2010 Second International Conference on Advances in Future Internet

NOL: Name Overlay Service for Improving Internet Routing Scalability

Yangyang Wang
Dept. Computer Science & Technology
Network Research Center
Tsinghua University
Beijing, China
wangyy-06@mails.tsinghua.edu.cn

Jun Bi
Network Research Center
Tsinghua University
Beijing, China
junbi@tsinghua.edu.cn

Jianping Wu
Dept. Computer Science & Technology
Network Research Center
Tsinghua University
Beijing, China
jianping@cernet.edu.cn

Abstract—**Internet routing system is the fundamental components of Internet. As the Internet growth, Internet routing system is facing the scaling issues of global routing table expanding due to the wide use of multi-homing, traffic engineering, and mobility. Existing proposed solutions need to change host protocol stack or routing architecture, hence, no incentive for practical deployment. In this paper, we describe a new mechanism for the scaling issue. This mechanism adding a name overlay layer on TCP/IP protocol stack, Comparing with current solutions, it is easier to be deployed. It benefits the Internet edge users from multi-homing, traffic engineering and mobility, and also facilitates the reduction the global routing table size.**

Keywords-Interne routing; scalability; multi-homing; mobility

I. Introduction

Internet routing system is the fundamental component of the whole Internet. It provides the basic end-to-end reachability for every device connected to Internet. As the Internet grows, the routing scaling issue has been concerned by many research communities and Internet operators [1]. The routing table size in the Default Free Zone (DFZ) of the Internet has been growing rapidly in recent years, which would consume more capacities of memory, computation and electric power. There are many factors that cause the routing table size growth. Edge networks widely adopt multi-homing for load balance or redundant connections, in which Provider-Independent (PI) address space are used than Provider-Allocated (PA) address space due to legacy address allocation and desire for avoid renumbering in changing upstream ISPs (Internet Service Providers). PI addresses can not be aggregated in upstream ISPs and many more specific prefixes are announced into global routing table. Edge networks and ISPs also announce more-specific prefixes for particular traffic engineering requirement. It can be imagined that large number of emerged mobile networks and IPv6 wide deployment will aggravate the routing scaling issue in future Internet.

Many solutions for routing scaling problem have been proposed. Some of them are clean-slate designs that need to change the whole routing architecture. Most solutions focus on how to enable actual effect on practice in a short term. However, these engineering methods need extra mapping system, relying on bilateral deployment, which bring a big barrier to deployment in real networks. These main proposed solutions will be summarized in detail in the following section. In this paper, we design a new mechanism called name overlay (NOL) service. Instead of focusing on the reducing the routing table size in transit ISP, it pays more attention to how to enable the edge networks obtain more benefits from multi-homing, traffic engineering, mobility in a rapid scaling Internet. It doesn't need extra mapping system, and can be deployed unilaterally and relieve the scaling issue by extending the NAT/NAPT device widely deployed in today's Internet.

This paper is organized as follows. In the Section 2, we introduce the existing related work and their problems. After that, we describe the basic mechanisms of name overlay service in Section 3. Its benefits and deployment road map are discussed in the Section 4. Last is the conclusion and future work.

II. Related Work

In this section, we will give detailed introduction to the related solutions for Internet routing scaling issue and their limitations.

These proposed solutions are mainly grouped into two categories: core-edge separation and core-edge elimination [2]. The typical core-edge separation solutions include LISP [3], eFIT [4], Ivip [5], etc. Core-edge separation solutions separate edge networks address space from transit networks address space, which prevent the PI prefixes of edge networks from propagating into the transit core so that reduce the global routing table size. However, it requires a mapping between the separated address spaces. We describe the packet delivery process as shown in Figure 1. Host A sends a packet to Host B with source address and destination address set to the IP address of host A and B respectively.

This work was supported in part by Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant 200800030034, and in part by the National Basic Research Program ("973" Program) of China under Grant 2009CB320501.

