Internet Engineering Task Force (IETF)
Request for Comments: 7934
BCP: 204
Category: Best Current Practice
ISSN: 2070-1721

L. Colitti
V. Cerf
Google
S. Cheshire
D. Schinazi
Apple Inc.

July 2016

Host Address Availability Recommendations

Abstract

This document recommends that networks provide general-purpose end hosts with multiple global IPv6 addresses when they attach, and it describes the benefits of and the options for doing so.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

In most aspects, the IPv6 protocol is very similar to IPv4. This similarity can create a tendency to think of IPv6 as 128-bit IPv4, and thus lead network designers and operators to apply identical configurations and operational practices to both. This is generally a good thing because it eases the transition to IPv6 and the operation of dual-stack networks. However, in some design and operational areas, it can lead to carrying over IPv4 practices that are limiting or not appropriate in IPv6 due to differences between the protocols.

One such area is IP addressing, particularly IP addressing of hosts. This is substantially different because unlike IPv4 addresses, IPv6 addresses are not a scarce resource. In IPv6, a single link provides over four billion times more address space than the whole IPv4 Internet [RFC7421]. Thus, unlike IPv4, IPv6 networks are not forced by address scarcity concerns to provide only one address per host. Furthermore, providing multiple addresses has many benefits, including application functionality and simplicity, privacy, and flexibility to accommodate future applications. Another significant benefit is the ability to provide Internet access without the use of Network Address Translation (NAT). Providing only one IPv6 address per host negates these benefits.

This document details the benefits of providing multiple addresses per host, and the problems with not doing so. It recommends that networks provide general-purpose end hosts with multiple global addresses when they attach and lists current options for doing so. It does not specify any changes to protocols or host behavior.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

2. Common IPv6 Deployment Model

IPv6 is designed to support multiple addresses, including multiple global addresses, per interface (see Section 2.1 of [RFC4291] and Section 5.9.4 of [RFC6434]). Today, many general-purpose IPv6 hosts are configured with three or more addresses per interface: a link-local address, a stable address (e.g., using 64-bit Extended Unique Identifiers (EUI-64) or Opaque Interface Identifiers [RFC7217]), one or more privacy addresses [RFC4941], and possibly one or more temporary or non-temporary addresses obtained using the Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [RFC3315].

In most general-purpose IPv6 networks, hosts have the ability to configure additional IPv6 addresses from the link prefix(es) without explicit requests to the network. Such networks include all 3GPP networks ([RFC6459], Section 5.2), in addition to Ethernet and Wi-Fi networks using Stateless Address Autoconfiguration (SLAAC) [RFC4862].

3. Benefits of Providing Multiple Addresses

Today, there are many host functions that require more than one IP address to be available to the host, including:

- Privacy addressing to prevent tracking by off-network hosts [RFC4941].

- Multiple processors inside the same device. For example, in many mobile devices, both the application processor and the baseband processor need to communicate with the network, particularly for technologies like I-WLAN [TS.24327] where the two processors share the Wi-Fi network connection.

- Extending the network (e.g., "tethering").

- Running virtual machines on hosts.

- Translation-based transition technologies such as 464XLAT (a combination of stateful and stateless translation) [RFC6877] that translate between IPv4 and IPv6. Some of these technologies require the availability of a dedicated IPv6 address in order to determine whether inbound packets are translated or native ([RFC6877], Section 6.3).

- Identifier-locator addressing (ILA) [ILA].

- Future applications (e.g., per-application IPv6 addresses [TARP]).

Two examples of how the availability of multiple addresses per host has already allowed substantial deployment of new applications without explicit requests to the network are:

- 464XLAT. 464XLAT is usually deployed within a particular network; in this model, the operator can ensure that the network is appropriately configured to provide the customer-side translator (CLAT) with the additional IPv6 address it needs to implement 464XLAT. However, there are deployments where the provider-side translator (PLAT) (i.e., NAT64) is provided as a service by a different network, without the knowledge or cooperation of the residential ISP (e.g., the IPv6v4 Exchange Service [IPv6v4]). This type of deployment is only possible because those residential ISPs provide multiple IP addresses to their users, and thus those users can freely obtain the extra IPv6 address required to run 464XLAT.

