INTERNET-DRAFT Michael Welzl
draft-welzl-pmtud-options-01.txt University of Innsbruck
Institute of Computer Science
Experimental 9. February 2003

PMTU-Options: Path MTU Discovery Using Options

Status of this Memo

Abstract

This document describes an experimental enhancement of Path MTU Discovery for special scenarios (e.g., tunnels, or to detect PMTU increases). It has the potential of reducing loss, speeding up convergence, reducing load in routers which would otherwise need to generate a large amount of ICMP messages, and alleviating certain additional problems (interactions with tunnels, Black Hole Detection). The idea is to use an IP Option which queries routers for their MTU before starting a Path MTU Discovery process. The result retrieved in this manner is used as an upper limit for Path MTU Discovery. To this end, it is fed back to the source either at the packetization layer (recommended) or at the IP layer.

Changes from draft-welzl-pmtud-options-00.txt:

- The addition of a "TTL-Check" field.

- The addition of a section on packet drops due to options.

- Update of section 5.1 on the impact of Slow Path processing.

- Update of references.

Table of Contents

1. Definitions

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.

Throughout this document, "Path MTU Discovery" (PMTUD) refers to the mechanism described in RFC 1191 [RFC1191] and "Packetization Layer Path MTU Discovery" (PLPMTUD) refers to the mechanism described in [draft-PLPMTUD]. The mechanism described in this document is called "Path MTU Discovery using Options" (PMTU-Options).

In the IP architecture, the choice of what size datagram to send is made by a protocol at a layer above IP. We refer to such a protocol as a "packetization protocol". Packetization protocols are usually transport protocols (for example, TCP) but can also be higher-layer protocols (for example, protocols built on top of UDP).

2. Introduction

This memo specifies how options can be used as an enhancement for PMTUD and PLPMTUD. The method resembles the mechanism described in [RFC1063]: a sender includes an IP Option containing the MTU of its outgoing link. Upon forwarding, each router compares the value with the MTUs of the links which are traversed by the datagram and updates the field if one of the MTUs of its links is smaller. If all routers support this scheme, the receiver has the correct MTU in the option and can communicate it back to the sender (preferably at the packetization layer instead of the IP layer as specified in [RFC1063]).

The main difference is that the mechanism described in this document does not rely on all routers along a path to support the IP option. Instead, when this scheme is carried out just before doing PMTUD or PLPMTUD, the result is used as an upper limit for the MTU of the path (i.e. the Path MTU will definitely not exceed the value obtained with this mechanism), no matter how many routers support it. This method has several potential advantages over standard PMTUD or PLPMTUD (listed in section 4) but also some issues (listed in section 5). Being an experimental specification, it is mainly intended for

special usage scenarios, some of which are described in section 6.

3. Specification

3.1. Probe MTU Option Format for IPv4

The "Probe MTU Option" for IPv4 that has routers update the MTU value (IP option number 11) is specified as in [RFC1063], with the exception of additional "TTL-Check" and "MTU Nonce" fields:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type = 11   |   Size = 8    |              MTU              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  TTL-Check    |                 MTU Nonce                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This option always contains the lowest MTU of all the links that have been traversed so far by the datagram.

A host that sends this option must initialize the MTU field to be the MTU of the directly-connected network. If the host is multi-homed, this should be for the first-hop network. Also, the host MUST set the TTL-Check field to a random value. It also also calculates and remembers TTL-Diff, the difference between the TTL value and the value of TTL-Check in the transmitted packet, as follows:

```
   TTL Diff = ( TTL - TTL-Check ) mod 256
```

The purpose of this calculation is to detect whether all routers along the path supported the option.

Each router that receives a datagram containing this option MUST compare the MTU field with the MTUs of the inbound and outbound links for the datagram. If either MTU is lower than the value in the MTU field of the option, the MTU field MUST be set to the lower MTU. (Note that routers conforming to RFC-1812 may not know either the inbound interface or the outbound interface at the time that IP options are processed. Accordingly, support for this option may require major router software changes). Additionally, a router MUST decrement the TTL-Check field on forwarding.

Any host receiving a datagram which contains this option should confirm that the value of the MTU field of the option is less than or equal to that of the inbound link, and if necessary, reduce the MTU field value, before processing the option.

If the receiving host is not able to accept datagrams as large as specified by the value of the MTU field of the option, then it should reduce the MTU field to the size of the largest datagram it can accept.

