Network Working Group
Internet-Draft
Expires: September 16, 2009

Christian Vogt
Ericsson
March 15, 2009

# Qualifying the Harmfulness of Address Translation
# draft-vogt-address-translation-harmfulness-02.txt

## Status of this Memo



## Copyright Notice



## Abstract

Address translation is widely considered harmful because it conflicts with design principles highly regarded within the Internet engineering community. Still, address translation has become common practice despite technical problems because it constitutes an easy-

to-deploy solution to a set of common operational needs. Since some of these needs will continue to exist in IP version 6, there is concern within the Internet engineering community about the potential proliferation of harmful technology from IP version 4 to IP version 6. This document investigates this concern. It compares feasible address translator designs with respect to the harmful impact they may have, explains why the problems of address translation, as used today, are to a significant extent entailed by the shortage of global addresses in IP version 4, and shows how the problems can be mitigated in IP version 6.

Table of Contents

## 1. Introduction

One of the design principles most well-heeded by the Internet engineering community is that addresses have end-to-end validity and do not change in packets en route. This principle is being challenged by the widespread use of address translation on the Internet. Address translators rewrite addresses in packets en route, typically at network borders, to satisfy network operators' desire for provider independence, topology concealment, or conservation of global addresses. The incentives for deploying address translation are strong, even though the technique, as used today for IP version 4, has profound drawbacks and hence is widely considered harmful. Since the purposes of address translation are partly independent of the IP version, there is concern within the Internet engineering community about the potential proliferation of harmful technology from IP version 4 to IP version 6.

This document investigates this concern by qualifying the harmfulness of feasible address translator designs in IP versions 4 and 6. The document makes four contributions to this end: First, it explains the purposes for which address translation is used, identifies the components of address translation that are necessary to achieve these purposes, and distinguishes two main address translator designs based on the components. Second, the document compares the impact of the two address translator designs and evaluates the cost of mitigating resulting problems. Third, it infers that many of the problems of address translation as deployed today can be attributed to address overloading, a technique that helps conserving global addresses, rather than because addresses are rewritten in packets en route. Fourth, the document argues that, while address overloading is inevitable in IP version 4 due to the shortage of global addresses [REF], address translation in IP version 6 does not require address overloading and could hence, if designed rightly, be considerably less problematic than address translation in IP version 4.

## 2. Purposes of Address Translation

Network operators frequently apply address translation to separate the "local" addresses they use inside their networks from the "global" addresses at which the networks are reachable from the Internet. They do this for any of the following three purposes:

- Provider independence: Network operators desire the flexibility to change providers at low cost, in order to avoid lock-in to any particular provider.

- Topology concealment: Network operators may want to hide a network's internal topology from the rest of the Internet for security reasons.

- Global address conservation: Network operators see increasing pressure to conserve global IP version 4 addresses due to the imminent runout of unallocated global IP version 4 addresses.

A use case of address translation to achieve provider independence is in networks that cannot afford, or are not eligible for, global provider-independent addresses. Most residential networks and small enterprise networks belong to this group. They must use addresses assigned by their providers and renumber in the event of a provider change. While renumbering can be a smooth process in sufficiently optimized networks, experience [REF] shows that the process often involves substantial manual labor, and is hence costly and time-consuming. For example, routers and servers typically have statically configured addresses and therefore have to be renumbered manually. Addresses may also have to be manually renumbered in applications, firewalls, and operations and management systems [REF]. Address translation eliminates the need to renumber without global provider-independent addresses. It enables network operators to use local provider-independent addresses internally, while retaining external reachability at global addresses assigned by a provider.

A security-related use case of address translation is for denial-of-service attack protection. The standard, network-topological assignment of addresses provides remote hosts with a means to infer the topology of a network. Attackers may use this information to identify attack targets. For example, a denial-of-service attack against a server may more easily be executed via a host on the server's link, and such a host can typically be identified based on comparing its address to the address of the server in question. Address translation can conceal the internal topology of a network, by mapping local and global addresses such that the topological structuring of local addresses cannot be derived from global addresses.