- /64 sharing [RFC7278]. When the topology supports it, this is a way to provide IPv6 tethering without needing to wait for network operators to deploy DHCPv6 Prefix Delegation (PD), which is only available in 3GPP release 10 or above ([RFC6459], Section 5.3).

4. Problems with Restricting the Number of Addresses per Host

Providing a restricted number of addresses per host implies that functions that require multiple addresses either will be unavailable (e.g., if the network provides only one IPv6 address per host, or if the host has reached the limit of the number of addresses available) or will only be available after an explicit request to the network is granted. Requiring explicit requests to the network has the following drawbacks:

- Increased latency, because a provisioning operation, and possibly human intervention with an update to the Service Level Agreement (SLA), must complete before the functionality is available.

- Uncertainty, because it is not known if a particular application function will be available until the provisioning operation succeeds or fails.

- Complexity, because implementations need to deal with failures and somehow present them to the user. Failures may manifest as timeouts, which may be slow and frustrating to users.

- Increased load on the network's provisioning servers.

Some operators may desire that their networks be configured to limit the number of IPv6 addresses per host. Reasons might include hardware limitations (e.g., Ternary Content-Addressable Memory (TCAM) size or size constraints of the Neighbor Cache table), business models (e.g., a desire to charge the network's users on a per-device basis), or operational consistency with IPv4 (e.g., an IP address management system that only supports one address per host). However, hardware limitations are expected to ease over time, and an attempt to generate additional revenue by charging per device may prove counterproductive if customers respond (as they did with IPv4) by using NAT, which results in no additional revenue, but leads to more operational problems and higher support costs.

5. Overcoming Limits Using Network Address Translation

When the network limits the number of addresses available to a host, this can mostly be overcome by end hosts by using NAT, and indeed in IPv4 the scarcity of addresses is often mitigated by using NAT on the host. Thus, the limits could be overcome in IPv6 as well by implementing NAT66 on the host.

Unfortunately, NAT has well-known drawbacks. For example, it causes application complexity due to the need to implement NAT traversal. It hinders development of new applications. On mobile devices, it reduces battery life due to the necessity of frequent keepalives, particularly for UDP. Applications using UDP that need to work on most of the Internet are forced to send keepalives at least every 30 seconds [KA]. For example, the QUIC protocol uses a 15-second keepalive [QUIC]. Other drawbacks of NAT are well-known and documented [RFC2993]. While IPv4 NAT is inevitable due to the limited amount of IPv4 space available, that argument does not apply to IPv6. Guidance from the Internet Architecture Board (IAB) is that deployment of IPv6 NAT is not desirable [RFC5902].

The desire to overcome the problems listed in Section 4 without disabling any features has resulted in developers implementing IPv6 NAT. There are fully stateful address+port NAT66 implementations in client operating systems today: for example, Linux has supported

NAT66 since 2012 [L66]. At least one popular software hypervisor also implemented NAT66 to work around these issues [V66]. Wide deployment of networks that provide a restricted number of addresses will cause proliferation of NAT66 implementations.

This is not a desirable outcome. It is not desirable for users because they may experience application brittleness. It is likely not desirable for network operators either, as they may suffer higher support costs, and even when the decision to provide only one IPv6 address per device is dictated by the network's business model, there may be little in the way of incremental revenue, because devices can share their IPv6 address with other devices. Finally, it is not desirable for operating system manufacturers and application developers, who will have to build more complexity, lengthening development time and/or reducing the time spent on other features.

Indeed, it could be argued that the main reason for deploying IPv6, instead of continuing to scale the Internet using only IPv4 and large-scale NAT44, is because doing so can provide all the hosts on the planet with end-to-end connectivity that is constrained not by accidental technical limitations, but only by intentional security policies.

