CLNP Path MTU Discovery

Status of this Memo

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts.

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

Please check the id-abstracts.txt listing contained in the internet-drafts Shadow Directories on nic.ddn.mil, nnscc.nsf.net, nic.nordu.net, ftp.nisc.sri.com, or munnari.oz.au to learn the current status of any Internet Draft.

Distribution of this memo is unlimited. Comments should be submitted to the tuba@lanl.gov mailing list.

Abstract

This memo describes a technique for dynamically discovering the maximum transmission unit (MTU) of an arbitrary CLNP path. The mechanism described here is applicable to both "pure-stack" OSI as well as TUBA/CLNP [6] environments, i.e., environments where Internet transport protocols (UDP and TCP) are operated over CLNP. This technique might not in all cases discover the optimum Path MTU, but it will always choose a Path MTU as accurate as, and in many cases more accurate than, the Path MTU that would be chosen by current practice.

Acknowledgements

The mechanism proposed here was first suggested by Geof Cooper, and incorporated into RFC 1191 [1], Path MTU Discovery, by Jeff Mogul and Steve Deering. The excellent work of these folks readily extends to CLNP-based internets. Thanks also to Steve Deering and Mike Shand, for their comments on early drafts.
1. Introduction

ISO/IEC 8473, Protocol for Providing the Connectionless Network Service, [2] is a network layer datagram protocol. As is the case for hosts in IP-based internets, a CLNP-based host that has a large amount of data to send to another CLNP-based host transmits that data as a series of CLNP datagrams. The desire to reduce or eliminate fragmentation is the same in CLNP-based internetworking environments as for IP [3]. (Refer to [4] for arguments against fragmentation.). It is thus desirable to define a mechanism that determines the largest size datagram that does not require fragmentation anywhere along the path from the source to the destination; this is referred to as the Path MTU (PMTU), and it is equal to the minimum of the MTUs of each hop in the path.

A shortcoming of the OSI protocol suite is the lack of a standard mechanism for a host to discover the PMTU of an arbitrary path. This document addresses this shortcoming by applying a mechanism demonstrated to be effective on IP-based internets.

ISO/IEC 8473 indicates that minimum subnetwork service data unit size an underlying service must offer to CLNP is 512 octets. This is as close as OSI comes to specifying a host requirement on what is referred to in Internet literature as a maximum segment size (MSS, [5]). The current practice in CLNP-based internets is to use the smaller of 512 and the first-hop MTU as the PMTU for any destination that is not connected to the same subnetwork as the source. This often results in the use of smaller CLNP datagrams than necessary, because it is increasingly the case that paths supporting CLNP offer a PMTU greater than 512. As is the case with IP, a host that sends CLNP datagrams smaller than the Path MTU allows wastes Internet resources and applications operating on that host are provided suboptimal throughput.

Future routing protocols may be required to provide accurate PMTU information within a routing domain, although perhaps not across multi-level routing hierarchies. Like IP networks, CLNP-based networks need a simple mechanism that discovers PMTUs without wasting resources within a routing domain, and in interdomain communications exchanges as well. The mechanism described here should serve the community until (and perhaps beyond) such time as routing protocol extensions are developed and deployed.

The initial mechanism described does not rely on changes to CLNP. Improvements in the mechanism can be achieved through the addition of a new option to the CLNP Error Report.
2. Protocol overview

The RFC 1191 technique of using the Don’t Fragment (DF) bit in the IP header to dynamically discover the PMTU of an IP path is easily extended to CLNP by using the Segmentation Permitted (SP) flag in the CLNP header. A source CLNP host initially assumes that the MTU of a path is the (known) MTU of its first hop, and sends all datagrams on that path with segmentation disabled (i.e., the SP = FALSE). If any of the datagrams are too large to be forwarded without fragmentation by some router along the path, that router will discard them and return a CLNP Error Report message with the Reason for Discard parameter set to the value indicating "segmentation needed but not permitted". Upon receipt of such a message (consistent with RFC 1191, this is referred to as a "Datagram Too Big" message), the source host reduces its assumed PMTU for the path. Since the mechanism relies on the generation of an Error Report message by a router along the path, hosts MUST NOT suppress error reporting (i.e., hosts MUST set the Error Report flag to TRUE in CLNP headers when attempting Path MTU discovery.