978-0-7695-4091-7/10 $26.00 © 2010 IEEE
DOI 10.1109/AFIN.2010.11

IEEE computer society

When the packet arrive at router R1 by local routing system, R1 will find that the B's IP is not reachable in global routing table, and it will look up the mapping system to find the globally routable address to host B. The mapping system answers the query with the IP address of router R3 and R4. Finally, R1 will select the R4 mapping entry (under traffic engineering or other operational considerations) and then encapsulate the packet in IP-in-IP tunneling with the outside source and destination set to the router R1 and R4. The Internet transit networks delivery the packet to destination R4 and R4 decapsulates the packet and route it to the right destination host B. After that, router R1 and R4 will cache the corresponding mapping entries for host A and host B that will be used in the following packet delivery between host A and B. During this process, mapping and tunneling are two key steps. The routers R1, R2, R3, R4, are named Ingress Tunneling Routers (ITRs) or Egress Tunneling Routers (ETRs), and this kind of solutions are also called Map-Encap solutions.

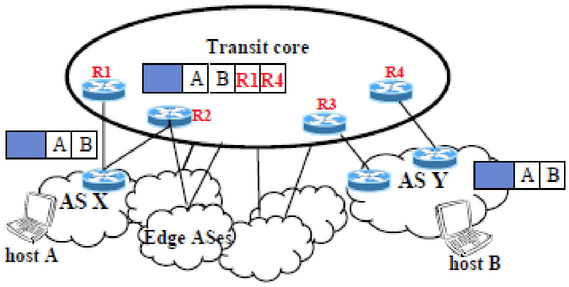

Figure 1. An example of core-edge separation solution.

The core-edge elimination solutions are mostly based on the ID/locator separation on end hosts. In today's Internet, IP addresses are considered being used with semantics overloading. An IP address is used as an identifier of one host end, as well as a locator where the host is attached to in the Internet. The typical core-edge elimination solutions based on ID/locator separation include HIP [6], Shim6 [7], name-based sockets [8], etc. In these solutions, a host will use an identifier in the transport sessions, while the IP layer use one or more IP addresses as locators. Therefore, hosts in a stub network with multiple provider networks can be assigned multiple PA addresses, which eliminate PI addresses from the core networks and improve the reduction of global routing table size.

However, there are many barriers to deploy these solutions in practice. Most of them need additional mapping system to map between core and edge network address spaces, or between IDs and locators. The construction, maintenance and cooperation of the global mapping system will introduce a large cost. Moreover, almost all of these solutions require either modifying host protocol stack (e.g., in core-edge elimination), or the bilateral deployment of ITRs/ETRs (e.g., in core-edge separation), which lead to complicated compatible issues in the process of increasingly deployment. Only their globally wide deployment of core-edge separation or elimination can take real effects.

In this paper, we present a name overlay (NOL) service, a new approach to improve routing scalability for the future Internet. It has no need to change existing TCP/IP stack and DNS system, and also no need for extra mapping system. It has better support for increasing deployment, and partial deployment will also bring corresponding benefits to some extent.

III. Name Overlay Architecture

The Figure 2 describes the protocol stack architecture with name overlay service. It adds a name overlay on the TCP/IP protocol stack. The main functions of name overlay layer include host name management and application session management based on host names. Applications can use NOL to execute host name configuration, registration and authentication, and also to initiate and manage transport connection channels by name.

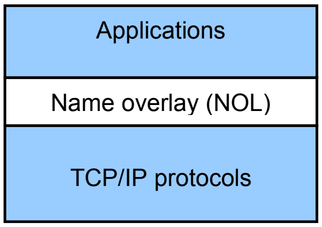

Figure 2. Architecture of Name Overlay.

Under the NOL, all data transports take place with traditional connections, such as using TCP/UDP protocols. The applications based on NOL can communicate with legacy applications by existing TCP/IP stack, or communicate with other NOL-enabled applications by NOL layer. Different from proposed host-based ID/Locator separation solutions, such as HIP, Shim6, and name-based stack, NOL doesn't need to change the existing TCP/IP stack, sockets and their packet formats. NOL can co-exist with the legacy infrastructure and the core-edges separation solutions. In the following section, we will describe in detail how it can be used to improve the routing scalability.

IV. Improvement For Routing System

This section will present how NOL works, and how it supports reduction of routing table size, multi-homing and traffic engineering. We will also discuss how NOL works with legacy hosts and domain name system of Internet.