The MTU Nonce field is a means to prevent attackers from hiding the fact that a router has updated the MTU field and provide some protection against broken downstream routers. It MUST be initialized with a random non-zero 24-bit number by the sender. The number MUST be kept for later comparison with the Nonce value which is fed back to the sender. A router which updates the MTU field in the option MUST set the MTU Nonce field to 0.

3.2. Feedback Format

IP Option processing is known to be a costly operation. To avoid placing this burden on routers along the backward path, feedback SHOULD be stored at the packetization layer; it MAY be stored at the IP layer in special cases (e.g. if PMTU-Options is used by a UDP based application). Header extensions are specified for IP, TCP, SCTP and DCCP. A host implementing PMTU-Options SHOULD react to this feedback in all of the supported protocols and provide the MTU value to the local PMTUD or PLPMTUD instance. The PMTUD or PLPMTUD instance MUST NOT increase a cached PMTU value in response to PMTU-Options feedback. If the MTU value from this feedback is greater than or equal to the cached MTU value and the MTU Nonce is set to 0, there is a chance that an intermediate node or the receiver misbehaved (due to broken software or because of an attack).

3.2.1. IP

The Reply MTU Option for IPv4 (IP option number 12) is specified as in [RFC1063], with the exception of additional "TTL-Diff" and "MTU Nonce" fields:

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   Type = 12   |    Size = 8   |              MTU              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    TTL-Diff   |                MTU Nonce                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This option is used to return the value learned from a Probe MTU

Option to the sender of the Probe MTU Option at the IP level. Since it causes unnecessary processing overhead in routers along the backward path, its usage is only recommended for rare special cases. In particular, it may be helpful for UDP-based applications which utilize PMTU-Options.

The first octet of this option contains the option type, identifying the IP option. The second octet of this option contains the size field, specifying the option length in octets. The size field is set to 8. The next three fields (two, one and three octets, respectively) contain the result of the MTU-Options process, TTL-Diff and the MTU Nonce; the MTU and MTU Nonce fields must be copied from the IPv4 Probe MTU Option by the receiver. The receiver must set the TTL-Diff field to (TTL - TTL-Check) mod 256, where "TTL" is the TTL field in the IP header.

3.2.2. TCP

The Reply MTU Option format for TCP over IPv4 is:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Kind      |   Length = 8  |              MTU              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    TTL-Diff   |                 MTU Nonce                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The first octet of this option contains the option kind, identifying the TCP option (to be specified by IANA). The second octet of this option contains the length field, specifying the option length in octets. The length field is set to 8. The MTU, TTL-Diff and MTU Nonce fields are similar to the specification in section 3.2.1.

3.2.3. SCTP

In SCTP [RFC2960], the sender is informed by the receiver about PMTU-Options feedback by including the Reply MTU chunk. This chunk looks 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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   Chunk Type  | Flags=00000000|      Chunk Length = 12        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              MTU              |    TTL-Diff   |   MTU Nonce   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     MTU Nonce (continued)     |          Padding (0)          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Note: The Reply MTU chunk is considered a Control chunk.

The Chunk Type field identifies the chunk; it consists of two high-order bits "11", indicating that a processing SCTP node which does not recognize this chunk type must skip this chunk and continue processing, but report in an ERROR Chunk using the "Unrecognized Chunk Type" cause of error. The sender of the Reply MTU chunk SHOULD send the MTU feedback via some other means (via a different active protocol or an IP option) in response to this ERROR Chunk. The trailing six bits are currently undefined (to be specified by IANA). Since this chunk has no specific flags, the Flags field is set to 0. The Chunk Length field is set to 12 (the length of the chunk including the Chunk Type, Flags and Length fields).

The MTU, TTL-Diff and MTU Nonce fields are similar to the specification in section 3.2.1; since the length of a SCTP chunk must be a multiple of 4 octets, the last octet is filled with 0 (padding).

3.2.4. DCCP

The Feedback MTU Option format for DCCP over IPv4 is:

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |   Length = 8  |              MTU              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    TTL-Diff   |                  MTU Nonce                    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Type field indentifies the option (number to be specified by IANA). The rest of the option is similar to the TCP option defined in section 3.2.2.

3.3 Host Operation

Generally, the PMTU-Options process is carried out before initiating a regular PMTUD or PLPMTUD process.