The conservation of global addresses provides a third use case for address translation, which is of common interest among operators of IP version 4 networks. The shortage of global IP version 4 addresses makes network expansion difficult, which in turn can have a negative impact on revenue. Address translation helps conserving global addresses because it allows multiple hosts with separate local addresses to share one global address.

## 3. Functional Components of Address Translation

In order to accomplish the purposes identified above, address translation incorporates two functional components:

- Address rewriting: The local and global addresses of a network are mapped, and swapped accordingly in packets leaving or entering the network.

- Address overloading: Multiple local addresses are mapped onto a single global address.

To enable demultiplexing of packets received at an overloaded global address back onto the right local address, address translators that use address overloading store address mappings as connection-specific "disambiguation state", and they use the connection initiator's port number in received packets as indexes into this state. To ensure uniqueness of this port number across all connections handled by an address translator, the port number may have to be translated. The port mapping is then stored as part of the corresponding disambiguation state.

Address rewriting affords provider independence and topology concealment. Provider independence is achieved through the decoupling of a network's local provider-independent addresses from the global addresses assigned by the network's provider. The local addresses consequently do not need to change if the network changes providers, thus eliminating the need to renumber. Topology concealment can be achieved either by overloading a single global address with a large set of local addresses, or by choosing, and keeping secret, a non-trivial permutation to be applied on relevant address bits during mapping.

Simple address rewriting without address overloading requires at least one global address per host, which maps one-to-one onto its corresponding global address. To conserve global addresses, it is necessary to have multiple hosts share one global address. This can be achieved by combining address rewriting with address overloading.

Given that address overloading is required for only part of the use cases of address translation, two types of address translation can be distinguished:

- One-to-one address translation, which consists of address rewriting without address overloading. This achieves provider independence and topology concealment.

- Many-to-one address translation, which combines address rewriting with address overloading. This achieves global address conservation in addition to provider independence and topology concealment.

These two types of address translation will be evaluated separately throughout the rest of this document.

## 4. Analysis of Problematic Impacts and Possible Solutions

Since address translation changes the way hosts are addressed and packets are forwarded, it has an impact on host reachability and network functioning. The analysis below explains this impact for one-to-one and many-to-one address translation, identifies problems that may arise from it, and examines the feasibility and cost of mitigating the problems.

### 4.1. Impact on Host Reachability

Since hosts behind an address translator effectively have at least two addresses -- a local address and a global address --, peers must have a means to discover one of these addresses that they can reach. Which of the host's addresses are reachable by a given peer then depends on the location of the peer. The peer must use the host's global address if all paths to the host lead through an address translator, and it should use the host's local address otherwise. Failure to choose the right address may lead to non-reachability of the host, or to sub-optimal routing, respectively.

Address translation must consequently be accounted for in both of the two main address discovery methods -- DNS-based address discovery and host-based address referrals. Authoritative DNS servers must refer peers to these addresses of a host that the peers can reach, preferably via an optimal path. Hosts must be able to determine their global addresses for address referrals in packets they send to peers, and they must recognize their global addresses in packets received. The following shows that, while both address discovery methods can be trimmed to accommodate address translation, the cost and reliability of suitable solutions in either case depends significantly on the type of address translation.

Appropriate configuration is sufficient to enable DNS-based address discovery in the presence of one-to-one address translation. Since without address overloading, a host's local and global addresses are both stable and unique, both can be associated with the host's name in the DNS through configuration of the authoritative DNS servers. This corresponds to common practice.

For many-to-one address translation, DNS-based address discovery is more expensive to enable. Since a host is reachable at a global address only after prior establishment of disambiguation state in an address translator, extra functionality is necessary in authoritative DNS servers to perform this state establishment prior to referring a peer to the global addresses of a host. An existing proposal [REF] calls for authoritative DNS servers to establish disambiguation state in a host's address translator when receiving a DNS query for the host's name, and for the address translator to bind this state to the peer's address and port number when receiving the first packet from the peer.

