Independent Submission
Request for Comments: 8507
Category: Historic
ISSN: 2070-1721

S. Deering
Retired
R. Hinden, Ed.
Check Point Software

December 2018

Simple Internet Protocol (SIP) Specification

Abstract

This document is published for the historical record. The Simple Internet Protocol was the basis for one of the candidates for the IETF's Next Generation (IPng) work that became IPv6.

The publication date of the original Internet-Draft was November 10, 1992. It is presented here substantially unchanged and is neither a complete document nor intended to be implementable.

The paragraph that follows is the Abstract from the original draft.

This document specifies a new version of IP called SIP, the Simple Internet Protocol. It also describes the changes needed to ICMP, IGMP, and transport protocols such as TCP and UDP, in order to work with SIP. A companion document [SIP-ADDR] describes the addressing and routing aspects of SIP, including issues of auto-configuration, host and subnet mobility, and multicast.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for the historical record.

This document defines a Historic Document for the Internet community. This is a contribution to the RFC Series, independently of any other RFC stream. The RFC Editor has chosen to publish this document at its discretion and makes no statement about its value for implementation or deployment. Documents approved for publication by the RFC Editor are not candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8507.

Table of Contents

1. Preface

This document is published for the historical record.

Simple IP (SIP) was the basis for one of the candidates for the IETF's Next Generation (IPng) work; see "The Recommendation for the IP Next Generation Protocol" [RFC1752]. The original 1992 Internet-Draft describing SIP is published here as part of the record of that work.

SIP evolved into SIP Plus [RFC1710], which was assessed as a candidate for IPng [RFC1752] and led eventually to the development of IPv6, first published as [RFC1883]. The current specification for IPv6 is [RFC8200].

The original Internet-Draft describing the Simple Internet Protocol was written by Steve Deering, and the Internet-Draft was posted on November 10, 1992. The contents of this document are unchanged from that Internet-Draft, except for clarifications in the Abstract, the addition of this section, modifications to the authors' information, the updating of references, removal of the IANA considerations, and minor formatting changes.

It should be noted that the original draft was not complete and that no attempt has been made to complete it. This document is not intended to be implementable.

2. Introduction

SIP is a new version of IP. Its salient differences from IP version 4 [RFC791], subsequently referred to as "IPv4", are:

- expansion of addresses to 64 bits,
- simplification of the IP header by eliminating some inessential fields, and
- relaxation of length restrictions on optional data, such as source-routing information.

SIP retains the IP model of globally-unique addresses, hierarchically-structured for efficient routing. Increasing the address size from 32 to 64 bits allows more levels of hierarchy to be encoded in the addresses, enough to enable efficient routing in an internet with tens of thousands of addressable devices in every office, every residence, and every vehicle in the world. Keeping the

addresses fixed-length and relatively compact facilitates high-performance router and host implementation, and keeps the bandwidth overhead of the SIP headers almost as low as IPv4.

The elimination of inessential fields also contributes to high-performance implementation, and to the likelihood of correct implementation. A change in the way that optional data, such as source-routing information, is encoded allows for more efficient forwarding and less stringent limits on the length of such data.

Despite these changes, SIP remains very similar to IPv4. This similarity makes it easy to understand SIP (for those who already understand IPv4), makes it possible to reuse much of the code and data structures from IPv4 in an implementation of SIP (including almost all of ICMP and IGMP), and makes it straightforward to translate between SIP packets and IPv4 packets for transition purposes [IPAE].

The subsequent sections of this document specify SIP and its associated protocols without much explanation of why the design choices were made the way they were. Appendix A presents the rationale for those aspects of SIP that differ from IPv4.

3. Terminology

system - a device that implements SIP.

router - a system that forwards SIP packets.

host - any system that is not a router.

link - a communication facility or medium over which systems can communicate at the link layer, i.e., the layer immediately below SIP.

interface - a system's attachment point to a link.

address - a SIP-layer identifier for an interface or a group of interfaces.

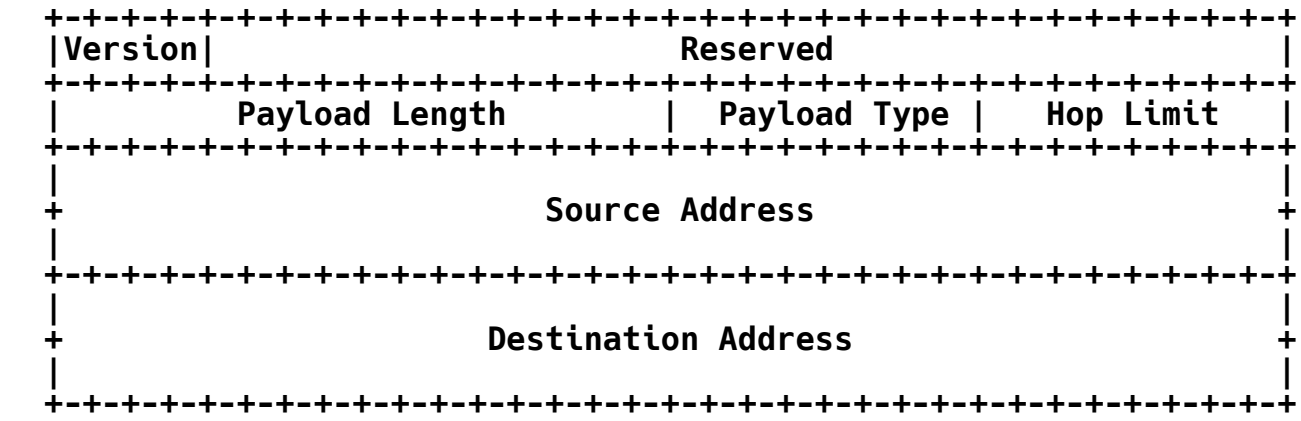
subnet - in the SIP unicast addressing hierarchy, a lowest-level (finest-grain) cluster of addresses, sharing a common address prefix (i.e., high-order address bits). Typically, all interfaces attached to the same link have addresses in the same subnet; however, in some cases, a single link may support more than one subnet, or a single subnet may span more than one link.

link MTU - the maximum transmission unit, i.e., maximum packet size in octets, that can be conveyed in one piece over a link (where a packet is a SIP header plus payload).

path MTU - the minimum link MTU of all the links in a path between a source system and a destination system.

packetization layer - any protocol layer above SIP that is responsible for packetizing data to fit within outgoing SIP packets. Typically, transport-layer protocols, such as TCP, are packetization protocols, but there may also be higher-layer packetization protocols, such as protocols implemented on top of UDP.

