Telecommun Syst (2013) 53:315–328
DOI 10.1007/s11235-013-9700-7

A conflict-insensitive NATed roaming framework using NAToD for proxy mobile IPv4 in WLANs

Wen-Kang Jia · Yaw-Chung Chen

Published online: 3 July 2013

Abstract Providing efficient mobility management in the current Internet is increasingly important due to the quick growth of wireless mobile users. The emerging Proxy Mobile IPv4 (PMIPv4) technique brings a possible solution for that purpose. Since NAT function is widely adopted in IPv4 environment nowadays because of lacking IPv4 addresses, the PMIPv4 interoperating with NAT must be considered. Unfortunately, owing to the possible conflict of private IP address, we encounter a problem in broadcasted point-to-multipoint wireless networks such as IEEE 802.11 networks. To address this issue, we proposed a novel Network Address Translation on Demand (NAToD) scheme, which can well interoperate with the PMIPv4 solution. With our scheme, single public IPv4 addresses can be shared by multiple mobile nodes in both home and foreign networks, low-latency handoff can be achieved, deployment cost can be reduced, and software upgrade can be avoided for mobile nodes in wireless LANs. Our work allows mobile users in WLAN to access Internet based on the advantages of both PMIPv4 and NAT.

Keywords Proxy Mobile IPv4 (PMIPv4) · Network-based localized mobility management (NetLMM) · Network Address Translation on Demand (NAToD)

W.-K. Jia (✉) · Y.-C. Chen
Department of Computer Science,
National Chiao Tung University, 1001 University Road,
300 Hsinchu, Taiwan, ROC
e-mail: wkchia@cs.nctu.edu.tw

Y.-C. Chen
e-mail: ycchen@cs.nctu.edu.tw

1 Introduction

With the quick advance in wireless Internet technologies, more and more IP-based user equipments are becoming mobile, and providing mobility support in the IP networks has been a long-standing challenge. The motivation of this work was to design a feasible technique that continues an IP session when a host has to change its IP address while in moving. The Proxy Mobile IPv4 (PMIPv4) [27] solution is firstly developed for several wireless wide-area networks. Indeed, the PMIPv4 and PMIPv6 [14] protocol is adopted as part of them (e.g. WiMAX [20], 3GPP LTE, 3GPP2 HRPDA [2], and so on). Unless the IPv6 has been widely deployed, it is essential to support mobility for IPv4 mobile nodes. In addition, mechanisms for dealing with overlapped private IPv4 addresses of mobile nodes and supporting separation of flows between the key components of PMIPv4, such as *Proxy Home Agent* (*PHA*), *Access Router* (*AR*), and especially *Mobile Node* (*MN*) are also required.

Nowadays *NAT* (*Network Address Translation*) [8, 35] mechanism has been widely adopted as a solution to accommodate the IPv4 address shortage problem [13], which is typical in PMIPv4 environment. In order to cope with this problem, most of the mobile devices connect to the Internet through the NAT mechanism. In case that *Care-of-Addresses* (*CoAs*) or *Home Addresses* (*HoAs*) are assigned in the private IP address space, problems regarding address overlapping and NAT traversal are likely to appear in a broadcasted wireless LANs [10], such as IEEE 802.11 network. Although these problems can be overcome by management in a local mobility environment, it is still a tough task to establish a global mobility environment.

In order to solve the IP conflicting problem in PMIPv4 that inter-operates with NAT in WLANs. We proposed an extension to PMIPv4 by integrating the *NAToD* (*NAT on de-*

mand) functions into the home agent. The topic is an important issue when convergence of wireless networks is becoming a reality.

2 Problem description and related works

2.1 Network-based local mobility management

The IETF has defined several client-based (host-based) mobility management protocols that intend to handle IP mobility for MNs. All IP mobility management protocols defined thus far require the involvement of IP layer in the MN. A variety of solutions, such as IETF Mobile IPv4 (MIPv4) [32] and Mobile IPv6 [21], Hierarchical Mobile IP [38] and its extension for the Regional Paging [15], Fast Handoff [25, 26], Cellular IP [5], HAWAII [33] and EMA [31] have been proposed. Given all these efforts, however, pervasive mobility service on the Internet anytime and anywhere is still not mature. Thus, integration and improvement of dissimilar and practicable mobility solutions becomes an essential issue in next-generation IP-based wireless networks [4].

Each of the solutions mentioned above has common problems. For example, MIPv4 incurs large handover and end-to-end latency, which makes it hard to support real-time multimedia applications. With MIPv4, network operators, mobile users and communication peers also need to upgrade their equipments to enable mobility support. Making such coordinated deployment across administrative boundaries has been proven to be an arduous task [22]. A satisfactory next-generation mobility management solution should solve all these major problems to get acceptance.

The Network-based Local Mobility Management (NetLMM) [19, 22, 23] working group of IETF has tasks in defining a self-titled protocol, in which local mobility is handled by network side without involvement of the MN. The idea is that a MN can move across multiple access routers without encountering a change in its IP address. Further, the NetLMM provides mobility support to a MN within a restricted portion and topologically localized network, and MN does not need to participate in any mobility related signaling. In other words, the NetLMM enables a mobility environment for all IP-based wireless equipments which lack built-in mobility capability, thereby hiding the mobility of the IP layer and higher layers.

An additional goal of NetLMM is to simplify the deployment, integrate with and enhance existing solutions if suitable, to the mutual benefit of service operators and end users. The key benefits of NetLMM include: decrease the complexity of MNs, enhance the capability for mobility, speedup the handoff procedure, and reduce the air-link consumption, etc. [23]. Such concept brings up PMIPv4 and PMIPv6 in addition to the legacy client mode (host mode) Mobile IP (CMIP) [1, 10].

2.2 Proxy mobile IPv4 (PMIPv4)

IPv6 is considered as the only practical long-term solution for IPv4 [8] address exhaustion problem, but before widespread deployment of IPv6, mobile users still suffers both the severest IPv4 address shortages in the wireless Internet and complex, impractical client based mobility management schemes. Extensions of PMIPv4 is therefore proposed to allow MIPv4 protocol operating within the network and enabling IPv4 hosts to roam without MIPv4 support, while the dedicated network entities provide mobility support on their behalf [2, 27]. The required mobility procedures are handled by the *Proxy Mobile Agent* (*PMA*), a new mobility entity in the wireless access network. It performs location registration and update analogous to regular MIPv4 procedures, but strictly omits any involvement of the MNs.

The PMA resides in (integrates with) the first AR or Base Station (BS) perceived by the MN, it is similar to *Foreign Agent* (*FA*) with DHCP [11] function. The PMA operates in the following manner: it detects an MN that is attached to the network and triggered by the regular network access procedure, initiates Mobile IPv4 registration with the HA on behalf of the MN. The basic PMIPv4 architecture and its handoff procedure are shown in Fig. 1 and Fig. 2. The PMA operation is triggered by the DHCP request message originated from MN, it then sends a *Proxy Registration ReQuest* (*PRRQ*) [27] message to the associated HA. The PRRQ contains the *pCoA* (*Proxy Care of Address*) of the serving PMA (collocated in FA in this case) and the original IP address of the MN. Therefore, the HA sets up the mobility binding entry for the MN after being assigned a HoA, then the HA will return the previously assigned HoA through the *Proxy Registration RePly* (*PRRP*) [27] message to the PMA. Otherwise, the HA may deny the registration because it is administratively prohibited. After the PRRP procedure, the PMA provides the IP address (HoA) to the MN in DHCP response, and establishes a bidirectional IP-in-IP tunnel [37] between the HA and PMA (using the IP address of the PMA as MN's pCoA). Afterwards, all incoming packets from the MN are intercepted, encapsulated and forwarded to the HA via the tunnel by PMA, which de-encapsulates the packets heading to the MN and delivers them using layer 2 forwarding. The registration procedure repeats whenever MN moves into the domain of another PMA; HA relocates the tunnel towards the target PMA and terminates the previous tunnel in parallel. The MN always maintains the same HoA during a session connection between itself and the CN. It is even unaware of its movement.