A. Support for reduction of routing tables

How to improve routing scalability? By and large, there are two ways. One is to totally adopt PA addressing and eliminate all PI addresses, i.e., core-edge elimination scheme. However, due to IPv4 address exhaustion, this way has better to be going on with IPv6. And, PA addressing also causes renumbering problem for edge networks, and edge networks prefer PI addressing than PA addressing. NOL is not a solution of the core-edge elimination scheme, but it can also support adopting core-edge elimination. In such situation, we can consider the names used in NOL as identifiers and IP addresses as locators. Different from ID/locator separation, such as Shim6, name-oriented stack, etc, the data transport between two NOL-enabled application ends is performed by application-level sessions associated with pairs of source and destination names respectively. One application session is created on multiple sessions in transport layer with multiple pairs of source and destination IP addresses (locators). Even though the IP addresses changed and transport layer sessions were broken, the application sessions based on the unaltered pair of names in NOL would be kept continued. Thus, the edge networks multihomed to multiple provider networks can deploy core-edge elimination scheme based on NOL and derive multiple aggregateable PA prefixes to reduce the prefixes number in global routing core.

Another way mainly proposed here for improving routing scalability is to allow edge networks to use PI prefixes to avoid renumbering, and prevent the PI address prefixes into transit core networks. As described in Figure 1, these core-edge separation solutions are the designs of this way. NOL service takes the similar way. It introduces a new type of gateway, called Name Transfer Relay (NTR), which can be considered as an extension based on today's widely used NAT/NAPT [9] devices. Similar to the ITR/ETR routers in core-edge separation, NTRs prevent the PI addresses of edge networks into upstream transit networks. As shown in Figure 3, Host A is located in the edge network 1.1.0.0/16. The two NTR routers prevent the prefix 1.1.0.0/16 from entering into transit core. Hence, in the global routing table of transit core, there is no entry for the prefix 1.1.0.0/16. In this way, the routing table size could be reduced. Behind the NTRs, host A can initiate access to the outside Internet in the same way as it is behind NAT/NAPT devices. In fact, both legacy and NOL-enabled applications behind NTRs can initiate access to the outside as usual. Address and/or port translation between blocked PI addresses and globally routable addresses is performed in the NTR routers.

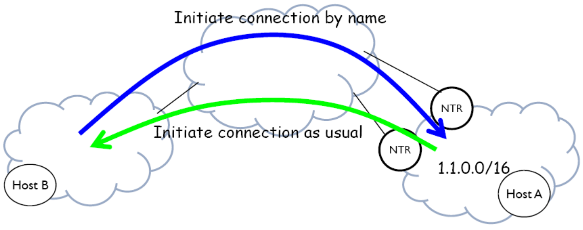

Figure 3. Prevent PI address using NTR routers.

But initiating access from outside Internet to host A is a complicated process. To access the hosts behind a NTR, we need to use NOL to traverse the NTR by name and initiate connections to the hosts behind it. We describe this process by the following example.

As showed in Figure 4, host B resides in an edge network with NTR routers deployed at network border. Host A is a host at outside Internet, and assumed to be NOL-enabled. Firstly, during the process of host B connecting to the Internet, it will obtain a name from the NTRs, or manually configure names and register them on the NTRs. The names have a format like email addresses, for example, "hostB@domain.net". The names to be used in the following steps must be also recorded by the NTRs. These names of hosts in the same edge network had better to be limited in the same one domain. For example, a host C that is in the same network as host B may have the name hostC@domain.net. Thus, we just query DNS for the "domain.net" for them in the following step, which will not increase the load of DNS.

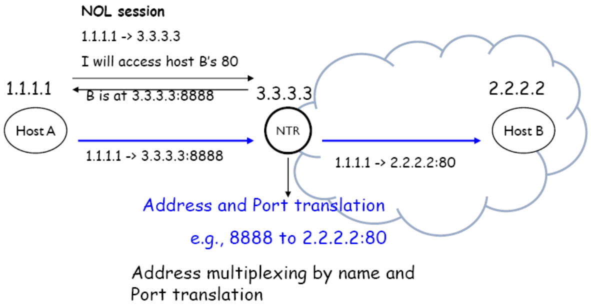

Figure 4. Traverse NTR from outside using NOL's name.

