INTAREA
Internet-Draft
Updates: RFC 4884 (if approved)
Intended status: Standards Track
Expires: September 3, 2017

R. Bonica
R. Thomas
Juniper Networks
J. Linkova
Google
C. Lenart
Verizon

March 2, 2017

Extended Ping (Xping)
draft-bonica-intarea-eping-04

Abstract

This document describes a new diagnostic tool called Extended Ping (Xping). Network operators execute Xping to determine the status of a remote interface. In this respect, Xping is similar to Ping. Xping differs from Ping in that it does not require network reachability between itself and remote interface whose status is being queried.

Xping relies on two new ICMP messages, called Extended Echo Request and Extended Echo Reply. Both ICMP messages are defined herein.

Requirements Language

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 RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 3, 2017.

Table of Contents

1. Problem Statement

Network operators use Ping [RFC2151] to determine whether a remote interface is operational. Ping sends an ICMP [RFC0792] [RFC4443] Echo message to the interface being probed and waits for an ICMP Echo Reply. If Ping receives the expected ICMP Echo Reply, it reports that the probed interface is operational.

In order for the ICMP Echo message to reach the probed interface, the probed interface must be addressed appropriately. IP addresses are scoped as follows:

- Global [RFC4291]
- Private [RFC1918]
- Link-local [RFC3927] [RFC4291]

Global addresses are the most widely scoped. A globally addressed interface can be reached from any node on the Internet. By contrast, link-local addresses are the least widely scoped. An interface whose only address is link-local can be reached from on-link interfaces only.

Network operators seek to decrease their dependence on widely-scoped interface addressing. For example:

- The operator of an IPv4 network currently assigns global addresses to all interfaces. In order to conserve scarce IPv4 address space, this operator seeks to renumber selected interfaces with private addresses.
- The operator of an IPv4 network currently assigns private addresses to all interfaces. In order to achieve operational efficiencies, this operator seeks to leave selected interfaces unnumbered.
- The operator of an IPv6 network currently assigns global addresses to all interfaces. In order to achieve operational efficiencies, this operator seeks to number selected interfaces with link-local addresses only [RFC7404]

When a network operator renumbers an interface, replacing a more widely scoped address with one that is less widely scoped, the operator also reduces the number of nodes from which Ping can probe the interface. Therefore, many network operators who rely on Ping remain dependant upon widely scoped interface addressing.

This document describes a new diagnostic tool called Extended Ping (Xping). Network operators use Xping to determine the status of a remote interface. In this respect, Xping is similar to Ping. Xping differs from Ping in that it does not require reachability between the probing node and the probed interface. Or, said another way, Xping does not require reachability between the node upon which it executes and the interface whose status is being queried.

Xping relies on two new informational ICMP messages, called Extended Echo Request and Extended Echo Reply. The Extended Echo Request message makes a semantic distinction between the destination interface and the probed interface. The destination interface is the

interface to which the Extended Echo Request message is delivered. It must be reachable from the probing node. The probed interface is the interface whose status is being queried. It does not need to be reachable from the probing node. However, the destination and probed interfaces must be local to one another (i.e., both interfaces must belong to the same node).

Because the Extended Echo Request message makes a distinction between the destination and probed interfaces, Xping can probe every interface on a node if it can reach any interface on the node. In many cases, this allows network operators to decrease their dependence on widely scoped interface addressing.

Network operators can use Xping to determine the operational status of the probed interface. They can also use Xping to determine which protocols (e.g., IPv4, IPv6) are active on the interface. However, they cannot use Xping to obtain other information regarding the interface (e.g., bandwidth, MTU). In order to obtain such information, they should use other network management protocols (e.g., SNMP, Netconf).

This document is divided into sections, with Section 2 describing the Extended Echo Request message and Section 3 describing the Extended Echo Reply message. Section 4 describes how the probed node processes the Extended Echo Request message and Section 5 describes the Xping application. Section 6 describes uses cases.

