Minimal Encapsulation within IP

Status of This Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

Abstract

   This document specifies a method by which an IP datagram may be
   encapsulated (carried as payload) within an IP datagram, with less
   overhead than "conventional" IP encapsulation that adds a second IP
   header to each encapsulated datagram. Encapsulation is suggested as
   a means to alter the normal IP routing for datagrams, by delivering
   them to an intermediate destination that would otherwise not be
   selected by the (network part of the) IP Destination Address field in
   the original IP header. Encapsulation may be serve a variety of
   purposes, such as delivery of a datagram to a mobile node using
   Mobile IP.

1. Introduction

   This document specifies a method by which an IP datagram may be
   encapsulated (carried as payload) within an IP datagram, with less
   overhead than "conventional" IP encapsulation [4] that adds a second
   IP header to each encapsulated datagram. Encapsulation is suggested
   as a means to alter the normal IP routing for datagrams, by
   delivering them to an intermediate destination that would otherwise
   not be selected by the (network part of the) IP Destination Address
   field in the original IP header. The process of encapsulation and
   decapsulation of a datagram is frequently referred to as "tunneling"
   the datagram, and the encapsulator and decapsulator are then
   considered to be the the "endpoints" of the tunnel; the encapsulator
   node is refered to as the "entry point" of the tunnel, and the
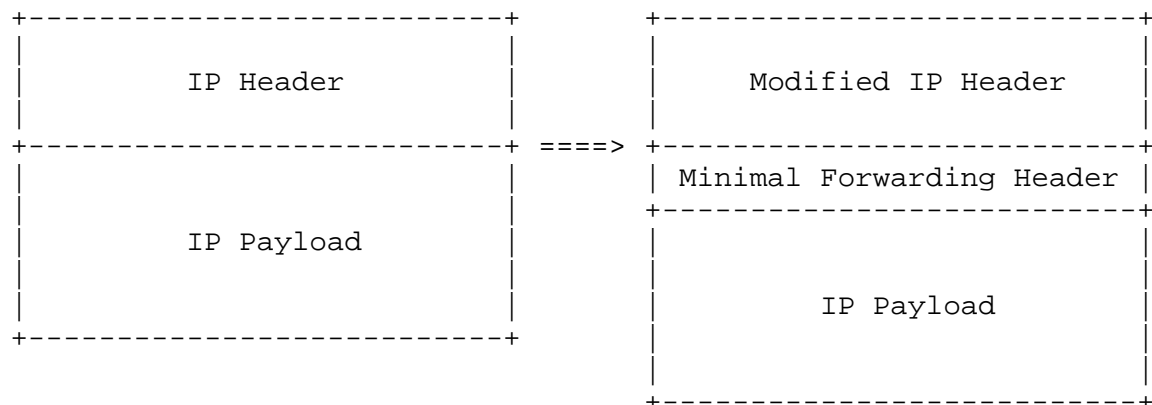   decapsulator node is refered to as the "exit point" of the tunnel.

2. Motivation

   The Mobile IP working group has specified the use of encapsulation as
   a way to deliver packets from a mobile node's "home network" to an
   agent that can deliver datagrams locally by conventional means to the
   mobile node at its current location away from home [5]. The use of
   encapsulation may also be indicated whenever the source (or an
   intermediate router) of an IP datagram must influence the route by
   which a datagram is to be delivered to its ultimate destination.
   Other possible applications of encapsulation include multicasting,
   preferential billing, choice of routes with selected security
   attributes, and general policy routing.

   See [4] for a discussion concerning the advantages of encapsulation
   versus use of the IP loose source routing option. Using IP headers
   to encapsulate IP datagrams requires the unnecessary duplication of
   several fields within the inner IP header; it is possible to save
   some additional space by specifying a new encapsulation mechanism
   that eliminates the duplication. The scheme outlined here comes from
   the Mobile IP Working Group (in earlier Internet Drafts), and is
   similar to that which had been defined in [2].

3. Minimal Encapsulation

   A minimal forwarding header is defined for datagrams which are not
   fragmented prior to encapsulation. Use of this encapsulating method
   is optional. Minimal encapsulation MUST NOT be used when an original
   datagram is already fragmented, since there is no room in the minimal
   forwarding header to store fragmentation information. To encapsulate
   an IP datagram using minimal encapsulation, the minimal forwarding
   header is inserted into the datagram, as follows:

```
   +--------------------------+         +--------------------------+
   |                          |         |                          |
   |        IP Header         |         |    Modified IP Header    |
   |                          |         |                          |
   +--------------------------+ ====>   +--------------------------+
   |                          |         | Minimal Forwarding Header |
   |                          |         +--------------------------+
   |       IP Payload         |         |                          |
   |                          |         |                          |
   |                          |         |        IP Payload        |
   +--------------------------+         |                          |
                                        |                          |
                                        +--------------------------+
```
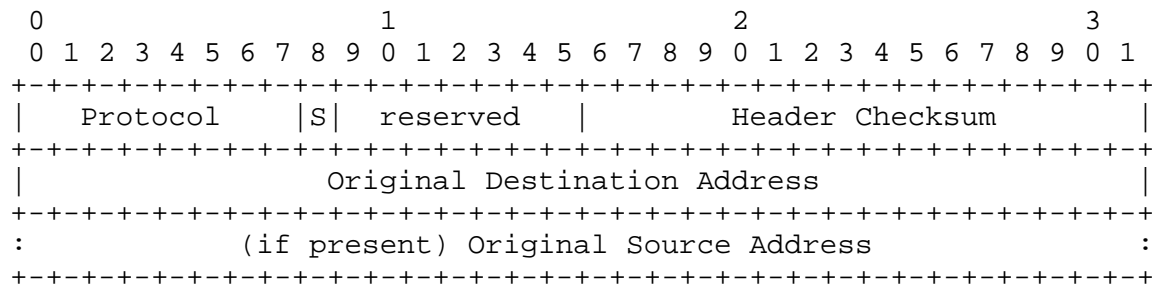
The IP header of the original datagram is modified, and the minimal
forwarding header is inserted into the datagram after the IP header,
followed by the unmodified IP payload of the original datagram (e.g.,
transport header and transport data). No additional IP header is
added to the datagram.