4. SIP Header Format



Version — 4-bit IP version number = decimal 6. <to be confirmed>

Reserved — 28-bit reserved field. Initialized to zero for transmission; ignored on reception.

Payload Length — 16-bit unsigned integer. Length of payload, i.e., the rest of the packet following the SIP header, in octets.

Payload Type — 8-bit selector. Identifies the type of payload, e.g., TCP.

Hop Limit — 8-bit unsigned integer. Decremented by 1 by each system that forwards the packet. The packet is discarded if Hop Limit is decremented to zero.

Source Address 64 bits. See "Addresses" section, following.

Destination Address 64 bits. See "Addresses" section, following.

5. Addresses

5.1. Text Representation of Addresses

SIP addresses are 64 bits (8 octets) long. The text representation of a SIP addresses is 16 hexadecimal digits, with a colon between the 4th and 5th digits, and optional colons between any subsequent pair of digits. Leading zeros must not be dropped. Examples:

0123:4567:89AB:CDEF

0123:456789ABCDEF

0123:456789AB:CDE:F

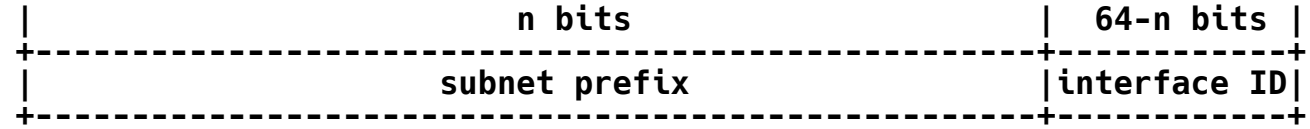
Programs that read the text representation of SIP addresses must be insensitive to the presence or absence of optional colons. Programs that write the text representation of a SIP address should use the first format above (i.e., colons after the 4th, 8th, and 12th digits), in the absence of any knowledge of the structure or preferred format of the address, such as knowledge of the format in which it was originally read.

The presence of at least one colon in the text representation allows SIP addresses to be easily distinguished from both domain names and the text representation of IPv4 addresses.

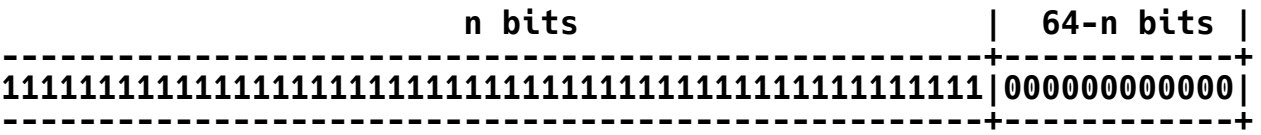
5.2. Unicast Addresses

A SIP unicast address is a globally-unique identifier for a single interface, i.e., no two interfaces in a SIP internet may have the same unicast address. A single interface may, however, have more than one unicast address.

A system considers its own unicast address(es) to have the following structure, where different addresses may have different values for n:



To know the length of the subnet prefix, the system is required to associate with each of its addresses a 'subnet mask' of the following form:

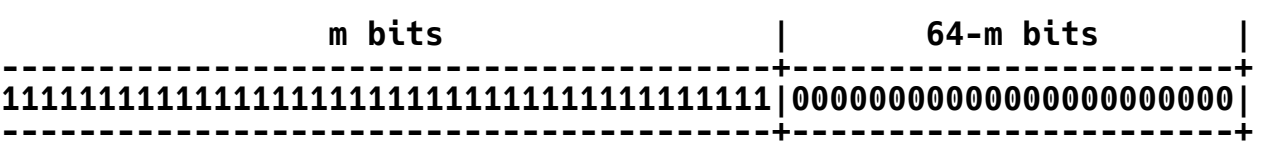


A system may have a subnet mask of all-ones, which means that the system belongs to a subnet containing exactly one system -- itself.

A system acquires its subnet mask(s) at the same time, and by the same mechanism, as it acquires its address(es), for example, by manual configuration or by a dynamic configuration protocol such as BOOTP [RFC951].

Hosts are ignorant of any further structure in a unicast address.

Routers may acquire, through manual configuration or the operation of routing protocols, additional masks that identify higher-level clusters in a hierarchical addressing plan. For example, the routers within a single site would typically have a 'site mask', such as the following:



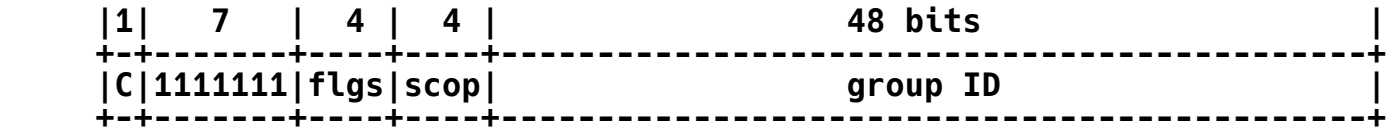
by which they could deduce the following structure in the site's addresses:

All knowledge by SIP systems of the structure of unicast addresses is based on possession of such masks -- there is no "wired-in" knowledge of unicast address formats.