In the second step, we assume that host A wants to access host B, and know the host B's name "hostB@domain.net". Then the host A NOL sends a query to DNS for the entry "domain.net", and then, DNS returns an IP of the NTR 3.3.3.3, (configured in DNS). And then, host A's NOL will initiate a NOL session to NTR 3.3.3.3, with B's full name "hostB@domain.net" and the port to be accessed, here is 80, the well-known web service port.

Then, the NTR look up B's name in the local register record table, and knows that B is at IP 2.2.2.2. Here we assume that the NTR has only one IP address 3.3.3.3 in its translating address pool. It will create a mapping entry from one port to 2.2.2.2, (for instance, port 8080 -> 2.2.2.2:80), and return the port (i.e., 8080) to host A. After that, the NOL session is finished. (If NTR has many addresses, create address-to-address translating entry or other type of translation entry).

And then, host A's NOL sends packet to 3.3.3.3:8888, and the NTR translate it to 2.2.2.2:80. Initiatial access from A to B is successful.

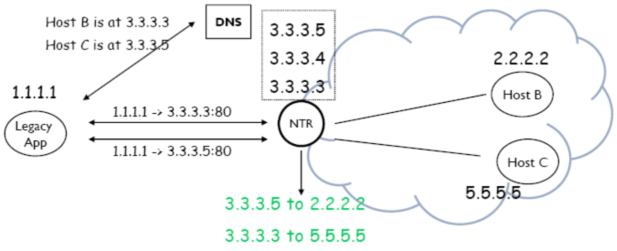

Figure 5. Legacy applications traverse NTR from outside.

For the legacy application accessing the servers located behind a NTR (e.g., host A runs legacy applications that does not adopt the NOL functions), the NTR can get globally routable PA addresses from upstream providers, and delegate these addresses to the public servers behind NTRs to enable the outside legacy applications to access the servers without port translation.

As shown in Figure 5, there is more than one address in the NTR address pool. The IP address 3.3.3.5 is delegated to the server host B 2.2.2.2, and 3.3.3.3 to the server 5.5.5.5. Corresponding map entries are also created in the NTR. DNS will direct the query for host B and C to 3.3.3.5 and 3.3.3.3 respectively, and the NTR will translate the packets sent to 3.3.3.5 and 3.3.3.3 to the real destination IP address 2.2.2.2 and 5.5.5.5.

B. Support for Multi-homing and traffic engineering

The above presents the basic elements and steps about how NOL service works. This section will discuss how to support multi-homing and traffic engineering through the NOL service.

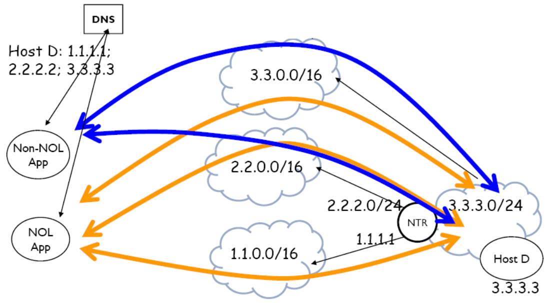

Figure 6. Multi-homing and traffic engineering in NOL service.

If an edge network multi-homes with several providers, and deploys NTRs to block its PI addresses into these providers. These NTRs can derive multiple PA addresses from the upstream providers and store them in their address pool. By DNS query or NOL session, any session that want to access the hosts behind the NTR can be directed to a specific PA address in the NTR address pool. As shown in Figure 6, the edge network 3.3.3.0/24 is multihomed with 2.2.0.0/16, 3.3.0.0/16, 1.1.0.0/16. Because prefix 3.3.3.0/24 can be aggregated in the network 3.3.0.0/16, thus, it is not necessary deploy NTR in the connection between the two networks. The deployed NTR prevent 3.3.0.0/24 from entering into the networks 2.2.0.0/16 and 1.1.0.0/16. However, the NTR can derive multiple PA addresses from them, such as 2.2.0.0/24 and one 1.1.1.1, stored in the NTR's address pool. And DNS can have three answers to the query for host D: 1.1.1.1, 2.2.2.2 and 3.3.3.3. The legacy applications that doesn't work with NOL will reach host D by 2.2.2.2 and 3.3.3.3, because these are specific IP address assigned to host D. Only NOL-enabled applications can access host D by multiplexing IP address 1.1.1.1 with the NOL help. Also, it can access host D by 2.2.2.2 and 3.3.3.3.