Since IP options require processing at every router along a path, it is important to ensure that no packets unnecessarily include IP options. Thus, a host implementing PMTU-Options must keep state in routing table entries and only initiate a PMTUD or PLPMTUD (and accompanying PMTU-Options) process when this information becomes stale. The process of purging stale PMTU information is specified in [RFC1191], sections 6.2 and 6.3. Additionally, the number of datagrams carrying IP options should be restricted with a fixed percentage of total datagrams that are sent by a host to ensure scalability. It can also be reduced by using PMTU-Options in special situations only; a discussion of such potential usage scenarios is provided in section 6.

It is assumed that the datagram size used when doing PMTU-Options is already some sensible value (e.g., the result of a recent PMTU process or the first-hop data-link MTU). When a Probe MTU Option is added to a datagram, it is prolonged by 8 octets (16 octets with IPv6). This number must therefore be taken into account when generating a datagram -- when doing PMTU-Options, the size of the datagram including the Probe MTU Option must not exceed the recent PMTU value. The correct size must be communicated to the packetization layer protocol (see [RFC1191], section 6.4, for a suggestion of how such communication could be implemented).

A host receiving a packet that carries the Probe MTU Option MUST feed back the information to the sender by copying the MTU and MTU Nonce fields and TTL-Diff as specified in section 3.2 to the next return datagram. To this end, it SHOULD inform a packetization layer protocol which is communicating with the originator of the Probe MTU Option. Alternatively, a host receiving a packet that carries the Probe MTU Option MAY use the MTU Reply IP Option; this method is not recommended because it involves unnecessary processing overhead in routers on the return path.

A host MUST be able to accept MTU feedback from IP and SHOULD be able to accept feedback from all packetization layer protocols. This feedback contains a MTU Nonce value which either has the same value as the MTU Nonce field in the original Probe MTU Option (a random number initialized by the sender), meaning that the initial MTU value in the Probe MTU Option was not changed by routers, or 0, meaning that the initial MTU value in the Probe MTU Option was changed by routers. If the PMTU upper limit from PMTU-Options feedback is

greater than the initial PMTU value stored at the sender, the latter value MUST be used. If, in such a case, the MTU Nonce was changed, there is a chance that an intermediate node or the receiver misbehaved (due to broken software or because of an attack).

The resulting PMTU upper limit at the PMTU-Options sender MUST be communicated to the local PMTUD or PLPMTUD instance. Note that because the options are placed on unreliable datagrams, the original sender will have to resend probes (possibly once per window of data) until it receives feedback.

If the value of TTL-Diff in the reply packet is equal to the stored value of TTL-Diff, all routers along the path supported the option. This means that the MTU value in the reply packet is not an upper limit for the PMTU but the actual end result; this fact SHOULD be communicated to the local PMTUD or PLPMTUD instance, which MAY then terminate immediately using the value from the packet carrying the MTU feedback.

A host that implements PMTU-Options SHOULD wait for feedback to arrive before initiating the subsequent PMTUD or PLPMTUD process, which remains unchanged except for one difference: the MTU during the process SHOULD not exceed the value received from PMTU-Options. Feedback from PMTU-Options SHOULD not be kept any longer -- it is only intended as an aid for the subsequent PMTUD or PLPMTUD process. Since packet drops can substantially delay the reception of feedback, a host MAY use a timer to initiate PTMUD or PLPMTUD even when no feedback has arrived; if such a timer is implemented, a means to configure this timer MUST be provided.

3.4. IPv6 Usage

Path MTU Discovery for IPv6 is specified in [RFC1981]. PMTU-Options can be combined with this PMTUD variant just like regular PMTUD or PLPMTUD; the mechanism should work with IPv6 without requiring any substantial changes. The following sections describe the IPv6 formats for the Probe MTU Option and feedback -- these formats differ from the IPv4 variants in that the MTU field is larger (4 instead of 2 octets) to support IPv6 jumbograms [RFC2675] and the MTU Nonce field is larger (8 instead of 4 octets) to ensure 8-octet alignment without wasting space for padding octets. Also, the TTL field in the IPv4 header, which is used for calculations related to TTL-Check, is the Hop Limit field in the case of IPv6.