The SIP Addressing and Routing document [SIP-ADDR] proposes two hierarchical addressing plans, one based on a hierarchy of SIP service providers, and one based on a geographic hierarchy.

5.3. Multicast Addresses

A SIP multicast address is an identifier for a group of interfaces. An interface may belong to any number of multicast groups. Multicast addresses have the following format:

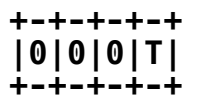


where:

C = IPv4 compatibility flag; see [IPAE].

1111111 in the rest of the first octet identifies the address as being a multicast address.

flgs is a set of 4 flags:



the high-order 3 flags are reserved, and must be initialized to 0.

T = 0 indicates a permanently-assigned ("well-known") multicast address, assigned by the global internet numbering authority.

T = 1 indicates a non-permanently-assigned ("transient") multicast address.

scop is a 4-bit multicast scope value:

0 reserved
1 intra-system scope
2 intra-link scope
3 (unassigned)
4 (unassigned)
5 intra-site scope
6 (unassigned)
7 (unassigned)
8 intra-metro scope
9 (unassigned)
A (unassigned)
B intra-country scope
C (unassigned)

D (unassigned)
E global scope
F reserved

group ID identifies the multicast group, either permanent or transient, within the given scope.

The "meaning" of a permanently-assigned multicast address is independent of the scope value. For example, if the "NTP servers group" is assigned a permanent multicast address with a group ID of 43 (hex), then:

7F01:000000000043 means all NTP servers on the same system as the sender.

7F02:000000000043 means all NTP servers on the same link as the sender.

7F05:000000000043 means all NTP servers at the same site as the sender.

7F0E:000000000043 means all NTP servers in the internet.

Non-permanently-assigned multicast addresses are meaningful only within a given scope. For example, a group identified by the non-permanent, intra-site multicast address 7F15:000000000043 at one site bears no relationship to a group using the same address at a different site, nor to a non-permanent group using the same group ID with different scope, nor to a permanent group with the same group ID.

5.4. Special Addresses

There are a number of "special purpose" SIP addresses:

The Unspecified Address: 0000:0000:0000:0000

This address shall never be assigned to any system. It may be used wherever an address appears, to indicate the absence of an address. One example of its use is in the Source Address field of a SIP packet sent by an initializing host, before it has learned its own address.

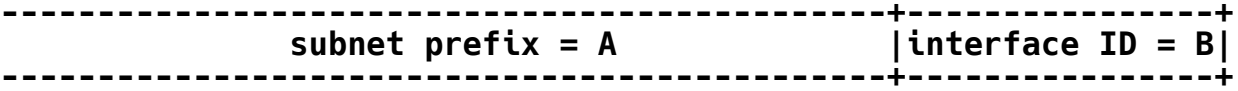
The Loopback Address: 0000:0000:0000:0001

This address may be used by a system to send a SIP packet to itself.

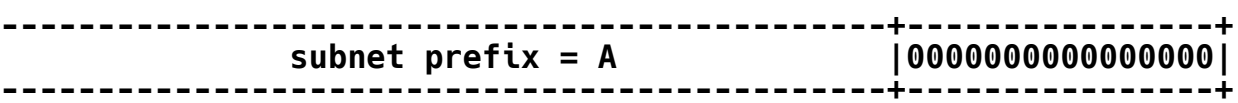
Anyone Addresses: <prefix><zero>

Addresses of this form may be used to send to the "nearest" system (according the routing protocols' measure of distance) that "knows" it has a unicast address prefix of <prefix>.

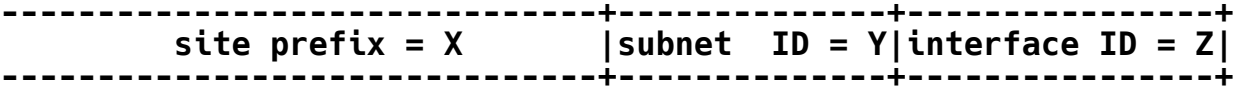
Since hosts know only their subnet prefix(es), and no higher-level prefixes, a host with the following address:



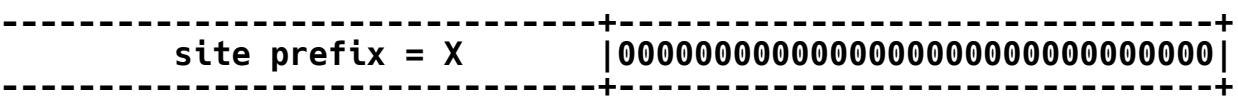
shall recognize only the following Anyone address as identifying itself:

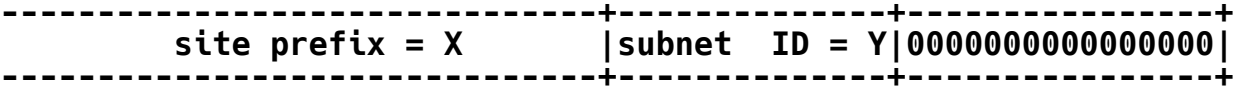


An intra-site router that knows that one of its addresses has the format:



shall accept packets sent to either of the following two Anyone addresses as if they had been sent to the router's own address:





Anyone Addresses work as follows:

If any system belonging to subnet A sends a packet to subnet A's Anyone address, the packet shall be looped-back within the sending system itself, since it is the nearest system to itself with the subnet A prefix. If a system outside of subnet A sends a packet to subnet A's Anyone address, the packet shall be accepted by the first router on subnet A that the packet reaches.

Similarly, a packet sent to site X's Anyone address from outside of site X shall be accepted by the first encountered router belonging to site X, i.e., one of site X's boundary routers. If a higher-level prefix P identifies, say, a particular service provider, then a packet sent to <P> <zero> from outside of provider P's facilities shall be delivered to the nearest entry router into P's facilities.

