Frontiers of Information Technology & Electronic Engineering
www.zju.edu.cn/jzus; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A scalable and efficient IPv4 address sharing approach in IPv6 transition scenarios

Guo-liang HAN[‡], Cong-xiao BAO, Xing LI

(*CERNET Center, Tsinghua University, Beijing 100084, China*)

E-mail: guoliang.taurus@gmail.com; congxiao@cernet.edu.cn; xing@cernet.edu.cn

Received Jan. 14, 2015; Revision accepted May 6, 2015; Crosschecked July 15, 2015

Abstract: IPv6 has been an inevitable trend with the depletion of the global IPv4 address space. However, new IPv6 users still need public IPv4 addresses to access global IPv4 users/resources, making it important for providers to share scarce global IPv4 addresses effectively. There are two categories of solutions to the problem, carrier-grade NAT (CGN) and 'A+P' (each customer sharing the same IPv4 address is assigned an excluded port range). However, both of them have limitations. Specifically, CGN solutions are not scalable and can bring much complexity in managing customers in large-scale deployments, while A+P solutions are not flexible enough to meet dynamic port requirements. In this paper, we propose a hybrid mechanism to improve current solutions and have deployed it in the Tsinghua University Campus Network. The real traffic data shows that our mechanism can utilize limited IPv4 addresses efficiently without degrading the performance of applications on end hosts. Based on the enhanced mechanism, we propose a method to help service providers make address plans based on their own traffic patterns and actual requirements.

Key words: IPv6 transition, Carrier-grade NAT (CGN), A+P, Address sharing, Dynamic switching
doi:10.1631/FITEE.1500022 **Document code:** A **CLC number:** TP393

1 Introduction

The global unallocated IPv4 address pool was depleted on Feb. 3, 2011, and the remaining addresses held by each regional Internet registry (RIR) are becoming exhausted quickly (Huston, 2014). Simply upgrading current IPv4 infrastructures to dual-stack can provide customers native IPv6 Internet access, but the solution cannot be incrementally deployed and brings much higher operating costs, which has not helped to stimulate the IPv6 evolution process. As a result, many large-scale service providers are constructing IPv6-only backbone networks to circumvent the shortage of IPv4 addresses (Fiocco, 2012). Meanwhile, their customers need to preserve connectivity with global IPv4 users/resources, which requires service providers forward IPv4 traffic in the IPv6 backbone (Chen *et al.*, 2006) and share scarce global IPv4 addresses among numerous customers. There have been a lot of solutions to the problem, which can be divided into two categories: carrier-grade NAT (CGN) solutions, e.g., Dual-Stack Lite (Durand *et al.*, 2011), and 'A+P' mode solutions (each customer sharing the same IPv4 address is assigned an excluded port range), e.g., mapping of address and port using encapsulation (MAP) (Troan *et al.*, 2014), mapping of address and port using translation (MAP-T) (Li *et al.*, 2014), 4rd (Després *et al.*, 2014), and lw4o6 (Cui *et al.*, 2014).

In CGN solutions, customers typically use private IPv4 addresses to access the IPv4 Internet. Global IPv4 addresses are shared as a pool in the

[‡] Corresponding author

ORCID: Guo-liang HAN, http://orcid.org/0000-0002-8921-4202 service provider NAT boxes which track the connection states of traversing sessions. For example, Dual-Stack Lite shares global IPv4 addresses in the address family translation router (AFTR). AFTR uses an extended NAT binding table to track states of tuples (endpoint IPv6 address, internal IPv4 address, internal IPv4 port, external IPv4 address, external IPv4 port). Obviously, the statistical multiplexing behavior helps CGN solutions utilize IPv4 addresses efficiently. However, since the size of this binding table is proportional to the total number of sessions of all customers, CGN solutions are not scalable to be deployed in large-scale service providers. It also has severe problems with traceability, state synchronization, logging, processing, and storage requirements as specified in Škoberne *et al.* (2014).

To solve the above problems, the 'A+P' mode solutions are commonly preferred (Cui *et al.*, 2014; Després *et al.*, 2014; Li *et al.*, 2014; Troan *et al.*, 2014). A+P adopts another IPv4 address sharing mechanism; i.e., available IPv4 addresses are split at the granularity of ports and distributed to customers. In other words, each customer can obtain a partial IPv4 address; i.e., it obtains an IPv4 address and an authorized port range. The port range of any two customers sharing the same IPv4 address does not overlap, so each customer can be uniquely identified by its pair (IPv4 address, port range). Compared with CGN solutions, the port restriction function in A+P scenarios is implemented on the customer side. Thus, the need of maintaining so many states on the provider side to differentiate customers is eliminated.

Regarding the port-range provision method, A+P solutions can be further classified into two categories. lw4o6 is one of stateful solutions, while MAP/MAP-T/4rd are representatives of stateless solutions. In stateful solutions, different customers sharing the same IPv4 address can be assigned different sizes of port ranges, but dynamic per-customer port ranges have to be maintained on the provider side. Thus, the complexity of management is increased. In stateless solutions, the entire port space is fairly distributed to customers sharing the same IPv4 address. Each customer obtains a distinct port set ID (PSID), which can be mapped algorithmically to an exclusive port set. In this way, service providers do not need to store any customer states on the provider side, and thus stateless solutions are more scalable and have better management and traceability capabilities. However, the fair allocation strategy is not flexible enough to adapt to dynamic port requirements of various customers and IPv4 addresses cannot be shared efficiently.

In this paper, we propose a hybrid IPv4 address sharing approach, which can combine the advantages of CGN and stateless A+P. As far as we are concerned, despite similar comments from the Internet Engineering Task Force (IETF) Sharing of an IPv4 Address (SHARA) Birds of a Feather (BOF) (Bajko *et al.*, 2009), the approach has neither been carefully designed nor quantitatively measured in prior study. Our contributions in this paper are listed below:

1. We propose a scalable IPv4 address sharing approach which can utilize limited IPv4 addresses efficiently without degrading the performance of applications on end hosts. To evaluate its performance, we deployed it in the Tsinghua University Campus Network (TUNET) and collected session statistics from thousands of campus devices. The evaluation shows the validity and efficiency of our approach.

2. Based on the approach and the evaluation results, we propose a method to help service providers make their own address plans. It is especially useful for providers who want to use limited IPv4 addresses to accommodate a large number of customers. For providers with enough ports to be assigned to their customers, the method can also provide some insights.