The PMTU discovery process ends when a host’s estimate of the PMTU is low enough that its datagrams can be delivered without fragmentation. Alternatively, the host could end the discovery process by enabling segmentation (SP = TRUE) in the datagram headers; it could do so, for example, because it is willing to have datagrams fragmented in some circumstances. Normally, the host continues to set SP = FALSE in all datagrams, so that if the route changes and the new PMTU is lower, the lower PMTU will be discovered.

2.1 Datagram Too Big message considerations

The Datagram Too Big message as originally specified in ICMP [7] did not report the MTU of the hop for which the rejected datagram was too big; the CLNP Error Report fails in this regard as well, so again, the source host cannot tell exactly how much to reduce its assumed PMTU given the information returned in the Error Report. To remedy this, a new option is defined for CLNP Error Reports in Appendix A. The Next-Hop-MTU option should convey the same semantics as the corresponding parameter in the ICMP header as specified in RFC 1191; i.e. This field is used to report the MTU of what RFC 1191 refers to as the "constricting (next) hop".

Although this is the only change needed for routers to fully support CLNP PMTU Discovery, it will not be possible to take advantage of this explicit feedback mechanism until all routers are upgraded, because the processing of CLNP options...
TUBA Working Group                    CLNP Path MTU Discovery

requires that Error Reports containing unrecognized options be (silently) discarded. Until such time as routers are updated, hosts may search for an accurate PMTU estimate by continuing to send datagrams with the SP = FALSE while varying datagram sizes. By using the search strategy described in Section 7, hosts can discover an optimum (or at least better) PMTU with good performance.

This memo recommends that all hosts that implement PMTU MUST implement both the search method described in section 7 and the option method described here and in Appendix A, with preference given the option, if present in the Error Report.

2.2 Path MTU changes

The MTU of a path may change over time, due to changes in the routing topology. Reductions of the PMTU are indicated by Datagram Too Big messages.

Hosts that choose to implement MTU discovery and cease the process by enabling segmentation (SP = TRUE) change the composition of the CLNP header, by forcing the addition of a segmentation part. RFC 1191 suggests that IP hosts that implement MTU discovery will normally continue to set the DF bit in all datagrams to detect PMTU changes resulting from routing changes; it is STRONGLY RECOMMENDED that under the same circumstances, CLNP hosts follow suit, and to continue to transmit datagrams in the discovery mode.

A host may periodically increase its assumed PMTU to detect increases in a PMTU. As is the case with IPv4, this will almost always result in CLNP datagrams being discarded and Datagram Too Big messages being generated, because in most cases the PMTU of the path will not have changed, so the increase "probe" should be done infrequently.

Note: this mechanism essentially guarantees that a CLNP host will not receive fragments from a peer doing PMTU Discovery, so a host that continues to operate in MTU discovery mode will interoperate with "segmentation-challenged" hosts; i.e., hosts that are unable to reassemble fragmented datagrams as a result of having implemented the non-segmenting subset rather than the full version of CLNP.

3. Host specification

When a host receives a Datagram Too Big message, it MUST reduce its estimate of the PMTU for the relevant path. The precise behavior of a host in this circumstance is not specified here, since different applications may have
different requirements, and different implementation architectures may favor different strategies.

After receiving a Datagram Too Big message, a host MUST avoid eliciting more such messages in the near future. The host has two choices; (1) reduce the size of the datagrams it sends along the path, or (2) set the segmentation flag in the CLNP header and use segmentation. A host MUST force the PMTU Discovery process to converge.

Hosts performing PMTU Discovery MUST detect decreases in Path MTU as fast as possible. Hosts MAY detect increases in PMTU, but since doing so requires sending datagrams larger than the current estimated PMTU, and since it is likely is that the PMTU will not have increased, this MUST be done at infrequent intervals. Consistent with RFC 1191 recommendations for IP, an attempt to detect an increase by sending a CLNP datagram larger than the current estimate MUST NOT be done less than 5 minutes after a Datagram Too Big message has been received for the given destination, or less than one minute after a previous, successful attempted increase. The recommended setting of these timers is twice their minimum values (10 and 2 minutes, respectively).