6. Options for Providing More Than One Address

Multiple IPv6 addresses can be provided in the following ways:

- Using Stateless Address Autoconfiguration (SLAAC) [RFC4862]. SLAAC allows hosts to create global IPv6 addresses on demand by simply forming new addresses from the global prefix(es) assigned to the link. Typically, SLAAC is used on shared links, but it is also possible to use SLAAC while providing a dedicated /64 prefix to each host. This is the case, for example, if the host is connected via a point-to-point link such as in 3GPP networks, on a network where each host has its own dedicated VLAN, or on a wireless network where every Media Access Control (MAC) address is placed in its own broadcast domain.

- Using stateful DHCPv6 address assignment [RFC3315]. Most DHCPv6 clients only ask for one non-temporary address, but the protocol allows requesting multiple temporary and even multiple non-temporary addresses, and the server could choose to provide multiple addresses. It is also technically possible for a client to request additional addresses using a different DHCP Unique Identifier (DUID), though the DHCPv6 specification implies that this is not expected behavior ([RFC3315], Section 9). The DHCPv6 server will decide whether to grant or reject the request based on information about the client, including its DUID, MAC address, and more. The maximum number of IPv6 addresses that can be provided in a single DHCPv6 packet, given a typical MTU of 1500 bytes or smaller, is approximately 30.

- DHCPv6 Prefix Delegation (PD) [RFC3633]. DHCPv6 PD allows the client to request and be delegated a prefix, from which it can autonomously form other addresses. If the prefix is shorter than /64, it can be divided into multiple subnets that can be further delegated to downstream clients. If the prefix is a /64, it can be extended via L2 bridging, Neighbor Discovery (ND) proxying [RFC4389], or /64 sharing [RFC7278], but it cannot be further subdivided, as a prefix longer than /64 is outside the current IPv6 specifications [RFC7421]. While the DHCPv6 Prefix Delegation specification [RFC3633] assumes that the DHCPv6 client is a router, DHCPv6 PD itself does not require that the client forward IPv6 packets not addressed to itself, and thus does not require that the client be an IPv6 router as defined in the IPv6 specification [RFC2460]. Also, in many cases (such as tethering, or hosting virtual machines), hosts are already forwarding IPv6 packets and thus operating as IPv6 routers as defined in the IPv6 specification [RFC2460].

	SLAAC	DHCPv6 IA_NA / IA_TA	DHCPv6 PD	DHCPv4
Can extend network	No+	No	Yes	Yes (NAT44)
Can number "unlimited" endpoints	Yes*	Yes*	No	No
Uses stateful, request-based assignment	No	Yes	Yes	Yes
Is immune to Layer 3 on-link resource exhaustion attacks	No	Yes	Yes	Yes

[*] Subject to network limitations, e.g., ND cache entry size limits.
[+] Except on certain networks, e.g., /64 sharing [RFC7278].

Table 1: Comparison of Multiple Address Assignment Options

7. Number of Addresses Required

If we itemize the use cases from Section 3, we can estimate the number of addresses currently used in normal operations. In typical implementations, privacy addresses use up to 7 addresses -- one per day ([RFC4941], Section 3.5). Current mobile devices sharing an uplink connection may typically support 8 downstream client devices, with each one requiring one or more addresses. A client might choose to run several virtual machines. Current implementations of 464XLAT require the use of a separate address. Some devices require another address for their baseband chip. Even a host performing just a few of these functions simultaneously might need on the order of 20 addresses at the same time. Future applications designed to use an address per application or even per resource will require many more. These will not function on networks that enforce a hard limit on the number of addresses provided to hosts. Thus, in general it is not possible to estimate in advance how many addresses are required.

8. Recommendations

In order to avoid the problems described above and preserve the Internet’s ability to support new applications that use more than one IPv6 address, it is RECOMMENDED that IPv6 network deployments provide multiple IPv6 addresses from each prefix to general-purpose hosts. To support future use cases, it is NOT RECOMMENDED to impose a hard limit on the size of the address pool assigned to a host. Particularly, it is NOT RECOMMENDED to limit a host to only one IPv6 address per prefix.