3.4.1. Probe MTU Option Format for IPv6

The option format specified in section 3.1 is a Hop-by-Hop Options header in the IPv6 case. The PMTU-Options Hop-by-Hop Options header is identified by a Next Header value of 0 in the IPv6 header, and has the following format:

```
 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
                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |  Option Type  |Opt Data Len=12|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              MTU                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   TTL-Check   |                                               |
+-+-+-+-+-+-+-+-+         7-octet MTU Nonce field               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Option Type field identifies the option; it consists of two high-order bits "00", indicating that a processing IPv6 node which does not recognize this option type must skip over this option and continue processing the header. The following bit "1" indicates that the Option Data field (MTU, TTL-Check and MTU Nonce) may change en-route. The trailing five bits are currently undefined (to be specified by IANA). The MTU field was stretched to 4 octets to support IPv6 jumbograms [RFC2675].

Since it may be assumed that, when either of the option-bearing headers are present, they carry a very small number of options -- usually only one [RFC2460] -- the MTU Nonce field was stretched to fit the 8-octet alignment (otherwise, a PadN Option of 6 octets would have to be used in most cases). This way, the security of PMTU-Options is enhanced. A complete Hop-by-Hop Options header containing this one option would look 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Next Header  | Hdr Ext Len=1 |  Option Type  |Opt Data Len=12|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              MTU                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   TTL-Check   |                                               |
+-+-+-+-+-+-+-+-+         7-octet MTU Nonce field               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.4.2. Feedback Format: IP

The option format specified in section 3.2.1 is a Destination Options header in the IPv6 case. The PMTU-Options Destination Options header is identified by a Next Header value of 60 in the immediately preceding header, and has the following format:

```
    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
                                   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                   |  Option Type  |Opt Data Len=12|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              MTU                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    TTL-Diff   |                                               |
   +-+-+-+-+-+-+-+-+         7-octet MTU Nonce field               +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Option Type field identifies the option; it consists of two high-order bits "00", indicating that a processing IPv6 node which does not recognize this option type must skip over this option and continue processing the header. The following bit "0" indicates that the Option Data field (MTU, TTL-Diff and MTU Nonce) does not change en-route. The trailing five bits are currently undefined (to be specified by IANA).

The second octet of this option contains the Opt Data Len field, specifying the length of the Option Data field in octets. The Opt Data Len field is set to 12. The next three fields (four, one and seven octets, respectively) contain the result of the MTU-Options process, TTL-Diff and the MTU Nonce; the MTU and MTU Nonce fields must be copied from the IPv6 Probe MTU Option by the receiver. The receiver must set the TTL-Diff field to (TTL - Hop Limit) mod 256, where "Hop Limit" is the Hop Limit field in the IPv6 header.

A complete Destination Options header containing this one option would look 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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Next Header  | Hdr Ext Len=1 |  Option Type  |Opt Data Len=12|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              MTU                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    TTL-Diff   |                                               |
   +-+-+-+-+-+-+-+-+         7-octet MTU Nonce field               +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.4.3. Feedback Format: TCP

The Reply MTU Option format for TCP over IPv6 is:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Kind       |  Length = 14  |              MTU              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         MTU (continued)       |    TTL-Diff   |   MTU Nonce   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      MTU Nonce (continued)    |     MTU Nonce (continued)     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      MTU Nonce (continued)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The first octet of this option contains the option kind, identifying the TCP option. The second octet of this option contains the length field, specifying the total option length in octets. The length field is set to 14. The MTU, TTL-Diff and MTU Nonce fields are similar to the specification in section 3.4.2.

3.4.4. Feedback Format: SCTP

In SCTP [RFC2960], the sender is informed by the receiver about PMTU-Options feedback by including the Reply MTU chunk. For IPv6, this chunk looks 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Chunk Type  | Flags=00000000|      Chunk Length = 16        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              MTU                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    TTL-Diff   |                                               |
+-+-+-+-+-+-+-+-+         7-octet MTU Nonce field               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Note: The Reply MTU chunk is considered a Control chunk.

The Chunk Type field identifies the chunk; it consists of two high-

order bits "11", indicating that a processing SCTP node which does not recognize this chunk type must skip this chunk and continue processing, but report in an ERROR Chunk using the "Unrecognized Chunk Type" cause of error. The sender of the Reply MTU chunk SHOULD send the MTU feedback via some other means (via a different active protocol or an IP option) in response to this ERROR Chunk. The trailing six bits are currently undefined (to be specified by IANA). Since this chunk has no specific flags, the Flags field is set to 0. The Chunk Length field is set to 16 (the length of the chunk including the Chunk Type, Flags and Length fields).

The MTU, TTL-Diff and MTU Nonce fields are similar to the specification in section 3.4.2.