2 Related work

Many prior studies have measured the port consumption statistics in large-scale networks. Durand (2009) measured 8000 subscribers behind the Cable Modem Termination System (CMTS) and the peak port consumption level was 40 000 ports, i.e., 5 ports per subscriber in each direction. Alcock (2008) measured the statistics data of the residential digital subscriber line (DSL) traffic of the New Zealand Internet Service Provider (ISP) and showed that the distribution of peak sessions is heavy-tailed. His group also gave useful statistics of outbound/inbound sessions and ports (Alcock *et al.*, 2010; Alcock and Nelson, 2011). As examples of specific application measurements, Schneider *et al.* (2008; 2009) showed that customers can generate hundreds of concurrent sessions by browsing websites or using on-line social networks. All the above measurements show the dynamic port requirements of different customers at different times. It is a critical demand for address sharing solutions to adapt to these requirements.

A recent survey paper introduced an exhaustive classification, comparison, and trade-off analysis of existing IPv4 address sharing mechanisms (Škoberne *et al.*, 2014). Stateless address mapping (SAM) (Després, 2009b) is a stateless address sharing approach different from A+P, and Després (2009a) presented a method for SAM to coexist with existing CGNs. Huston (2009) presented a method to combine CGN and A+P to support legacy non-(A+P) customer premise equipments (CPEs), without discussing inter-operation between the two data paths. All of the above solutions are not flexible in allocating dynamic port ranges while preserving scalability.

In the IETF SHARA BOF meeting, Bagnulo (2009) and Bajko *et al.* (2009) demonstrated three dimensions for address sharing approaches to adapt to different deployment models: designing an appropriate address compression ratio and corresponding compression techniques, encapsulating mechanisms, and signaling mechanisms. They also stated that CGN can be used in A+P scenarios to support 'out-of-port-range packets', but no further design or evaluation was proposed. Also, they did not present the possible effects on performance and management of the hybrid model.

Ripke *et al.* (2010) proposed two dynamic mechanisms, i.e., reuse and increase, for A+P solutions to delegate available ports, and evaluated impacts of port assignment strategies and TIME_WAIT timeout values. The mechanisms can adapt to the dynamic port requirements and share IPv4 addresses efficiently. However, dynamic per-customer states need to be maintained at the carrier side. If they were deployed in the large-scale network, the port requirements of different customers would vary frequently, and the solution would trigger problems of scalability, traceability, and the additional cost of signaling.

3 A scalable and efficient address sharing approach

Our approach is based on stateless A+P solutions (Li *et al.*, 2014; Troan *et al.*, 2014). As stated, stateless A+P solutions have several advantages.

1. End-to-end transparency

A+P offloads the port mapping function onto the CPE of end customers, providing customers enough flexibility to configure their own port rules based on their own applications. It is especially useful for peer-to-peer (P2P) applications or other applications requiring inbound connections.

2. Scalability

'Stateless' means that the session/customer states can be aggregated into a small and static table so that costs of corresponding insertion/query/deletion operations will not increase with the number of customers or the total number of sessions. The stateless characteristic makes the core translation/tunneling devices more robust in large-scale networks. They do not have to synchronize dynamic states with other load-balancing devices. Moreover, they have higher CPU power utilization efficiency and less signaling overhead.

3. Traceability

Compared with the current address model where each customer has at least one IPv4 address, all address sharing solutions require additional port information to trace back specific customers (Ford *et al.*, 2011). Among all these solutions, the stateless A+P solutions can achieve the best traceability. The administrator can use the port mapping algorithm (as specified in Section 3.2.1) to locate customers precisely, without the need to explore the massive size of session logs and dynamic port allocation leases.

4. Worm isolation

In stateless A+P solutions, if one of the end hosts behind the port-restricted A+P gateway is infected with a worm which occupies all the available ports, it cannot spread to other customers thanks to the static port range allocation. Such behavior makes stateless A+P solutions robust and fault-tolerant.

To adapt to the dynamic port requirement, we use the stateful data path DS-Lite to dynamically compensate for the stateless A+P data path. In the general case, the A+P gateway uses the algorithm specified in Section 3.2.2 to multiplex available ports efficiently. When the available ports of the A+P customer are used up and the current packet cannot be mapped to an external port, the A+P gateway resorts to the alternative data path; i.e., it forwards the packet to the CGN directly. Since the application behaviors of different customers are independent, their peak requirement of ports would have a high probability to differ in time points. Meanwhile, according to our measurement (in Section 3.2.3), the requirement of ports is always heavy-tailed and most customers require very few ports at each time point. Consequently, in stateless A+P scenarios, if providers adjust the port quota to satisfy only the 'basic' requirements and redirect those 'heavy' sessions to CGN, the address sharing efficiency can be improved significantly while maintaining scalability and traceability. The framework will be presented in detail in the following subsections.

3.1 Architecture

Fig. 1 shows the architecture of our approach. Each A+P customer has a gateway to perform flexible forwarding operations. When the gateway is connected to the IPv6 access network, it acquires an IPv4 address and a PSID of the customer, and other essential parameters to perform stateless IPv4/IPv6 address mappings from the DHCPv6 server. The provision process is outside the scope of this paper. Readers can refer to Mrugalski *et al.* (2015) as an example, and other provision methods are also possible. If there is a DS-Lite AFTR (Durand *et al.*, 2011) server in the carrier side network, the DHCPv6 server also provides the FQDN name of the AFTR server to the A+P gateway (Hankins and Mrugalski, 2011).

When the IPv4 application on the end host behind the gateway initiates a connection with its server/peer in the IPv4 Internet, the gateway uses its strategy module to decide whether to allocate the session an external port or redirect it to the alternative DS-Lite path directly. If the former decision is made, the packet will undergo the stateless A+P path, i.e., double translation (Li *et al.*, 2014) or encapsulation (Troan *et al.*, 2014). If the latter decision is made, the Basic Bridging BroadBand element (B4 module) (Durand *et al.*, 2011) will encapsulate the packet and forward it to the AFTR server. The key components of the strategy module are the efficient NAPT module and the strategy layer, which will be illustrated in Sections 3.2.2 and 3.2.4, respectively.

