefore does not scale with increasing number of mobile nodes in the domain.

***Hierarchical mobile IP (HMIP)[14]*:** This scheme introduces the concept of "regional registration", which performs registrations locally in the visited domain instead of to a remote HA. This optional extension to MIP introduces a new network entity called Gateway Foreign Agent (GFA) that manages host mobility within the domain. The visited domain supports two level hierarchy of foreign agents: at the top level there is at least one GFA. Beneath a GFA, there are one or more regional FA (RFA). When the MN first arrives at a visited domain, it registers with HA. At this registration, the home network generates a registration key that is then distributed to MN and GFA. When MN moves from across FAs in the visited domain, it uses this key to register to the local GFA. When the MN moves across GFAs or to a new domain, the global HA is informed. Unlike Hawaii, HMIP uses tunneling between the GFA and FA.

***Intra-domain mobility protocol (IDMP)*:**

IDMP [9] is similar to hierarchical mobile IP, except that it also allows multiple mobility agents (similar to GFA in HMIP) for load balancing, and can use DHCP for signaling. IDMP is also well suited for NAT domains where the mobility agent also interacts with the NAT device for IP address translation.

C. Host Identity Protocol (HIP) Architecture

The motivation for HIP architecture stems from fundamental limitation of IP address mentioned in Section III-A [19], [6], [18], namely that it currently embodies the dual role of host locator and endpoint identifier. Each IP address names a topological location in the Internet, thereby acting as a routing direction vector, or locator. At the same time, the IP address names the physical network interface currently located at the point-of-attachment, thereby acting as a endpoint name. HIP architecture introduces a new namespace called Host Identity (HI) namespace to eliminate the confounding of IP and DNS namespace. The Host Identity is cryptographic in nature; it is a 128-bit public key of an asymmetric key-pair and serves as the endpoint identifier. IP address continues to act as locator. Thus, the endpoint names and locators are separated from each other, allowing decoupling of internetworking and transport layer. The HIP architecture introduces a Host Identity Protocol [18] and associated packet payload between transport and internetworking layer. The HIP protocol defines a cryptographic exchange, called the HIP base exchange, between the communicating endpoints and provides for limited forms of trust between systems. It enhances mobility, multi-homing and dynamic IP renumbering, aids in protocol translation/transition and reduces certain types of denial-of-service (DoS) attacks [18].

Since HIP binds transport associations to Host Identities and not to IP addresses, mobility and multi-homing can be supported easily. As the end-host roams, the IP address can change dynamically without affecting transport association. The HIP architecture defines a 32-bit Local Scope Identifier (LSI) to facilitate using Host Identities in existing protocols and APIs such as sockets. It also introduces a new network element called rendezvous server to enable communication to mobile nodes. The mobile nodes register and continuously update their current network layer addresses with the rendezvous server to enable other nodes to locate them.

D. Application level mobility

Session Initiation Protocol (SIP [12]) is also proposed for application level mobility [11] in IP telephony and 3GPP, but can not be used as generic connection migration protocol (e.g., to preserve TCP sessions on mobility). The MN re-REGISTERs with the global home SIP server when the MN moves. The existing calls are modified using re-INVITE to update the media transport address after mobility. One advantage is that it can work without any mobility support from the lower layer.

E. Transport level mobility

A number of proposals are available to support mobility at the transport layer or the socket layer however all of them suffer from a serious drawback that the CN needs to be modified.

***Virtual-NAT*:** More recently, VirtualNAT [27] has been proposed for connection migration of TCP connections when a host moves from one location to another. It uses the concept of two addresses, a fixed virtual IP and a changing actual routable IP. The translation is done in the client network stack. Explicit signaling messages are exchanged between the two connecting endpoints for mobility. The proposal does not deal with external NAT devices. It requires modification at both the end-points for connection migration.

***Real-specific IP (RSIP)*:** Although, RSIP is not for mobility, it allows a node to query the NAT device for its external IP and port. The private node uses a tunnel to the NAT where the inner actual IP packet contains the actual destination IP in the remote domain and the actual source IP that the NAT has allocated for this private host.