The incoming session from legacy or NOL-enabled applications can be directed to a specific NTR by DNS answers for names. In addition, the initial session from NOL applications can be redirected from one NTR to other appropriate NTR by the coordination in the NTRs' overlay network. These mechanisms provide some supports for traffic engineering. But it relies on the interaction with DNS, which maybe cause some update costs to DNS.

C. NOL naming space compatible with legacy Internet

Here, we will discuss the name space after adding and mixing the NOL service with the DNS. From the Figure 5 and Figure 6, we can see that DNS returns IP addresses of NTRs that are not the actual address of the desired destination hosts. This name resolution arrangement may cause some confusion for legacy applications. For example, as shown in Figure 6, only the IP addresses 2.2.2.2 and 3.3.3.3 are accessible for both NOL-enabled and legacy applications. The 1.1.1.1 is accessible only to NOL-enabled applications (by NOL session). If DNS returns the three IP addresses for the legacy ends, the IP address 1.1.1.1 would make the legacy end confused. This kind of NTR's IP address is only fetched and recognized by NOL layer. For example, if a NOL-enabled application wants to access a web server, which has a FQDN name www.webserv.com, and a NOL name webserv@domain.net. For the FQDN name, DNS will return the server's IP addresses that are accessible without NOL. For NOL name, the NOL layer will send a query only for "domain.net" of its NOL name "webserv@domain.net". The DNS will return the IP address of a proper NTR. And then, NOL can connect to the NTR router and tell it that "I want to talk with the host or service named "webserv". Thus, the web server could have two names www.webserv.com and webserv@domain.net. The latter is only used by NOL. But it is supported by DNS and doesn't modify the DNS mechanism (e.g., no need to add new record type to DNS).

D. Support for Mobility

NOL service supports mobility in two keys: 1) keeping application-level continuity; 2) notifying position change. Mobility takes effect only between NOL-enabled ends.

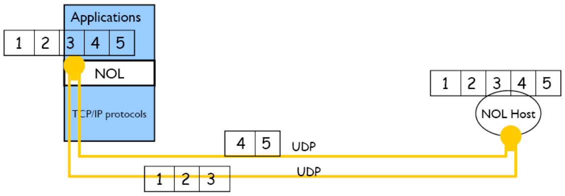

Figure 7. Applications data transport over multiple traditional TCP/IP transport channels (here shows UDP, it also may be TCP/SCTP, etc.) .

NOL layer considers the traditional TCP/IP transport connections as transport channels. By overlay on TCP/IP stack, NOL layer can isolate the application data transport process from the underlying transport channel updates due to the IP addresses change. NOL keeps application continuity by setting breakpoints and sequence numbers in data stream, as shown in Figure 7.

We assume that two NOL-enabled ends are communicating with each other. One of them moves to a new IP attachment point, while the other end keeps the original place. Before the application session switching to the new pair of IP attachment points, the end moved to new position will inform the other unmoved end of its new IP attachment point information by a NOL notification on original transport channels. This information includes new IP address, and/or new port number, etc. After that, the original transport channels become broken, and new channels are created, and the application-level session is switched to these new channels.

It may happen sometimes that all the original channels are broken before creating new channels, and there is no active channel used to send NOL notification. For example, two communicating ends move in the mean time. In this situation, the both ends only know the names of the other side. Thus, the application data transport session must restart from zero.

V. Discussion of Benefits and Challenges

In this section, we will summarize the benefits of NOL and challenges to its costs.

A. *Benefits*

1) No need to change TCP/IP stack, sockets and DNS system. It doesn't impact the legacy applications. This will facilitate its increasing deployment. For the popular applications, such as Skype, MSN, BitTorrent, NOL can be installed with these popular applications upgrade.

