Title:	Document Version:
Deliverable D2.4 **Alternative approaches to intra-site stability**	1.2

Project Number:	Project Acronym:	Project Title:
035167	RiNG	Routing in Next Generation

Contractual Delivery Date:	Actual Delivery Date:	Deliverable Type* - Security**:
31/12/2007	15/11/2008	PU

* Type:	P – Prototype, R – Report, D – Demonstrator, O – Other
** Security Class:	PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

Responsible and Editor/Author:	Organization:	Contributing WP:
Stig Venaas	UNINETT	WP2

Authors (organizations):

Abstract:

This document describes alternative approaches to intra-site stability. With the current way of doing inter-domain routing, a high degree of address aggregation is required for the system to scale. The current thinking is that most sites deploying IPv6 need to use Provider Assigned addresses, so that only one or a few prefixes are needed per provider in the global routing system. This obviously keeps the global routing tables at a minimum and allows for good scalability. The downside to this is that sites need to renumber whenever they change providers. This means that an organization or an enterprise may not be able to effectively choose the cheapest or best provider. With IPv4 and the use of NAT one could have internal addressing that could remain unchanged while the external provider assigned addresses changed. In this document we will consider IPv6 Unique Local Addresses as an alternative way of providing stable internal addressing. We will show that this can indeed be done, but that there are also some associated costs and issues.

Keywords:

IPv6, Addressing, Scalability, Stability, ULA

Revision History

The following table describes the main changes done in the document since its creation.

Revision	Date	Description	Author (Organization)
v1.0	25/12/2007	Document creation	Stig Venaas (UNINETT)
v1.1	25/12/2007	Minor editorial fixes	César Olvera (Consulintel)
v1.2	15/11/2008	Final Review	Jordi Palet (Consulintel)

Executive Summary

In this deliverable we consider the use of IPv6 Unique Local Addresses (ULA) to allow internal addressing and communications of a site to be independent of the global addresses assigned by an ISP, and only use the global addresses for external communications. If the site uses only provider assigned addresses, then internal communications, configuration of internal services etc. will be affected if the site changes providers. The obvious solution would be for the site to have provider independent addresses, but these addresses can generally not be aggregated in the global routing tables and widespread use of these would cause an explosion in the size of the global routing tables. This is partly a problem for IPv4 as well. However, for IPv4 the use of Network Address Translation (NAT) offers some help, in that the addresses used internally in a site are kept completely separate from the external addresses. There are many issues with NAT, and it should be avoided if possible for IPv6.

This deliverable discusses how hosts in a site may be multi-addresses, having both ULA and provider assigned addresses for internal and external communications respectively. IPv6 is designed to allow multi-addressing, including mechanisms for address selection. We show that there are benefits to using such multi-addressing, but that there are also issues. We also consider the option of using private IPv4 space for internal communications. We show that these methods may indeed reduce the cost of changing providers, mainly in terms of management, but also avoiding disruptions of internal communications and services. However, we also show that there is some added complexity, increasing the overall management costs.

We will see that whether these methods are useful, depends on the amount and complexity of internal communications versus how often the site may change providers.

Table of Contents

1. INTRODUCTION

When deploying IPv6 most organisations/sites have to use provider assigned (PA) addresses. This allows for good aggregation in the global routing tables since the number of routes in the global routing table can be on the order of the number of providers rather than the number of organisations or sites. The downside of this, is that an organisations/site need to renumber (change the IPv6 addresses of their hosts) when changing providers. Renumbering has a significant cost, which again means that buying connectivity from a provider is a long term investment, and that changing to a slightly cheaper provider may indeed cost more than staying.

From an end-user perspective the optimal solution would be to own a set of addresses that are provider independent (PI), where one could keep using the same addresses independent of the provider. However, due to the impact this would have on routing, there are limitations on who can get such addresses.

For IPv4 it is pretty common to make use of network address translation (NAT). While NAT has several problems, one major benefit is that one can change providers (and possibly be multi-homed) and only alter the external addressing. The internal addressing can be completely independent of the external addressing, and need not be altered.