Anyone addresses are most commonly used in conjunction with the SIP source routing header, to cause a packet to be routed via one or more specified "transit domains", without the need to identify individual routers in those domains.

The value zero is reserved at each level of every unicast address hierarchy, to serve as an Anyone address for that level.

The Reserved Multicast Address: 7F0s:0000:0000:0000

This multicast address (with any scope value, s) is reserved, and shall never be assigned to any multicast group.

The All Systems Addresses: 7F01:0000:0000:0001
7F02:0000:0000:0001

These multicast addresses identify the group of all SIP systems, within scope 1 (intra-system) or 2 (intra-link).

The All Hosts Addresses: 7F01:0000:0000:0002
7F02:0000:0000:0002

These multicast addresses identify the group of all SIP hosts, within scope 1 (intra-system) or 2 (intra-link).

The All Routers Addresses: 7F01:0000:0000:0003
7F02:0000:0000:0003

These multicast addresses identify the group of all SIP routers, within scope 1 (intra-system) or 2 (intra-link).

A host is required to recognize the following addresses as identifying itself: its own unicast addresses, the Anyone addresses with the same subnet prefixes as its unicast addresses, the Loopback address, the All Systems and All Hosts addresses, and any other multicast addresses to which the host belongs.

A router is required to recognize the following addresses as identifying itself: its own unicast addresses, the Anyone addresses with the same subnet or higher-level prefixes as its unicast addresses, the Loopback address, the All Systems and All Routers addresses, and any other multicast addresses to which the host belongs.

6. Packet Size Issues

SIP requires that every link in the internet have an MTU of 576 octets or greater. On any link that cannot convey a 576-octet packet in one piece, link-specific fragmentation and reassembly must be provided at a layer below SIP.

> (Note: this minimum link MTU is NOT the same as the one in IPv4. In IPv4, the minimum link MTU is 68 octets [[RFC791], page 25]; 576 octets is the minimum reassembly buffer size required in an IPv4 system, which has nothing to do with link MTUs.)

From each link to which a system is directly attached, the system must be able to accept packets as large as that link's MTU. Links that have a configurable MTU, such as PPP links [RFC1661], should be configured with an MTU of 600 octets or greater.

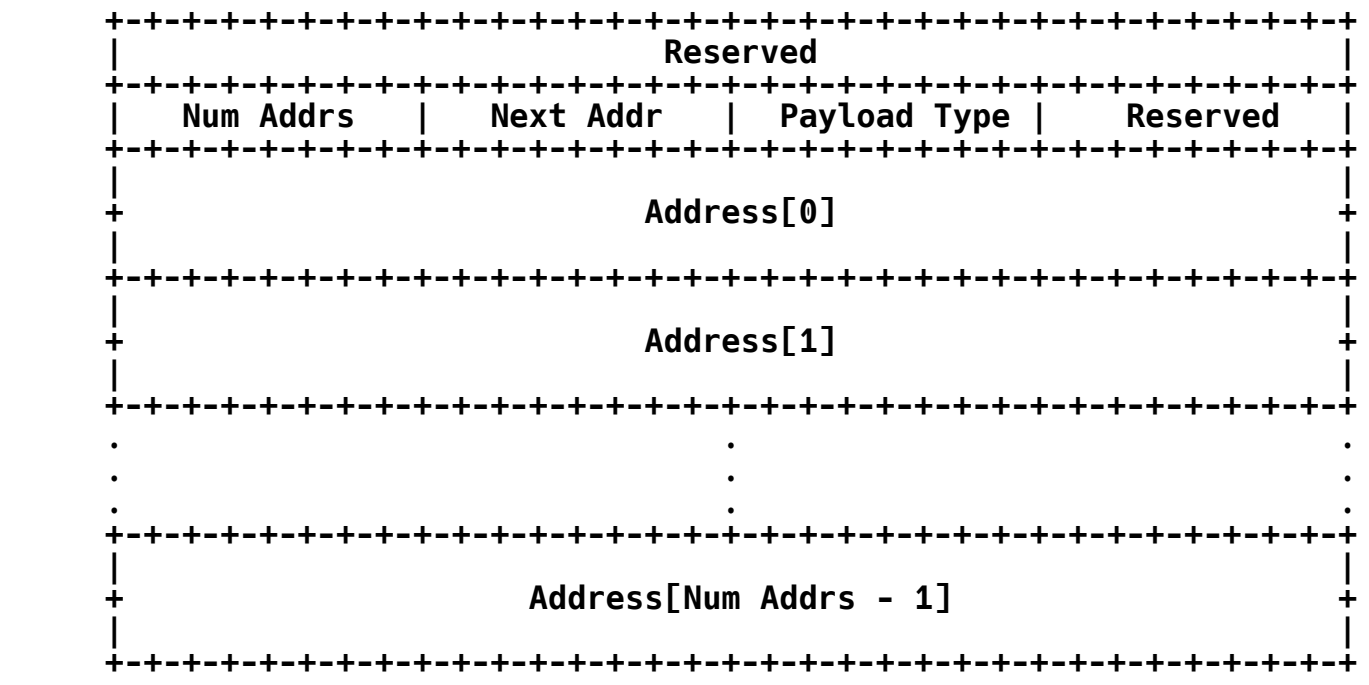
SIP systems are expected to implement Path MTU Discovery [RFC1191], in order to discover and take advantage of paths with MTU greater than 576 octets. However, a minimal SIP implementation (e.g., in a boot ROM) may simply restrict itself to sending packets no larger than 576 octets, and omit implementation of Path MTU Discovery.

Path MTU Discovery requires support both in the SIP layer and in the packetization layers. A system that supports Path MTU Discovery at the SIP layer may serve packetization layers that are unable to adapt to changes of the path MTU. Such packetization layers must limit themselves to sending packets no longer than 576 octets, even when sending to destinations that belong to the same subnet.