In encapsulating the datagram, the original IP header [6] is modified
as follows:

  - The Protocol field in the IP header is replaced by protocol
     number 55 for the minimal encapsulation protocol.

  - The Destination Address field in the IP header is replaced by the
     IP address of the exit point of the tunnel.

  - If the encapsulator is not the original source of the datagram,
     the Source Address field in the IP header is replaced by the IP
     address of the encapsulator.

  - The Total Length field in the IP header is incremented by the
     size of the minimal forwarding header added to the datagram.
     This incremental size is either 12 or 8 octets, depending on
     whether or not the Original Source Address Present (S) bit is set
     in the forwarding header.

  - The Header Checksum field in the IP header is recomputed [6] or
     updated to account for the changes in the IP header described
     here for encapsulation.

Note that unlike IP-in-IP encapsulation [4], the Time to Live
(TTL) field in the IP header is not modified during encapsulation;
if the encapsulator is forwarding the datagram, it will decrement
the TTL as a result of doing normal IP forwarding. Also, since
the original TTL remains in the IP header after encapsulation,
hops taken by the datagram within the tunnel are visible, for
example, to "traceroute".

The format of the minimal forwarding header is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Protocol   |S|  reserved   |          Header Checksum      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Original Destination Address                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:            (if present) Original Source Address               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Protocol

   Copied from the Protocol field in the original IP header.

Original Source Address Present (S)

      0   The Original Source Address field is not present. The
          length of the minimal tunneling header in this case is
          8 octets.

      1   The Original Source Address field is present. The
          length of the minimal tunneling header in this case is
          12 octets.

reserved

   Sent as zero; ignored on reception.

Header Checksum

   The 16-bit one's complement of the one's complement sum of all
   16-bit words in the minimal forwarding header. For purposes of
   computing the checksum, the value of the checksum field is 0.
   The IP header and IP payload (after the minimal forwarding
   header) are not included in this checksum computation.

Original Destination Address

   Copied from the Destination Address field in the original IP
   header.

Original Source Address

   Copied from the Source Address field in the original IP header.
   This field is present only if the Original Source Address
   Present (S) bit is set.

When decapsulating a datagram, the fields in the minimal forwarding
header are restored to the IP header, and the forwarding header is
removed from the datagram. In addition, the Total Length field in
the IP header is decremented by the size of the minimal forwarding
header removed from the datagram, and the Header Checksum field in
the IP header is recomputed [6] or updated to account for the changes
to the IP header described here for decapsulation.

The encapsulator may use existing IP mechanisms appropriate for
delivery of the encapsulated payload to the tunnel exit point. In
particular, use of IP options are allowed, and use of fragmentation
is allowed unless the "Don't Fragment" bit is set in the IP header.
This restriction on fragmentation is required so that nodes employing
Path MTU Discovery [3] can obtain the information they seek.

4. Routing Failures

The use of any encapsulation method for routing purposes brings with
it increased susceptibility to routing loops. To cut down the
danger, a router should follow the same procedures outlined in [4].

5. ICMP Messages from within the Tunnel

ICMP messages are to be handled as specified in [4], including the
maintenance of tunnel "soft state".

6. Security Considerations

Security considerations are not addressed in this document, but are
generally similar to those outlined in [4].

7. Acknowledgements

The original text for much of Section 3 was taken from the Mobile IP
draft [1]. Thanks to David Johnson for improving consistency and
making many other improvements to the draft.

References

    [1] Perkins, C., Editor, "IPv4 Mobility Support", Work in Progress,
        May 1995.

    [2] David B. Johnson. Scalable and Robust Internetwork Routing
        for Mobile Hosts. In Proceedings of the 14th International
        Conference on Distributed Computing Systems, pages 2--11, June
        1994.

    [3] Mogul, J., and S. Deering, "Path MTU Discovery", RFC 1191,
        November 1990.

    [4] Perkins, C., "IP Encapsulation within IP", RFC 2003,
        October 1996.

    [5] Perkins, C., Editor, "IP Mobility Support", RFC 2002,
        October 1996.

    [6] Postel, J., Editor, "Internet Protocol", STD 5, RFC 791,
        September 1981.

Author's Address

    Questions about this memo can be directed to:

    Charles Perkins
    Room H3-D34
    T. J. Watson Research Center
    IBM Corporation
    30 Saw Mill River Rd.
    Hawthorne, NY 10532

    Work: +1-914-784-7350
    Fax: +1-914-784-6205
    EMail: perk@watson.ibm.com

    The working group can be contacted via the current chair:

    Jim Solomon
    Motorola, Inc.
    1301 E. Algonquin Rd.
    Schaumburg, IL 60196

    Work: +1-847-576-2753
    EMail: solomon@comm.mot.com