2. ICMP Extended Echo Request

The ICMP Extended Echo Request message is defined for both ICMPv4 and ICMPv6. Like any ICMP message, the ICMP Extended Echo Request message is encapsulated in an IP header. The ICMPv4 version of the Extended Echo Request message is encapsulated in an IPv4 header, while the ICMPv6 version is encapsulated in an IPv6 header.

Figure 1 depicts the ICMP Extended Echo Request message.

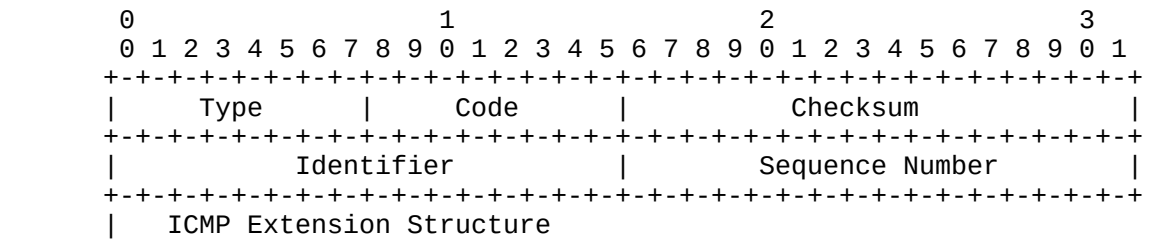

Figure 1: ICMP Extended Echo Request Message

IP Header fields:

- Source Address: The Source Address MUST be valid IPv4 or IPv6 unicast address belonging to the sending node.

- Destination Address: Identifies the destination interface (i.e., the interface to which this message will be delivered).

ICMP fields:

- Type: Extended Echo Request. The value for ICMPv4 is TBD by IANA. The value for ICMPv6 is also TBD by IANA.

- Code: 0

- Checksum: For ICMPv4, see RFC 792. For ICMPv6, see RFC 4443.

- Identifier: An identifier to aid in matching Extended Echo Replies to Extended Echo Requests. May be zero.

- Sequence Number: A sequence number to aid in matching Extended Echo Replies to Extended Echo Requests. May be zero.

- ICMP Extension Structure: Identifies the probed interface, by name, index or address.

If the ICMP Extension Structure identifies the probed interface by address, that address can be a member of any address family. For example:

- An ICMPv4 Extended Echo Request message can carry an ICMP Extension Structure that identifies the probed interface by IPv4 address

- An ICMPv4 Extended Echo Request message can carry an ICMP Extension Structure that identifies the probed interface by IPv6 address

- An ICMPv6 Extended Echo Request message can carry an ICMP Extension Structure that identifies the probed interface by IPv4 address

- An ICMPv6 Extended Echo Request message can carry an ICMP Extension Structure that identifies the probed interface by IPv6 address

Section 7 of [RFC4884] defines the ICMP Extension Structure. As per RFC 4884, the Extension Structure contains exactly one Extension Header followed by one or more objects. When applied to the ICMP Extended Echo Request message, the ICMP Extension Structure contains one or two instances of the Interface Identification Object (Section 2.1).

In most cases, a single instance of the Interface Identification Object can identify the probed interface. However, two instance are required when neither uniquely identifies a interface (e.g., an IPv6 link-local address and an IEEE 802 address).

2.1. Interface Identification Object

The Interface Identification Object identifies the probed interface by name, index, or address. Like any other ICMP Extension Object, it contains an Object Header and Object Payload. The Object Header contains the following fields:

- Class-Num: Interface Identification Object. Value is TBD by IANA

- C-type: Values are: (1) Identifies Interface By Name, (2) Identifies Interface By Index, and (3) Identifies Interface By Address

- Length: Length of the object, measured in octets, including the object header and object payload.