Finally, when the mobility binding entry for a MN in HA is updated, the HA may send a *Registration Revocation* [27] to the previous serving PMA (i.e. specific to the Foreign Agent entity) in order to reclaim unused resources in an expeditious manner, then the previous serving PMA

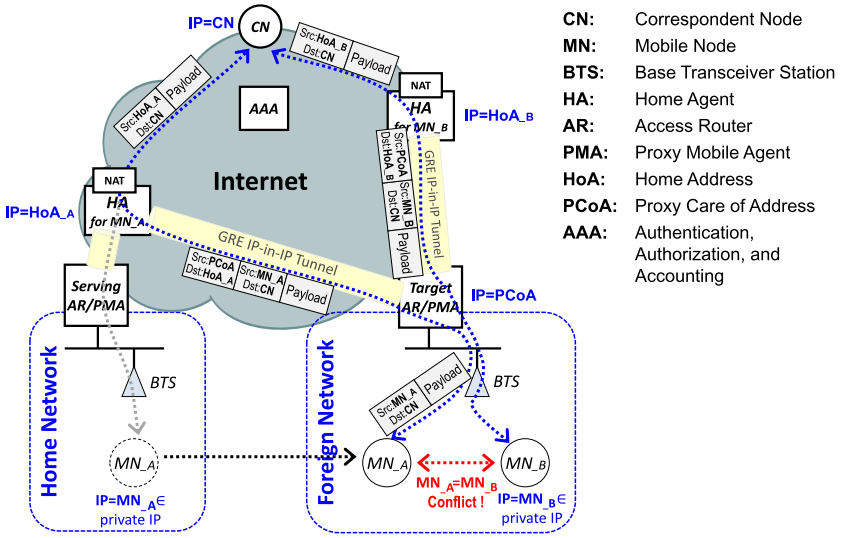

Fig. 1 The architecture of PMIPv4 with NAT support

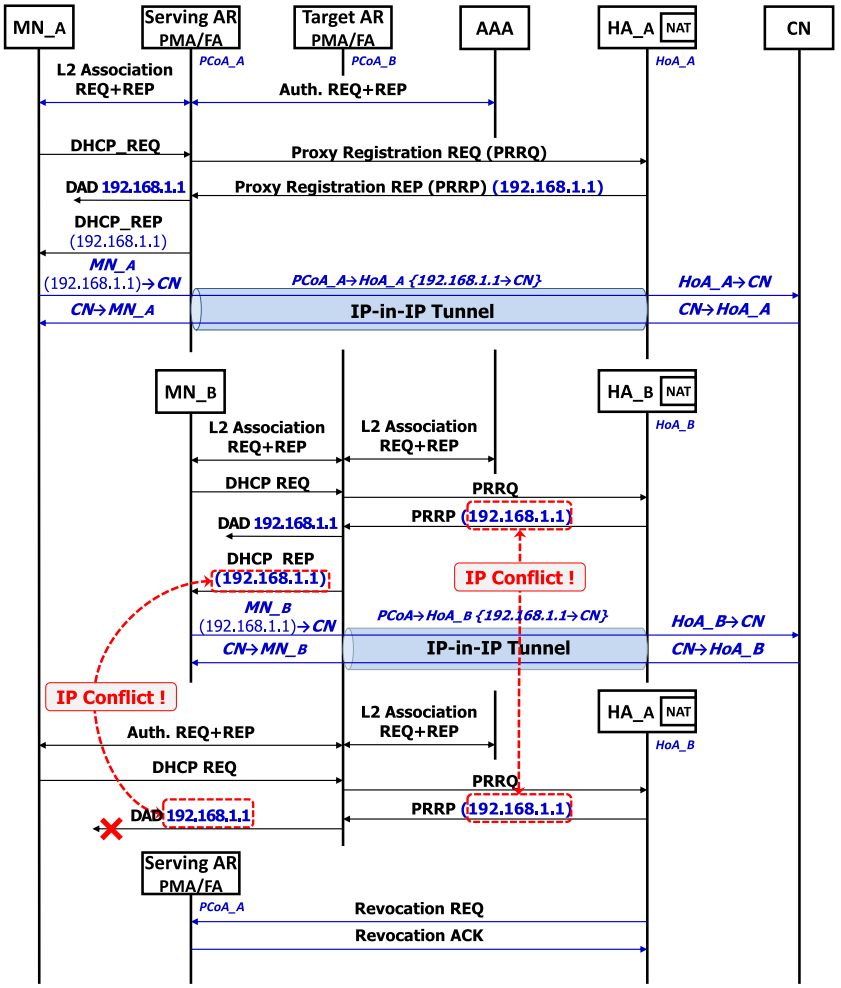

Fig. 2 PMIPv4 handoff procedure and its duplicated IP problem on NAT environment

sends revocation acknowledgement to the HA. Now the target PMA acts as the serving PMA to replace the previous one.

The PMIPv4 described in [27] asserts several benefits: additional mobility support for unmodified hosts; reduction of the handoff signaling transmitted over the wireless link

and in the network, and the support for heterogeneous handoffs. Since there is no route optimization in IPv4 networks, all the traffic to/from the MN always goes through the HA, even when the CN and MN are in the same subnet. It remains identical with host-based MIPv4 except that the mobility signaling is no longer required on the air links. Extensions to the Mobile IPv4 registration and reply messages are needed to accommodate location change. This approach introduces additional tunneling overhead, and requires considerable extensions to the HA [27].

2.3 IP conflict problem description of NAT over PMIPv4

Before the widespread IPv6 deployment by *Internet service provider* (*ISP*) to provide sufficient address space, NAT was devised as a short-term solution to the IPv4 address exhaustion problem [8]. As the name implies, it translates an IP address from a private space into the public address used on the Internet, so that more internal hosts can access external networks with fewer public IPv4 addresses by repeatedly using private IP addresses. NAT functionality can be built into a device such as a router that sits between an upstream provider (e.g. an ISP) and a local network. An even denser deployment of IPv4 NAT is perceived nowadays, and NAT is still the essential technique for extending the life cycle of IPv4 [8].

A MN behind the NAT can only play the client role, as opposed to either a client or server in the end-to-end model that characterized the original Internet. Also due to the quick advance in broadband technologies and media streaming applications nowadays, the NATs may become the performance bottleneck of the common Internet access [8].

PMIPv4 is designed for newer non-broadcasted *Wireless Wide Area Network* (*WWAN*) and *Wireless Metropolitan Area Network* (*WMAN*) environments such as IEEE 802.16 [2, 28]. The point-to-multipoint broadcasting control message is transmitted in block in the above type of networks, in which the layer-2 point-to-point connection-oriented service is mandatory between BS and the MNs [30]. It means no ARP [6] broadcasting among MNs, and even the private IP address conflict detection [9] in layer-3 can be ignored. The BS can recognize the individual layer-2 connection whether it comes from foreign MNs or home MNs by *Connection IDentifier* (*CID*) and handle them in appropriate data paths respectively [18, 28]. This overcomes the obstacle of PMIPv4 working with NAT in WWAN environments.

However, if we deploy the PMIPv4 with NAT in a broadcasted point-to-multipoint WLAN environment (e.g. IEEE 802.11), challenges may occur. Firstly, the MN may use the private IP address as a HoA. Since private IP address can be repeatedly used in different domains, once a MN moved to a foreign network, it was possible that the foreign network had used the same private IP address as MN's. This may cause two problems: (1) the foreign MN's HoA (a private IP address) conflicts with the home MN's HoA, or conflicts with another foreign MN's HoA, especially the network administrators are used to configure the similar private IP subnets (e.g. 192.168.0.x). Therefore all of them will receive warning message of duplicated IP; (2) Even if the BS attempted to filter the ARP broadcasting across the MNs, it is still hard for the PMA (FA) to distinguish the connection of foreign MN from that of home MN, because they have same private IP address. Although PMA still can distinguish the connection by either MAC address or other manner, it will increase the complexity of the PMA. According to the above discussion, the NAT conflict problem is still an open issue for PMIPv4 environments [10].