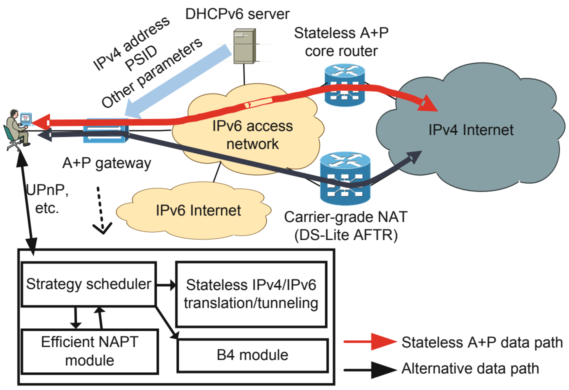

Fig. 1 Architecture of the hybrid address sharing approach

The architecture also supports inbound connections, which are often required by P2P applications (Alcock and Nelson, 2011). Related issues will be discussed in Section 3.2.2.

3.2 Important building blocks

3.2.1 Generalized modulus algorithm

To facilitate the stateless manner of the A+P core router, the generalized modulus algorithm (GMA) (Troan *et al.*, 2014) is used as one of the basic building blocks to aggregate and manage A+P customers efficiently. GMA specifies a bijective mapping between each PSID and the corresponding authorized port range. In other words, each customer needs only to acquire a PSID rather than a specified port range, and can still use Eq. (1) to work out its authorized port range:

$$\begin{aligned} &P_{\text{PSID}} = \{j \cdot R \cdot M + \text{PSID} \cdot M + i| \\ &i \in [0, M-1], j \in [1, 65\,536/(R \cdot M) - 1]\}. \end{aligned} \tag{1}$$

There are two important parameters in GMA: sharing ratio R and contiguous parameter M. The entire 65 000 port space is fairly distributed to R customers, with the system ports reserved. If an IPv4 address has a sharing ratio R, at most R customers can share this IPv4 address with different PSIDs. The parameter M scatters the port set of each customer into several smaller port sets, each of which has M contiguous ports. The use of M facilitates the applications requiring contiguous ports (e.g., RTP/RTCP) and gives providers enough flexibility to manage the port range of each customer. Note that GMA does not use the well-known ports 0–1023 specified by IANA.

For example, if the provider needs to use 256 IPv4 addresses (a C class subnet) to serve 4000 customers, R is set to $\lceil 4000/256 \rceil = 16$; i.e., at most 16 customers share one IPv4 address and the range of each customer's PSID is 0 to 15. Supposing $M = 64$, the authorized port ranges of each PSID are shown in Table 1. The port range of each customer is split into several segments, each of which has 64 consecutive ports. Providers can customize the value of M to adapt to the requirements of their customers.

Table 1 An example of GMA

PSID	Port range
0	{1024, 1025, ..., 1087}, ..., {64 512, 64 513, ..., 64 575}
1	{1088, 1089, ..., 1151}, ..., {64 576, 64 577, ..., 64 639}
...	...
15	{1984, 1985, ..., 2047} ..., {65 472, 65 473, ..., 65 535}

GMA guarantees that different customers (with different PSIDs) sharing the same IPv4 address have distinct port ranges. Consequently, in the context of GMA, the pair (IPv4 address, PSID) is the unique identity of each customer in the whole Internet. An explicit external port accompanied with the IPv4 address can determine the corresponding customer accurately with Eq. (2), giving the advantage of traceability:

$$\mathrm{PSID} = \left\lfloor \frac{\mathrm{Port}}{M} \right\rfloor \% R. \tag{2}$$

Compared with managing specific port ranges, managing PSIDs is much more convenient and scalable. In a specific domain, PSIDs can be aggregated much like the aggregation of IPv4 addresses. In this way, providers can use very few rules to define the processing behaviors of several customer groups, hence a stateless and scalable solution.

3.2.2 Efficient NAPT module

In our approach, we use the port-restricted network address port translation (NAPT) (Srisuresh and Egevang, 2001) deployed in the A+P gateway to guarantee that the external ports of each customer are in the scope of authorized port ranges as specified by GMA. On that basis, we explore possible ways to improve its efficiency of sharing IPv4 addresses; i.e., limited external ports are used to accommodate as many sessions as possible. It may be optional, but it is especially useful for providers with very few IPv4 addresses but a large number of customers.

NAPT is a well-known and widely deployed technique. First we declare some terminologies for the convenience of further discussion:

1. ses: a specific session tracked by the NAPT module;

2. S(ses): the local pair of ses (internal address, internal port);

3. D(ses): the pair of ses (remote address, remote port);

4. pt(ses): the transport layer protocol of ses;

5. P(ses): the external port (in the authorized port set A) assigned by NAPT;

6. f: the NAPT mapping function which maps ses to P(ses): $f(\mathrm{ses}) = P(\mathrm{ses}) \in A$.

To improve the efficiency of the NAPT module, there are two kinds of adjustable factors: the port selection algorithm and timeout values of different TCP/UDP states.

The port selection algorithm is the strategy of choosing an available port for each new outbound session. The IETF BEHAVE Working Group contributed three best current practice (BCP) documents (Audet and Jennings, 2007; Guha *et al.*, 2008; Srisuresh *et al.*, 2009) to address basic NAPT implementation requirements for UDP applications, TCP applications, and ICMP sessions, respectively. Specifically, they showed that an NAT MUST has an 'endpoint-independent mapping' behavior to help real-time multimedia or P2P applications communicate across NAPT boxes more easily. The 'endpoint-independent mapping' behavior specifies that different sessions with the same S(ses) must be treated as a group and allocated with the same external port. The behavior implies the following mandatory port selection requirement: For a new session s, if there exists an active session s_0 such that $S(s_0) = S(s)$ and $\mathrm{pt}(s_0) = \mathrm{pt}(s)$, we must have

$$\begin{aligned} f(S(s), D(s), \mathrm{pt}(s)) &= f(S(s_0), D(s_0), \mathrm{pt}(s_0)) \\ &= P(s_0). \end{aligned}$$

Meanwhile, different groups can multiplex the same external port as long as they have different destinations. We call it 'destination multiplexing', which is optional. To guarantee the correctness of f, the following constraint must be conformed to: for each two active sessions s_1, s_2, if $D(s_1) = D(s_2)$ and $\mathrm{pt}(s_1) = \mathrm{pt}(s_2)$ (here obviously $S(\mathrm{s}_1) \neq S(\mathrm{s}_2)$; otherwise, they are the same session), we must have

$$f(S(s_1), D(s_1), pt(s_1)) \neq f(S(s_2), D(s_2), pt(s_2)).$$

That is, there exists a reverse mapping function g such that $g(P(\mathrm{ses}), D(\mathrm{ses}), \mathrm{pt}(\mathrm{ses})) = S(\mathrm{ses})$.