Host-based address referrals naturally require special host support to function in the presence of address translation. Hosts must be enabled to discover their global addresses, and they must use and recognize their global addresses in referrals they send and receive. Furthermore, hosts behind a many-to-one address translator may have to establish demultiplexing state in the address translator prior to sending an address referral. This is necessary when the address referral itself is sent via an overlay instead of to the peer directly, and hence cannot establish the necessary demultiplexing state in the address translator. Applications that may send address referrals via overlays include those that use server-assisted peer-to-peer protocols or the Session Initiation Protocol [REF].

While the use and recognition of global addresses is specific to the application protocol, standard methods [REF] exists for hosts to discover their global addresses. These methods were designed for many-to-one address translation, as it is the prevailing address translation type in the existing Internet, and they are hence appropriate to establish disambiguation state in address translators. They discover a global address by inquiring of infrastructure what a packet's source address looks like after translation.

Although the existing solutions to enable address discovery in the presence of many-to-one address translation would likewise apply to one-to-one address translation, solutions can be simpler in the case of one-to-one address translation. Since one-to-one address translation does not overload addresses, address discovery methods for one-to-one address translation do not have to establish disambiguation state in address translators. For example, a simplified method for hosts to discover their global addresses is for access routers to announce address mapping rules, based on which hosts derive their global addresses given their local addresses. In the case where addresses are translated by swapping their prefix, a mapping rule could be as simple as a pair of local and global address prefixes. This discovery method would be similar to the existing practice of auto-configuring addresses based on on-link address

prefixes announced by access routers.

Both DNS-based and referral-based address discovery are consequently more expensive and less reliable for many-to-one address translation than they can be for one-to-one address translation. Apart from requiring extra complexity, the establishment of disambiguation state during address discovery assumes that connection initiation happens shortly after address discovery, which may not always be the case. Disambiguation state must expire when found unused to make room for new connections. So if the first packet from a connection arrives at the address translator after the corresponding disambiguation state has expired, connection initiation fails. For the same reason, peers cannot be configured with the global address of a host, since the necessary disambiguation state would likely be unavailable at the time the peer initiates a connection to this address.

## 4.2. Impact on Network Functioning

The addition of new components to a network, such as in the form of address translators, can impact the functioning of the network. Two potential problems that may arise are the loss of generic forwarding support for all connection types, and the loss of network robustness. Forwarding support can be reduced to certain connection types if the new component relies on connection-specific properties. This is the case for many-to-one address translators, which rely on modifiable port numbers. Packets without port numbers are dropped, and so are packets with port numbers that are not modifiable due to encryption and authentication. Network robustness can be reduced if the new component constitues a single point of failure. This is also the case for address translators. Address translators perform a function that connections depend on, and hence, if not provisioned redundantly, limit a network's ability to reroute traffic in the event of failures. The following shows that, while both problems can be effectively mitigated, the cost and reliability of suitable solutions in either case depends significantly on the type of address translation.

Loss of generic forwarding support for all connection types is a problem that is peculiar to many-to-one address translation, because address overloading requires packets to carry port numbers modifiable by an address translator for use as indexes into disambiguation state. One-to-one address translation does not limit forwarding support since it does not rely on port numbers or other connection-specific properties.

A common method [REF] to retain support for all connection types in many-to-one address translation is to tunnel packets end-to-end in an extra layer of UDP. This enables connections without accessible port

numbers to pass through many-to-one address translators, because the extra UDP header in packets adds the port numbers needed by the address translators. Unfortunately, UDP tunneling cannot be expected to always be available. It requires support at both ends of a connection, independent of whether the address of the peer is translated or not. So if either the host which address is being translated or its peer does not support UDP tunneling, connections without modifiable port numbers cannot pass through many-to-one address translator.

