Network Working Group
Request for Comments: 3879
Category: Standards Track

C. Huitema
Microsoft
B. Carpenter
IBM

September 2004

Deprecating Site Local Addresses

Status of this Memo

Copyright Notice

Abstract

This document describes the issues surrounding the use of IPv6 site-local unicast addresses in their original form, and formally deprecates them. This deprecation does not prevent their continued use until a replacement has been standardized and implemented.

1. Introduction

For some time, the IPv6 working group has been debating a set of issues surrounding the use of "site local" addresses. In its meeting in March 2003, the group reached a measure of agreement that these issues were serious enough to warrant a replacement of site local addresses in their original form. Although the consensus was far from unanimous, the working group confirmed in its meeting in July 2003 the need to document these issues and the consequent decision to deprecate IPv6 site-local unicast addresses.

Site-local addresses are defined in the IPv6 addressing architecture [RFC3513], especially in section 2.5.6.

The remainder of this document describes the adverse effects of site-local addresses according to the above definition, and formally deprecates them.

Companion documents will describe the goals of a replacement solution and specify a replacement solution. However, the formal deprecation allows existing usage of site-local addresses to continue until the replacement is standardized and implemented.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

2. Adverse Effects of Site Local Addresses

Discussions in the IPv6 working group outlined several defects of the current site local addressing scope. These defects fall in two broad categories: ambiguity of addresses, and fuzzy definition of sites.

As currently defined, site local addresses are ambiguous: an address such as FEC0::1 can be present in multiple sites, and the address itself does not contain any indication of the site to which it belongs. This creates pain for developers of applications, for the designers of routers and for the network managers. This pain is compounded by the fuzzy nature of the site concept. We will develop the specific nature of this pain in the following section.

2.1. Developer Pain, Scope Identifiers

Early feedback from developers indicates that site local addresses are hard to use correctly in an application. This is particularly true for multi-homed hosts, which can be simultaneously connected to multiple sites, and for mobile hosts, which can be successively connected to multiple sites.

Applications would learn or remember that the address of some correspondent was "FEC0::1234:5678:9ABC", they would try to feed the address in a socket address structure and issue a connect, and the call will fail because they did not fill up the "site identifier" variable, as in "FEC0::1234:5678:9ABC%1". (The use of the % character as a delimiter for zone identifiers is specified in [SCOPING].) The problem is compounded by the fact that the site identifier varies with the host instantiation, e.g., sometimes %1 and sometimes %2, and thus that the host identifier cannot be remembered in memory, or learned from a name server.

In short, the developer pain is caused by the ambiguity of site local addresses. Since site-local addresses are ambiguous, application developers have to manage the "site identifiers" that qualify the

addresses of the hosts. This management of identifiers has proven hard to understand by developers, and also hard to execute by those developers who understand the concept.

2.2. Developer Pain, Local Addresses

Simple client/server applications that do share IP addresses at the application layer are made more complex by IPv6 site-local addressing. These applications need to make intelligent decisions about the addresses that should and shouldn't be passed across site boundaries. These decisions, in practice, require that the applications acquire some knowledge of the network topology. Site local addresses may be used when client and server are in the same site, but trying to use them when client and server are in different sites may result in unexpected errors (i.e., connection reset by peer) or the establishment of connections with the wrong node. The robustness and security implications of sending packets to an unexpected end-point will differ from application to application.

Multi-party applications that pass IP addresses at the application layer present a particular challenge. Even if a node can correctly determine whether a single remote node belongs or not to the local site, it will have no way of knowing where those addresses may eventually be sent. The best course of action for these applications might be to use only global addresses. However, this would prevent the use of these applications on isolated or intermittently connected networks that only have site-local addresses available, and might be incompatible with the use of site-local addresses for access control in some cases.

In summary, the ambiguity of site local addresses leads to unexpected application behavior when application payloads carry these addresses outside the local site.

2.3. Manager Pain, Leaks

The management of IPv6 site local addresses is in many ways similar to the management of RFC 1918 [RFC1918] addresses in some IPv4 networks. In theory, the private addresses defined in RFC 1918 should only be used locally, and should never appear in the Internet. In practice, these addresses "leak". The conjunction of leaks and ambiguity ends up causing management problems.

Names and literal addresses of "private" hosts leak in mail messages, web pages, or files. Private addresses end up being used as source or destination of TCP requests or UDP messages, for example in DNS or trace-route requests, causing the request to fail, or the response to arrive at unsuspecting hosts.

The experience with RFC 1918 addresses also shows some non trivial leaks, besides placing these addresses in IP headers. Private addresses also end up being used as targets of reverse DNS queries for RFC 1918, uselessly overloading the DNS infrastructure. In general, many applications that use IP addresses directly end up passing RFC 1918 addresses in application payloads, creating confusion and failures.