RFC 1191 recommends that a host MUST never reduce its estimate of the PMTU below 68 octets (the value of 68 octets guarantees that 8 octets of data can be transmitted given an IPv4 header of 60 octets, see RFC 791). CLNP implementations SHOULD NOT allow the MTU size to be configured to be less than 512 octets. A CLNP host SHOULD NEVER reduce its estimate of the PMTU below 512 octets.

3.1. TCP MSS Option

A host performing CLNP PMTU Discovery must obey the rule that it not send datagrams larger than 512 octets unless it has permission from the receiver. For TCP connections, this means that a TUBA/CLNP host must not send datagrams larger than 74 octets plus the Maximum Segment Size (MSS) sent by its peer.

Note: In RFC 879, the TCP MSS is defined to be the relevant IP datagram size minus 40, where 40 represents what is referred to as the "liberal or optimistic" assumption regarding TCP and IP header size (20 octets each); the default of 576 octets for the maximum IP datagram size in this scenario yields a default of 536 octets for the TCP MSS. Using CLNP, with a correspondingly liberal and optimistic assumption about CLNP header size (54 octets), the default CLNP MSS of 512 octets yields a default of 438 octets for the TCP MSS.
Hosts SHOULD not lower the value they send in the MSS option; doing so prevents the PMTU Discovery mechanism from discovering PMTUs larger than the default TCP MSS. For TUBA/CLNP hosts, the TCP MSS option should be 74 octets less than the size of the largest datagram the host is able to reassemble (MMS_R, as defined in RFC 1122 [8]). In many cases, this will be the architectural limit of 65461 (65535 – 74) octets. A host MAY send an MSS value derived from the MTU of its connected network (the maximum MTU over its connected networks, for a multi-homed host); this should not cause problems for PMTU Discovery, and may dissuade a broken peer from sending enormous datagrams.

Note: RFC 1191 recommends that hosts refrain from sending an MSS greater than the architectural limit of 65535 minus the IP header size. This recommendation applies for TUBA/CLNP hosts as well (i.e., do not use a value greater than 65461).

4. Router specification

When a router is unable to forward a datagram because (a) the datagram length exceeds the MTU of the next-hop network, (b) segmentation is disabled (SP = FALSE), and (c) the Suppress Error Reports flag is reset, the router MUST attempt to return an Error Report message to the source of the datagram, with the Reason for Discard parameter code set to indicate "segmentation required but not permitted".

To support MTU discovery, all routers MUST recognize the option specified in Appendix A and are STRONGLY ENCOURAGED to be capable of generating the option. Having all the routers recognize the option will allow the option to be returned to the host engaged in MTU discovery. (It is recommended that a router’s ability to generate the option be operator-configurable; generation of the option can then be implemented in an incremental fashion.).

5. Host processing of Error Report messages

RFC 1191 outlines several possible strategies a host may follow upon receiving a Datagram Too Big message from a router that has not implemented the next-hop-MTU parameter. This section describes the strategies as they apply to TUBA/CLNP hosts; however, the discussion here is limited to the strategies that RFC 1191 identifies as tractable.

The simplest thing for a CLNP host to do in response to a Datagram Too Big message is to assume that the PMTU is the minimum of its currently-assumed PMTU and 512, and to enable segmentation (SP = TRUE) in datagrams sent on that path.
Thus, the host falls back to the same PMTU as it would choose under current practice. This strategy terminates quickly and does no worse than existing practice, but it fails to avoid fragmentation in some cases, and fails to make the most efficient utilization of the internetwork in other cases. More sophisticated strategies involve "searching" for an accurate PMTU estimate, by continuing to send datagrams with SP = FALSE while varying datagram sizes.