> (Note: Unlike IPv4, it is unnecessary in SIP to set a "Don't Fragment" flag in the packet header in order to perform Path MTU Discovery; that is an implicit attribute of every SIP packet. Also, those parts of the RFC-1191 procedures that involve use of a table of MTU "plateaus" do not apply to SIP, because the SIP version of the "Datagram Too Big" message always identifies the exact MTU to be used.)

7. Source Routing Header

A Payload Type of <TBD> in the immediately preceding header indicates the presence of this Source Routing header:



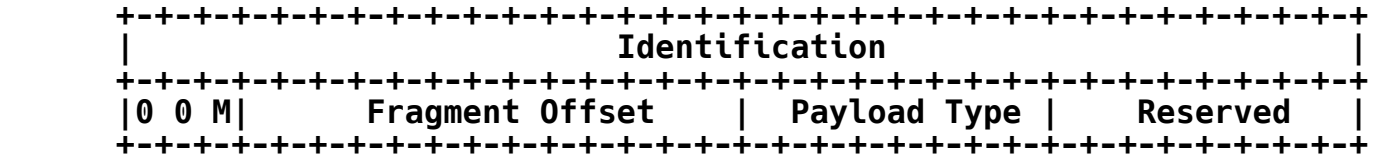
Reserved	Initialized to zero for transmission; ignored on reception.
Num Addrs	Number of addresses in the Source Routing header.
Next Addr	Index of next address to be processed; initialized to 0 by the originating system.
Payload Type	Identifies the type of payload following the Source Routing header.

A Source Routing header is not examined or processed until it reaches the system identified in the Destination Address field of the SIP header. In that system, dispatching on the Payload Type of the SIP (or subsequent) header causes the Source Routing module to be invoked, which performs the following algorithm:

- If Next Addr < Num Addrs, swap the SIP Destination Address and Address[Next Addr], increment Next Addr by one, and re-submit the packet to the SIP module for forwarding to the next destination.
- If Next Addr = Num Addrs, dispatch to the local protocol module identified by the Payload Type field in the Source Routing header.
- If Next Addr > Num Addrs, send an ICMP Parameter Problem message to the Source Address, pointing to the Num Addrs field.

8. Fragmentation Header

A Payload Type of <TBD> in the immediately preceding header indicates the presence of this Fragmentation header:



Identification	A value that changes on each packet sent with the same Source Address, Destination Address, and preceding Payload Type.
M flag	1 = more fragments; 0 = last fragment.
Fragment Offset	The offset, in 8-octet chunks, of the following payload, relative to the original, unfragmented payload.
Payload Type	Identifies the type of payload following the Fragmentation header.
Reserved	Initialized to zero for transmission; ignored on reception.

The Fragmentation header is NOT intended to support general, SIP-layer fragmentation. In particular, SIP routers shall not fragment a SIP packet that is too big for the MTU of its next hop, except in the special cases described below; in the normal case, such a packet results in an ICMP Packet Too Big message being sent back to its source, for use by the source system's Path MTU Discovery algorithm.

The special cases for which the Fragmentation header is intended are the following:

- A SIP packet that is "tunneled", either by encapsulation within another SIP packet or by insertion of a Source Routing header en-route, may, after the addition of the extra header fields, exceed the MTU of the tunnel's path; if the original packet is 576 octets or less in length, the tunnel entry system cannot respond to the source with a Packet Too Big message, and therefore must insert a Fragmentation header and fragment the packet to fit within the tunnel's MTU.

- An IPv4 fragment that is translated into a SIP packet, or an unfragmented IPv4 packet that is translated into too long a SIP packet to fit in the remaining path MTU, must include the SIP Fragmentation header, so that it may be properly reassembled at the destination SIP system.

Every SIP system must support SIP fragmentation and reassembly, since any system may be configured to serve as a tunnel entry or exit point, and any SIP system may be destination of IPv4 fragments. All SIP systems must be capable of reassembling, from fragments, a SIP packet of up to 1024 octets in length, including the SIP header; a system may be capable of assembling packets longer than 1024 octets.

Routers do not examine or process Fragmentation headers of packets that they forward; only at the destination system is the Fragmentation header acted upon (i.e., reassembly performed), as a result of dispatching on the Payload Type of the preceding header.

Fragmentation and reassembly employ the same algorithm as IPv4, with the following exceptions:

- All headers up to and including the Fragmentation header are repeated in each fragment; no headers or data following the Fragmentation header are repeated in each fragment.

- the Identification field is increased to 32 bits, to decrease the risk of wraparound of that field within the maximum packet lifetime over very high-throughput paths.

The similarity of the algorithm and the field layout to that of IPv4 enables existing IPv4 fragmentation and reassembly code and data structures to be re-used with little modification.

9. Changes to Other Protocols

Upgrading IPv4 to SIP entails changes to the associated control protocols, ICMP and IGMP, as well as to the transport layer, above, and possibly to the link-layer, below. This section identifies those changes.

9.1. Changes to ICMP

SIP uses a subset of ICMP [[RFC792], [RFC950], [RFC1122], [RFC1191], [RFC1256]], with a few minor changes and some additions. The presence of an ICMP header is indicated by a Payload Type of 1.

One change to all ICMP messages is that, when used with SIP, the ICMP checksum includes a pseudo-header, like TCP and UDP, consisting of the SIP Source Address, Destination Address, Payload Length, and Payload Type (see section 8.3).

There are a set of ICMP messages called "error messages", each of which, for IPv4, carries the IPv4 header plus 64 bits or more of data from the packet that invoked the error message. When used with SIP, ICMP error messages carry the first 256 octets of the invoking SIP packet, or the entire invoking packet if it is shorter than 256 octets.

For most of the ICMP message types, the packets retain the same format and semantics as with IPv4; however, some of the fields are given new names to match SIP terminology.