If the gateway is allocated very few available ports, destination multiplexing should be always enabled to accommodate more sessions. There are many port selection algorithms to realize destination multiplexing, but we claim that they have the same effect on multiplex external ports. In fact, if some algorithm cannot allocate an external port for the incoming packet of a new session ses, it implies that there have been enough active sessions ($|A|$) with the same destination $D(\text{ses})$ and that allocating any external port to ses would violate the above constraint. In this case, any other algorithm cannot accept the packet. To prevent the possible side effects caused by multiplexing a lot of sessions on a specific external port (e.g., issues proposed by Ramaiah and Tate (2008) and Wing (2008)), we use the following port selection algorithm:

1. Try to choose an unused external port;

2. If all external ports are in use, among the ports with all sessions having completed their TCP handshakes, find the one with the least number of sessions;

3. If each external port has at least one session with uncompleted TCP handshakes, resort to the strategy layer in Section 3.2.4.

Choosing appropriate timeout values is also important. It helps the gateway clear inactive sessions in time and accept new sessions. If the timeout values are set overly long, those already expired sessions may prevent the new sessions from being accommodated. If they are set too short, the algorithm may clear some active sessions, causing subsequent packets to be dropped.

The three BCP documents (Audet and Jennings, 2007; Guha *et al.*, 2008; Srisuresh *et al.*, 2009) also contain requirements of the NAPT module to track states of TCP/UDP/ICMP sessions of end hosts in order not to degrade the performance of applications. Our efficient NAPT module follows the requirements. Guha *et al.* (2008) specified that an NAT UDP mapping timer MUST NOT expires in less than 2 min unless the destination port is in the well-known port range. However, Alcock *et al.* (2010) showed that there are a large number of short-lived UDP connections and 2 min for all sessions would reduce the port utilization efficiency. Alcock *et al.* (2010) suggested setting a short expiry timeout value for UDP sessions with only one single outgoing packet. We adopt this suggestion in our approach. Audet and Jennings (2007) specified the suggested timeout values for TCP sessions. Specifically, 'established connection idle-timeout' (denoted by TO_1) must be at least 2 h 4 min, 'transitory connection idle-timeout' (denoted by TO_2) must be at least 4 min, and the TIME_WAIT state timeout (denoted by TO_3) may be configurable. Ripke *et al.* (2010) verified the high correlation between the port consumption and the TIME_WAIT timeout values. However, setting TO_3 too small would cause the overlap of connections and increase their failure rates. In our approach, we set TO_3 to 30 s. If the new sessions cannot be accommodated, the NAPT module resorts to the strategy layer to explore alternative paths.

To support inbound connections, we choose to perform 'endpoint-independent filtering' for TCP sessions as specified in Audet and Jennings (2007).

We also claim that "DNS sessions should not be handled by the NAPT module". Many previous studies have pointed out that source port randomization is essential for mitigating the DNS cache poisoning attack (Kaminsky, 2008; Herzberg and Shulman, 2013) but contradicts the 'endpoint-independent mapping' requirement. Besides, DNS sessions always have the same destination but different sources, and are thus difficult to multiplex. Therefore, leaving the DNS sessions to specific DNS proxy software (e.g., BIND or DNSMASQ) not only decouples the security problem with the efficiency of the NAPT module, but also eliminates the unnecessary difficulty in multiplexing DNS sessions.

The NAPT module mentioned here is especially efficient in two respects. On the one hand, 'destination multiplexing' helps sessions with different remote pairs (address, port) multiplex the same external port more easily. It implies that applications with different destination addresses can multiplex the same port-set easily. Even if the customer has many end hosts, the NAPT module can perform well as long as the behaviors of those hosts are asynchronous. On the other hand, the 'endpoint-independent mapping' behavior eliminates the pressing shortages of external ports to accommodate a lot of P2P sessions with only few source ports.

3.2.3 Diversity of external port requirement

The efficient NAPT module alone cannot adapt to the dynamic port requirements of various customers. To make it clearer, we collected the IP traffic at the egress of the Tsinghua University Campus Network (TUNET) for 5 d and evaluated the port consumption of each TUNET customer. Specifically, we used the efficient NAPT module specified in Section 3.2.2 to generate active 5-tuple sessions of all TUNET customers (there were a total of 387 390 distinct on-line customers) during the five days and these active sessions were sampled every 5 min. At each sample time point t, we used Eq. (3) to calculate the port requirement of each customer i:

$$\mathrm{REQ}(i,t)=\max\{\mathrm{REQ}_{\mathrm{tcp}}(i,t),\mathrm{REQ}_{\mathrm{udp}}(i,t)\}. \quad (3)$$

The distribution of port requirements is as shown in Fig. 2.

We observe that the port requirement at each time point is heavy-tailed. More than 99.6% (1 − 0.4%) customers at each time point require fewer than 128 external ports. More than 95% (1−5%) customers at each time point require fewer than 16 external ports. More than 90% (1 − 10%) customers at each time point require fewer than 8 external ports. Meanwhile, customers with peak port requirements always differ in time points. Therefore, providers can adjust the port quota to satisfy only the 'ordinary' customers at each time point. When those 'heavy' customers have sessions that cannot be assigned with an external port by the NAPT module, they can be redirected to the DS-Lite path, which will be illustrated in Section 3.2.4. For example, if the port quota was set to 128, at each time point, at most 0.4% TUNET customers would have to resort to the DS-Lite path for some of their sessions.

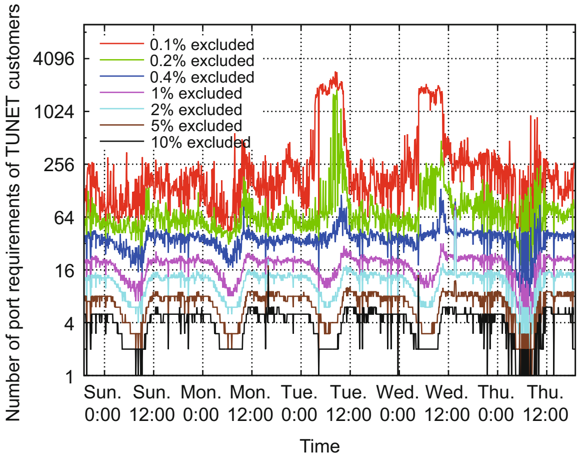

Fig. 2 Port requirement distribution at each sample time point

3.2.4 Strategy layer

As illustrated before, we use the DS-Lite data path in the strategy layer to compensate for the stateless A+P data path dynamically. As demonstrated by the evaluation results in Section 4, this mechanism can help stateless A+P networks significantly increase the address sharing efficiency while preserving scalability and traceability.