Due to the drawbacks imposed by requiring explicit requests for address space (see Section 4), it is RECOMMENDED that the network give the host the ability to use new addresses without requiring explicit requests. This can be achieved either by allowing the host to form new addresses autonomously (e.g., via SLAAC) or by providing the host with a dedicated /64 prefix. The prefix MAY be provided using DHCPv6 PD, SLAAC with per-device VLANs, or any other means.

Using stateful address assignment (DHCPv6 IA_NA or IA_TA) to provide multiple addresses when the host connects (e.g., the approximately 30 addresses that can fit into a single packet) would accommodate current clients, but it sets a limit on the number of addresses available to hosts when they attach and therefore limits the development of future applications.

9. Operational Considerations

9.1. Host Tracking

Some network operators -- often operators of networks that provide services to third parties such as university campus networks -- are required to track which IP addresses are assigned to which hosts on their network. Maintaining persistent logs that map user IP addresses and timestamps to hardware identifiers such as MAC addresses may be used to attribute liability for copyright infringement or other illegal activity.

It is worth noting that this requirement can be met without using DHCPv6 address assignment. For example, it is possible to maintain these mappings by monitoring the IPv6 neighbor table: routers typically allow periodic dumps of the Neighbor Cache via the Simple Network Management Protocol (SNMP) or other means, and many can be configured to log every change to the Neighbor Cache. Using SLAAC with a dedicated /64 prefix for each host simplifies tracking, as it does not require logging every address formed by the host, but only the prefix assigned to the host when it attaches to the network. Similarly, providing address space using DHCPv6 PD has the same tracking properties as DHCPv6 address assignment, but allows the network to provide unrestricted address space.

Many large enterprise networks are fully dual stack and implement address monitoring without using or supporting DHCPv6. The authors are directly aware of several networks that operate in this way, including the Universities of Loughborough, Minnesota, Reading, Southampton, and Wisconsin, and Imperial College London, in addition to the enterprise networks of the authors’ employers.

It should also be noted that using DHCPv6 address assignment does not ensure that the network can reliably track the IPv6 addresses used by hosts. On any shared network without Layer 2 (L2) edge port security, hosts are able to choose their own addresses regardless of what address provisioning methodology the network operator believes is in use. The only way to restrict the addresses used by hosts is to use L2 security mechanisms that enforce that particular IPv6 addresses are used by particular link-layer addresses (for example, Source Address Validation Improvement (SAVI) [RFC7039]). If those mechanisms are available, it is possible to use them to provide tracking; this form of tracking is more secure and reliable than server logs because it operates independently of how addresses are allocated. Finally, tracking address information via DHCPv6 server logs is likely to become decreasingly viable due to ongoing efforts to improve the privacy of DHCPv6 and MAC address randomization [RFC7844].

9.2. Address Space Management

In IPv4, all but the world’s largest networks can be addressed using private space [RFC1918], with each host receiving one IPv4 address. Many networks can be numbered in 192.168.0.0/16, which has roughly 65 thousand addresses. In IPv6, that is equivalent to a /48, with each host receiving a /64 prefix. Under current Regional Internet Registry (RIR) policies, a /48 is easy to obtain for an enterprise network. Networks that need a bigger block of private space use 10.0.0.0/8, which has roughly 16 million addresses. In IPv6, that is equivalent to a /40, with each host receiving a /64 prefix. Enterprises of such size can easily obtain a /40 under current RIR policies.

In the above cases, aggregation and routing can be equivalent to IPv4: if a network aggregates per-host IPv4 addresses into prefixes of length /32 - n, it can aggregate per-host /64 prefixes into the same number of prefixes of length /64 - n.

Currently, residential users typically receive one IPv4 address and a /48, /56, or /60 IPv6 prefix. While such networks do not provide enough space to assign a /64 per host, such networks almost universally use SLAAC, and thus do not pose any particular limit to the number of addresses hosts can use.

Unlike IPv4 where addresses came at a premium, in all of these networks there is enough IPv6 address space to supply clients with multiple IPv6 addresses.