The changes to specific message types are as follows:

Destination Unreachable

The following Codes have different names when used with SIP:

1 - destination address unreachable (IPv4 "host unreachable")
7 - destination address unknown (IPv4 "dest. host unknown")
2 - payload type unknown (IPv4 "protocol unreachable")
4 - packet too big (IPv4 "fragmentation needed and DF set")

The following Codes retain the same names when used with SIP:

```
  3 - port unreachable
  5 - source route failed
  8 - source host isolated
 13 - communication administratively prohibited
```

The following Codes are not used with SIP:

```
  0 - net unreachable
  6 - destination network unknown
  9 - comm. with dest. network administratively prohibited
 10 - comm. with dest. host administratively prohibited
 11 - network unreachable for type of service
 12 - host unreachable for type of service
```

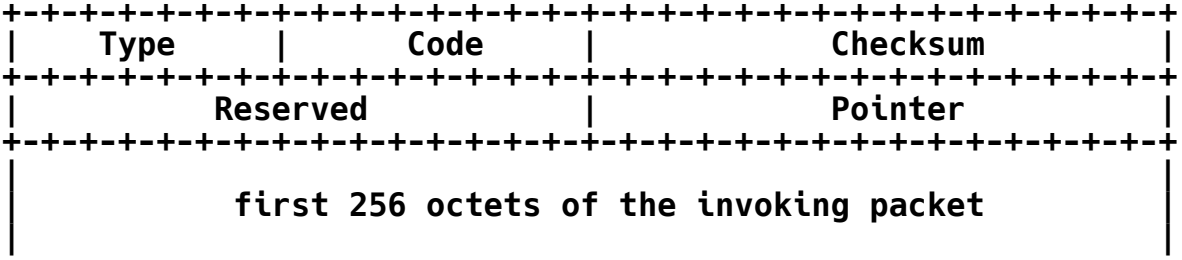
For "packet too big" messages (Code 4), the minimum legal value in the Next-Hop MTU field [RFC1191] is 576.

Time Exceeded

The name of Code 0 is changed to "hop limit exceeded in transit".

Parameter Problem

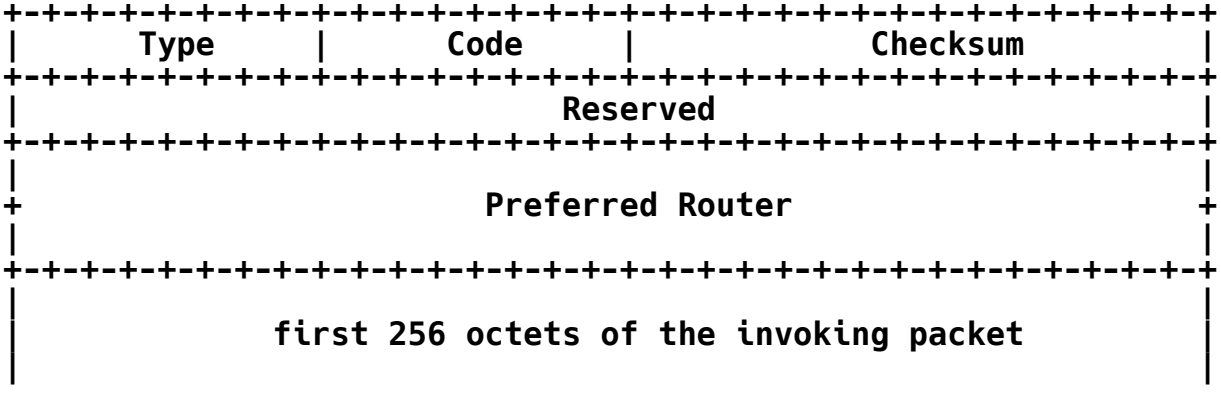
The Pointer field is extended to 16 bits and moved to the low-order end of the second 32-bit word, as follows:



Redirect

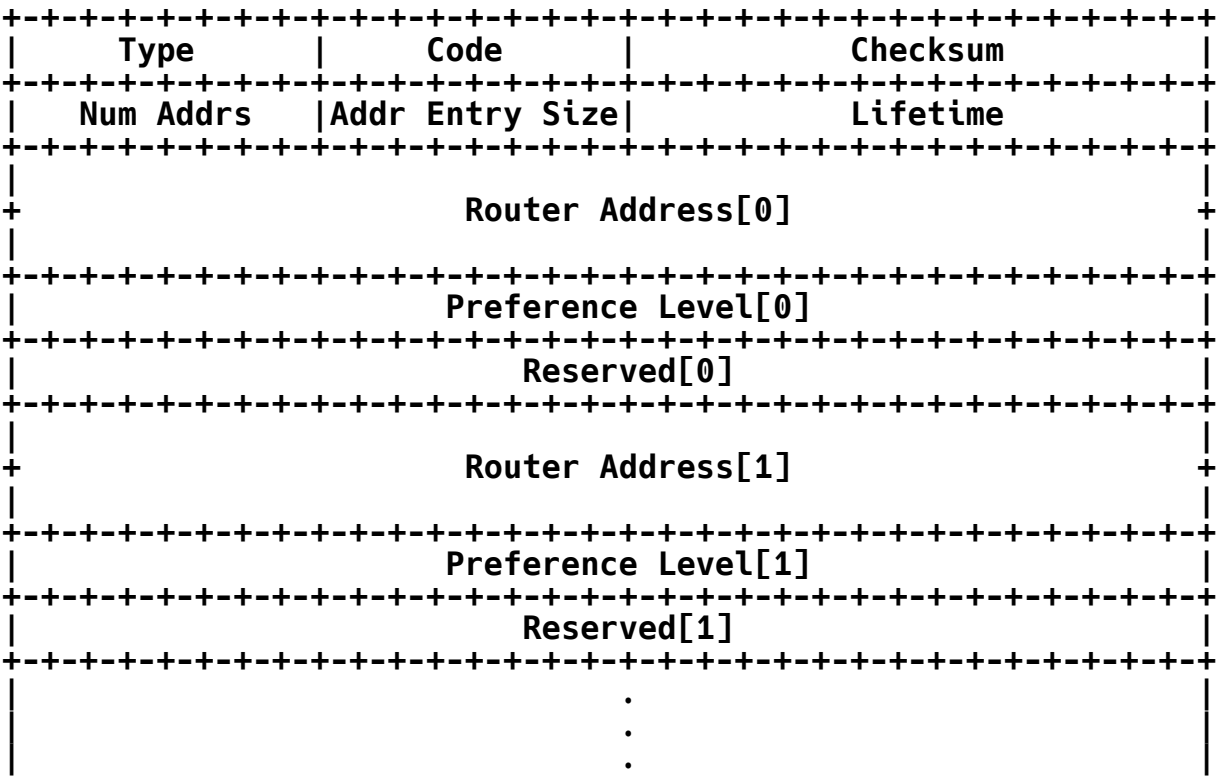
Only Code 1 is used for SIP, meaning "redirect packets for the destination address".