If the Interface Identification Object identifies the probed interface by name, the object payload contains the human-readable interface name. The interface name SHOULD be the full MIB-II ifName [RFC2863], if less than 255 octets, or the first 255 octets of the ifName, if the ifName is longer. The interface name MAY be some other human-meaningful name of the interface. The interface name MUST be represented in the UTF-8 charset [RFC3629] using the Default Language [RFC2277].

If the Interface Identification Object identifies the probed interface by index, the length is equal to 8 and the payload contains the MIB-II ifIndex [RFC 2863].

If the Interface Identification Object identifies the probed interface by address, the payload is as depicted in Figure 2.

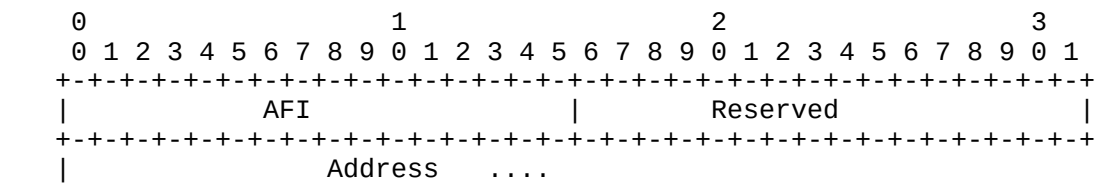

Figure 2: Interface Identification Object - C-type 3 Payload

Payload fields are defined as follows:

- Address Family Identifier (AFI): This 16-bit field identifies the type of address represented by the Address field. All values found in the IANA registry of Address Family Numbers (available from <http://www.iana.org>) are valid in this field. Implementations MUST support values (1) IPv4, (2) IPv6, (6) IEEE 802, (16389) 48-bit MAC and (16390) 64-bit MAC. They MAY support other values.

- Reserved: This 16-bit field MUST be set to zero and ignored upon receipt.

- Address: This variable-length field represents an address associated with the probed interface.

3. ICMP Extended Echo Reply

The ICMP Extended Echo Reply message is defined for both ICMPv4 and ICMPv6. Like any ICMP message, the ICMP Extended Echo Reply message is encapsulated in an IP header. The ICMPv4 version of the Extended Echo Reply message is encapsulated in an IPv4 header, while the ICMPv6 version is encapsulated in an IPv6 header.

Figure 3 depicts the ICMP Extended Echo Reply message.

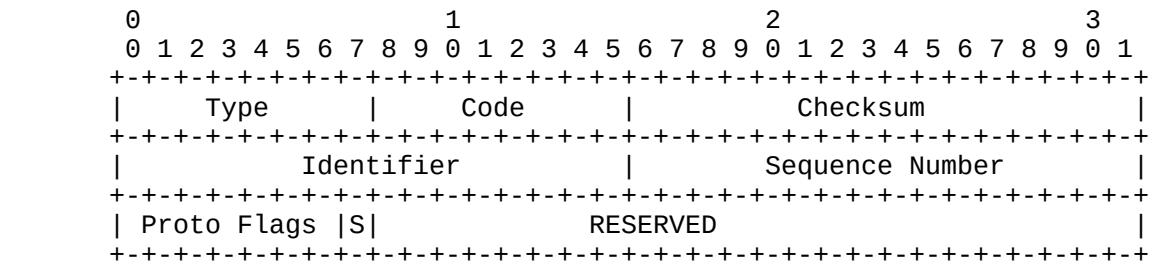

Figure 3: ICMP Extened Echo Reply Message

IP Header fields:

- Source address: Copied from the Destination Address field of the invoking Extended Echo Request message.

- Destination address: Copied from the Source Address field of the invoking Extended Echo Request message.

ICMP fields:

- Type: Extended Echo Reply. The value for ICMPv4 is TBD by IANA. The value for ICMPv6 is also TBD by IANA.

- Code: (0) No Error, (1) Malformed Query, (2) No Such Interface, (3) Multiple Interfaces Satisfy Query

- Checksum: For ICMPv4, see RFC 792. For ICMPv6, see RFC 4443.