Practically, in order to avoid such situation, a *Duplicate IP address detection* (*DAD*) [9] procedure has been implemented. Once a duplicated IP was detected, the recursive procedures (including PRRP, PRRQ and DAD) continue until the address conflict is eliminated. These procedures introduce a large number of network attachment and long handoff latency.

This work is motivated by the issues described above, we take the development of the PMIPv4 with NAT services in IEEE 802.11 infrastructure mode as an example, our work can be further modified for different WLAN environments, and this part is left as the future work.

3 Proposed scheme

3.1 Architecture and operation of NAToD

The *Network Address Translation on Demand* (*NAToD*) mechanism is first addressed in [7] as a substitute for traditional NATs. Its original goal is improving the packet transmission performance for the Internet access. Due to its remarkable properties, NAToD becomes a promising candidate for NAT applications.

The NAToD is a cross-layer network function working in both data-link and transport layer (it can be said to bypass the network layer). The NAToD works in bridge mode, and does not take any IP address (including public and private IP address). The default gateway of endpoints in the internal network also points to the router through NAToDs, which neither translate any IP address nor modify any network layer header. Since the internal hosts have already used the external unique public IP address directly, every internal host uses the duplicate public IP address repeatedly. Therefore the NAToD cannot distinguish IP addresses of the internal hosts, and all packets will be using the 48/64 bits source MAC address to tell them from each other instead. The original idea in this design regards that traditional NATs

must always resolve the source/destination IP address and source/destination port, lookup the translation table, update translation table, replace source IP and/or source port, recalculate the checksum and send the packet out. It will waste a lot of processing time and cause NATs the bottleneck of packet flows. In NAToDs, it has the same procedure for checking each packet, but only a small portion of packet headers needs to be translated. The probability of *Translation on Demand* (*ToD*) for a port number is very low. It is needed only when the so-called "*Source Port Collision* (*SPC*)" occurs: Multiple endpoints in the internal network connect to the same external host and access the same service port at the same time; and coincidentally, two or more of the randomly selected source port numbers for these internal hosts happen to be the same. In such case, the collided source port number of the latter session should be translated to a new randomly-selected free port number in NAToD, and added to the translation table to accomplish correct translation for return packets. Theoretically, the probability of SPC is less than $1/2^{16}$ when two internal hosts are connecting to the same external host and the same service port. This probability will increase when the number of internal host increases, or the internal hosts have frequent access to a specific external host. If there were 256 hosts in the internal segment, the SPC probability is still less than $1/256$. In most cases, multiple internal hosts won't open the same source port for transmitting their packets during the same time period. In other words, more than 99 % of the sessions/packets are transparent to the NAToD and do not need IP header translation. The majority of the packets do not need translation and recalculation of the checksum. By simplifying the translation process using NAToD mechanism, we are able to improve the throughput and reduce the forwarding latency of NAT services.

For example, when the first packet is sent by internal host A, the packet will obtain a randomly selected port number 1024, pass through the NAToD without changing its IP header and arrive at the external host. Similarly, the packet responded by the external host will get through the NAToD without translation and reach the internal host A. While this session is active, internal hosts B using the same IP address as host A, which sets up a connection with same external host too. The source port number used by B is selected randomly as 1025. It could be distinguished from the connection using 1024 as source port number before by recording them in the NAToD. Still, it does not need any translation for setting up the second record in the NAToD translation table and transmitting this packet to the external host. When an internal host C tries to set up the third connection with same external host, the NAToD will check two existed records of the internal network. If it conflicts with the second record, the returned packet won't be able to reach the correct internal host C. In order to solve this problem, it must replace the conflicted source port number from 1025 to, for example, 9523 (randomly generated), so that those return packets destined to client C can alleviate confliction and reach the correct destination. It is similar to the traditional NAPT, but most of the time it does not need to change the IP header. Since the port number of the source has already been changed, the IP header checksum of the packet must be recalculated.

On the design of the NAToDs, the function of proxy ARP [6] is required, and network administrators must ensure that the MAC address of any host in both internal and external networks is unique (MAC addresses should inherent unique). When an internal host broadcasts the ARP request packet to look for the external host, the NAToD will forward the packet to the external network and vice versa: when the ARP reply packet comes back, the NAToD forwards it to the internal network too.

3.2 PMIPv4 with NAToD initial network attachment

Due to the special property of NAToD, it is an inherent cross-layer fast handoff mechanism for IPv4 mobility management, especially it cooperates with PMIPv4. Our design integrates the NAToD functions within the HA in PMIPv4 networks as presented in Fig. 3. Thus, every bidirectional traffic flow from MNs will pass through and be processed by HA with NAToD entity, also all addresses acquisition procedure will be handled by HA.

Basically, our approach does not affect the original network attachment procedure of PMIPv4. There are three distinct phases: Firstly, the MN establishes L2 link with the base station (e.g. access point in IEEE 802.11, not shown) and performs access authentication/authorization with the PMA/AR. In this phase, the MN may perform the EAP [3] (e.g. PPP [36] or IEEE 802.1x [17]) between the ARs. The AR acts as the *NAS* (*Network Access Server*) in this phase. Therefore, the AR makes exchange of *Authentication, Authorization, and Accounting* (*AAA*) messages within the network management infrastructure to perform authentication and authorization for the MN. As part of this phase, the AAA server may retrieve some information such as user's profile, handset type, assigned home agent address, and other capabilities of the MN.

Secondly, the MN attempts to obtain an IP address via a DHCP or PPP. This triggers PMIPv4 which assigns/authorizes the IP address and handles forwarding between the PMA and HA. Specifically, the DHCP client (built-in on the MN) sends the DHCP discovery message to the DHCP relay agent (built-in on the PMA) or DHCP server, the DHCP relay agent or DHCP server will send the DHCP ACK message to the DHCP client after PMIPv4 signaling has been completed.

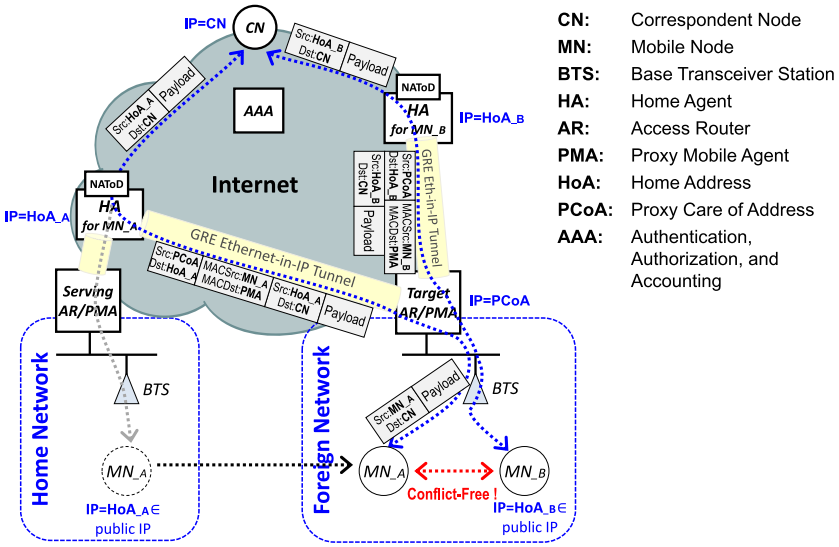

Fig. 3 The architecture of PMIPv4 with NAToD support

However, in order to speed up the network attachment and handoff process, the address acquirement procedure may be omitted: let's statically assign public IP addresses to each mobile node, thus the DHCP or PPP procedures are no longer required. In addition, since the conflict problem is inherent in the NAToD and ignored, the DAD procedure will be no longer required. As a consequence, both the network attachment and handoff latency could be significantly decreased through this way.

Thirdly, when the previous phase (DHCP or normal datagram) is completed, the PMA sends a PRRQ message to the HA. The PRRQ contains the pCoA of the serving PMA, HoA and MAC address of the MN in our manner. Therefore, The HA sets up the mobility binding entry for the MN after assigning a former HoA, note that HoA is a duplicate public IP address on NAToD mechanism. The HA may also assign a GRE [12] key to PMA in this phase (if GRE tunneling is used between the PMA and HA). If the request is authorized, both configuration parameters of MN and PMA can be carried by the PMIPv4 messages.