The leakage issue is largely unavoidable. While some applications are intrinsically scoped (e.g., Router Advertisement, Neighbor Discovery), most applications have no concept of scope, and no way of expressing scope. As a result, "stuff leaks across the borders". Since the addresses are ambiguous, the network managers cannot easily find out "who did it". Leaks are thus hard to fix, resulting in a lot of frustration.

2.4. Router Pain, Increased Complexity

The ambiguity of site local addresses also creates complications for the routers. In theory, site local addresses are only used within a contiguous site, and all routers in that site can treat them as if they were not ambiguous. In practice, special mechanisms are needed when sites are disjoint, or when routers have to handle several sites.

In theory, sites should never be disjoint. In practice, if site local addressing is used throughout a large network, some elements of the site will not be directly connected for example, due to network partitioning. This will create a demand to route the site-local packets across some intermediate network (such as the backbone area) that cannot be dedicated for a specific site. In practice, this leads to an extensive use of tunneling techniques, or the use of multi-sited routers, or both.

Ambiguous addresses have fairly obvious consequences on multi-sited routers. In classic router architecture, the exit interface is a direct function of the destination address, as specified by a single routing table. However, if a router is connected to multiple sites, the routing of site local packets depends on the interface on which the packet arrived. Interfaces have to be associated to sites, and the routing entries for the site local addresses are site-dependent. Supporting this requires special provisions in routing protocols and techniques for routing and forwarding table virtualization that are normally used for VPNs. This contributes to additional complexity of router implementation and management.

Network management complexity is also increased by the fact that though sites could be supported using existing routing constructs--such as domains and areas--the factors driving creation and setting the boundaries of sites are different from the factors driving those of areas and domains.

In multi-homed routers, such as for example site border routers, the forwarding process should be complemented by a filtering process, to guarantee that packets sourced with a site local address never leave the site. This filtering process will in turn interact with the forwarding of packets, for example if implementation defects cause the drop of packets sent to a global address, even if that global address happen to belong to the target site.

In summary, the ambiguity of site local addresses makes them hard to manage in multi-sited routers, while the requirement to support disjoint sites and existing routing protocol constructs creates a demand for such routers.

2.5. Site is an Ill-Defined Concept

The current definition of scopes follows an idealized "concentric scopes" model. Hosts are supposed to be attached to a link, which belongs to a site, which belongs to the Internet. Packets could be sent to the same link, the same site, or outside that site. However, experts have been arguing about the definition of sites for years and have reached no sort of consensus. That suggests that there is in fact no consensus to be reached.

Apart from link-local, scope boundaries are ill-defined. What is a site? Is the whole of a corporate network a site, or are sites limited to single geographic locations? Many networks today are split between an internal area and an outside facing "DMZ", separated by a firewall. Servers in the DMZ are supposedly accessible by both the internal hosts and external hosts on the Internet. Does the DMZ belong to the same site as the internal host?

Depending on whom we ask, the definition of the site scope varies. It may map security boundaries, reachability boundaries, routing boundaries, QOS boundaries, administrative boundaries, funding boundaries, some other kinds of boundaries, or a combination of these. It is very unclear that a single scope could satisfy all these requirements.

There are some well known and important scope-breaking phenomena, such as intermittently connected networks, mobile nodes, mobile networks, inter-domain VPNs, hosted networks, network merges and splits, etc. Specifically, this means that scope *cannot* be mapped

into concentric circles such as a naive link/local/global model. Scopes overlap and extend into one another. The scope relationship between two hosts may even be different for different protocols.

In summary, the current concept of site is naive, and does not map operational requirements.

3. Development of a Better Alternative

The previous section reviewed the arguments against site-local addresses. Obviously, site locals also have some benefits, without which they would have been removed from the specification long ago. The perceived benefits of site local are that they are simple, stable, and private. However, it appears that these benefits can be also obtained with an alternative architecture, for example [Hinden/Haberman], in which addresses are not ambiguous and do not have a simple explicit scope.

Having non-ambiguous address solves a large part of the developers' pain, as it removes the need to manage site identifiers. The application can use the addresses as if they were regular global addresses, and the stack will be able to use standard techniques to discover which interface should be used. Some level of pain will remain, as these addresses will not always be reachable; however, applications can deal with the un-reachability issues by trying connections at a different time, or with a different address. Speculatively, a more sophisticated scope mechanism might be introduced at a later date.

Having non ambiguous addresses will not eliminate the leaks that cause management pain. However, since the addresses are not ambiguous, debugging these leaks will be much simpler.