9.3. Addressing Link-Layer Scalability Issues via IP Routing

The number of IPv6 addresses on a link has a direct impact on networking infrastructure nodes (routers, switches) and other nodes on the link. Setting aside exhaustion attacks via L2 address spoofing, every (L2, IP) address pair impacts networking hardware requirements in terms of memory, Multicast Listener Discovery (MLD) snooping, solicited node multicast groups, etc. Many of these costs are incurred by neighboring hosts.

Hosts on such networks that create unreasonable numbers of addresses risk impairing network connectivity for themselves and other hosts on the network, and in extreme cases (e.g., hundreds or thousands of addresses) may even find their network access restricted by denial-of-service protection mechanisms.

We expect these scaling limitations to change over time as hardware and applications evolve. However, switching to a dedicated /64 prefix per host can resolve these scaling limitations. If the prefix is provided via DHCPv6 PD, or if the prefix can be used by only one link-layer address (e.g., if the link layer uniquely identifies or authenticates hosts based on MAC addresses), then there will be only one routing entry and one ND cache entry per host on the network. Furthermore, if the host is aware that the prefix is dedicated (e.g., if it was provided via DHCPv6 PD and not SLAAC), it is possible for the host to assign IPv6 addresses from this prefix to an internal virtual interface such as a loopback interface. This obviates the need to perform Neighbor Discovery and Duplicate Address Detection on the network interface for these addresses, reducing network traffic.

Thus, assigning a dedicated /64 prefix per host is operationally prudent. Clearly, however, it requires more IPv6 address space than using shared links, so the benefits provided must be weighed with the operational overhead of address space management.

10. Security Considerations

As mentioned in Section 9.3, on shared networks using SLAAC, it is possible for hosts to attempt to exhaust network resources and possibly deny service to other hosts by creating unreasonable numbers (e.g., hundreds or thousands) of addresses. Networks that provide access to untrusted hosts can mitigate this threat by providing a dedicated /64 prefix per host. It is also possible to mitigate the threat by limiting the number of ND cache entries that can be created for a particular host, but care must be taken to ensure that the network does not prevent the legitimate use of multiple IP addresses by non-malicious hosts.

Security issues related to host tracking are discussed in Section 9.1.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

11.2. Informative References

[ILA] Herbert, T., "Identifier-locator addressing for network virtualization", Work in Progress, draft-herbert-nvo3-ila-02, March 2016.

[IPv6v4] Japan Internet Exchange, "IPv6v4 Exchange Service", April 2013, <http://www.jpix.ad.jp/en/service/ipv6v4.html>.

[KA] Roskind, J., "Quick UDP Internet Connections", November 2013, <http://www.ietf.org/proceedings/88/slides/slides-88-tsvarea-10.pdf>.

[L66] McHardy, P., "netfilter: ipv6: add IPv6 NAT support", Linux commit 58a317f1061c894d2344c0b6a18ab4a64b69b815, August 2012, <https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/?id=58a317f1061c894d2344c0b6a18ab4a64b69b815>.

[QUIC] Hamilton, R., Iyengar, J., Swett, I., and A. Wilk, "QUIC: A UDP-Based Secure and Reliable Transport for HTTP/2", Work in Progress, draft-tsvwg-quic-protocol-02, January 2016.

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <http://www.rfc-editor.org/info/rfc1918>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC2993] Hain, T., "Architectural Implications of NAT", RFC 2993, DOI 10.17487/RFC2993, November 2000, <http://www.rfc-editor.org/info/rfc2993>.

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <http://www.rfc-editor.org/info/rfc3315>.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, DOI 10.17487/RFC3633, December 2003, <http://www.rfc-editor.org/info/rfc3633>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <http://www.rfc-editor.org/info/rfc4291>.

[RFC4389] Thaler, D., Talwar, M., and C. Patel, "Neighbor Discovery Proxies (ND Proxy)", RFC 4389, DOI 10.17487/RFC4389, April 2006, <http://www.rfc-editor.org/info/rfc4389>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <http://www.rfc-editor.org/info/rfc4862>.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <http://www.rfc-editor.org/info/rfc4941>.