The Redirect header is modified for SIP, to accommodate the 64-bit address of the "preferred router" and to retain 64-bit alignment, as follows:



Router Advertisement

The format of the Router Advertisement message is changed to:



The value in the Addr Entry Size field is 4, and all of the Reserved fields are initialized to zero by senders and ignored by receivers.

Router Solicitation

No changes.

Echo and Echo Reply

No changes.

The following ICMP message types are not used with SIP:

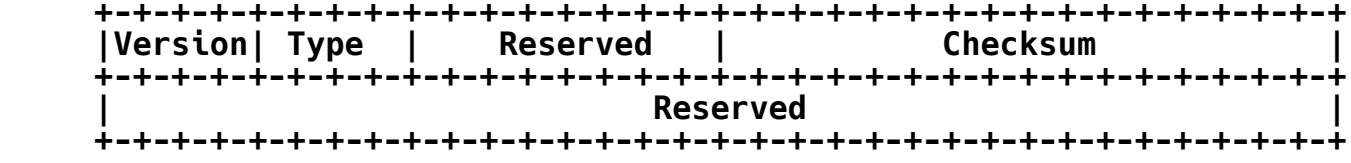
- Source Quench
- Timestamp
- Timestamp Reply
- Information Request
- Information Reply
- Address Mask Request
- Address Mask Reply

9.2. Changes to IGMP

SIP uses the Internet Group Management Protocol, IGMP [RFC1112]. The presence of an IGMP header is indicated by a Payload Type of 2.

When used with SIP, the IGMP checksum includes a pseudo-header, like TCP and UDP, consisting of the SIP Source Address, Destination Address, Payload Length, and Payload Type (see section 8.3).

The format of an IGMP Host Membership Query message becomes:



The format of an IGMP Host Membership Report message becomes:

For both message types, the Version number remains 1, and the Reserved fields are set to zero by senders and ignored by receivers.

9.3. Changes to Transport Protocols

The service interface to SIP has some differences from IPv4's service interface. Existing transport protocols that use IPv4 must be changed to operate over SIP's service interface. The differences from IPv4 are:

- Any addresses passed across the interface are 64 bits long, rather than 32 bits.
- The following IPv4 variables are not passed across the interface: Precedence, Type-of-Service, Identifier, Don't Fragment Flag
- SIP options have a different format than IPv4 options. (For SIP, "options" are all headers between, and not including, the SIP header and the transport header. The only IPv4 option currently specified for SIP is Loose Source Routing.
- ICMP error messages for SIP that are passed up to the transport layer carry the first 256 octets of the invoking SIP packet.

Transport protocols that use IPv4 addresses for their own purposes, such as identifying connection state or inclusion in a pseudo-header checksum, must be changed to use 64-bit SIP addresses for those purposes instead.

For SIP, the pseudo-header checksums of TCP, UDP, ICMP, and IGMP include the SIP Source Address, Destination Address, Payload Length, and Payload Type, with the following caveats:

- If the packet contains a Source Routing header, the destination address used in the pseudo-header checksum is that of the final destination.
- The Payload Length used in the pseudo-header checksum is the length of the transport-layer packet, including the transport header.
- The Payload Type used in the pseudo-header checksum is the Payload Type from the header immediately preceding the transport header.
- When added to the pseudo-header checksum, the Payload Type is treated as the left octet of a 16-bit word, with zeros in the the right octet, when viewed in IP standard octet order.

- If either of the two addresses used in the pseudo-header checksum has its high-order bit set to 1, only the low-order 32-bits of that address shall be used in the sum. The high-order bit is used to indicate that the addressed system is an IPv4 system, and that the low-order 32-bits of the address contain that system's IPv4 address [IPAE].

The semantics of SIP service differ from IPv4 service in three ways that may affect some transport protocols:

(1) SIP does not enforce maximum packet lifetime. Any transport protocol that relies on IPv4 to limit packet lifetime must take this change into account, for example, by providing its own mechanisms for detecting and discarding obsolete packets.

(2) SIP does not checksum its own header fields. Any transport protocol that relies on IPv4 to assure the integrity of the source and destinations addresses, packet length, and transport protocol identifier must take this change into account. In particular, when used with SIP, the UDP checksum is mandatory, and ICMP and IGMP are changed to use a pseudo-header checksum.

(3) SIP does not (except in special cases) fragment packets that exceed the MTU of their delivery paths. Therefore, a transport protocol must not send packets longer than 576 octets unless it implements Path MTU Discovery [RFC1191] and is capable of adapting its transmitted packet size in response to changes of the path MTU.

9.4. Changes to Link-Layer Protocols

Link-layer media that have an MTU less than 576 must be enhanced with a link-specific fragmentation and reassembly mechanism, to support SIP.