The HA will return the HoA and the GRE key through the PRRP message to the PMA, if the registration is permitted. Then the PMA provides the IP address (HoA) to the MN in DHCP response, and the forwarding path (tunnel) for the HoA between the PMA and HA is established [13]. Note that the MAC address of MNs will also be included in this tunneling protocol, so the tunnel type is transparent Ethernet bridging (0x6558) instead of the IP-in-IP (0x0800) as in the conventional PMIPv4. At this step, the MN's IP protocol stack is still configured as the original HoA that has a tunneling between the AR/PMA and HA. Thus MN can access the Internet through this duplicated HoA that is shared with other MNs located in anywhere via NAToD mechanism. All IP address translation procedures are completed in HA. Regardless of whether the PMA belongs to home network or visited network, the initial network attachment procedure is similar to that mentioned previously.

3.3 PMIPv4 with NAToD handoff

When a base station detects that a MN has moved into the visited network, authentication and authorization will be performed again firstly when MN leaves the serving AR and attaches to the target AR in the foreign networks. The successful authentication or first datagram will trigger the PMIPv4 signaling. The target PMA sends a PRRQ to the HA. The PRRQ contains the pCoA of the target PMA and the MAC address of MN. Afterwards, The HA updates the existing mobility binding entry for the MN and returns the original HoA fetched from the binding entry to the target PMA through the PRRP message.

The MN's HoA can also be statically assigned or obtained through the aforementioned DHCP method; note that the HoA is never modified as that in NAToD architecture. In general, The MN's IP protocol stack may detect layer-2 link down and up after the access re-authentication, and attempt to validate its IP address connectivity by DCHP, gratuitous ARP [6, 9] or ICMP. In the last phase, the forwarding path between target AR and HA is set up for the MN to send and receive IP packets using the same HoA anchored to the HA.

Since the visiting MNs carry their HoAs when they join a foreign network which is also a NAToD environment, the IP conflict won't be a problem because the AR never cares about it. Besides, with the NAToD, both the home and foreign networks are native public IP conflict environment, and private IP address is not taken in NAToD function. PMA can easily distinguish the traffic of visiting and home MNs by

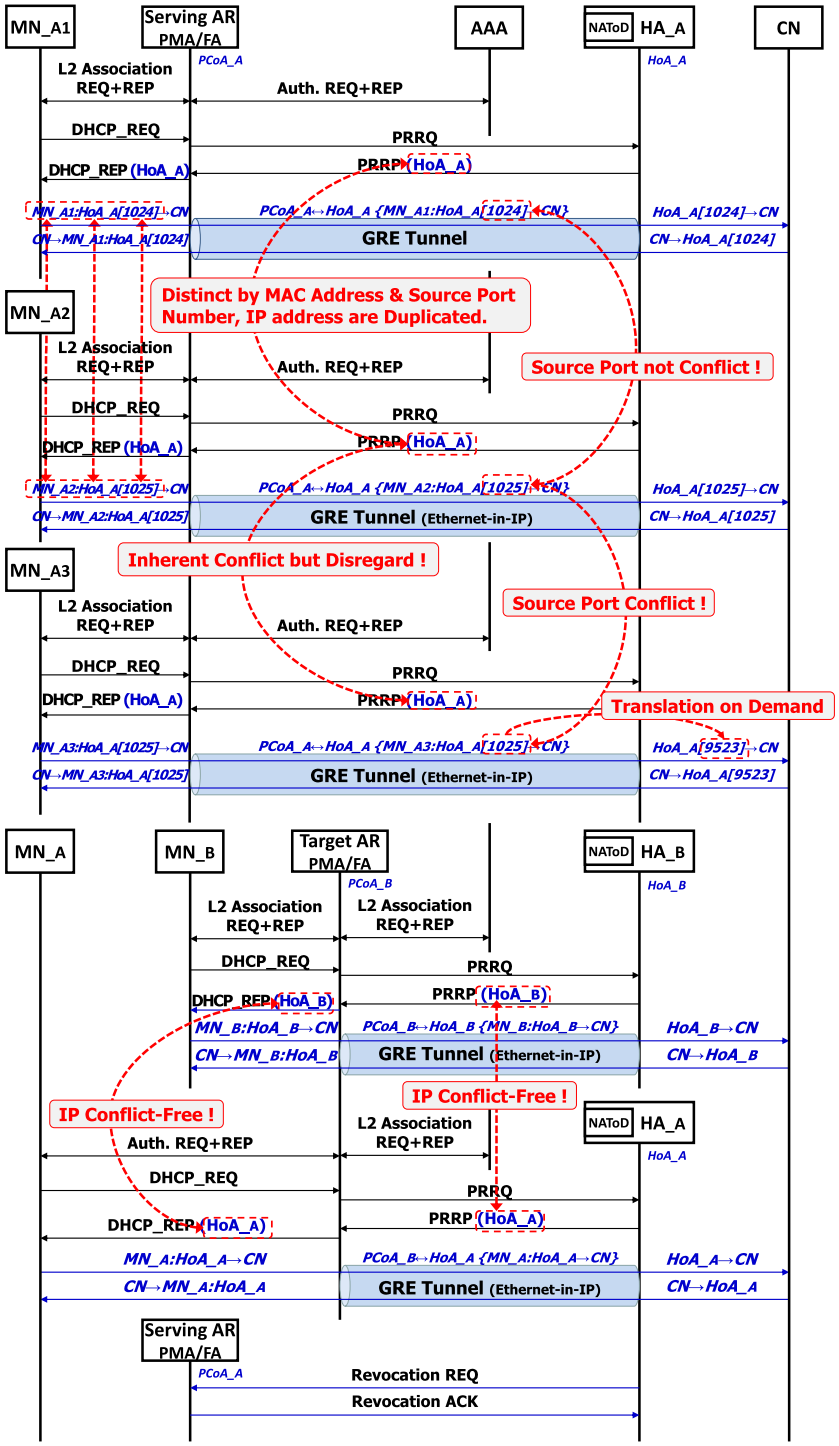

Fig. 4 The handoff operations of PMIPv4 with NAToD

their source IP address (HoAs). All traffics originated from the visiting MNs will be quickly recognized and forwarded to the corresponding HA through Ethernet-in-IP tunneling by the target AR. Since the HA's IP address is exactly the same as the visiting MN's HoA, the target AR acquires the HA's location information directly without needing any extra procedure.

Through this method, a MN can continue to communicate with CNs during handoff by using its unmodified HoA which is shared with other MNs. The IP sessions between the MNs and CNs can still be kept alive because TCP/UDP binding information has never been changed in both sides. As a result, NAToD could be a feasible NAT solution for PMIPv4.

4 Performance analysis

In the foreign network environment which needs a lot of connections to home network in the PMIPv4 mobile management domain, our proposed scheme feature three contributions: firstly, it reduces the amount of the IPv4 address usage and prevent address conflict problem from happening; secondly, it reduces the time for address acquiring, duplicated IP checking. Even the ARP table checking time could be omitted when the proposed scheme is deployed on PMA, thus the handoff latency can be reduced; thirdly, it also reduces the time of NAT packet checking, IP header replacement, IP/TCP/UDP header checksum recalculation. ARP table checking could also be omitted too when our proposed scheme is embedded in a HA, thus the NAT packet forwarding latency can be reduced. These features of PMIPv4 are based on NAToD mechanism with simple data structure, fast handoff latency, and low processor loading.

4.1 Usage of IP addresses

We firstly analyzed the IPv4 address usage of the proposed scheme, and compare it to the original PMIPv4 scheme. Our proposed scheme uses only one address in the home network and this address is occupied by the home agent. All MNs will share this address for communications everywhere, even if they roam to the foreign networks. Consider that a real environment usually adopt an IP address pool to a single NAT equipment, the maximum address occupancy is bounded by $O(N(h) + N(a))$, and the minimum address occupancy is bounded by $o(N(H))$, where $N(h)$ denotes the number of home agents, $N(a)$ denotes the number of PMAs/ARs, and $N(H)$ denotes the number of home networks existing in access networks.