Having non ambiguous addresses will solve a large part of the router issues: since addresses are not ambiguous, routers will be able to use standard routing techniques, and will not need different routing tables for each interface. Some of the pain will remain at border routers, which will need to filter packets from some ranges of source addresses; this is however a fairly common function.

Avoiding the explicit declaration of scope will remove the issues linked to the ambiguity of the site concept. Non-reachability can be obtained by using "firewalls" where appropriate. The firewall rules can explicitly accommodate various network configurations, by accepting of refusing traffic to and from ranges of the new non-ambiguous addresses.

One question remains, anycast addressing. Anycast addresses are ambiguous by construction, since they refer by definition to any host that has been assigned a given anycast address. Link-local or global anycast addresses can be "baked in the code". Further study is required on the need for anycast addresses with scope between link-local and global.

4. Deprecation

This document formally deprecates the IPv6 site-local unicast prefix defined in [RFC3513], i.e., 1111111011 binary or FEC0::/10. The special behavior of this prefix MUST no longer be supported in new implementations. The prefix MUST NOT be reassigned for other use except by a future IETF standards action. Future versions of the addressing architecture [RFC3513] will include this information.

However, router implementations SHOULD be configured to prevent routing of this prefix by default.

The references to site local addresses should be removed as soon as practical from the revision of the Default Address Selection for Internet Protocol version 6 [RFC3484], the revision of the Basic Socket Interface Extensions for IPv6 [RFC3493], and from the revision of the Internet Protocol Version 6 (IPv6) Addressing Architecture [RFC3513]. Incidental references to site local addresses should be removed from other IETF documents if and when they are updated. These documents include [RFC2772, RFC2894, RFC3082, RFC3111, RFC3142, RFC3177, and RFC3316].

Existing implementations and deployments MAY continue to use this prefix.

5. Security Considerations

The use of ambiguous site-local addresses has the potential to adversely affect network security through leaks, ambiguity and potential misrouting, as documented in section 2. Deprecating the use of ambiguous addresses helps solving many of these problems.

The site-local unicast prefix allows for some blocking action in firewall rules and address selection rules, which are commonly viewed as a security feature since they prevent packets crossing administrative boundaries. Such blocking rules can be configured for any prefix, including the expected future replacement for the site-local prefix. If these blocking rules are actually enforced, the deprecation of the site-local prefix does not endanger security.

6. IANA Considerations

IANA is requested to mark the FEC0::/10 prefix as "deprecated", pointing to this document. Reassignment of the prefix for any usage requires justification via an IETF Standards Action [RFC2434].

7. Acknowledgements

The authors would like to thank Fred Templin, Peter Bieringer, Chirayu Patel, Pekka Savola, and Alain Baudot for their review of the initial version of the document. The text of section 2.2 includes 2 paragraphs taken from a version by Margaret Wasserman describing the impact of site local addressing. Alain Durand pointed out the need to revise existing RFC that make reference to site local addresses.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC3513] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003.

8.2. Informative References

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC2772] Rockell, R. and R. Fink, "6Bone Backbone Routing Guidelines", RFC 2772, February 2000.

[RFC2894] Crawford, M., "Router Renumbering for IPv6", RFC 2894, August 2000.

[RFC3082] Kempf, J. and J. Goldschmidt, "Notification and Subscription for SLP", RFC 3082, March 2001.

[RFC3111] Guttman, E., "Service Location Protocol Modifications for IPv6", RFC 3111, May 2001.

[RFC3142] Hagino, J. and K. Yamamoto, "An IPv6-to-IPv4 Transport Relay Translator", RFC 3142, June 2001.

[RFC3177] IAB and IESG, "IAB/IESG Recommendations on IPv6 Address", RFC 3177, September 2001.

[RFC3316] Arkko, J., Kuijpers, G., Soliman, H., Loughney, J., and J. Wiljakka, "Internet Protocol Version 6 (IPv6) for Some Second and Third Generation Cellular Hosts", RFC 3316, April 2003.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

[RFC3493] Gilligan, R., Thomson, S., Bound, J., McCann, J., and W. Stevens, "Basic Socket Interface Extensions for IPv6", RFC 3493, February 2003.

[Hinden/Haberman] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", Work in Progress, June 2004.

[SCOPING] Deering, S., Haberman, B., Jinmei, T., Nordmark, E., and B. Zill, "IPv6 Scoped Address Architecture", Work in Progress, August 2004.

9. Authors' Addresses

Christian Huitema
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399
USA

EMail: huitema@microsoft.com

Brian Carpenter
IBM Corporation
Sauemerstrasse 4
8803 Rueschlikon
Switzerland

EMail: brc@zurich.ibm.com

10. Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.