Fig. 3 shows the structure of the strategy layer. It can be separated by three units: strategy input interface, pre-processing unit, and post-processing unit. The interface unit can be forked by customers to formulate flexible strategies to adapt to the dynamic situations and requirements. These strategies are stored in the rule table which typically contains port forwarding rules and sessions to be delivered to the DS-Lite path. End hosts can also use UPnP or NAT-PMP (Ford *et al.*, 2011) to add port forwarding rules to the rule table.

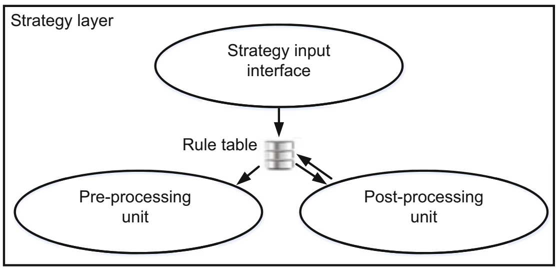

Fig. 3 Structure of the strategy layer

The pre-processing unit handles the outgoing packet before the NAPT module and chooses the subsequent module to process the packet. Algorithm 1 shows the detailed steps of the pre-processing unit.

If the packet is rejected by the NAPT module and cannot be assigned an external port, the post-processing unit typically redirects it to the DS-Lite path in order not to drop the packet, as illustrated in Algorithm 2. The structure of the strategy layer is scalable to accept other flexible strategies.

In the next section, we will use the real traffic data of TUNET to show the validity of our approach and the possible effects of different port quotas.

4 Evaluation

We implemented the above mentioned scalable and efficient address sharing approach and deployed it in the Tsinghua University Campus Network. The evaluation topology is as shown in Fig. 4a.

Algorithm 1 Pre-processing unit

```
Require: an outgoing packet pkt
Ensure:
Find the port forwarding rule (rule) for pkt
if rule exists then
    Map the source pair (address, port) according to rule
    Deliver pkt to the stateless IPv4/IPv6 translation/tunneling module
else
    Check whether some previous packet ppkt of the same session traversed the DS-Lite path
    if ppkt exists then
        Deliver pkt to the B4 module
    else
        Deliver pkt to the NAPT module
    end if
end if
```

Algorithm 2 Post-processing unit

```
Require: the outgoing packet pkt and the return
         code ret from the NAPT module
Ensure:
if ret == SUCCESS then
    Deliver pkt to the stateless IPv4/IPv6 translation/tunneling module
else
    if the conservative strategy is applied then
        Redirect pkt to the B4 module
        Track the states of the session and provide an interface for the pre-processing module to query
    end if
end if
```

4.1 Preparation

As prior work, the stateless A+P core router has been deployed between the pure IPv6 backbone network CNGI-CERNET2 and the pure IPv4 backbone network CERNET (Li *et al.*, 2011). To evaluate the framework proposed in this study, we further deployed the DS-Lite AFTR between the CNGI-CERNET2 and the IPv4 Internet. We also attached the 'aggregated A+P gateway' to the TUNET IPv6 network which is connected to CNGI-CERNET2. The aggregated A+P gateway is designed for campus customers who typically have no home gateways. The design also provides us with the convenience of measuring the session tracking states of various A+P gateways. The structure of the aggregated A+P gateway is as illustrated in Fig. 4b.

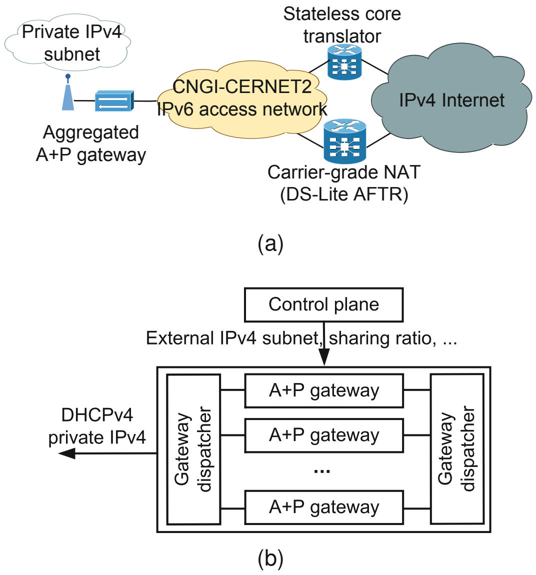

Fig. 4 The evaluation architecture: (a) evaluation topology; (b) structure of the aggregated A+P gateway

The aggregated A+P gateway is made up of many parallel A+P gateways, each of which has an external IPv4 address and a distinct PSID allocated by the control plane. The external IPv4 information (address, PSID) is encoded in the private IPv4 address, which is then allocated to wireless customers by DHCP such that the address mapping process is stateless. For example, if we have a /30 public IPv4 subnet, and the sharing ratio R is 1024, the maximum number of A+P gateways will be $4 \times 1024 = 4096$. The DHCPv4 module then generates a /20 ($32 - \log_2 4096 = 20$) private subnet and allocates the private addresses to wireless customers. Each address within the private subnet can be algorithmically mapped to a dedicated A+P gateway. When an outgoing packet arrives at the aggregated A+P gateway, the 'gateway dispatcher' module will work out the corresponding A+P gateway based on the source address. When an incoming packet arrives, the 'gateway dispatcher' module will work out the corresponding A+P gateway based on the destination pair (address, port).

To evaluate our approach proposed in Section 3, we applied the mechanism in every A+P gateway and adjusted the sharing ratio R on the control plane of the aggregated A+P gateway. Each R lasted from 22:00 on the first day to 22:00 on the next day. Table 2 shows the detailed R settings during the evaluation. During the six days, we dumped the 5-tuple sessions tracked by the NAPT modules (stateless path) and the DS-Lite AFTR server (stateful path). Fig. 5 shows the number of concurrent online customers on the stateless path during the six days. We analyzed the impact of different sharing ratios (i.e., different port quotas), which will be presented in the next sections.