A good search strategy is one that obtains an accurate estimate of the PMTU without causing many packets to be lost in the process. The "MTU Plateau" strategy recommended in RFC 1191 for IP applies to CLNP hosts. The strategy begins with the assumption that there are relatively few MTU values in use in the Internet, so the search can be constrained to include only the MTU values that are likely to appear. Mogul and Deering make the assumption that designers tend to choose MTUs in similar ways, so they collect groups of similar MTU values and use the lowest value in the group as a search "plateau", suggesting that it is better to underestimate an MTU by a few per cent than to overestimate it by one.

Section 7 provides a table of representative MTU plateaus for use in PMTU estimation, derived from RFC 1191, but extended to include technologies that have emerged since its publication. With this table, convergence is as good as binary search in the worst case, and is far better in common cases. Since the plateaus lie near powers of two, if an MTU is not represented in this table, the algorithm will not underestimate it by more than a factor of 2.

In RFC 1191, Mogul and Deering note that any search strategy must have some "memory" of previous estimates in order to choose the next one, and suggest that the information available in the Datagram Too Big message itself can be used for this purpose. Like ICMP Destination Unreachable messages, CLNP Error report messages contain the header of the original datagram, which contains the Total Length of the datagram too big to be forwarded without fragmentation (note: when SP = FALSE, the total length of the CLNP datagram is recorded in the Segment Length field). Since this Total Length may be less than the current PMTU estimate, but is nonetheless larger than the actual PMTU, it may be a good input to the method for choosing the next PMTU estimate.

Consistent with the strategy recommended for IP in RFC 1191, CLNP hosts shall use as the next PMTU estimate the greatest plateau value that is less than the returned Total Length field.
6. Host implementation

The RFC 1191 discussion of how PMTU Discovery is implemented in host software is relevant here. The issues that are applicable to CLNP MTU Discovery include:

- What layer or layers implement PMTU Discovery?
- Where is the PMTU information cached?
- How is stale PMTU information removed?
- What must transport and higher layers do?

6.1. Layering

In the IP architecture, the choice of what size datagram to send is made by a transport or higher layer protocol, i.e., a layer above IP. Mogul and Deering call such protocols "packetization protocols", and explain how implementing PMTU Discovery in the packetization layers simplifies some of the inter-layer issues, but has several drawbacks, and conclude that the IP layer should store PMTU information and that the ICMP layer should process received Datagram Too Big messages.

In OSI, the functions ascribed to ICMP and IP are both provided in the network layer. The division of function between the packetization and network layer changes slightly. The packetization layers must still respond to changes in the Path MTU by changing the size of the datagrams they send, and must also be able to specify when segmentation of datagrams is not permitted (SP = FALSE). (As is the case with IP, the network (CLNP) layer does not simply set SP = FALSE in every packet, since it is possible that a packetization layer, i.e., UDP or an application outside the kernel, is unable to change its datagram size.)

To support this layering in CLNP, packetization layers require an extension of the network service interface defined in [8]. The extension provides a way to learn of changes in the value of MMS_S, the "maximum send transport-message size", which is derived from the Path MTU by subtracting the minimum CLNP header size (52 octets). This interaction might take the form of an OSI network service primitive; i.e., an N-MSS_S-CHANGE.indication. (For completeness, one may wish to extend the N-UNITDATA.request primitive in [9] to allow transport-entities to control the setting of the SP flag.)

6.2. Storing PMTU information

The general guidelines for storing PMTU information are the same for CLNP as IP. The network (CLNP) layer should associate each PMTU value that it has learned with a specific
path, identified by a source address, a destination address, a CLNP quality-of-service, and if implemented, a security classification. This association can be stored as a field in the routing table entries. A host will not have a route for every possible destination, but it should be able to cache a per-host route for every active destination (A requirement already imposed by the need to process ES-IS Redirects [10].)

PMTU storage guidelines for IP also apply to CLNP. When the first packet is sent to a host for which no per-host route exists, a route is chosen either from the set of per-network routes, or from the set of default routes. The PMTU fields in these route entries should be initialized to be the MTU of the associated first-hop data link, and must never be changed by the PMTU Discovery process. (PMTU Discovery only creates or changes entries for per-host routes). The PMTU associated with the initially-chosen route is presumed to be accurate until a Datagram Too Big message is received.