Contrarily, with original PMIPv4 scheme, the maximum address occupancy is bounded by $O(N(h) + N(a) + (1 + \omega)N(m))$, where $N(m)$ denotes the number of active mobile agents, and ω denotes the average roaming frequency factor of mobile agents.

We also investigate the impact of the end-to-end throughput during the continuous movement of mobile nodes. We setup an arbitrary roaming PMIPv4 network environment with 20 home networks, which are constructed as hexagonal cells. Each home network has one home agent, one AR, and 20 MNs. All MNs are set to arbitrarily roam within adjacent six cells with different roaming frequency factors which is the percentage of roaming MNs. MNs move to a foreign networks, stay there for a certain period of time and then move again; the handoff occurs randomly, and the period length is normally distributed. The model is more suitable to the movement pattern in mobile networks that may be typical in future Internet.

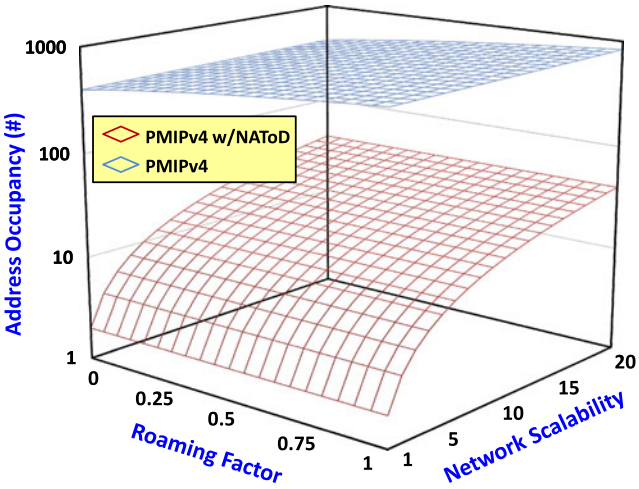

Fig. 5 The address occupancy between PMIPv4 and the proposed scheme

Figure 5 shows the address occupancy of the proposed scheme and original PMIPv4 in different network scalabilities and roaming frequencies. Compare to our proposed scheme in high roaming frequencies, the address occupancy of the original PMIPv4 is several-fold. Obviously, we can observe that massive IP addresses are saved by our proposed scheme in a global PMIPv4 network. This may be one of the most significant contributions of our study.

4.2 Handoff latency

We firstly conducted an experiment to observe the handoff latency variation when a MN attaches to a PMA. The experiment environment is organized as in Fig. 6: Two Access Points (APs) are running in IEEE 802.11g mode and connected to different LMAs. The two LMAs belong to different manage domains and connect to an upper tier router by Ethernet. HA is located in the same subnet with LMA1 and AP1. MN moves and performs handover from AP1 to AP2. The CN is connected to the upper tier router with two hop distance and also connected by Ethernet. The layer-3 handoff procedure is initiated immediately after layer-2 handoff procedure, which costs 500 ms. During the handoff, the MNs send constant bit rate UDP traffic to CN at about 500 Kbps sustained rate with 1280 byte UDP packet and 20 ms inter-packet duration. We use sniffer to observe the sequence number of UDP packets which reflect the service disruption period during handoff.

The results are shown in Fig. 7. Since both the address acquisition and duplicated address detection procedures are omitted in the proposed scheme, the handoff latency is about 510 ms only, which is almost the same as

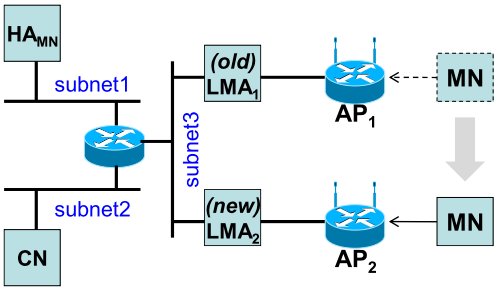

Fig. 6 Network topology under consideration

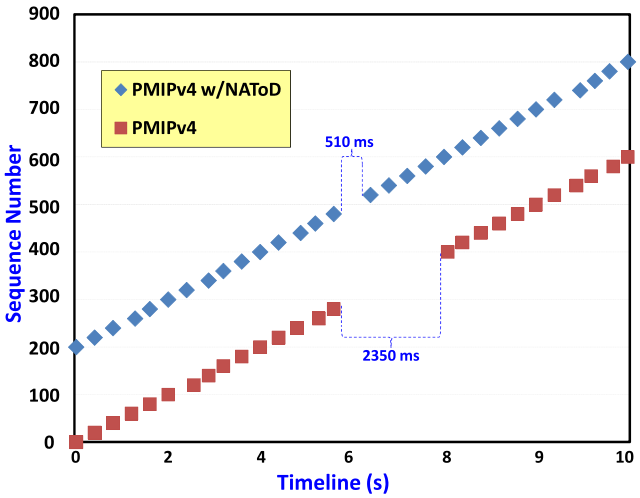

Fig. 7 Service disruption during handoff period

that of layer-2 handoff. In contrast, the original PMIPv4 needs a long period RTT between HA and PMA, for waiting the PRRP in the address acquiring stage, and a long period to ensure the availability of designated address in the DAD stage, thus it totally needs 2350 ms during a layer-3 handoff. Obviously, there is a big gap between our proposed scheme and the original PMIPv4 in handoff latency.

In the proposed scheme, two modes are used to perform layer-3 network attachment: static assignment and DHCP. Notes that results are based on IP static assignment manner, once DHCP is employed, the handoff latency of the proposed scheme will be decreased to similar with original PMIPv4's results because both address acquisition and DAD procedures are still required.

4.3 Lookup performance of translation table in HA

As described before, during the process of IP address translation the NATs should dynamically set up a NAT translation table (NATTT) for both traffic directions. Once an outgoing packet arrives, the NAT must check the corresponding session packets with NATTT's entries. If there is a match, it will replace the source IP and port number that has already been assigned before. Otherwise it will assign a new source IP and port randomly, and add this new record to NATTT, then forward the packet to the external port. Once an incoming packet arrives, the same checking procedures will be performed; if hit, recover the destination IP and port number from packet according to the original mapping, then forward the packet to the internal port; otherwise, drop the packet. All operations mentioned above are processed in IP layer. Before the packets are really forwarded to the physical network interface, the layer 2 encapsulation is necessary, and the corresponding MAC address will be fetched from ARP table. Therefore, it needs to lookup table at least twice in the traditional NAT operation.

The NAToD should also dynamically set up a set of NAToD translation mapping table, abbreviated as NAToDTT, during the process of translation. The purpose and functionality of the NAToDTT are same as NATTT, but the data structure of the table is different. For outbound packets, the NAToD will check the corresponding session packets with NAToDTT entries. It checks fields of source MAC address, destination IP, local source port and destination port number in the table. If a conflict is detected, it will assign a global source port randomly and record it in NAToDTT to accommodate its return packet. For those inbound packets, the NAToD will perform the same checking procedure to see whether the address has been translated before or not. If yes, it uses the original source port and MAC address for transmitting the return packet. Otherwise it will drop the packet. All translations will be performed on demand by NAToD while forwarding the packets. It does not have to deal with the binding problem of the IP layer and MAC layer. We can consider this design as combining ARP table and NATTT into one NATODTT. In Fig. 8, the NATTT entries have at least 18 bytes including 4-byte local source IP address, 4-byte destination IP address, 2-byte local source port number, 2-byte destination port number, 4-byte global source IP address, and 2-byte global source port number. It also shows the corresponding 4-byte local IP address and 6-byte MAC address in the ARP table.

In addition, the NAToD can lookup NAToDTT and find MAC address for the transmission of the packets directly. It needs neither to check the ARP Table nor to re-encapsulate MAC address. In fact, the ARP Table will be used only for connecting to the external network, but not the internal network. From the data structure point of view, both the space complexity and time complexity of NAToDTTs would be better than that of NATs.