Table 2 Sharing ratio settings during the evaluation

Day	R	Quota	Day	R	Quota
1	256	255	4	2048	31
2	512	127	5	4096	15
3	1024	63	6	8192	7

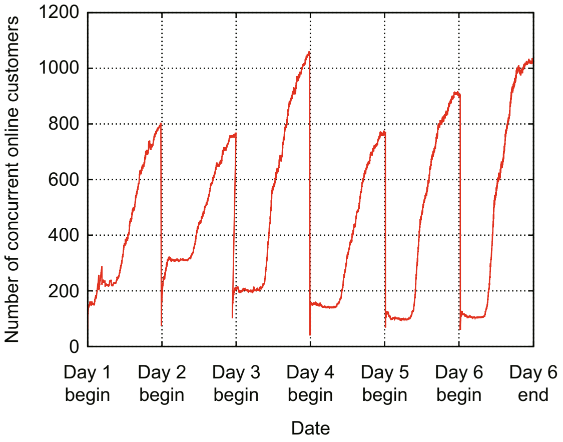

Fig. 5 Number of concurrent online customers (stateless path)

4.2 Impact on the DS-Lite traffic

Fig. 6 shows the numbers of active TCP and UDP sessions tracked by the DS-Lite AFTR during the six days. It also shows the number of customers with at least one active session on the DS-Lite path.

In the first two days (the port quotas of each customer are 255 and 127, respectively), the TCP sessions of all customers barely need the DS-Lite path, which implies the port quota is sufficient. When the port quota reduces to 63, some TCP applications resort to the DS-Lite path, but the number of such customers is limited to 3 at the same time, and the number of DS-Lite TCP sessions is bursty. It implies that there are very few customers requiring more than 63 external TCP ports at the same time, and our approach can satisfy the bursty requirements of these customers without reducing the address sharing efficiency of all customers. When the TCP port quota is further reduced, more customers have to use the DS-Lite path. When the quota is set to 7 (Day 6), there are more than 80 customers requiring the DS-Lite path. Although the number is relatively small compared with the number of concurrent on-line customers (by less than 10%), it implies that many TCP applications require more than 7 external ports.

Compared with TCP sessions, the number of DS-Lite UDP sessions is less correlated with the port quota settings. Even when the quota is set to 127, there are still some UDP applications of very few customers which require more than 800 (680 + 127) external ports. When the quota is set to 7, there are at most 12 customers requiring at most 1000 DS-Lite

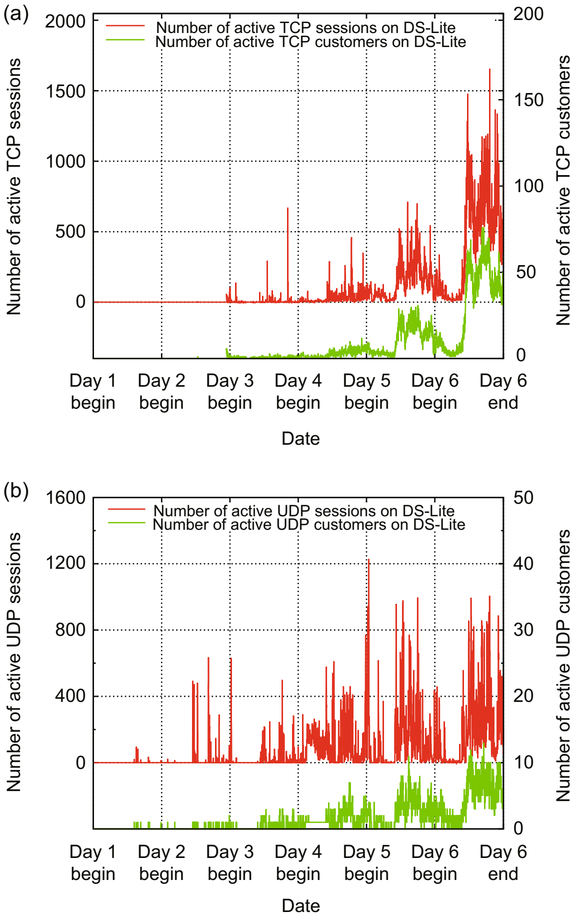

Fig. 6 Impact of the sharing ratio on DS-Lite traffic (stateful path): (a) DS-Lite TCP sessions and customers; (b) DS-Lite UDP sessions and customers

UDP sessions concurrently. Considering the policy tracking UDP sessions in Section 3.2.2, it implies that there are many short-lived UDP sessions and the UDP port requirements are more distributed across different time periods and different customers.

4.3 Impact on distribution of NAPT port requirements

For another perspective showing the impact of different sharing ratios, Fig. 7 illustrates the distribution of port consumption tracked by the NAPT modules measured in the aggregated A+P gateway. The distribution of the first day shows the unbounded distribution of port requirements since there are hardly any customers requiring the DS-Lite path. When smaller port quotas are exerted on A+P customers, the distribution of the top 20% customers is squeezed and more customers have to resort to the DS-Lite path (the port consumption reaches the port quota).

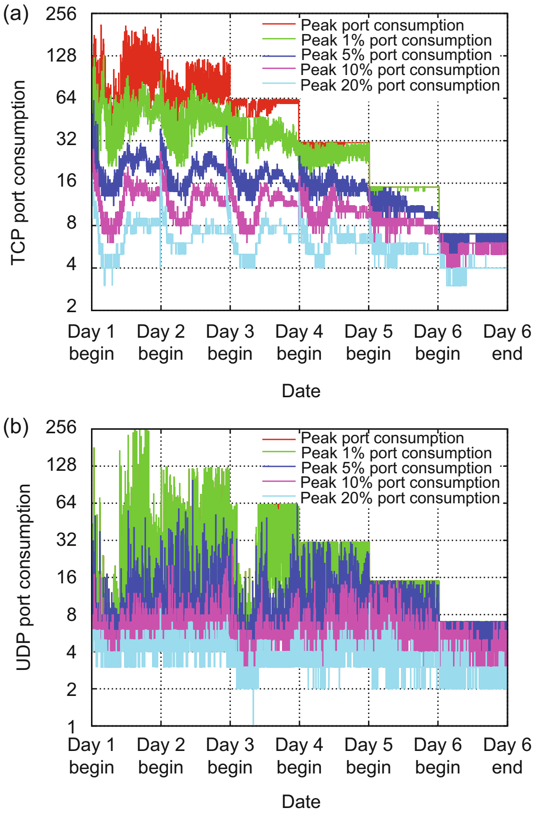

Fig. 7 Impact of the sharing ratio on NAPT port requirement distribution (stateless path): (a) NAPT TCP port distribution; (b) NAPT UDP port distribution

4.4 Impact on the data path of applications

To further explore the application types on the DS-Lite path, we measured the distributions of remote ports on the NAPT path and on the DS-Lite path. Fig. 8 shows the distribution of remote TCP ports in the last four days (there were no TCP DS-Lite sessions in the first two days).