- Identifier: Copied from the Identifier field of the invoking Extended Echo Request packet.

- Sequence Number: Copied from the Sequence Number field of the invoking Extended Echo Request packet.

- Proto Flags: Each bit in this field represents a protocol. The bit is set if the S-bit is set and the corresponding protocol is running on the probed interface. Bit mappings are as follows: Bit 0 (IPv4), Bit 1 (IPv6), Bit 2 (Ethernet), Bits 3-7 (Reserved)

- S Bit: This bit is set if the Code field is equal to No Error (0) and the probed interface is active. Otherwise, this bit is clear.

- Reserved: This field MUST be set to zero and ignored upon receipt.

4. ICMP Extended Echo and Extended Echo Reply Processing

When a node receives an ICMP Extended Echo Request message and any of the following conditions apply, the node MUST silently discard the incoming message:

- The node does not recognize ICMP Extended Echo Request messages
- The node has not explicitly enabled ICMP Extended Echo functionality
- The node has not explicitly enabled the incoming ICMP Extended Echo Request type (i.e., by ifName, by IfIndex, by Address)
- The incoming ICMP Extend Echo Request carries a source address that is not authorized for the incoming ICMP Extended Echo Request type
- The Source Address of the incoming messages is not a unicast address

Otherwise, when a node receives an ICMPv4 Extended Echo Request, it MUST format an ICMP Extended Echo Reply as follows:

- Don’t Fragment flag (DF) is 1
- More Fragments flag is 0
- Fragment Offset is 0
- TTL is 255
- Protocol is ICMP

When a node receives an ICMPv6 Extended Echo Request, it MUST format an ICMPv6 Extended Echo Reply as follows:

- Hop Limit is 255
- Next Header is ICMPv6

In either case, the responding node MUST:

- Copy the source address from the Extended Echo Request message to the destination address of the Extended Echo Reply

- Copy the destination address from the Extended Echo Request message to the source address of the Extended Echo Reply
- Set the DiffServ codepoint to CS0 [RFC4594]
- Set the ICMP Type to Extended Echo Reply
- Copy the Identifier from the Extended Echo Request message to the Extended Echo Reply
- Copy the sequence number from the Extended Echo Request message to the Extended Echo Reply
- Set the Code field as described Section 4.1
- If the Code Field is equal to No Error (0) and the probed interface is active, set the S-Bit. Otherwise, clear the S-Bit.
- If the S-bit is set, set Protocol Flags as appropriate. Otherwise, clear all Protocol Flags.
- Set the checksum appropriately
- Forward the ICMP Extended Echo Reply to its destination

The status of the probed interface is determined exactly as if it had been probed by a directly connected neighbor using traditional ping.

4.1. Code Field Processing

The following rules govern how the Code should be set:

- If the query is malformed, set the Code to Malformed Query (1)
- Otherwise, if the ICMP Extension Structure does not identify any local interfaces, set the Code to No Such Interface (2)
- Otherwise, if the ICMP Extension Structure identifies more than one local interfaces, set the Code to Multiple Interfaces Satisfy Query (3)
- Otherwise, set the code to No Error (0)

5. The Xping Application

The Xping application accepts input parameters, sets a counter and enters a loop to be exited when the counter is equal to zero. On each iteration of the loop, Xping emits an ICMP Extended Echo

Request, decrements the counter, sets a timer, waits for the timer to expire. If an expected ICMP Extended Echo Reply arrives while Xping is waiting for the timer to expire, Xping relays information returned by that message to its user. However, on each iteration of the loop, Xping waits for the timer to expire, regardless of whether an Extended Echo Reply message arrives.

Xping accepts the following parameters:

- Count
- Wait
- Source Interface Address
- Hop Count
- Destination Interface Address
- Probed Interface Identifier

Count is a positive integer whose default value is 3. Count determines the number of times that Xping iterates through the above-mentioned loop.