3.4.5. Feedback Format: DCCP

The Reply MTU Option format for DCCP over IPv6 is:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |  Length = 14  |              MTU              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         MTU (continued)       |    TTL-Diff   |   MTU Nonce   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      MTU Nonce (continued)    |     MTU Nonce (continued)     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      MTU Nonce (continued)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Type field indentifies the option (number to be specified by IANA). The rest of the option is similar to the TCP option defined in section 3.4.3.

3.5. IP Tunnels

If a network node that performs IP-in-IP tunneling (be it because of IPsec, IPv6 or for any other purpose) encapsulates a datagram carrying the Probe MTU Option, it should copy this option to the outer IP header, no matter how many headers there are in between. If the network node at the "other side of the tunnel" (the node that unpackages an IP datagram by removing the outer header) receives a datagram carrying the Probe MTU Option in the (outer) IP header, it should subtract the length of the outer and

all intermediate headers from the value in the (outer header) option and then copy it into the inner header option on unpackaging.

Such nodes SHOULD NOT change the MTU Nonce field.

4. Potential Advantages

The PMTU-Options mechanism has a number of potential advantages:

4.1. Reducing Packet Loss

Probing for the Path MTU means that at some point, a datagram with a size exceeding the Path MTU must be sent with the DF bit set to 1. Such a datagram will be dropped, which normally requires the data it carried to be retransmitted. This retransmission is additional traffic which is caused by PMTUD or PLPMTUD.

The result of PMTU-Options is to be interpreted as an upper limit for the Path MTU. If it turns out to be the Path MTU during the subsequent PMTUD or PLPMTUD process, the Path MTU is detected without causing packet loss.

4.2. Circumventing Black Hole Detection

PMTUD is known to have a problem called "Black Hole Detection" [RFC2923], which happens when a PMTU sender does not receive an ICMP Fragmentation Needed message even though the size of a datagram with DF set to 1 has exceeded the MTU of a link. This may happen if, for instance, routers or firewalls are misconfigured. PLPMTUD is robust against this failure because it does not rely on ICMP.

Since the information from PMTU-Options is explicit even when the size of a datagram has not exceeded the MTU of a link, the "Black Hole Detection" problem of PMTUD could be circumvented in a special case that is best explained by means of a simple example:

```
        MTU A  ------  MTU B  ------  MTU C  ------  MTU A
Sender --------| R1 |---------| R2 |---------| R3 |--------- Receiver
               ------         ------         ------
```

In this example, a datagram traverses routers R1, R2 and R3, and MTU A > MTU B > MTU C. PMTUD has converged a long time ago, and a path

change has occured. Not having noticed any problems, the host now probes for a larger MTU. Routers R1 and R2 are misconfigured not to send ICMP "Fragmentation Needed" messages -- the size is increased and exceeds MTU B, but the sender never notices this because R1 simply drops the datagram.

Now let us assume that PTMU-Options is used. The sender first submits a datagram carrying a Probe MTU Option; R3, which is properly configured, updates the value in the option (originally A) with C. Since PMTUD uses this value as an upper limit, it never exceeds the MTU of the link between routers R1 and R2, and Black Hole Detection does not occur.

4.3. Other Problems with ICMP Fragmentation Needed

In addition to the danger of leading to the Black Hole Detection problem, utilizing ICMP Fragmentation Needed messages has the following additional problems:

- Generating an ICMP packet requires memory to be allocated, header fields to be initialized etc., which is a costly operation for a router.
- An ICMP message is additional signaling traffic that consumes network capacity.
- An ICMP message can be dropped, e.g. as the result of congestion on the return path.

By replacing the generation of an ICMP Fragmentation Needed message with a simple option field update, PMTU-Options circumvents these problems.

4.4. Circumventing Problems with IP Tunnels

Sometimes, the DF bit is ignored by network nodes that encapsulate IP packets: the total length of a packet with additional headers may exceed the Path MTU of the tunnel even if this is not the case without the extra headers. Also, the inner nodes of a tunnel are often invisible to the data flow that is carried through the tunnel. It would therefore be up to the network nodes at the edge of the tunnel to perform PMTUD or PLPMTUD and fragment a packet appropriately, if necessary. Simply ignoring the DF bit is an attractive alternative.

If the PMTU-Options mechanism is supported by routers along a tunnel

path and IP Options are copied to the outer IP header, it is possible to detect a potentially smaller MTU in the tunnel, thereby decreasing the chance of fragmentation in the tunnel.