For links on which ARP is used by IPv4, the identical ARP protocol is used for SIP. The low-order 32-bits of SIP addresses are used wherever IPv4 addresses would appear; since ARP is used only among systems on the same subnet, the high-order 32-bits of the SIP addresses may be inferred from the subnet prefix (assuming the subnet prefix is at least 32 bits long). [This is subject to change -- see Appendix B.]

10. Security Considerations

<to be done>

11. Acknowledgments

The author acknowledges the many helpful suggestions and the words of encouragement from Dave Clark, Dave Crocker, Deborah Estrin, Bob Hinden, Christian Huitema, Van Jacobson, Jeff Mogul, Dave Nichols, Erik Nordmark, Dave Oran, Craig Partridge, Scott Shenker, Paul Tsuchiya, Lixia Zhang, the members of End-to-End Research Group and the IPAE Working Group, and the participants in the big-internet and sip mailing lists. He apologizes to those whose names he has not explicitly listed. [If you want to be on the list in the next draft, just let him know!]

Editor's note: Steve Deering was employed by the Xerox Palo Alto Research Center in Palo Alto, CA USA when this work was done.

12. Informative References

[IPAE] Crocker, D. and R. Hinden, "IP Address Encapsulation (IPAE): A Mechanism for Introducing a New IP", Work in Progress, draft-crocker-ip-encaps-01, November 1992.

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/info/rfc791>.

[RFC792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/info/rfc792>.

[RFC950] Mogul, J. and J. Postel, "Internet Standard Subnetting Procedure", STD 5, RFC 950, DOI 10.17487/RFC0950, August 1985, <https://www.rfc-editor.org/info/rfc950>.

[RFC951] Croft, W. and J. Gilmore, "Bootstrap Protocol", RFC 951, DOI 10.17487/RFC0951, September 1985, <https://www.rfc-editor.org/info/rfc951>.

[RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, DOI 10.17487/RFC1112, August 1989, <https://www.rfc-editor.org/info/rfc1112>.

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <https://www.rfc-editor.org/info/rfc1122>.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <https://www.rfc-editor.org/info/rfc1191>.

[RFC1256] Deering, S., Ed., "ICMP Router Discovery Messages", RFC 1256, DOI 10.17487/RFC1256, September 1991, <https://www.rfc-editor.org/info/rfc1256>.

[RFC1661] Simpson, W., Ed., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, DOI 10.17487/RFC1661, July 1994, <https://www.rfc-editor.org/info/rfc1661>.

[RFC1710] Hinden, R., "Simple Internet Protocol Plus White Paper", RFC 1710, DOI 10.17487/RFC1710, October 1994, <https://www.rfc-editor.org/info/rfc1710>.

[RFC1752] Bradner, S. and A. Mankin, "The Recommendation for the IP Next Generation Protocol", RFC 1752, DOI 10.17487/RFC1752, January 1995, <https://www.rfc-editor.org/info/rfc1752>.

[RFC1883] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 1883, DOI 10.17487/RFC1883, December 1995, <https://www.rfc-editor.org/info/rfc1883>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[SIP-ADDR] Deering, S., "Simple Internet Protocol (SIP) Addressing and Routing", Work in Progress, November 1992.

Appendix A. SIP Design Rationale

<this section still to be done>

Fields present in IPv4, but absent in SIP:

Header Length	Not needed; SIP header length is fixed.
Precedence & Type of Service	Not used; transport-layer Port fields (or perhaps a to-be-defined value in the Reserved field of the SIP header) may be used for classifying packets at a granularity finer than host-to-host, as required for special handling.
Header Checksum	Not used; transport pseudo-header checksum protects destinations from accepting corrupted packets.

Need to justify:

change of Total Length -> Payload Length, excluding header
change of Protocol -> Payload Type
change of Time to Live -> Hop Limit
movement of fragmentation fields out of fixed header
bigger minimum MTU, and reliance on PMTU Discovery

Appendix B. Future Directions

SIP as specified above is a fully functional replacement for IPv4, with a number of improvements, particularly in the areas of scalability of routing and addressing, and performance. Some additional improvements are still under consideration:

- ARP may be modified to carry full 64-bit addresses, and to use link-layer multicast addresses, rather than broadcast addresses.

- The 28-bit Reserved field in the SIP header may be defined as a "Flow ID", or partitioned into a Type of Service field and a Flow ID field, for classifying packets deserving of special handling, e.g., non-default quality of service or real-time service. On the other hand, the transport-layer port fields may be adequate for performing any such classification. (One possibility would be simply to remove the port fields from TCP & UDP and append them to the SIP header, as in XNS.)

- A new ICMP "destination has moved" message may defined, for re-routing to mobile hosts or subnets, and to domains that have changed their address prefixes.

- An explicit Trace Route message or option may be defined; the current IPv4 traceroute scheme will work fine with SIP, but it does not work for multicast, for which it has become very apparent that management and debugging tools are needed.

- A new Host-to-Router protocol may be specified, encompassing the requirements of router discovery, black-hole detection, auto- configuration of subnet prefixes, "beaconing" for mobile hosts, and, possibly, address resolution. The OSI End System To Intermediate System Protocol may serve as a good model for such a protocol.

- The requirement that SIP addresses be strictly bound to interfaces may be relaxed, so that, for example, a system might have fewer addresses than interfaces. There is some experience with this approach in the current Internet, with the use of "unnumbered links" in routing protocols such as OSPF.

- Authentication and integrity-assurance mechanisms for all clients of SIP, including ICMP and IGMP, may be specified, possibly based on the Secure Data Network System (SNDS) SP-3 or SP-4 protocol.

Authors' Addresses

Stephen E. Deering
Retired
Vancouver, British Columbia
Canada

Robert M. Hinden (editor)
Check Point Software
959 Skyway Road
San Carlos, CA 94070
USA

Email: bob.hinden@gmail.com