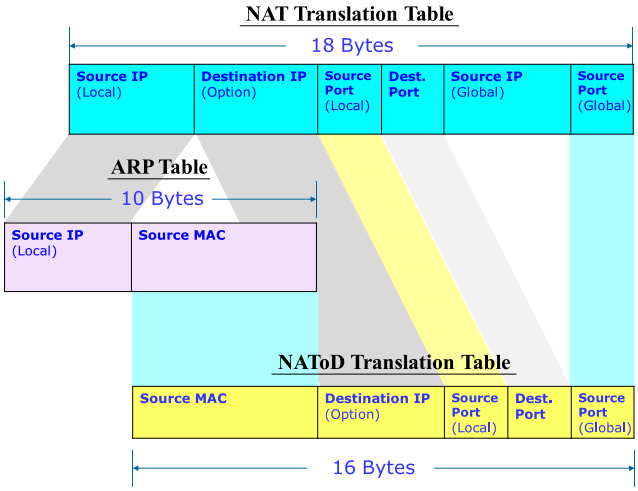

Fig. 8 Comparison of NAT+ARP and NAToD table structure

4.4 The probability of translation and forwarding latency in HA

We can calculate the probability of a fixed source port conflict $P(k)$ during continuous session Φ in k_{th} times as:

$$P(k) = \frac{\beta^k \binom{\Phi}{k+1} \binom{\alpha\varepsilon(2^{16}-\eta)-\alpha}{\Phi-k-1} (\Phi-k-1)!}{\binom{\Phi}{1} \binom{\alpha\varepsilon(2^{16}-\eta)-1}{\Phi-1} (\Phi-1)!},$$

$$k = 2, 3 \ldots \Phi - 1 \tag{1}$$

where α denotes the size of IP addresses pool, β denotes the number of internal hosts, η denotes reserved region of source port number, and ε denotes the number of external host and service (hosts $\times$ services). Equation (1) shows the conditional probability. The denominator is the number of cases that no two identical cases exist during the continuous session Φ, and the numerator is the number of cases that exactly k identical cases appear during the continuous session. In Eq. (1), we found that the most important factor contributing to the probability is Φ, and β the second. It is observed that the probability of conflicts with any given source port exponentially decreases as the number of conflicts increases.

We can calculate the conditional probability P that at least one source port conflicts with others during continuous session Φ as:

$$P = 1 - \frac{\beta^{\Phi} \binom{\alpha\varepsilon(2^{16}-\eta)}{\Phi}}{\binom{\alpha\beta\varepsilon(2^{16}-\eta)}{\Phi}} \tag{2}$$

Where the numerical analysis shows the number of cases that no two are the same, and the numerator is the number of cases that at least two source ports are conflicted.

Equation (2) shows that if the number of continuous sessions was not large enough, the probability of conflicts increases in polynomial for continuous sessions; however, when number of continuous sessions is large enough, the probability decreases in polynomial.

We modified Eq. (2) by considering the probability of each number k of conflicts and calculated the expected number during each continuous session. The expected value of the translation $E[k]$ (a.k.a. "average translation percentage") in concurrent session Φ is as follows:

$$E[k] = \sum_{i=1}^{\Phi-1} \frac{i\beta^{i+1}\alpha\varepsilon(2^{16}-\eta)\binom{\Phi}{i+1}\binom{\alpha\varepsilon(2^{16}-\eta)-1}{\Phi-i-1}}{\binom{\alpha\beta\varepsilon(2^{16}-\eta)}{\Phi}\Phi!} \times \frac{(\Phi-i-1)!}{\binom{\alpha\beta\varepsilon(2^{16}-\eta)}{\Phi}\Phi!} \tag{3}$$

We use MATLAB [29] for setting up NAToD emulation model to estimate the probability of performing packet translation. There are four controllable parameters in our design, the number of concurrent sessions, the number of internal clients, the number of the external servers, and the number of service types in each server in the external network. The simulation scenario is: (1) using a single IP address for the NAToD's IP address pool, (2) 1,000 clients in the internal network, (3) 10, 100, 1000 servers $\times$ number of service ports in the external network, (4) 100,000 continuous (concurrent) sessions. We initiate a simulation to set up TCP/UDP sessions from all internal clients to several service types on several external servers randomly. We run 100,000 times of simulation totally and keep all sessions alive simultaneously. It represents that we will set up the number of internal clients $\times$ 65,536 session records in the NAToDTT. Most of the NATTT entries in a common commercial network equipment are usually designed for only 2,048~4,096 sessions. It means that the number of simultaneous sessions in the simulation is far larger than the general cases used in the practical applications.

In this research we run simulations with 3 scenarios: "Traditional NAT", "Restricted NAToD" and NAToD. It is for the load testing to find NAToD's translation frequency and its performance. The simulation result is presented in Fig. 9. At the end of simulation, the NAToDs have processed only 30 (0.03 %) sessions that need IP address translation in the scenario of 1000 servers; and only 332 (0.33 %) sessions that need IP translation in the scenario of 100 servers. In the scenario of 10 servers, we find just only 15,065 (15 %) sessions that need address translation. It shows that the efficiency of translation is improved apparently. We also assume that same percentage of the packet flow need to be translated when all clients share a single public IP address.

But this simulation has two restrictions: Firstly, two sessions that are translated after collision may still get the same source port number, but the probability should be very low and can be neglected; Secondly, in most TCP/IP protocol

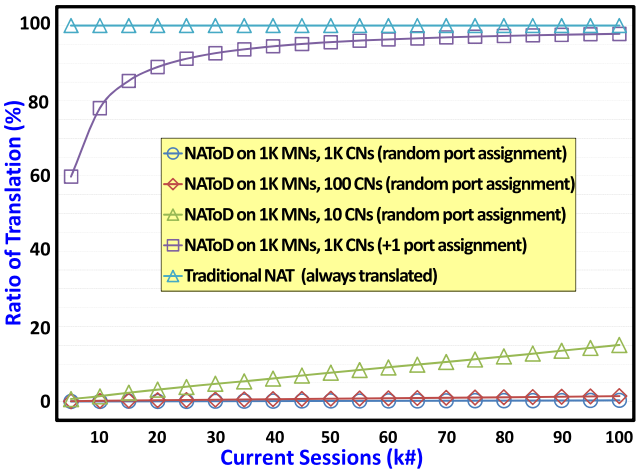

Fig. 9 The translation probability of NAToDs and NATs

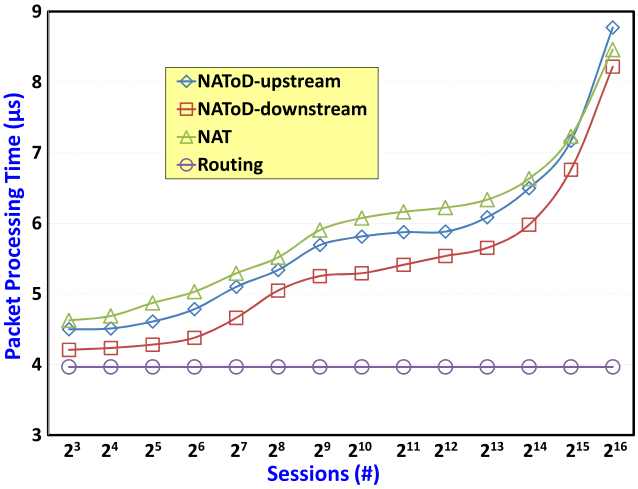

Fig. 10 Packet processing time vs. sessions

stack designed on various OS platforms, the source port number may not be generated randomly. Taking Microsoft Windows system as an example, it increases the port number starting from port 2,000. With "Restricted NAToD", it shows the special scenario that port number assigned always starts from the same value and increases progressively for all clients. In Fig. 9 we can observe that 97,792 (97.8 %) sessions need to be translated in the scenario of 1K servers. Our works are in full accordance with our numerical analysis earlier.