5. Discussion of Issues

In what follows, we discuss some obvious problems with PMTU-Options:

5.1. Slow Path Processing

IP packets carrying options are known to be processed in the "Slow Path" (software, as opposed to the hardware-only "Fast Path") in most normal IP router. This means that packets carrying the Probe MTU Option will experience a notable delay along the forward path. It is therefore not recommended to use datagrams that belong to a PMTU-Options process for round-trip time estimation; this makes it questionable whether PMTU-Options should be used at the beginning of a TCP connection.

In IETF and IRTF mailing lists, the impact of option processing has been claimed to be immense (e.g., slowing down packets by factor 100) on several occasions. While this may be true for packet processing in a single router, a recent measurement study has shown that the impact on end-to-end delay can be much less severe [WeRo]. The study encompassed two separate tests -- one in July/August 2002 and one in August/September 2003. In each of these tests, a ping carrying a NOP IP Option was sent to a host from the list at [TBIT], followed by a ping without the option; this process was repeated 100 times per host, and there was a pause of 1 second in between all pings -- thus, these measurements do not say anything about the (possibly different) behavior of routers when they are flooded with a large number of packets that contain IP options. Additionally, a traceroute was carried out for each host.

4427 and 4401 hosts reacted to pings with options, with a number of hops ranging from 6 to 34 (most hosts were in the range of 14 - 25 hops). The pings traversed 5726 unique router addresses (not necessarily as many different machines because different interfaces on a single router may show up as different addresses in traceroute) in 2002 and 5194 in 2003. The measurements from hosts that answered with options (approximately 1/4) were ignored in the final statistics because the backward path is unknown. The rest of the data was weighted based on the frequency of occurence (a router which shows up 100 times in the measurements is 100 times less important than a router which shows up only once), path length and variance (to

diminish the effect of queuing delay).

The final result was that on average, slow path processing in routers caused an additional delay of 10% (2002) and 7% (2003) along the forward path if a packet contains a NOP option. These results appear to concur with the results reported in [FrJo], where similar measurements are described.

5.2. Dropped Packets

In addition to the resulting impact that slow path processing has on the round-trip time, the measurements reported in [WeRo] revealed an alarming fact: in the 2003 test run, 3507 out of 7908 hosts reacted to regular pings but did not react when a NOP option was used. At this point, it is unclear who dropped the pings that carried options: routers along the paths to the hosts, firewalls, the hosts themselves, or routers along the backward paths (because the hosts may have included the option in their responses). Also, the reaction to different types of packets (e.g., TCP SYN) is unknown.

For PMTU-Options, this means that the mechanism can easily lead to packet drops, in particular if the receiver is not known to support it. This may have adverse effects on the packetization layer if these drops are interpreted as a sign of congestion. This is one particular reason to consider PMTU-Options as experimental and propose its usage for special scenarios only.

5.3. Placing Additional Burden on Routers

PMTUD and PLPMTUD only require the "problematic" router (router attached to a link with an MTU that was just exceeded) to do substantial extra work (notably, all the routers on the return path from the "problematic" router to the sender are involved in forwarding an ICMP Fragmentation Needed message in the case of standard PMTUD [RFC1191]). PMTU-Options involves all routers along the path by having them process IP options. In other words, while the burden for an individual router is smaller (processing of the Probe MTU Option is probably a less costly operation than generating an ICMP Fragmentation Needed message), the burden for the whole network is perhaps greater.

Since the processing overhead caused by the Probe MTU option in routers is unknown, it is important to limit the amount of such packets in a network; clearly, PMTU-Options should not be used for each and every new TCP connection but in special scenarios only.

5.4. Motivating Deployment

From the perspective of a single node, there is no immediate gain when deploying PMTU-Options; at least the sender, receiver and a router (ideally attached to the link with the Path MTU -- otherwise the only benefit of PMTU-Options could be faster PMTUD convergence because it starts with a smaller value) must participate for the mechanism to be beneficial. However, the same is true for Explicit Congestion Notification (ECN) [RFC3168], a deployment overview of which is given at [ECN].

PMTU-Options has the following motivating deployment factor: a router with a particularly small MTU will typically need to send a large number of ICMP packets. This is where PMTU-Options deployment would be most beneficial because it might lead to reduced CPU load. From the perspective of an end host where PMTU-Options is used, such routers are exactly the ones that should be updated because they have small MTUs.