When a Datagram Too Big message is received, the network layer determines a new estimate for the Path MTU. If a per-host route for this path does not exist, then one is created (as if a per-host ES-IS Redirect is being processed; the new route uses the same first-hop router as the current route). If the PMTU estimate associated with the per-host route is higher than the new estimate, then the value in the routing entry is changed.

The packetization layers must be notified about decreases in the PMTU (for example, through an implementation equivalent of the primitive earlier described). Any packetization layer instance (for example, a TCP connection) that is actively using the path must be notified if the PMTU estimate is decreased. Even if the Datagram Too Big message contains an original datagram header that refers to a UDP packet, the TCP layer must be notified if any of its connections use the given path. (The same would be true for CLTP and TP-4 connections in OSI internets.)

The packetization layer instance that sent the CLNP datagram that elicited the Datagram Too Big message should be notified that its datagram has been dropped, even if the PMTU estimate has not changed, so that it may retransmit the dropped datagram. This notification can be asynchronously generated by the network (CLNP) layer, or the notification can be postponed until the packetization instance next attempts to send a CLNP datagram larger than the PMTU estimate. In the latter approach, if one assumes that an N-UNITDATA.request is used to model the request to send a datagram, and the primitive is extended to include the ability to twiddle the
SP flag, and the datagram is larger than the PMTU estimate, the send function should fail and return a suitable error indication. In RFC 1191, Mogul and Deering suggest that this approach may be more suitable to a connectionless packetization layer (such as one using UDP), which may be hard to "notify" from the ICMP (or network) layer; this should not be the case for CLNP, however, if so, the normal timeout-based retransmission mechanisms would be used to recover from the dropped datagrams.

Mogul and Deering are careful to note that the notification to the packetization layer instances using the path about the change in the PMTU is distinct from the notification of a specific instance that a packet has been dropped. The latter should be done as soon as practical (i.e., asynchronously from the point of view of the packetization layer instance), while the former may be delayed until a packetization layer instance wants to create a packet. Retransmission should be done for only those packets that are known to be dropped, as indicated by a Datagram Too Big message. This applies to CLNP Path MTU discovery for TUBA/CLNP environments as well.

6.3. Purging stale PMTU information

RFC 1191 provides guidelines for aging PMTU information. Similar guidelines apply for TUBA/CLNP MTU discovery.

Because (under normal circumstances) a host performing CLNP PMTU Discovery always disables segmentation, a stale PMTU value (one that is too large) will be discovered almost immediately once a datagram is sent to the given destination. No such mechanism exists for determining that a stored PMTU value is too small, so an implementation SHOULD "age" cached PMTU values. When a PMTU value has not decreased for some time (on the order of 10 minutes), the PMTU estimate SHOULD be set to the first-hop data-link MTU, and the packetization layers should be notified of the change. This will cause the complete PMTU Discovery process to take place again.

Note: an implementation should provide a means for changing the timeout duration, including setting it to "infinity". In RFC 1191, Mogul and Deering cite the example of hosts attached to an FDDI network, which is then attached to the rest of the Internet via a slow serial line; such hosts will never discover a larger, non-local PMTU, so they should not be subjected to dropped datagrams every 10 minutes.

An upper layer MUST not retransmit datagrams in response to an increase in the PMTU estimate, since this increase never comes in response to an indication of a dropped datagram.
RFC 1191 and this memo recommend that PMTU aging be implemented by adding a timestamp field to the routing table entry. This field SHOULD be initialized to a "reserved" value that indicates that the PMTU has never been changed. Whenever the PMTU is decreased in response to a Datagram Too Big message, the timestamp is set to the current time. Once a minute thereafter, a timer-driven procedure should run through the routing table, and for each entry whose timestamp is not "reserved" and is older than the timeout interval,

- set the PMTU estimate to the MTU of the associated first hop

- notify the packetization layers using this route of the increase.