F. Comparing with MobileNAT

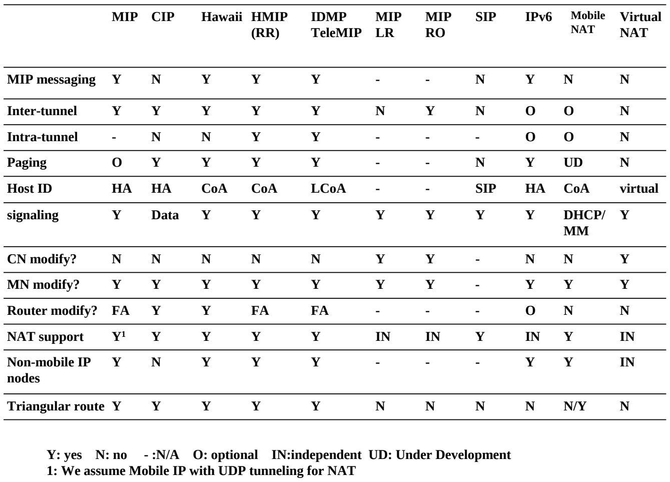
Fig. 14 highlights some of the differences between MOBILENAT and other approaches. In particular, most of the Mobile IP based approaches, including micro-mobility protocols, require changes in the routing infrastructure, per-host routing or deploying the foreign agents. On the contrary, MobileNAT requires no changes to routing infrastructure, except requiring an AN which is a NAT function commonly found in domain egress routers and control path entities in the form of a modified DHCP server and a Mobility manager.

In the common case, where end-host MN does not offer long-lived sessions and initiates session to internet hosts, MobileIP still requires a public Home address (HAddr) and a home agent (HA) to provisioned a-priori. The packet forwarding in this case is always via remote HA and though dynamic HA allocation is possible, typically it happens only at the time of MIP registration. The MIP protocol requires a client software in both foreign agent per-subnet and co-located-FA modes. The MobileNAT packet forwarding for common case does not require any apriori configuration. Also, MobileNAT implementation is possible without a specialized client software.

When the MN moves within a domain across subnets, in pure MIP, registrations must be sent to the remote HA. In MobileNAT, the A_v lease renewal and obtaining new A_p is a fast operation as it is done locally in the domain.

The inter-domain mobility case with MobileNAT has packet forwarding similar to triangular routing in MIP; the Home-NAT is much like HA in Mobile IP, and Visited-NAT is like FA. The difference is: The Home-NAT to MN association is dynamic; it breaks when all flows of the MN close or when a domain specific policy dictated lease period expires without lease renewal. Also, the Home-NAT is dynamically selected based on the domain where the packet flows are commenced. Therefore, unless MN moves extremely fast over large distances and keeps very long lived connections, in common-case network path between Home-NAT and Visited-NAT is short and therefore, packet forwarding is efficient. This is in contrast to MIP where the HA can be very remote from FA.

Unlike MOBILENAT, some other optimized approaches, such as MIP-LR, MIP-RO, Virtual NAT, need changes in CN. Moreover, MOBILENAT assumes an external NAT device, and provides choice between tunneling and translation. It uses the existing protocols like DHCP and ICMP for signaling. It can co-exit with other IP and Mobile IP nodes.



	MIP	CIP	Hawaii	HMIP (RR)	IDMP TeleMIP	MIP LR	MIP RO	SIP	IPv6	Mobile NAT	Virtual NAT
MIP messaging	Y	N	Y	Y	Y	-	-	N	Y	N	N
Inter-tunnel	Y	Y	Y	Y	Y	N	Y	N	O	O	N
Intra-tunnel	-	N	N	Y	Y	-	-	-	O	O	N
Paging	O	Y	Y	Y	Y	-	-	N	Y	UD	N
Host ID	HA	HA	CoA	CoA	LCoA	-	-	SIP	HA	CoA	virtual
signaling	Y	Data	Y	Y	Y	Y	Y	Y	Y	DHCP/ MM	Y
CN modify?	N	N	N	N	N	Y	Y	-	N	N	Y
MN modify?	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y
Router modify?	FA	Y	Y	FA	FA	-	-	-	O	N	N
NAT support	Y[1]	Y	Y	Y	Y	IN	IN	Y	IN	Y	IN
Non-mobile IP nodes	Y	N	Y	Y	Y	-	-	-	Y	Y	IN
Triangular route	Y	Y	Y	Y	Y	N	N	N	N	N/Y	N