6. Discussion of Usage Scenarios

Since the Probe MTU Option places an additional burden on routers via IP option processing and the additional delay from Slow Path processing can falsify a round-trip time estimation, it is questionable whether the mechanism should be used at the beginning of a standard TCP connection. Being an experimental PMTUD enhancement, PMTU-Options is rather intended to be used under special circumstances -- depending on the importance of the aforementioned advantages as opposed to the gravity of the aforementioned issues with the mechanism. In what follows, some examples of potential usage scenarios are given:

6.1. Detecting PMTU Increases

Since the PMTU may change (e.g., when a routing change occurs) and even become larger while a long-lasting connection is active, [RFC1191] describes a method to probe for increased MTUs (which should be done rarely). The recommended method increases the packet size according to a table specified in [RFC1191]; alternatively, the "aged" cached PMTU values may be reset to the first-hop data-link MTU. Since chances are high that the PMTU did not change and either of these process will therefore immediately exceed the current PMTU, it may be recommendable to use PMTU-Options before increasing a cached PMTU value.

6.2. RTT-robust Transport Protocols

Transport protocols that do not rely on round-trip time estimates as heavily as TCP or SCTP may be a good fit for PMTU-Options. In particular, this includes DCCP [draft-DCCP], where the importance of round-trip time estimates depends on the Congestion Control ID (CCID), and UDP, where no round-trip time estimation is specified. Currently, PMTUD is unavailable for applications that utilize UDP -- it is up to such applications to find out about the ideal size of a datagram. PMTUD or PLPMTUD may however be provided to such applications in the future (the issue has been discussed in the PMTUD Working Group). Assuming that it is available to applications running on top of UDP, it may be recommendable for such an application to use PMTU-Options depending on its requirements.

6.3. Tunnels

The operation of PMTU-Options across tunnels is specified in section 3.5, the potential advantage of this kind of operation is described in section 4.4. In addition to the viewpoint of an application traversing a tunnel without wanting PMTUD to fail, the endpoints of a tunnel may sometimes also need to determine the MTU in between them (the viewpoint being a tunnel with endpoints which do not care about actual application endpoints) [draft-TunnelMTU]. In such a case, using PMTU-Options may be recommendable due to the potentially controlled (or known) tunnel environment and sporadic MTU determination at tunnel endpoints.

7. Related Work

This work can be seen as an extension of the basic idea in [RFC1063]. A related document is [draft-PMTUDv6], where the idea of utilizing options for PMTUD is described for IPv6. This is the only other text that the author is aware of where a Probe MTU option is combined with regular PMTUD. Some of the design choices in this document (e.g., the MTU Nonce) were based on [RFC3168] and [draft-QS].

8. Security Considerations

Since MTU-Options requires routers to change a value in packets and must therefore always be placed in the outermost IP header to remain functioning, it cannot be fully protected with IPsec; however, as the explicit MTU update to the sender originates from the receiver of an end-to-end connection and not an intermediate router as with ICMP Fragmentation Needed messages, the receiver can be authenticated

using the Encapsulation Security Payload (ESP) header [RFC2406] or the Authentication Header (AH) [RFC2402]. For this purpose, the Probe MTU Option is classified as mutable and the Reply MTU Option is classified as immutable. The mutable header field (MTU field in the Probe MTU Option) is where IPsec cannot help -- it cannot prevent intermediate attackers from sending false data. The following attacks from such a malicious node must be considered:

- Sending a MTU value that is too large:

  A host MUST ignore feedback containing an MTU that is larger than the MTU it initially wrote into the option. If a router reduces the MTU value, it MUST set the MTU Nonce to 0 -- to hide this fact, a malicious node would have to guess the original MTU Nonce, which has a 1/(2^32) chance of success (1/2^128 with IPv6). If a malicious node ignores the MTU Nonce and still increases the MTU value, the attacker can only succeed if the new value is smaller than the (unknown) original value from the sender. Even then, the value received from this option is only used as an upper limit for PMTUD and PLPMTUD, which will still work in case of such an attack. Only the benefit of PMTU-Options can be lost.

- Sending a MTU value that is too small:

  This attack, which is similar to an attack with ICMP "Fragmentation needed" messages carrying a value that is too small, can degrade the performance of a sender. PMTU-Options does not provide a mechanism to prevent this attack. This is one of the most important reasons to consider it experimental: the result of PMTU-Options should merely be used as a hint.