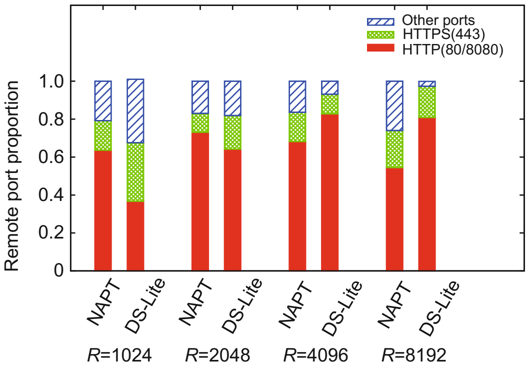

Fig. 8 Application mix of the NAPT path and the DS-Lite path

When R = 8192, the HTTP/HTTPS applications contributed 97.4% of the TCP DS-Lite sessions and the proportion of other applications was only 2.6%. When R = 4096, the proportion of HTTP/HTTPS applications was still very high (93.1%). It implies that many web applications require more than 15 ports as studied in Schneider *et al.* (2008; 2009). When the port quota increased to 31 and 63, there were not so many concurrent DS-Lite TCP sessions and the proportion of HTTP/HTTPS applications on the DS-Lite path decreased. From the distribution we can observe that HTTP/HTTPS applications (typically generated by browsers) dominated the DS-Lite path because they typically establish many concurrent sessions with the same destination to cache possible next-step links on the same page. For UDP sessions, DNS was the dominant protocol on the DS-Lite path. This is because some customers use manually configured DNS addresses (IPv4) and these DNS sessions cannot be multiplexed by our efficient NAPT module as specified in Section 3.2.2.

The evaluation results validate the efficiency and flexibility of our approach. They also show that different sharing ratios have different impacts on the traffic and the applications of customers. In the next section, we will propose a method to help service providers choose an appropriate sharing ratio and make their own address plans.

5 Making flexible address plans

Our address sharing approach, which provides a tuner between the stateless A+P solutions and the stateful CGN solutions, can help providers make flexible address plans according to their actual requirements. In fact, if R is so small that the port quota is sufficient for every customer, the approach will be a stateless A+P solution. If R is set to 32 768, the port quota of each customer is 0 and the approach will be equivalent to the CGN solution. If R is set between the above-mentioned extreme values, the approach can combine the advantages of stateless A+P solutions and CGN solutions. Fig. 9 shows the impacts of tuning sharing ratios.

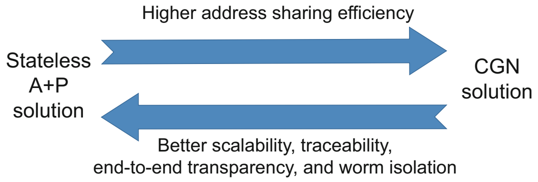
Higher address sharing efficiency
Stateless A+P solution
CGN solution
Better scalability, traceability, end-to-end transparency, and worm isolation

Fig. 9 Impacts of tuning sharing ratios

When the sharing ratio is tuned larger, the IPv4 address sharing efficiency will be improved and providers can use the same set of external IPv4 addresses to accommodate more customers. However, as more and more customers have to use the DS-Lite path, the DS-Lite AFTR will be heavily loaded and the 'end-to-end' transparency compromised. When there are many on-line customers with sessions of two data paths, the traceability will also be compromised to some extent because the probability of tracing back to the DS-Lite address pool can no longer be neglected. Conversely, when the sharing ratio is set smaller, the approach will have a better scalability, traceability, 'end-to-end' transparency, and worm isolation capability but a relatively low address sharing efficiency. Based on our approach, service providers can choose an appropriate sharing ratio and design their own address plans.

6 Discussion and application scope

Our approach is especially useful for providers who want to use limited IPv4 addresses to accommodate a large number of customers. For providers with enough ports to be assigned to their customers, it may not be necessary.

Apart from the scenarios of forwarding IPv4 traffic in the IPv6 backbone networks, our approach can easily be applied in other IPv4 address sharing scenarios, e.g., the stateless IPv4/IPv6 translation scenario (Li *et al.*, 2011).

Note that our hybrid address sharing mechanism cannot solve the basic address sharing issues proposed in Ford *et al.* (2011). For example, tracing back some specific customer also needs the pair (external address, external port) instead of the external address alone. If some customers or providers are not satisfied with any address sharing solutions, upgrading to IPv6 will be a good choice.

As mentioned in Section 3.2.4, the approach is scalable to accept other flexible strategies. For example, the customer may configure the data path (or some timeout values) of some sessions beforehand to meet the requirement of some specific applications.

7 Conclusions and future work

The hybrid address sharing approach proposed in this paper can use limited IPv4 addresses efficiently without degrading the performance of applications on end hosts. By deploying the hybrid approach, providers can easily combine the advantages of stateless A+P solutions and the stateful CGN solutions. We have deployed the approach in the Tsinghua University Campus Network for more than one year. Recently, we also deployed the approach in the CERNET Corporation network (enterprise network) and Wuxi Telecom network (broadband network). The real evaluation and deployment show the validity of our approach.

The approach is especially useful for providers who want to use limited IPv4 addresses to accommodate a large number of customers. For instance, for the over 240 million students in China (http://www.stats.gov.cn/tjsj/zxfb/201502/t20150226_685799.html), by using our approach, we can choose 1024 as the sharing ratio and provide all of them with scalable and flexible Internet access service. In future work, we will further expand the deployment scale and investigate other possible strategies to satisfy the requirements of prevalent applications on numerous customers.

References

Alcock, S., 2008. Research into the Viability of Service-Provider NAT. Available from http://www.wand.net.nz/∼salcock/someisp/flow_counting/result_page.html [Accessed on Jan. 8, 2015].

Alcock, S., Nelson, R., 2011. Measuring and characterising inbound sessions in residential DSL traffic. Proc. Australasian Telecommunication Networks and Applications Conf., p.1-6. [doi:10.1109/ATNAC.2011.6096628]

Alcock, S., Nelson, R., Miles, D., 2010. Investigating the impact of service provider NAT on residential broadband users. Proc. IEEE INFOCOM.