Y: yes N: no - :N/A O: optional IN:independent UD: Under Development

1: We assume Mobile IP with UDP tunneling for NAT

Fig. 14. Comparison of different mobility approaches

G. *Comparison of MobileNAT with Host Identity Protocol Architecture*

The Host Identity Protocol (HIP) architecture [17] and associated protocol [18] are quite ambitious efforts. The MOBILENAT has much narrower objectives of incrementally and elegantly modifying existing internet infrastructure to support intra/inter-domain mobility across heterogeneous address spaces. The virtual address A_v in MOBILENAT resembles a Host Identity or more specifically the LSI in HIP; it is a 32-bit number that has variable amount of routing significance. Also, unlike Host Identifier, A_v has no cryptographic significance and therefore, does not need a HIP-protocol with its base exchange and also, does not provide any protection against DoS attacks. Benefits of HIP are realized only when both communicating end-points support HIP-base exchange and packet payload. On the contrary, MOBILENAT works well with non-mobile and mobile end-hosts with and without MOBILENAT support. Also, to gain full benefit of HIP, new infrastructure in the form of rendezvous servers and Public Key Infrastructure (PKI) or Domain Name Servers (DNS) that store HIs are required. MOBILENAT too requires new network element in the form of Mobility Manager (MM), modified DHCP server that stores/allocates A_v and modified Anchor Node (AN). The Mobility Manager performs functions somewhat analogous to the rendezvous server. In summary, the HIP architecture represents a research effort for next-generation internet whereas benefits of MOBILENAT can be realized in short-term in the current internet.

VIII. CONCLUSIONS

New wireless access technologies such as 802.11 networks will enable large scale deployment of public hotspot and wide-area wireless data networks and rapid growth in mobile end-devices. For most of these devices, transiently allocated IP addresses instead of permanently assigned Home IP addresses will be sufficient under most common circumstances. Given this, public networks may use private addresses coupled with NATs to tackle the explosion in the number of IP devices. In this paper, we propose a new scheme called MOBILENAT for supporting efficient micro and macro mobility of devices across such private and public heterogeneous address spaces. Our technique uses a fixed unique virtual IP address for host identification and a dynamic, unique actual IP address for routing within the domain. Address translation is performed both at the client in the shim-layer and at the NAT device in the Anchor Node (AN). We use existing protocols like DHCP and ICMP for signaling and introduce a domain wide mobility manager to co-ordinate inter-domain and intra-domain mobility. Our use of MIDCOM framework allows separation of control path functions in the mobility manager from the per-packet address translation in the NAT device.

Our scheme introduces a novel concept of dynamic home agent on a per-connection basis. The biggest advantage of our scheme is that unlike several micro-mobility schemes reported in literature, it does not require any change in the routing infrastructure in the domain or does not need any foreign agent. It co-exists with Mobile IP and is easy to deploy. Our prototype implementation of MOBILENAT architecture clearly demonstrates feasibility of our architecture.

IX. ACKNOWLEDGMENTS

Salim Virani helped with the Windows intermediate driver. Luca Salgarelli provided insights on the Linux netfilter and iptables modules. Girish Chandranmenon helped in the TCP/IP checksum computation. Henning Schulzrinne provided valuable comments on the initial architecture.

REFERENCES

[1] TIA/EIA/IS-835B - cdma2000 Wireless IP Network Standard. , Third Generation Partnership Program 2 (3GPP2), 2000.

[2] General Packet Radio Service (GPRS) Service Description (Stage 2). TS 122 060, ETSI, 2002.

[3] http://udhcp.busybox.net/. Technical Memorandum, April 2002.

