Summary of name overlay service for scalable Internet routing

Yangyang Wang, Wei Zhang, Jun Bi
Network Research Center of Tsinghua University

This document briefly describes a proposal for the scaling issue, including the basic idea, its features different with existing solutions, its pros and cons.

Overview and basic idea about NOL

This proposal presents a new mechanism called name overlay service. The basic idea is to add a name overlay (NOL) on the TCP/IP stack. It could be implemented as an API library. The NOL functions include: 1) host names configuration, registration and authentication; 2) Initiate and manage transport connection channels (i.e., TCP/IP connections) by name; 3) keep application data transport continuity for mobility.

Under the NOL, all data transports use traditional connections, such as TCP/IP protocols. The applications based on NOL (we called NOL applications in the following) can communicate with legacy applications or with other NOL applications.

Different from proposed host-based ID/Locator split solutions, such as HIP, Shim6, and name-oriented stack, NOL doesn't need to change the existing TCP/IP stack, sockets and their packet formats. Hence, NOL can co-exist with the legacy infrastructure and the core-edges separation solutions (e.g., APT, LISP, Six/one, Ivip, etc.).

How to improve routing scalability?

How to improve routing scalability? There are two ways. One is to adopt PA (Provider-allocated) addressing at host-level. However, due to IPv4 address exhaustion, this way has better to be going on with IPv6. And, PA addressing also causes renumbering problem in edge networks.

Another way proposed here for improving routing scalability is to block the PI (Provider-independent) addresses into transit networks while allow edge networks to use PI addresses to avoid **renumbering**. To achieve this goal, we introduce a new type of gateway NTR (Name Transfer Relay), which can be seen as an extension based on today's widely used NAT/NAPT devices. Similar to the ITR/ETRs in LISP, NTRs block the PI addresses of edge networks into upstream transit networks. It performs address and/or port translation between blocked PI addresses and globally routable addresses. Both legacy and NOL applications behind a NTR can access to the outside as usual. To access the hosts behind a NTR, we need to use NOL traverse the NTR by name and initiate connections to the hosts behind it. For the legacy application accessing the servers located behind a NTR, 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 address translation. In addition, we can also deploy NOL proxy to enable legacy applications to traverse NTRs by name.

How does NOL support traffic engineering, multi-homing and mobility?

1) Scalable support for traffic engineering:

The incoming session from legacy or NOL applications can be directed to a specific NTR by DNS answer for names. In addition, the session from NOL applications can be redirected from one NTR to other appropriate NTRs at the beginning. These mechanisms provide some support for traffic engineering.

2) Scalable support for multi-homing

A network multi-homing with several providers can deploy NTRs to block its PI addresses into provide networks. And the NTRs can be allocated PA addresses from the upstream providers and store them in the address pool of NTRs. 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.

3) Scalable support for mobility

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

Summary: Its Advantages and features different with existing solutions

1) No need to change TCP/IP stack, sockets and DNS system.

This makes its deployment very easy. NOL can be installed with popular applications upgrade, and It don't impact the legacy applications.

2) No need for extra mapping system.

3) Don't increase the load of DNS system drastically.

The name used in NOL should be in a domain hierarchy For example, the name form could be like a email address "hostname@domain" or like a URL "host.domain". We just need to query DNS for the "domain", and the corresponding NTRs know the IP addresses of the "hostname" in that domain.

4) NOL applications can communicate with legacy applications.

5) NOL cooperating with distributed NTR gateways can benefit applications from multi-path routing.

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 PI addresses into transit network by unilaterally deploying NTR. It can contribute to control global routing table growth. NOL applications and proxies can traverse NTR to access the inside endpoints. The LISP solution need bilateral deployment of ITRs and ETRs to run tunneling process, NOL only need unilateral

deployment like a NAT.
8) NOL can be compatible with existing solutions, such as APT, LISP, Ivip, etc. and it can cooperate with them.

Summary: Its limitations and costs

1) Legacy applications have trouble with initiating access to the servers behind NTR. Such trouble can be resolved by deploying NOL proxy for legacy hosts or delegating globally routable PA addresses in the NTR address pool for these servers.
2) It may increase the number of entries of DNS, but not drastic because that it only increases DNS entries in **domains** granularity not hosts. The DNS entries will not only be increased, but its dynamic 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.

Conclusion

Name overlay service is to enable the Internet edge to gain benefits from multi-homing, traffic engineering, and mobility in a large and scaling Internet. It can meet most of goals described in the document “draft-irtf-rrg-design-goals-01” [http://tools.ietf.org/html/draft-irtf-rrg-design-goals-01] and almost all the constraints [http://www.firstpr.com.au/ip/ivip/RRG-2009/constraints/] except that some very strict statements (such as 9). The name overlay service implies that maybe we can learn and explore a practical and low-cost approach for the scaling issue from NATs (NAT/NAPT or NAT66) and the session layer of the OSI model.