4.5 The processing latency of NATed packets in HA

In the following we discuss the performance of the NAToD mechanism; we analyze the packet processing and end-to-end latency of the proposed scheme in this subsection. We compare the performance of processing NAToDs and NAT packets in a home agent, which is based on a one-MIPS processor. Note that in PMIPv4 environments, traditional NAT still encounters address confliction problem, in which NAT program codes are executed with the assumption that the search time (hash table lookup) of the NAT translation table and processing time will be influenced by the memory access performance of the NAT function. Generally, the more entries in the table, the longer search time and processing time will take. The procedure of ARP table lookup also faces the same situation. In addition, recalculation of checksum increases the latency and consumes extra processor power. Again, comparing with NAToDs under the same condition, we assume that NAToD translation table consumes same search time as NAT translation does, but the search time of the ARP table can be totally neglected at backward traffics because the MAC addresses have been piggybacked in tunneled packets. We could expect that both the processor utilization and packet forwarding delay of NAToD will be reduced significantly compared with the original NAT.

Figure 10 presents the average packet processing time obtained from the reciprocal of the above results. The line of "NAT", "NAToD-upstream" and "NAToD-downstream" is the time cost for traditional NAT, NAToD with outgoing traffic (forward) and NAToD with incoming traffic (backward). The line of "Routing" denotes the processing time of normal forwarding without NAT procedures. The packet processing time in NAT and NAToD layer grows when the number of sessions gets larger. For incoming (backward) traffics, NAToD scheme does not require to look up routing and ARP table, it gets better average performance than NAT even when the number of sessions reaches 65536 (2^{16}). From the above result, the end-to-end forwarding latency between MNs and CNs is expected to decrease by our proposed scheme.

4.6 The signaling cost and protocol overhead

Since the address acquisition and duplicated address detection procedures are both omitted, the signaling cost and power consumption in PMIPv4 networks could benefit from our proposed scheme. In terms of protocol overhead, since Ethernet header is necessary for tunneling the data packets, the extra data such as delivery IP header (outer header) for tunneling, Ethernet header and tunnel header are all considered as overhead, the analysis result shows that protocol overhead in PMIPv4 networks would be increased a little in our proposed scheme.

In this simulation, we set the payload size from 64 to 1024 bytes; the overhead is 40 bytes for IPv4, 8 bytes for UDP, 8 bytes for tunnel, and 18 bytes for Ethernet header and trailer. The only difference between our proposed scheme and the original PMIPv4 is that the tunnel consists of an Ethernet frame rather than an IP packet. Figure 11 represents the comparison of protocol overhead between the original PMIPv4 and PMIPv4 with NAToD support. A minor overhead is introduced so that the effective

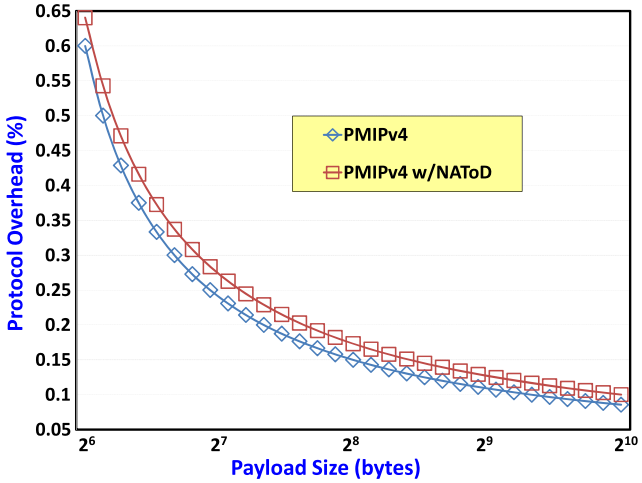

Fig. 11 The protocol overhead between PMIPv4 and proposed scheme

payload size is decreased, this is the only tiny drawback of our proposed scheme.

5 Pros and cons

Although NAToD solves the major problems in PMIPv4 deployment, but there is still uncertainty of some minor problems, which are discussed as follows:

5.1 How to avoid duplicate IP warning?

In standard IP over point-to-multipoint network environment, an internal host will continuously receive the broadcast packets (e.g. ARP) sent by each other and monitor whether the duplicate IP address appears in the same network segment. Once a duplicate IP address is detected [9], the operating system will show the "Duplicate IP Warning" message in user's console. The internal hosts that adopt the duplicate IP address may encounter such IP conflict situation in NAToD architecture. Although this situation does not affect the normal operation of NAToD, it is still an issue to be solved. There are several feasible solutions: (1) disabling this warning message in the user's operating system; (2) using broadcast filter; and (3) combining port-based virtual LAN (VLAN) design in wireless access point to filter out the ARP broadcast packets from each internal hosts. Through this method, each internal host in its individual VLAN will no longer receive the ARP broadcast from other VLANs.

5.2 How to assign duplicate IP address expressly?

There are two ways to assign duplicate IP address in the internal network, one is to establish all MNs with the same IP address manually, the other is using special designed DHCP server to assign duplicate IP address on purpose. The latter would be easier to mobile users. How to realize it deserves further study.

5.3 How to make connection between internal mobile nodes?

There are various application environments, such as public library, network coffee shop, computer classroom, and hotspot areas which provide Internet access services. For NAToD-enabled environments, it will be especially suitable to deal with inside invading or paralyzed attack under the wireless network environment. Since all hosts of internal network adopt the same global unique IP address, they obviously cannot set up the connection to each other as usual, and it will break the threaten that comes from the internal network effectively. The traditional NAT is a mechanism that protects the external network accessing to the internal network normally. However, the NAToD is a mechanism that protects both the external network accessing to internal network, as well as internal network accessing to internal network at the same time normally. This design that avoids the attack coming from other internal hosts in same network segment is important for the network service on WLAN environment.

Localised addressing Regarding how to connect internal hosts with same IP address in the NAToD environment, the answer is using the proxy server, which prepares several local domain names, pseudo IP addresses to fake both sides by Proxy ARP function, translate both IP addresses and relay the traffic to the other side. Thus, the internal mobile nodes with same outside public address can still identify and communicate with each other using call by unique domain names [4, 8]. If we want to put NAToD design into commercial products, this extended part should be an issue for follow-up study [6]. Actually, NAToD improves translating efficiency for the outside directory but tradeoff the poor translating efficiency for the inside directory.

5.4 What's the NAT type of NAToD?

The major difference between NAToD and NAT is that the former works under the transparent mode when it processes the outbound packets. While NAToD is processing the inbound packets, there is little difference between NAT and NAToD on the mapping method. According to NAT variation definition in [34], NAToDs can also work in those four main types: (1) Full Cone; (2) Restricted Cone; (3) Port Restricted Cone; and (4) Symmetric. This is basically similar to NATs.

The NAToDs is recommended to work on the symmetric mode when the session is on the SPC state with ToD being active. The NAToD can solve the restriction for some

specific protocol described in [16], such as IPSec [24]. The reason it cannot work normally behind NATs is because it fails in integrity check due to altered IP header. NAToD may solve this problem because most packets still keep their original IP header. However, IPSec still works abnormally behind NAToD with very low probability because SPC cannot be totally avoided in NAToD.

Regarding the restriction on FTP Port Mode or H.323 with VoIP application caused by external hosts attempting to set up new TCP/UDP connection via new port number [16], NAToD is unable to solve such problem, and it is the fundamental limitation of NAToD.

Moreover, some protocols such as SNMP, RSVP and H.323 also encounter problems [16, 35], these protocols cannot operate normally because the payload fields carry the IP address information, and this also cannot be solved efficiently when packets pass through traditional NAT. This problem may be solved using the NAToD mechanism, because NAToD might not replace the IP header, but it is not a complete NAT traversal solution and not guaranteed to work.

6 Conclusions

This paper presents a novel approach to provision NAT function interoperating with PMIPv4, a network-based, intra-domain local mobility management scheme. Using experiments based on a working prototype, we demonstrate that with NAToD, mobile users can immediately experience the benefits of seamless mobility without any software upgrade on their mobile terminals. PMIPv4 inherently achieves very low handoff latency, which makes uninterrupted mobile real-time applications possible. We have extended NAToD to support mobility when users move across wireless network domains which lack IPv4 address.