PMTU estimates may disappear from the routing table if the per-host routes are removed; this can happen in response to an ES-IS Redirect message, or because certain routing-table daemons delete old routes after several minutes. Also, on a multi-homed host a topology change may result in the use of a different source interface. When this happens, if the packetization layer is not notified then it may continue to use a cached PMTU value that is now too small. RFC 1191 and this memo suggest that the packetization layer be notified of a possible PMTU change whenever a Redirect message causes a route change, and whenever a route is deleted from the routing table.

6.4. TCP layer actions

RFC 1191 provides guidelines for TCP layers when Path MTU discovery is being performed. Similar guidelines apply for TUBA/CLNP MTU discovery.

The TCP layer must track the PMTU for the destination of a connection; it should not send datagrams that would be larger than this. A simple implementation could ask the network (CLNP) layer for this value (using a TUBA/CLNP equivalent of the GET_MAXSIZES interface described in [8]) each time it created a new segment, but this could be inefficient. Moreover, TCP implementations that follow the "slow-start" congestion-avoidance algorithm [11] typically calculate and cache several other values derived from the PMTU. It may be simpler to receive asynchronous notification when the PMTU changes, so that these variables may be updated.

A TCP implementation must also store the MSS value received from its peer (which defaults to 440), and not send any segment larger than this MSS, regardless of the PMTU.
When a Datagram Too Big message is received, it implies that a datagram was dropped by the router that sent the Error Report message. It is sufficient to treat this as any other dropped segment, and wait until the retransmission timer expires to cause retransmission of the segment. If the PMTU discovery process requires several steps to estimate the right PMTU, this could delay the connection by many round-trip times. Alternatively, the retransmission could be done in immediate response to a notification that the Path MTU has changed, but only for the specific connection specified by the Datagram Too Big message. The datagram size used in the retransmission should be no larger than the new PMTU.

Note: Retransmissions MUST not be sent in response to every Datagram Too Big message. A burst of oversized segments will give rise to several such messages and hence several retransmissions of the same data; if the new estimated PMTU is still wrong, the process repeats, and there is an exponential growth in the number of superfluous segments sent. This means that the TCP layer must be able to recognize when a Datagram Too Big notification actually decreases the PMTU that it has already used to send a datagram on the given connection, and should ignore any other notifications.

Many TCP implementations now incorporate "congestion avoidance" and "slow-start" algorithms to improve performance [11, 12]. Unlike a retransmission caused by a TCP retransmission timeout, a retransmission caused by a Datagram Too Big message should not change the congestion window. It should, however, trigger the slow-start mechanism (i.e., only one segment should be retransmitted until acknowledgements begin to arrive again).

TCP performance can be reduced if the sender’s maximum window size is not an exact multiple of the segment size in use (this is not the congestion window size, which is always a multiple of the segment size). In many systems (such as those derived from 4.2BSD), the segment size is often set to 1024 octets, and the maximum window size (the "send space") is usually a multiple of 1024 octets, so the proper relationship holds by default. If PMTU Discovery is used, however, the segment size may not be a submultiple of the send space, and it may change during a connection; this means that the TCP layer may need to change the transmission window size when PMTU Discovery changes the PMTU value. The maximum window size should be set to the greatest multiple of the segment size (PMTU - 74) that is less than or equal to the sender’s buffer space size.

PMTU Discovery does not affect the value sent in the TCP MSS.
option, because that value is used by the other end of the connection, which may be using an unrelated PMTU value.

6.5. Issues for other transport protocols

Some transport protocols (such as OSI TP4 [13]) are not allowed to repacketize when doing a retransmission; once an attempt is made to transmit a datagram of a certain size, its contents cannot be split into smaller datagrams for retransmission. In such a case, the original CLNP datagram should be retransmitted with segmentation enabled, allowing it to be fragmented as necessary to reach its destination. Subsequent datagrams, when transmitted for the first time, should be no larger than allowed by the Path MTU, and should have the SP = FALSE.