[4] M. Buddhikot, G. Chandranmenon, S. Han, Y. W. Lee, S. Miller, and L. Salgarelli. Integration of 802.11 and Third Generation Wireless Data Networks. In *IEEE INFOCOM 2003*, April. 2003.

[5] A.T. Campbell and J. Gomez. IP Micro-Mobility Protocols. *IEEE Wireless Communications*, 4:45–54, October 2001.

[6] J. Chiappa. Endpoints and Endpoint Names: A Proposed Enhancement to the Internet Architecture. Work in progress, 1999. http://users.exis.net/ jnc/tech/endpoints.txt.

[7] R. Droms and Ed. W. Arbaugh. Authentication for DHCP Messages. RFC 3118, IETF, June 2001.

[8] C. Perkins (Editor). IP Mobility Support for IPv4. RFC 3344, IETF, August 2002.

[9] A. Misra et al. IDMP-based Fast handoffs and paging in IP-based cellular Networks. *3GWireless 2001*, May 2001.

[10] A. T. Campbell et al. Comparison of IP Micro-Mobility Protocols. *IEEE Wireless Communications*, 9:54–64, Feb 2002.

[11] F. Vakil et al. Supporting Mobility for TCP with SIP. draft-itsumo-sipping-mobility-tcp-00.txt, IETF, December 2001.

[12] J. Rosenberg et al. ed. Session Initiation Protocol. RFC 3261, IETF, June 2002.

[13] E. Gerich. Guidelines for Management of IP Address Space. RFC 1466, IETF, May 1993.

[14] E. Gustafsson, A. Jonsson, and C. Perkins. Mobile IPv4 Regional Registration. Work in progress -Internet Draft, November 2003. draft-ietf-mobileip-reg-tunnel-08.txt.

[15] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. RFC 2401, IETF, November 1998.

[16] H. Levkowetz and S. Vaarala. Mobile IP Traversal of Network Address Translation (NAT) Devices. RFC 3519, IETF, April 2003.

[17] R. Moskowitz and P. Nikander. Host Identity Protocol Architecture. Work in progress -Internet Draft, September 2003. draft-moskowitz-hip-arch-05.txt.

[18] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson. Host Identity Protocol. Work in progress -Internet Draft, October 2003. draft-moskowitz-hip-08.txt.

[19] P. Bhagwat and C. Perkins and S. Tripathi. Network Layer Mobility: An Architecture and Survey. *IEEE Personal Communications*, pages 54–64, 1996.

[20] Charles Perkins. IP encapsulation within IP. RFC 2003, IETF, October 1996.

[21] R. Ramjee, L. Li, T. F. La Porta, and S. Kasera. IP paging service for mobile hosts. In *Mobile Computing and Networking*, pages 332–345, 2001.

[22] R. Ramjee, K. Varadhan, L. Salgarelli, S. R. Thuel, S. Wang, and T. La Porta. HAWAII: A Domain-Based Approach for Supporting Mobility in Wide-Area Wireless Networks. *IEEE/ACM Transactions on Networking*, 4:45–54, June 2002.

[23] Y. Rekhter, B. Moskowitz, D. Karrenberg, G.J. de Groot, and E. Lear. Address Allocation for Private Internets. RFC 1918, IETF, February 1996.

[24] W. Simpson. IP in IP Tunneling. RFC 2003, IETF, October 1995.

[25] P. Srisuresh and K. Egevang. Traditional IP Network Address Translator (Traditional NAT). RFC 3022, IETF, January 2001.

[26] P. Srisuresh, J. Kuthan, J. Rosenberg, A. Molitor, and A. Rayhan. Middlebox communication architecture and framework. RFC 3303, IETF, August 2002.

[27] G. Su and J. Nieh. Mobile Communication with Virtual Network Address Translation. *Department of Computer Science, Columbia University*, CUCS-003–2, Feb 2002.

[28] R. P. Swale, P. A. Mart, P. Sijben, S. Brim, and M. Shore. Middlebox Communications (midcom) Protocol Requirements. RFC 3304, IETF, August 2002.

[29] A. G. Valko. Cellular IP- A New Approach to Internet Host Mobility. *ACM Computer Communication Review*, Jan 1999.