[RFC5902] Thaler, D., Zhang, L., and G. Lebovitz, "IAB Thoughts on IPv6 Network Address Translation", RFC 5902, DOI 10.17487/RFC5902, July 2010, <http://www.rfc-editor.org/info/rfc5902>.

[RFC6434] Jankiewicz, E., Loughney, J., and T. Narten, "IPv6 Node Requirements", RFC 6434, DOI 10.17487/RFC6434, December 2011, <http://www.rfc-editor.org/info/rfc6434>.

[RFC6459] Korhonen, J., Ed., Soininen, J., Patil, B., Savolainen, T., Bajko, G., and K. Iisakkila, "IPv6 in 3rd Generation Partnership Project (3GPP) Evolved Packet System (EPS)", RFC 6459, DOI 10.17487/RFC6459, January 2012, <http://www.rfc-editor.org/info/rfc6459>.

[RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, DOI 10.17487/RFC6877, April 2013, <http://www.rfc-editor.org/info/rfc6877>.

[RFC7039] Wu, J., Bi, J., Bagnulo, M., Baker, F., and C. Vogt, Ed., "Source Address Validation Improvement (SAVI) Framework", RFC 7039, DOI 10.17487/RFC7039, October 2013, <http://www.rfc-editor.org/info/rfc7039>.

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <http://www.rfc-editor.org/info/rfc7217>.

[RFC7278] Byrne, C., Drown, D., and A. Vizdal, "Extending an IPv6 /64 Prefix from a Third Generation Partnership Project (3GPP) Mobile Interface to a LAN Link", RFC 7278, DOI 10.17487/RFC7278, June 2014, <http://www.rfc-editor.org/info/rfc7278>.

[RFC7421] Carpenter, B., Ed., Chown, T., Gont, F., Jiang, S., Petrescu, A., and A. Yourtchenko, "Analysis of the 64-bit Boundary in IPv6 Addressing", RFC 7421, DOI 10.17487/RFC7421, January 2015, <http://www.rfc-editor.org/info/rfc7421>.

[RFC7844] Huitema, C., Mrugalski, T., and S. Krishnan, "Anonymity Profiles for DHCP Clients", RFC 7844, DOI 10.17487/RFC7844, May 2016, <http://www.rfc-editor.org/info/rfc7844>.

[TARP] Gleitz, PM. and SB. Bellovin, "Transient Addressing for Related Processes: Improved Firewalling by Using IPv6 and Multiple Addresses per Host", In Proceedings of the Eleventh Usenix Security Symposium, August 2001, <https://www.usenix.org/legacy/events/sec01/gleitz.html>.

[TS.24327] 3GPP, "Mobility between 3GPP Wireless Local Area Network (WLAN) interworking (I-WLAN) and 3GPP systems; General Packet Radio System (GPRS) and 3GPP I-WLAN aspects; Stage 3", 3GPP TS 24.327, June 2011, <http://www.3gpp.org/DynaReport/24327.htm>.

[V66] Oracle, "What’s New in VirtualBox 4.3?", October 2013, <https://blogs.oracle.com/fatbloke/entry/what_s_new_in_virtualbox>.

Acknowledgements

The authors thank Tore Anderson, Brian Carpenter, David Farmer, Wesley George, Geoff Huston, Erik Kline, Victor Kuarsingh, Shucheng (Will) Liu, Shin Miyakawa, Dieter Siegmund, Mark Smith, Sander Steffann, Fred Templin, and James Woodyatt for their input and contributions.

Authors' Addresses

Lorenzo Colitti
Google
Roppongi 6-10-1
Minato, Tokyo 106-6126
Japan

Email: lorenzo@google.com

Vint Cerf
Google
1875 Explorer Street
10th Floor
Reston, VA 20190
United States of America

Email: vint@google.com

Stuart Cheshire
Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
United States of America

Email: cheshire@apple.com

David Schinazi
Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
United States of America

Email: dschinazi@apple.com