The Sun Network File System (NFS) uses a Remote Procedure Call (RPC) protocol [14] that, in many cases, sends datagrams that must be fragmented even for the first-hop link. This might improve performance in certain cases, but it is known to cause reliability and performance problems, especially when the client and server are separated by routers. NFS implementations SHOULD use PMTU Discovery whenever routers are involved. Most NFS implementations allow the RPC datagram size to be changed at mount-time (indirectly, by changing the effective file system block size), but might require some modification to support changes later on.

Also, since a single NFS operation cannot be split across several UDP datagrams, certain operations (primarily, those operating on file names and directories) require a minimum datagram size that may be larger than the PMTU. NFS implementations SHOULD NOT reduce the datagram size below this threshold, even if PMTU Discovery suggests a lower value. (In this case datagrams should not be sent with segmentation disabled.)

6.6. Management interface

In RFC 1191, Mogul and Deering suggest that an implementation provide a way for a system utility program to:

- Specify that PMTU Discovery not be done on a given route
- Change the PMTU value associated with a given route

The former can be accomplished by associating a flag with the routing entry; when a packet is sent via a route with this flag set, the IP layer leaves the DF bit clear no matter what the upper layer requests. The same can be provided for CLNP.
PMTU discovery; when a packet is sent via a route with a "supress PMTU discovery" flag set, the CLNP layer leaves the SP flag reset irrespective of upper layer requests. (The implementation should also provide a way to change the timeout period for aging stale PMTU information.)

7. Likely values for Path MTUs

The algorithm recommended in section 5 for "searching" the space of Path MTUs is based on a table of values that severely restricts the search space. In RFC 1191, Mogul and Deering describe a table of MTU values that represented all major data-link technologies in use in the Internet.

In this memo, Table 7-1 has been revised to consider technologies that have been introduced to the Internet since the publication of RFC 1191. The author has also removed technologies that seem unlikely transmission media for CLNP; notably, 1822/ARPANET, ARCNET, SLIP, Experimental Ethernets, and WIDEBAND. Implementors should also make it convenient for customers without source code to update the table values in their systems.

<table>
<thead>
<tr>
<th>Plateau</th>
<th>MTU</th>
<th>Comments</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>65535</td>
<td>65535</td>
<td>Official maximum MTU</td>
<td>RFC 791</td>
</tr>
<tr>
<td></td>
<td>65535</td>
<td>Official maximum NSDU</td>
<td>ISO 8348</td>
</tr>
<tr>
<td></td>
<td>65535</td>
<td>Hyperchannel</td>
<td>RFC 1044</td>
</tr>
<tr>
<td>32000</td>
<td>17914</td>
<td>Just in case</td>
<td>(RFC 1191)</td>
</tr>
<tr>
<td>17914</td>
<td>9180</td>
<td>SMDS</td>
<td>RFC 1209</td>
</tr>
<tr>
<td></td>
<td>9180</td>
<td>ATM over AAL5</td>
<td>RFC iii</td>
</tr>
<tr>
<td>9180</td>
<td>8166</td>
<td>IEEE 802.4</td>
<td>RFC 1042</td>
</tr>
<tr>
<td>8166</td>
<td>4464</td>
<td>IEEE 802.5 (4Mb max)</td>
<td>RFC 1042</td>
</tr>
<tr>
<td></td>
<td>4352</td>
<td>FDDI (Revised)</td>
<td>RFC 1188</td>
</tr>
<tr>
<td>4352</td>
<td>1600</td>
<td>Frame Relay (recommended)</td>
<td>RFC 1490</td>
</tr>
<tr>
<td>1600</td>
<td>1600</td>
<td>X.25 Networks</td>
<td>RFC 1356</td>
</tr>
<tr>
<td>1500</td>
<td>1500</td>
<td>Ethernet Networks</td>
<td>RFC 894</td>
</tr>
<tr>
<td>1492</td>
<td>1492</td>
<td>Point-to-Point (default)</td>
<td>RFC 1548</td>
</tr>
<tr>
<td></td>
<td>512</td>
<td>NETBIOS</td>
<td>RFC 1088</td>
</tr>
<tr>
<td>1492</td>
<td>512</td>
<td>Minimum SNSDU size</td>
<td>ISO 8473</td>
</tr>
<tr>
<td></td>
<td>512</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7-1: CLNP MTUs in the Internet
Table 7-1 lists data links in order of decreasing MTU, and groups them so that each set of similar MTUs is associated with a "plateau" equal to the lowest MTU in the group. As indicated in RFC 1191, the values in the table, especially for higher MTU levels, will not remain valid forever; they are presented here as an implementation suggestion, NOT as a specification or requirement. Implementors should use up-to-date references to pick a set of plateaus. It is important that the table not contain too many entries or the process of searching for a PMTU might waste Internet resources.