To increase the robustness of networks with address translators, one must ensure that connections do not fail due to a single address translator. Depending on the type of address translation deployed, this may require one or both of the following two components:

- Redundancy of address translators: Redundant provisioning of address translators on alternative paths is necessary to protect against failure of address translators. It enables failover of connections from one path to another.

- Refreshes of disambiguation state: Disambiguation state in many-to-one address translators must be refreshed periodically throughout the lifetime of the corresponding connection to prevent premature disposal of disambiguation state. Idle connections may otherwise be unable to resume.

Redundant provisioning is straightforward for one-to-one address translators. Since they operate without connection-specific state, redundantly provisioned one-to-one address translators can substitute for each other without prior synchronization, and hence can be deployed independently on alternative paths. On the other hand, redundantly provisioned many-to-one address translators, which do maintain disambiguation state, must be continuously synchronized.

Many-to-one address translation in addition requires periodic refreshes of disambiguation state to prevent premature state removal for connections that are idle, though still active. Methods to refresh disambiguation state either ensure that connections periodically exchange "keep-alive" packets end to end [REF], or they introduce infrastructure [REF] with which hosts behind address translators can exchange such packets. Unfortunately, neither of these methods can be expected to always be available due to their demanding requirements. The methods either require support at both ends of a connection, or they require special infrastructure plus support in the host which address is being translated. If neither requirement is met, connections that go idle temporarily may be unable to resume afterwards due to premature removal of disambiguation state. Disambiguation state refreshes are not

necessary for one-to-one address translation, since this functions without disambiguation state.

## 5. Conclusion

This document has shown that the harmfulness of address translation depends significantly on whether or not global addresses are overloaded with mappings to multiple local addresses. Although address translation with and without address overloading can have an impact on host reachability and network functioning, resulting problems have been found to be especially intractable and expensive to solve if address overloading is used.

Impacts on host reachability in one-to-one address translation, which functions without address overloading, can be compensated for by appropriate configuration of authoritative DNS servers and special support for address referrals in hosts behind an address translator. Many-to-one address translation, which does perform address overloading, in addition requires the hosts and authoritative DNS servers to interact with address translators. This is necessary to establish disambiguation state that will subsequently permit an address translator to demultiplex packets received at an overloaded global address back onto the right local address. Impacts on network functioning in one-to-one address translation can be compensated for by provisioning address translators redundantly. Many-to-one address translation in addition requires synchronization of disambiguation state across redundantly provisioned address translators, periodic state refreshes, and UDP tunneling of connections that lack modifiable port numbers address translators could use to index into their disambiguation state.

Compensating for the impacts of address translation is hence significantly more expensive, both in deployment and in administration, when address translation is many-to-one compared to when it is one-to-one. The higher complexity involved furthermore constitutes a source of potential failure on its own. And extra requirements for hosts and their peers reduce the likelihood that sufficient functionality will be available when needed.

The superiority of one-to-one address translation over many-to-one address translation naturally leads to the question whether the former can satisfy the demand for address translation, as it has become apparent through the wide deployment of address translators in today's Internet. Clearly, this depends on the availability of global addresses. One-to-one address translation, which requires one global address per local address, is unsuitable for IP version 4, where the shortage of global addresses necessitates the use of

address overloading. For IP version 6, however, one-to-one address translation is suitable, as the sufficient number of global addresses here makes address overloading dispensable. Address translation in IP version 6 could hence, if designed without address overloading, be considerably less harmful than address translation in IP version 4.

## Appendix A. Acknowledgment

The author would like to thank Gonzalo Camarillo, James Kempf, Tony Li, Joel Halpern, Suresh Krishnan, Zoltan Turanyi, and Wassim Haddad for their reviews of this document and their valuable comments.

This document was generated using the xml2rfc tool.

## Author's Address

Christian Vogt
Ericsson Research
200 Holger Way
San Jose, CA 95134
United States

Email: christian.vogt@ericsson.com