For IPv6 there are enough addresses for all hosts to get globally unique routable addresses, which means that no NAT is required and one can get full end-to-end connectivity. We will here look at how one can without the use of NAT, have relatively stable addressing for internal communications, and how that reduces the cost of changing providers.

One obvious way to try to solve this problem is the use of Unique Local Addresses (ULA) [1], for internal communications, and that is the focus of this document. There is some on-going work, in e.g. the IRTF, on new routing and addressing solutions, including various so-called locator identifier split solutions. These solutions may allow hosts and sites to have stable addresses that are used as identifiers. It remains to be seen whether such solutions eventually get deployed. It is clear that they are many years ahead.

2. UNIQUE LOCAL ADDRESSES

Unique Local IPv6 Unicast Addresses (ULA) are specified in RFC 4193 [1]. These are intended to be globally unique, used for local communications, and not to be routable on the global Internet.

2.1 ULA Format

The format of the ULA addresses is defined as follows:

7 bits	1	40 bits	16 bits	64 bits
Prefix	L	Global ID	Subnet ID	Interface ID

The prefix is **FC00::/7**, Global ID is a more or less globally unique identifier, and L is set to **1** meaning the identifier is locally assigned. We will discuss that below. The Subnet ID is used for addressing subnets. This ID is 16 bits, so a site may pick a Global ID, and for that value, have up to 65536 subnets.

2.2 ULA Global ID

RFC 4193 specifies how Global ID can be locally assigned. In this case the L-bit is set to 1. For local assignment one must use a pseudo-random algorithm, one is suggested in the RFC. The reason for using such an algorithm is to avoid, or at least make it unlikely, that the ID clashes with the ID of another network. There may be two sites in the world with the same Global ID, but given that they should be for local use and not on the global Internet, it is unlikely that two sites with the same ID come in conflict with each other. The use of this pseudo-random ID means that if networks using these addresses merge, or a host moves between two networks, etc., the two networks are very unlikely to have the same Global ID. These addresses can be regarded mostly like the IPv4 private address space in RFC 1918, but the pseudo-random ID means that one is unlikely to have the clashes experienced between sites using the same IPv4 private address space.

There is an on-going discussion whether one should also specify centrally assigned Global IDs, where the L-bit would be 0. One could then guarantee global uniqueness. This is an interesting discussion, but not all that relevant to our discussion in this document. It would further reduce the risks of conflicts where two sites are using the same Global ID, but we believe the risk is already sufficiently small.

2.3 Routing of ULA

The Unique Local Addresses are not to be routable on the global Internet. The main reasons for that is that they would generally not be aggregated, resulting in a large number of /48 prefixes in the global routing system; this would also cause problems if two sites connected to the Internet happened to choose the same Global ID. However, they can be routed between small groups of cooperating sites. An attempt to use it between large groups increases the risk of conflicting IDs.

2.4 ULA scope and reachability

ULA addresses are by default to be treated as globally scoped addresses. They are however only locally reachable. This may cause problems for some applications. There may be a need to give ULA preference over other types of global unicast addresses. One way of doing that is by controlling what addresses can be looked up in the DNS, see next section. There are also address selection mechanisms that we will discuss later.

2.5 ULA and DNS

The use of ULA may have a significant impact on the use of DNS in a site. Generally speaking one would typically register all the addresses of all the hosts in the DNS, where everyone from everywhere could look up any host in the DNS and find the exact same set of addresses for the host. However, the use of ULA changes this (as with the use of private IPv4 addresses).

First of all, one should not put ULA addresses in the global DNS. Since they are not globally reachable, but are of global scope, an application anywhere on the Internet looking up the host's addresses, may try to contact the host using a ULA address. If the address is not reachable, it may take a long time before the application times out and tries another address; some applications may indeed never try another address. In the worst case, the same ULA Global ID may be used in the site where the application resides, and the application may contact the wrong host. Hence, hosts should only be registered with their globally reachable unicast addresses in the global DNS. Hosts that never are to be reached from the global Internet may not even need to be in the global DNS.