- Lying about the number of routers that supported the option:

  The number of routers that supported the option can be faked by altering TTL-Check or TTL-Diff. The only plausible attack based on this value would be to claim that all routers along the path supported the option, as this might cause the sender to use the MTU value in the reply packet right away without starting a PMTUD or PLPMTUD process. However, since the initial value of TTL-Check is a random number, the chance of a malicious node guessing the right value of TTL-Check is at most 1/256.

9. IANA Considerations

This specification reuses the obsolete IPv4 option numbers 11 and 12. It requires two IPv6 Option Type numbers (with leading bits "001" and "000"), a TCP option number, a SCTP Chunk Type number (with leading

bits "00") and a DCCP option number.

10. Normative References

[RFC1191] Mogul, J.C., and Deering, S.E., "Path MTU discovery", RFC 1191, November 1990.

[RFC1981] McCann, J., Deering, S. and Mogul, J., "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

[draft-PLPMTUD] Mathis, M., Heffner, J. and Lahey, K., "Path MTU Discovery", Internet-draft draft-ietf-pmtud-method-00.txt, October 19, 2003.

[RFC1435] Knowles, S., "IESG Advice from Experience with Path MTU Discovery.", RFC 1435, March 1993.

[RFC2460] Deering, S., and Hinden, R., "Internet Protocol, Version 6 (IPv6)", RFC 2460, December 1998.

[RFC2923] Lahey, K., "TCP Problems with Path MTU Discovery", RFC 2923, September 2000.

[RFC2960] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., and Paxson, V., "Stream Control Transmission Protocol", RFC 2960, October 2000.

[draft-DCCP] Kohler, E., Handley, M., Floyd, S., and Padhye, J., "Datagram Congestion Control Protocol (DCCP)", Internet-draft draft-ietf-dccp-spec-05.txt, October 2003.

[RFC2402] Kent, S., and Atkinson, R., "IP Authentication Header", RFC 2402, November 1998

11. Informative References

[RFC1063] Mogul, J.C., Kent, C.A., Partridge, C., and McCloghrie, K., "IP MTU discovery options.", RFC 1063, July, 1988.

[WeRo] Welzl, M., and Rossi, M., "On The Impact of IP Option Processing", Preprint-Reihe des Fachbereichs Mathematik-Informatik (technical report), No. 15, 2003. Available from http://www.welzl.at/ip-options

[FrJo] Fransson, P., and Jonsson, A., "The Need for an Alternative to IPv4-Options", RVK (RadioVetenskap och Kommunikation), Stockholm, Sweden, pp. 162-166, June 2002.

[RFC2675] Borman, D., Deering, S., and Hinden, R., "IPv6 Jumbograms", RFC 2675, August 1999.

[draft-TunnelMTU] Templin, F., "Dynamic MTU Determination for IPv6-in-IPv4 Tunnels", Internet-draft draft-ietf-templin-tunnelmtu-06.txt, November 2003. Currently available from: http://www.geocities.com/osprey67/tunnelmtu-06.txt

[RFC3168] Ramakrishnan, K., Floyd, S., and Black, D., "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[ECN] The ECN website, http://www.icir.org/floyd/ecn.html

[draft-PMTUDv6] Park, S. D., and Lee, H., "The PMTU Discovery for IPv6 Using Hop-by-Hop Option Header", Internet-draft draft-park-pmtu-ipv6-option-header-00.txt, March 2003 (expired).

[draft-QS] Jain, A., and Floyd, S., "Quick-Start for TCP and IP", Internet-draft draft-amit-quick-start-02.txt, October 2002 (expired). Available from http://www.icir.org/floyd/quickstart.html

[TBIT] The TBIT website, http://www.icir.org/tbit/

[RFC2406] Kent, S., and Atkinson, R., "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

12. Acknowledgements

The author would like to thank everybody who contributed to this document via helpful discussions. In particular, this includes: Eddie Kohler, Simon Leinen, Matt Mathis, Michael Richardson and Fred Templin.

13. Author's Address

Michael Welzl
University of Innsbruck
Institute fuer Informatik
Technikerstr. 25/7
A-6020 Innsbruck, Austria

Phone: +43 (512) 507-6110
Fax: +43 (5122) 507-2977
Email: michael.welzl@uibk.ac.at
Web: http://www.welzl.at

14. Full Copyright Statement