PMIPv4 is defined as an emerging protocol for both WiMAX and LTE networks for mobility. We first discuss the possibility to adopt PMIPv4 on WLANs such as WiFi (IEEE 802.11) networks. Beside the application in PMIPv4 to solve the private IP address overlapping problem, the PMIPv4 cooperating with NAToD also features higher performance, better security, and simpler NAT traversal and so forth. The performance issue of NATs has become a great challenge to the network administrators in large-scale, high speed networks. The NAToD mechanism not only can be implemented as our approach within HA in a PMIPv4 environment, but also can provide a single function for IP sharing combined with the routers, switches, firewalls, home gateways, IP phones, etc. These products can be value-added by enabling NAToD features. The NAToD could benefit the network appliances design with NAT behavior.

Despite the optimistic prediction of a rapid IPv4 depletion and IPv6 deployment, so far the majority of user equipments still operates with the IPv4 protocol suite [8], and by default is not able to perform any mobility procedures. Before widespread deployment of IPv6, mobile users still suffers both the severest IPv4 address shortages in the wireless Internet and complex, impractical client based mobility schemes. Thus, the PMIPv4 cooperate with NAToD is a feasible solution for most kind of IPv4 mobile Internet applications, just like the original NAT environment. Although our proposed scheme does not intend to encourage the network application or delay the retirement schedule of IPv4, it will nevertheless be an inevitable result.

Regardless of the layer upon which mobility rides, the end result of our efforts will deliver a new approach of mobile communications on WLANs—one in which subscribers are always connected, able to seamlessly access services regardless of whether they are located in a home network, or move to any foreign networks. Additionally, our approach also allows the subscriber to move points of attachment without requiring mobility capacity, and therefore, neither the subscriber nor the corresponding device will be aware that mobility has happened. This novel approach relies on PMIPv4 and NAToD to support mobility, and it provides a various mobility services options to future mobile Internet.

References

1. 3GPP2 (2007). *CMIP based inter-AGW handoff*. 3GPP2 X.S0054-210-0.
2. 3GPP2 (2008). *Network PMIP support*. 3GPP2 X.S0061-0 Ver1.0.
3. Aboba, B., et al. (2004). *Extensible Authentication Protocol (EAP)*. IETF RFC 3748.
4. Atkinson, R., Bhatti, S., & Hailes, S. (2009). ILNP: mobility, multi-homing, localised addressing and security through naming. *Telecommunications Systems*, *42*(3–4), 273–291.
5. Campbell, A., Gomez, J., Kim, S., Valko, A., & Wan, C. (2000). Design, implementation and evaluation of cellular IP. *IEEE Personal Communications.*
6. Carl-Mitchell, S., & Quarterman, J. S. (1984). *Using ARP to implement transparent subnet gateways*. IETF RFC1027.
7. Chen, W. S., & Jia, W. K. (2006). An IP shared device based on the Network Port Translation (NPT). *Journal of Internet Technology*, *7*(1), 85–93.
8. Chen, Y. C., & Jia, W. K. (2009). Challenge and solutions of NAT traversal for ubiquitous and pervasive applications on the Internet. *The Journal of Systems and Software*, *82*(10), 1620–1626.
9. Cheshire, S. (2008). *IPv4 address conflict detection*. IETF RFC 5227.
10. Damic, D. (2007). *Comparison and evaluation of network-based IP mobility management schemes*. Intl Conf. of Telecommunications.
11. Droms, R. (1997). *Dynamic Host Configuration Protocol*. IETF RFC 2131.
12. Farinacci, D., et al. (2000). *Generic Routing Encapsulation (GRE)*. IETF RFC 2784.
13. Frikha, M., & Maale, L. (2006). Micro mobility in the IP networks. *Telecommunications Systems*, *31*(4), 337–352.

14. Gundavelli, S., et al. (2008). *Proxy mobile IPv6*. IETF RFC 5213.
15. Haverinen, H., & Malinen, J. (2000). *Mobile IP regional paging*. Internet draft: draft-haverinen-mobileip-reg-paging-00.
16. Holdrege, M., & Srisuresh, P. (2001). *Protocol complications with the IP network address translator*. IETF RFC3027.
17. IEEE. (2004). *802.1X-2004—port based network access control*.
18. IETF. *IP over IEEE 802. 16 networks (16ng) working group*.
19. IETF. *Network-based localized mobility management (NetLMM) working group*.
20. Jeon, H., Jeong, S., & Riegel, M. (2009). *Transmission of IP over Ethernet over IEEE 802. 16 networks*. IETF RFC 5692.
21. Johnson, D., Perkins, C., & Arkko, J. (2004). *Mobility support in IPv6*. IETF RFC 3775.
22. Kempf, J. (2007). *Problem statement for network-based localized mobility management (NETLMM)*. IETF RFC 4830.
23. Kempf, J. (Ed.) (2007). *Goals for Network-based Localized Mobility Management (NETLMM)*. IETF RFC 4831.
24. Kent, S., & Atkinson, R. (1998). *Security architecture for the Internet Protocol*. IETF RFC 2401.
25. Koodli, R. (2005). *Fast handovers for mobile IPv6*. IETF RFC 4068.
26. Koodli, R., & Perkins, C. (2007). *Mobile IPv4 fast handovers*. IETF RFC 4988.
27. Leung, K., Dommety, G., Yegani, P., & Chowdhury, K. (2010). WiMAX forum/3GPP2 proxy mobile IPv4. IETF RFC 5563.
28. Madanapalli, S., Park, S Chakrabarti, S., & Montenegro, G. (2009). *Transmission of IPv4 packets over the IP convergence sublayer of IEEE 802.16.* IETF RFC 5948.
29. MATLAB. http://www.mathworks.com/.
30. Mrugalski, T., & Wozniak, J. (2010). In *Analysis of IPv6 handovers in IEEE 802. 16 environment. Telecommunication systems* (Vol. 45, pp. 191–204).
31. O'Neill, A., Tsirtsis, G., & Corson, S. (2000). *Edge mobility architecture*. Internet draft: draft-oneill-ema-02.
32. Perkins, C. (2002). *IP mobility support for IPv4*. IETF RFC 3344.
33. Ramjee, R., Varadhan, K., Salgarelli, L., Thuel, S. R., Wang, S.-Y., & La Porta, T. (2002). HAWAII: a domain-based approach for supporting mobility in wide-area wireless networks. *IEEE/ACM Transactions on Networking*, *10*(3), 396–410.
34. Rosenberg, J., Weinberger, J., Huitema, C., & Mahy, R. (2003). *STUN—simple traversal of User Datagram Protocol (UDP) through Network Address Translators (NATs)*. IETF RFC3489.
35. Senie, D. (2002). *Network Address Translator (NAT)—friendly application design guidelines*. IETF RFC 3235.
36. Simpson, W. (1994). *The Point-to-Point Protocol (PPP)*. IETF RFC 1661.
37. Simpson, W. (1995). *IP in IP tunneling*. IETF RFC 1853.
38. Soliman, H., et al. (2005). *Hierarchical mobile IPv6 mobility management (HMIPv6)*. IETF RFC 4140.

Wen-Kang Jia received his Ph.D. degree from the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan. Before returned to school, he had been a senior engineer and manager since 1991 in various networking areas including ICT Manufacturer, Network Integrator, and Telecomm Service Provider. His research interests include TCP/IP protocol design, IP mobility, IP convergence, error resilience coding, multimedia communications, NAT traversal, routing and switching, multicasting and broadcasting, teletraffic engineering, P2P Networks, and wireless networks. He is a member of IEEE.

Yaw-Chung Chen received his B.S. degree from National Chiao Tung University, Hsinchu, Taiwan, his M.S. degree from Texas A&M University, Kingsville, Texas, USA, and his Ph.D. degree from Northwestern University, Evanston, Illinois, USA. In 1986 he joined AT& T Bell Laboratories, Naperville, Illinois, where he worked on various exploratory projects. He joined National Chiao Tung University, Hsinchu, Taiwan as an associate professor in 1990. He is currently a professor in the Department of Computer Science. He is also acting as the Director of Information Industry Institute/ National Chiao Tung University Joint Research Center. His research interests include wireless media streaming, mobility management, P2P systems and green computing. He is a senior member of IEEE and a member of ACM.