Inside a site using ULA, one would obviously like to be able to look up the ULA addresses of a host in the DNS; perhaps all addresses of the host. We will discuss this later.

The wish to find ULA addresses when looking up some fully qualified domain name (FQDN) in the DNS internally, but not externally, leads to the deployment of so-called split DNS. The use of split DNS is likely to make DNS harder to manage and increase management cost. It may also be tricky to make a host always get the right information. E.g., a mobile host visiting an external network should normally get only the external view. While if it establishes a VPN connection to its home site, one may want it to have an internal view where lookups return ULA addresses, so that these addresses can be used through the tunnel.

3. ULA FOR INTRA-SITE STABILITY

We will consider how one may use ULA for intra-site stability. The idea is that one can use ULA for internal communications, so that when one changes providers and the provider assigned (PA) addresses change, not only on-going internal communications, but also configuration files and applications that deal with internal communications, remain unaffected. We will discuss how this might be done, to what degree it can provide stability and simplify management when changing external addressing, and also possible issues, including increased complexity and management. We will see that there are benefits, but there is also a cost.

The main problems with renumbering, change of addressing, are perhaps configuration files that have hardcoded addresses, application clients and servers that need to be restarted due to long time caching of addresses, and sessions both at the transport layer (like TCP), and the application layer that may be reset. By using stable addressing internally one can hopefully keep things more stable, and also limit the amount of configuration changes.

For end-to-end connectivity all hosts in the site that are to communicate with the outside world will need to have a global address. These are the PA addresses that may need to change. In our case, we also assume that all hosts in the site have a ULA address from a given ULA /48 prefix that the site is using (based on some ULA Global ID). Note that it is not a strict requirement that all hosts in the site have a ULA, some may and some may not, but we assume all to simplify our discussion. Note that when using stateless address autoconfiguration all hosts on a given link will get addresses within the same prefixes. In general the easiest is probably to give all hosts both global and private addresses. This includes having internal routing for both ULA and PA addresses, and announcing both on all (most) links.

The site will probably need to add ULA addresses in the DNS for FQDNs of internal hosts. As discussed in 2.5 the site will then need to deploy split DNS since the ULA addresses should not be visible externally. As discussed, this adds some complexity. There are however several DNS server implementations supporting this, and the amount of management work may not increase, depending on the management systems used.

3.1 Address selection

Let us now consider the case where all the internal hosts in the site both have a ULA address (internal addressing) and one or more PA addresses (external addressing). How can one we ensure that internal addresses are used for internal communications, while external addresses are to be used for external communications? Basically we would like to use internal addressing if both hosts have addresses from the same /48 ULA prefix, otherwise external addressing. One might want to do this for ULA more generally if there are multiple cooperating sites using ULA.

IPv6 implementations should do address selection according to RFC 3484 [2]. The RFC defines the default behaviour, but allows for this to be modified. Whether this behaviour can be modified varies between the different implementations. RFC 3484 distinguishes between IPv4 and IPv6, 6to4 versus native, and also takes scoping into account. However, it has no knowledge of ULA, and ULA addresses are treated as being of global scope. That is, the mechanism makes no distinction between ULA and PA addresses.

If say a client tries to reach a server, it may look up the server's FQDN in the DNS and obtain both a ULA and a PA address. The client will need to choose which address to try to connect to (in which order to try them). Assuming both the addresses are valid, not deprecated etc, the rules that come into play for choosing ULA versus PA will be "longest matching prefix" and if that makes no difference, the order the addresses were returned from the DNS. Longest prefix match ensures that if the client has only a ULA address, then it will prefer the server's ULA address. If the client only has a PA address, then it is fairly certain that the server's PA address will get a better match. But what happens if the client and the server both have ULA and PA addresses?