2) No need for extra mapping systems. NOL doesn't need mapping system that is an important component in the proposed core-edge separation or elimination solutions. This will greatly reduce the barrier for wide deployment.

3) NOL-enabled applications can communicate with legacy applications by the traditional TCP/IP stack. It means that NOL can be compatible with legacy applications in the Internet.

4) Don't increase the load of DNS system drastically. The name used in NOL should be in a domain hierarchy. It is just like an email address "hostname@domain.net". We only query DNS for the "domain.net" in NOL service, and the corresponding NTRs know the specific IP addresses of the "hostname" in that domain.

5) NOL cooperating with multiple distributed NTR gateways can benefit applications from multi-path routing. This will facilitate improving the access performance of content-distributed networks (CDN) or Internet content provider networks, such as Google, Youtube, eBay, etc.

6) NOL layer considers the traditional TCP/IP transport connections as transport channels. By overlay on TCP/IP stack, NOL can make the application data transport process isolated from the underlying transport channels update due to IP addresses change, which keeps application continuity for mobility.

7) We can prevent more specific PI prefixes into transit network by unilaterally deploying NTR routers. It can contribute to the control of global routing table growth. NOL applications and proxies can traverse NTR to access the behind end systems. The LISP solution need bilateral deployment of ITRs and ETRs to run tunneling process, NOL only needs unilateral deployment of NTRs.

8) NOL can be compatible with existing core-edge separation solutions, such as eFIT, LISP, Ivip, etc., and it can cooperate with them.

B. *Challenges*

1) Legacy applications have trouble with traversing NTRs to access to the hosts behind NTRs. Such problems can be resolved by deploying NOL proxy for legacy hosts or delegating globally routable PA addresses for these servers in the NTR addressing pool.

2) It may increase the number of entries in DNS, but it is not drastic. Because it only increases DNS entries in domains granularity, not hosts granularity. The DNS entries will not only be increased, but its dynamical upates might be agitated as well. However the scalability and performance of DNS is guaranteed by name hierarchy and cache mechanism.

3) Address translating cost on NTRs. The NTRs need to keep the related mapping tables, and perform IP address and port number translation per packet, which will increase the costs of CPU and memory resources.

VI. Conclusion and Future Work

Name overlay service focus on enabling the Internet edge to gain benefits from multi-homing, traffic engineering, and mobility in a large and scaling Internet, and then, reduce and control the global routing table size growth after its wide deployment. It doesn't need extra global mapping system and no change to existing TCP/IP stack. The name overlay service may mean that we can learn and explore a practical and low-cost approach for the routing scaling issue from NATs (NAT/NAPT or NAT66 [10]) and the session layer of the OSI model [11]. And the name overlay service may also cause the evolution of DNS for future Internet. Further prototype evaluation and improvement in real Internet are the future work.

References

[1] D. Meyer, L. Zhang, and K. Fall, "Report from the IAB Workshop on Routing and Addressing", RFC4984

[2] Dan Jen, Michael Meisel, et al. "Towards A New Internet Routing Architecture: Arguments for Separating Edges from Transit Core". HotNets 2008

[3] D. Farinacci, V. Fuller, D. Meyer, and D. Levis. Locator/ID Separation Protocol (LISP). draft-farinacci-lisp-12, Sep. 2009.

[4] The eFIT project, http://netlab.cs.memphis.edu/efit/index.php

[5] Ivip, http://www.firstpr.com.au/ip/ivip/ .

[6] R. Moskowitz and P. Nikander, "Host Identity Protocol (HIP) Architecture", RFC 4423

[7] E. Nordmark and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6" , RFC5533

[8] Name-based sockets, http://christianvogt.mailup.net/pub/2009/vogt-2009-name-oriented-sockets.pdf

[9] P. Srisuresh and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC2663

[10] M. Wasserman and F.Baker, "IPv6-to-IPv6 Network Address Translation (NAT66)", draft-mrw-behave-nat66-02

[11] X.200 : Information technology - Open Systems Interconnection - Basic Reference Model: The basic model, http://www.itu.int/rec/T-REC-X.200-199407-I/en