Audet, F., Jennings, C., 2007. Network Address Translation (NAT) Behavioral Requirements for Unicast UDP. RFC 4787. [doi:10.17487/RFC4787]

Bagnulo, M., 2009. Sharing of an IPv4 Address. Available from http://www.ietf.org/proceedings/74/shara.html [Accessed on Jan. 8, 2015].

Bajko, G., Boucadair, M., Bush, R., *et al.*, 2009. Overview of Shared Address Solution Space. Available from http://www.ietf.org/proceedings/74/slides/shara-9.pdf [Accessed on Jan. 8, 2015].

Chen, M., Li, X., Li, A., *et al.*, 2006. Forwarding IPv4 traffics in pure IPv6 backbone with stateless address mapping. Proc. 10th IEEE/IFIP Network Operations and Management Symp., p.260-270. [doi:10.1109/NOMS.2006.1687557]

Cui, Y., Sun, Q., Boucadair, M., *et al.*, 2014. Lightweight 4over6: an Extension to the DS-Lite Architecture. Available from https://tools.ietf.org/html/draft-cui-softwire-b4-translated-ds-lite-05 [Accessed on Jan. 8, 2015].

Després, R., 2009a. Port-Range Based IPv4 Address Space Extension—a Static Approach Based on SAM. Available from http://www.ietf.org/proceedings/74/slides/shara-7.pdf [Accessed on Jan. 8, 2015].

Després, R., 2009b. Scalable Multihoming across IPv6 Local-Address Routing Zones Global-Prefix/Local-Address Stateless Address Mapping (SAM). Available from https://tools.ietf.org/html/draft-despres-sam-03 [Accessed on Jan. 8, 2015].

Després, R., Jiang, S., Penno, R., *et al.*, 2014. IPv4 Residual Deployment via IPv6—a Stateless Solution (4rd).

Durand, A., 2009. Dual-Stack Lite. Available from http://lacnic.net/documentos/lacnicxii/presentaciones/flip6/02_Alain_Durand.pdf [Accessed on Jan. 8, 2015].

Durand, A., Droms, R., Woodyatt, J., *et al.*, 2011. Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion. RFC 6333.

Fiocco, A., 2012. Two Months after World IPv6 Launch, Measuring IPv6 Adoption: 6lab.cisco.com/stats. Available from http://blogs.cisco.com/news/two-months-after-world-ipv6-launch-measuring-ipv6-adoption-6lab-cisco-comstats [Accessed on Jan. 8, 2015].

Ford, M., Boucadair, M., Durand, A., *et al.*, 2011. Issues with IP Address Sharing. RFC 6269. [doi:10.17487/RFC6269]

Guha, S., Biswas, K., Ford, B., *et al.*, 2008. NAT Behavioral Requirements for TCP. RFC 5382. [doi:10.17487/RFC5382]

Hankins, D., Mrugalski, T., 2011. Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite. RFC 6334. [doi:10.17487/RFC6334]

Herzberg, A., Shulman, H., 2013. Socket overloading for fun and cache-poisoning. Proc. 29th Annual Computer Security Applications Conf., p.189-198. [doi:10.1145/2523649.2523662]

Huston, G., 2009. NAT++: address sharing in IPv4. *Int. Proto. J.*, **13**(2):1-10.

Huston, G., 2014. IPv4 Address Report. Available from http://www.potaroo.net/tools/ipv4/index.html [Accessed on Jan. 8, 2015].

Kaminsky, D., 2008. Black Ops 2008: It's the End of the Cache as We Know It. Black Hat USA.

Li, X., Bao, C., Chen, M., *et al.*, 2011. The China Education and Research Network (CERNET) IVI Translation Design and Deployment for the IPv4/IPv6 Coexistence and Transition. RFC 6219. [doi:10.17487/RFC6219]

Li, X., Bao, C., Dec, W., *et al.*, 2014. Mapping of Address and Port Using Translation (MAP-T). Available from https://tools.ietf.org/html/draft-ietf-softwire-map-t-08 [Accessed on Jan. 8, 2015].

Mrugalski, T., Troan, O., Farrer, I., *et al.*, 2015. DHCPv6 Options for Configuration of Softwire Address and Port Mapped Clients. Available from https://tools.ietf.org/html/draft-ietf-softwire-map-dhcp-12 [Accessed on Jan. 8, 2015].

Ramaiah, A., Tate, P., 2008. Effects of Port Randomization with TCP TIME-WAIT State.

Ripke, A., Winter, R., Brunner, M., *et al.*, 2010. The impact of port-based address-sharing on residential broadband access networks. Proc. IEEE Global Telecommunications Conf., p.1-6. [doi:10.1109/GLOCOM.2010.5683449]

Schneider, F., Agarwal, S., Alpcan, T., *et al.*, 2008. The new web: characterizing AJAX traffic. Proc. 9th Int. Conf. on Passive and Active Network Measurement, p.31-40. [doi:10.1007/978-3-540-79232-1_4]

Schneider, F., Feldmann, A., Krishnamurthy, B., *et al.*, 2009. Understanding online social network usage from a network perspective. Proc. 9th ACM SIGCOMM Conf. on Internet Measurement, p.35-48. [doi:10.1145/1644893.1644899]

Škoberne, N., Maennel, O., Phillips, I., *et al.*, 2014. IPv4 address sharing mechanism classification and tradeoff analysis. *IEEE/ACM Trans. Netw.*, **22**(2):391-404. [doi:10.1109/TNET.2013.2256147]

Srisuresh, P., Egevang, K., 2001. Traditional IP Network Address Translator (Traditional NAT). RFC 3022. [doi:10.17487/RFC3022]

Srisuresh, P., Ford, B., Sivakumar, S., *et al.*, 2009. NAT Behavioral Requirements for ICMP. RFC 5508.

Troan, O., Dec, W., Li, X., *et al.*, 2014. Mapping of Address and Port with Encapsulation (MAP). Available from http://tools.ietf.org/html/rfc7597 [Accessed on Jan. 8, 2015].

Wing, D., 2008. Dynamic TCP Port Reuse for Large Network Address and Port Translators. Available from http://tools.ietf.org/html/draft-wing-behave-dynamic-tcp-port-reuse-00 [Accessed on Jan. 8, 2015].