If we look at two hosts using stateless address autoconfiguration [3] that both reside on the same link, where there are two /64 prefixes on the link, one ULA and one PA, then the longest prefix match rule will not make a distinction between using ULA for both destination and source, or PA for both destination and source. Also, many sites are likely to use say a /48 prefix for the PA addresses, matching the ULA prefix length. For ease of maintenance, the site may then use the exact same bits for the remaining bits of the addresses. Then longest prefix match will not give a preference for ULA versus PA for any pair of hosts in the site.

This means that when deploying both ULA and PA, the order returned by the DNS is likely to be the deciding factor. Hence one may be able to prefer ULA over PA for internal communications if the DNS sorts ULA addresses first. Perhaps an easier method is to only have ULA addresses in the internal DNS.

Note that some implementations allow the policy table in RFC 3484 to be customised. This means that the host administrator may be able to add a rule for preference of the site's ULA prefix (and possibly for ULA prefixes of cooperating sites that also are known to be reachable). However, not all implementations do this, and it is also implementation specific how exactly to change this table. There is some on-going work for using e.g. DHCPv6 [4] to configure the policy table and also for changing the default table. At the time of writing, it is basically left to the administrator to figure out whether and how the default table may be changed on the different platforms. Provided that the table is updated to prefer ULA, one can safely also put PA addresses in the internal DNS.

Address selection can get complicated for some applications. If for instance the site is streaming some video, the source address need to be PA if the stream is to go outside the site, while if internal one probably would prefer ULA. This can not be determined automatically. Also, a user in the site wants to receive the stream needs to know which address to use (although this could be via SDP or a URL etc). The user cannot necessarily just use the FQDN of the streaming host. This could lead the client to expect data from the wrong address; in particularly if a PA address is used for the stream and only ULA addresses are in the internal DNS.

4. IPv4 ADDRESSES FOR INTRA-SITE STABILITY

Sites deploying IPv6 will generally do it alongside IPv4, so that both protocols can be used. This may change in the long term. One common scenario will probably be to use IPv4 NAT with private addresses internally, and IPv6 with end-to-end connectivity and PA addresses. Using the private IPv4 addresses for internal connectivity would provide stable addressing.

Address selection would get difficult in this case. For external hosts one would like to prefer IPv6 over IPv4 (at least in the case with IPv4 NAT), while for internal hosts one would like IPv4 over IPv6. In this scenario one would also end up using split DNS (to have the private IPv4 addresses only visible in internal DNS), and one could consider solving this by only having IPv4 addresses for hosts in internal DNS.

To only use IPv4 for intra-site connectivity would be a bad idea in the long term though. The site may at some point in the future get some IPv6-only devices or prefer to make some part of the site IPv6-only. But this could be a possible path for a site that is currently using IPv4 NAT and private addressing internally.

5. CONCLUSIONS

We have discussed how one may use stable addressing like ULA, or perhaps even private IPv4 address space, to provide intra-site addressing to be independent of the PA addresses used for external connectivity. This means that internal addresses can remain fixed and be independent of any renumbering caused by a change of providers. The benefits are that internal connections are not reset at the time of renumbering, but perhaps more importantly that configuration of internal services need not be changed. On the other hand, the use of the internal addressing will add complexity like split DNS and possibly result in address selection issues as well.

Whether these techniques are useful, depends on how often one changes providers, on the number of services deployed in the site, and the amount of internal communication. They add complexity in day-to-day operations, but make it easier to change providers, in particular if there are many internal services deployed.

6. REFERENCES

[1] R. Hinden, B. Haberman., “Unique Local IPv6 Unicast Addresses”, 2005, Internet Engineering Task Force (IETF), RFC 4193

[2] R. Draves, “Default Address Selection for Internet Protocol version 6 (IPv6)”, 2003, Internet Engineering Task Force (IETF), RFC 3484

[3] S. Thomson, T. Narten, “IPv6 Stateless Address Autoconfiguration”, 1998, Internet Engineering Task Force (IETF), RFC 2462

[4] R. Droms, Ed., J. Bound, B. Volz, T. Lemon, C. Perkins, M. Carney, “Dynamic Host Configuration Protocol for IPv6 (DHCPv6)”, 2003, Internet Engineering Task Force (IETF), RFC 3315