Wait is a positive integer whose minimum and default values are 1. Wait determines the duration of the above-mentioned timer, measured in seconds.

Source Interface Address specifies the source address of ICMP Extended Echo Request. The Source Interface Address MUST be a unicast address and MUST identify an interface that is local to the probing node.

The destination Interface Address identifies the interface to which the ICMP Extended Echo Request message is sent. It can be an IPv4 or IPv6 address. If it is an IPv4 address, Xping emits an ICMPv4 message. If it is an IPv6 address, Xping emits an ICMPv6 message.

The probed interface is the interface whose status is being queried. If the probed interface identifier is not specified, the Xping application invokes the traditional Ping application and terminates. If the probed interface identifier is specified, it can be any of the following:

- an interface name

- an address from any address family (e.g., IPv4, IPv6, IEEE 802, 48-bit MAC, 64-bit MAC)
- an ifIndex

The probed interface identifier can have any scope. For example, the probed interface identifier can be:

- an IPv6 address, whose scope is global
- an IPv6 address, whose scope is link-local
- an interface name, whose scope is node-local
- an ifIndex, whose scope is node-local

If the probed interface identifier is an address, it does not need to be of the same address family as the destination interface address. For example, Xping accepts an IPv4 destination interface address and an IPv6 probed interface identifier.

6. Use-Cases

In the use cases below, Xping can be used to determine the operational status of a forwarding interface. Other management protocols (e.g., SNMP) might also be used to obtain this information. However, we assume that those management protocols are not viable options, either because they are too heavyweight or they are not supported on the relevant nodes.

6.1. Unnumbered Interfaces

An IPv4 network contains many routers. On each router, a loopback interface is numbered from global address space and all forwarding interfaces are unnumbered. Network operations staff need a tool that they can execute on any router in the network to determine the operational status of any forwarding interface in the network.

6.2. Link-local Interfaces

An IPv6 network contains many routers. On each router, a loopback interface is numbered from global address space and some or all forwarding interfaces are numbered from link-local address space. Network operations staff need a tool that they can execute on any router in the network to determine the operational status of any forwarding interface in the network.

6.3. Unadvertised Interfaces

A network contains many routers. On each router, the loopback interface and all forwarding interfaces are numbered from global address space. However, some forwarding interfaces do not participate in any routing protocol nor are they advertised by any routing protocol. Network operations staff need a tool that they can execute on any router in the network to determine the operational status of any forwarding interface in the network.

7. Updates to RFC 4884

Section 4.6 of RFC 4884 provides a list of extensible ICMP messages (i.e., messages that can carry the ICMP Extension Structure). This document adds the ICMP Extended Echo message and the ICMP Extended Echo Reply message to that list.

8. IANA Considerations

This document requests the following actions from IANA:

- Add an entry to the "ICMP Type Number" registry, representing the Extended Echo Request. This entry has one code (0).

- Add an entry to the "Internet Control Message Protocol version 6 (ICMPv6) Parameters" registry, representing the Extended Echo Request. This entry has one code (0).

- Add an entry to the "ICMP Type Number" registry, representing the Extended Echo Reply. This entry has the following codes: (0) No Error, (1) Malformed Query, (2) No Such Interface, (3) Multiple Interfaces Satisfy Query. Protocol Flag Bit mappings are as follows: Bit 0 (IPv4), Bit 1 (IPv6), Bit 2 (Ethernet), Bits 3-15 (Reserved).

- Add an entry to the "Internet Control Message Protocol version 6 (ICMPv6) Parameters" registry, representing the Extended Echo Reply. This entry has the following codes: (0) No Error, (1) Malformed Query, (2) No Such Interface, (3) Multiple Interfaces Satisfy Query. Protocol Flag Bit mappings are as follows: Bit 0 (IPv4), Bit 1 (IPv6), Bit 2 (Ethernet), Bits 3-15 (Reserved).