7.1. A better way to detect PMTU increases

Rather than detecting increases in the PMTU value by periodically increasing the PMTU estimate to the first-hop MTU, it is possible to periodically increase a PMTU estimate to the lesser of the next-highest value in the plateau table or the first-hop MTU. If the increased estimate is wrong, at most one round-trip time is wasted before the correct value is rediscovered. If the increased estimate is still too low, a higher estimate will be attempted somewhat later.

Because it may take several such periods to discover a significant increase in the PMTU, a short timeout period should be used after the estimate is increased, and a longer timeout be used after the PMTU estimate is decreased because of a Datagram Too Big message. For example, after the PMTU estimate is decreased, the timeout should be set to 10 minutes; once this timer expires and a larger MTU is attempted, the timeout can be set to a much smaller value (say, 2 minutes). In no case should the timeout be shorter than the estimated round-trip time, if this is known.

8. Security considerations

A malicious party could cause problems if it could stop a victim from receiving legitimate Datagram Too Big messages, but in this case there are simpler denial-of-service attacks. Other, more likely forms of denial-of-service attacks against an IP host attempting MTU discovery are based on tampering with the value announced in the ICMP NEXT-HOP-MTU parameter (see also Appendix A).

9. References


Author’s Address

David M. Piscitello
Core Competence, Inc.
1620 Tuckerstown Road
Dresher, PA 19025 USA
dave@corecom.com
Appendix A. NEXT-HOP-MTU parameter for CLNP Error Reports

To support Path MTU Discovery more efficiently, a new parameter is defined for CLNP Error Reports. The "Next-Hop-MTU" parameter has the same semantics as the corresponding parameter in the ICMP header as specified in RFC 1191; i.e., this field shall be used to report the "constricting (next) hop" MTU. As part of its specification, ISO/IEC 8473 MUST indicate that a router MUST include the MTU of the constricting next-hop network in the new parameter in the Error Report header. The format of the parameter is:

```
0          1          2          3
01234567 89012345 67890123 45678901
+--------+--------+--------+--------+
|   Code | Length |   (value of)    |
|11000010|  (4)   |  Next-Hop-MTU   |
+--------+--------+--------+--------+
```

The value of the Next-Hop MTU field is the size in octets of the largest CLNP datagram that could be forwarded, along the path of the original datagram, without being fragmented at this router. The size includes the CLNP header and data, and does not include any lower level headers. This field MUST never contain a value less than 512. When a host receives a Datagram Too Big message, it MUST reduce its estimate of the PMTU for the relevant path, based on the value of the Next-Hop-MTU field in the Error Report.

The specification of this parameter introduces additional security considerations for PMTU Discovery. CLNP Path MTU Discovery mechanism will be vulnerable to the same denial-of-service attacks as IP. Both attacks are based on a malicious party sending false Datagram Too Big messages to a host. The RFC 1191 description of these attacks is repeated here.

In the first attack, the false message indicates a PMTU much smaller than reality. This should not entirely stop data flow, since the victim host should never set its PMTU estimate below the absolute minimum. Since the minimum MTU is 512, this has less impact than with IP but is nonetheless intrusive. In the other attack, the false message indicates a larger PMTU than reality. If believed, this could cause temporary blockage as the victim sends datagrams that will be dropped by some router. The host would discover its mistake within one RTT, by receiving Datagram Too Big messages, but frequent repetition of this attack could cause many discards. A host should never raise its estimate of the PMTU based on a Datagram Too Big message, so should not be vulnerable to this attack.