- Add an entry to the "ICMP Extension Object Classes and Class Sub-types" registry, representing the Interface Identification Object. It has C-types Reserved (0), Identifies Interface By Name (1), Identifies Interface By Index (2), Identifies Interface By Address (3)

Note to RFC Editor: this section may be removed on publication as an RFC.

9. Security Considerations

The following are legitimate uses of Xping:

- to determine the operational status of an interface
- to determine which protocols (e.g., IPv4, IPv6) are active on an interface

However, malicious parties can use Xping to obtain additional information. For example, a malicious party can use Xping to discover interface names. Having discovered an interface name, the malicious party may be able to infer additional information. Additional information may include:

- interface bandwidth
- the type of device that supports the interface (e.g., vendor identity)
- the operating system version that the above-mentioned device executes

Understanding this risk, network operators establish policies that restrict access to ICMP Extended Echo functionality. In order to enforce these polices, nodes that support ICMP Extended Echo functionality MUST support the following configuration options:

- Enable/disable ICMP Extended Echo functionality. By default, ICMP Extend Echo functionality is disabled.
- Define enabled query types (i.e., by ifName, by ifIndex, by Address). By default, all query types are disabled.
- For each enabled query type, define the prefixes from which ICMP Extended Echo Request messages are permitted
- For each interface, determine whether ICMP Echo Request messages are accepted

When a node receives an ICMP Extended Echo Request message that it is not configured to support, it MUST silently discard the message. See Section 4 for details.

In order to protect local resources, implementations SHOULD rate-limit incoming ICMP Extended Echo Request messages.

10. Acknowledgements

Thanks to Jeff Haas, Carlos Pignataro, Jonathan Looney and Joe Touch for their thoughtful review of this document.

11. References

11.1. Normative References

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <http://www.rfc-editor.org/info/rfc792>.

[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>.

[RFC2277] Alvestrand, H., "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, DOI 10.17487/RFC2277, January 1998, <http://www.rfc-editor.org/info/rfc2277>.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <http://www.rfc-editor.org/info/rfc2863>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <http://www.rfc-editor.org/info/rfc3629>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, DOI 10.17487/RFC4443, March 2006, <http://www.rfc-editor.org/info/rfc4443>.

[RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, DOI 10.17487/RFC4884, April 2007, <http://www.rfc-editor.org/info/rfc4884>.

11.2. Informative References

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

[RFC2151] Kessler, G. and S. Shepard, "A Primer On Internet and TCP/IP Tools and Utilities", FYI 30, RFC 2151, DOI 10.17487/RFC2151, June 1997, <http://www.rfc-editor.org/info/rfc2151>.

[RFC3927] Cheshire, S., Aboba, B., and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses", RFC 3927, DOI 10.17487/RFC3927, May 2005, <http://www.rfc-editor.org/info/rfc3927>.

[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>.

[RFC4594] Babiarz, J., Chan, K., and F. Baker, "Configuration Guidelines for DiffServ Service Classes", RFC 4594, DOI 10.17487/RFC4594, August 2006, <http://www.rfc-editor.org/info/rfc4594>.

[RFC7404] Behringer, M. and E. Vyncke, "Using Only Link-Local Addressing inside an IPv6 Network", RFC 7404, DOI 10.17487/RFC7404, November 2014, <http://www.rfc-editor.org/info/rfc7404>.

Authors' Addresses

Ron Bonica
Juniper Networks
2251 Corporate Park Drive
Herndon, Virginia 20171
USA

Email: rbonica@juniper.net

Reji Thomas
Juniper Networks
Elnath-Exora Business Park Survey
Bangalore, Karnataka 560103
India

Email: rejithomas@juniper.net

Jen Linkova
Google
1600 Amphitheatre Parkway
Mountain View, California 94043
USA

Email: furry@google.com

Chris Lenart
Verizon
22001 Loudoun County Parkway
Ashburn, Virginia 20147
USA

Email: chris.lenart@verizon.com