Matt Mathis
John Heffner
PSC
Kevin Lahey
Freelance
Feb 23, 2003

Packetization Layer Path MTU Discovery
draft-mathis-plpmtud-00.txt

Status of this Memo

Abstract

This document describes a new Packetization Layer MTU probing algorithm which is not subject to the problems associated with the current path MTU discovery algorithms [RFC1191, RFC1981, RFC2923]. The general strategy of the new algorithm is to start with a small MTU and probe upward, testing successively larger MTUs by probing with single packets. If the probe is successfully delivered, then the MTU is raised. If the probe is lost, it is treated as an MTU limitation and not as a congestion signal.

Table of Contents

1. Introduction

Path MTU discovery (PMTUD) as described in RFC1191 and RFC1981

depends on ICMP messages from the network. For a variety of reasons, these messages may not be correctly generated or propagated back to the end host, causing connection failure [RFC2923]. This draft proposes a robust method of performing path MTU discovery from TCP which does not depend on messages from the network. These procedures should be applicable to other transport- or application-level Packetization protocols which implement similar features.

The lower layers need only to be consistent about what packet sizes are acceptable. Media that has parametric limitation (e.g. MTU bounds due to limited clock stability) must include explicit mechanisms to consistently reject packets that might otherwise be nondeterministically delivered.

Classic ICMP-layer PMTUD, when working properly, can speed the discovery of the correct PMTU.

In addition, packetization-layer PMTUD (PLPMTUD) can be extended with heuristics to use other criteria to select PMTU. For example, on a path that is so congested that the fair share window is only 5 KB, TCP may be better behaved with 512-byte packets than with 1500-byte packets since with the larger packets the window would be too small to trigger Fast Retransmit.

PLPMTUD is defined by two independent algorithms. The "probing method" which is will specified in this document, defines the manner in which a candidate MTU may be validated or invalidated. The "probing strategy" describes a suggested approach to choosing MTUs to probe. It is only loosely described here and is subject to future research and improvement.

The general strategy is to start with a small MTU and probe upward, testing successively larger sizes by probing with single packets. If the probe is successfully delivered, then the MTU is raised. If the probe is lost, it is treated as an MTU limitation and not as a congestion signal.

2. TODO list

This Internet-Draft is a partial update of an earlier informally published document. It still needs to be revised to:

- Restructure the document and use the "packetization layer" and "network layer" terminology from RFC1191, to make it less TCP specific.

- Collect all of the TCP specific details into one section. Describe the general principles that apply to all transport pro tocols.

- Fold all of RFC1191, RFC1981 and related lore into this docu ment. (longer term)

- This document should throughly address persistent timeouts. When a TCP or other transport connection suddenly experiences persistent timeouts, several competing recovery strategies might be invoked at each level in the protocol stack, including restarting network interfaces, trying alternate first hop routers, using smaller MTU's, etc. All of these individual strategies need to be tied into a single unified multi-level strategy.

- We need to consider robustness under a number of pathological conditions, such as when there is multi-path routing over paths with different MTUs.

  Please send comments and suggestions to mtu@psc.edu.

3. Context and terminology

This algorithm is built on top of TCP. It's basic design is portable to other protocols, including application protocols over RTP or UDP and SCTP. It is light weight enough where it is not mandatory that MSS information be passed between successive TCP connections to the same remote host. It does not incur excessive overhead for each con nection to discover the maximum MTU on its own.

In TCP it can be inconvenient to compute the largest possible segment size given a particular MTU due the presence of variable length options, such as TCP SACK. MSS probing minimizes this problem by choosing the segment sizes and testing if the link can support trans mission of the resulting IP packet. It is recommended that the test packet is padded with the maximal length variable options.

Note that we use the term Maximal Transmission Unit to mean the largest possible IP packet. e.g. the largest possible layer 2 pay load. Most link layer standards organizations use MTU to mean the largest possible total layer two frame, including the layer two header.

MSS probing can be adapted to other, non-TCP protocols. In

particular, MSS probing can be adapted to tunneling protocols if the tunnel endpoints have a mechanism to detect and report missing pack ets.

4. Probe method

A new "candidate MSS" is tested by sending one "probe segment", which is larger than the current MSS.

Before a probe can be sent the following criteria MUST be met: There connection MUST have at least the candidate MSS worth of pending data. The connection MUST be using the current MSS, as defined by having received at least one acknowledgment for a recent non-probe segment at the current MSS. This implicitly limits successful probes to once per two round trips. [Making the algorithm robust in the presence of multi-path routing is likely to require an additional RTT.]

Failed and inconclusive probes must be more widely spaced than the normal AIMD congestion interval for the current average window size. This is enforced by keeping a "probe count down" which is decremented on each non-probe segment sent. Probes MUST NOT be sent before the probe countdown reaches zero.

After a probe segment has been sent (of size candidate MSS), the sub sequent segment(s) MUST be sent as though the probe segment was not over sized. Thus if the probe segment is lost, it will leave a hole that is exactly one current MSS. We refer to this potential hole as the probe gap. Note that the length of the probe segment is deter mined by the candidate MSS under consideration, but the length of the probe gap is the current MSS. [This has been shown to be more restrictive than necessary.]

The candidate MSS MUST be strictly smaller than three times the cur rent MSS. Thus the probe segment fully covers at most one subsequent segment. The second subsequent segment is at most partially covered by the probe segment. This guarantees that the segments following the probe segment will cause at most one superfluous duplicate acknowledgment.

The TCP MUST be using Fast-Retransmit and SACK or new Reno, such that isolated lost segments will normally be retransmitted without the spurious retransmission of any additional segments.

During the probe, all of the normal retransmission, recovery and con gestion control machinery is in effect except if just the probe gap is retransmitted (and no other segments) the normal multiplicative cwnd reduction is suppressed. If any other segments are

retransmitted, all normal cwnd reductions MUST take place.

The probe is completed when the acknowledgments sequence advances past the probe gap. If the probe gap was not retransmitted the probe was successful. If the probe gap was retransmitted and there were no other retransmissions, the candidate MSS failed. If there were any other retransmissions the probe was inconclusive.

If the probe was successful, the current MSS is updated to the candi date MSS. If cwnd and other congestion state variables are kept in packets, they MUST be rescaled by the change in MSS, to preserve the current window size in bytes.

If the probe failed or was inconclusive the probe count down is set to COUNTDOWN_SCALE times the square of the current window size in packets.

If an RFC1191 style ICMP "Can't" fragment message is received, it is used to compute a MSS limit by deducting the TCP/IP header sizes (including options) from the MTU reported in the ICMP message. If the MSS limit is between the current MSS and candidate MSS, the cur rent MSS is updated from the MSS limit, otherwise the message is ignored. If the current MSS is updated, then the probe strategy is forced into to monitor state described below.

5. Probe strategy

The probe strategy described here is a recommended baseline algo rithm. It is not presented in formal standards language because the probe strategy can include heuristics to help select an optimal MSS for a given path. As a consequence there is opportunity for future improvements to this algorithms.

The probing strategy has three major states: search, monitor and sus pend. During the search state, it sequentially searches for the largest MSS that the path can support. Once the path MSS has been discovered, the probing algorithm enters the monitor state where it probes infrequently to detect if the path MSS has become larger. If the MSS probing persistently fails it may be desirable to suspend path MSS probing and heuristically select one of the common default MSSs: 576, 1280, or 1500 Bytes.

The recommended search strategy is a multi-phase scan: First, a coarse scan for the approximate path MSS using factor of 2 steps starting at 1024 Bytes until a probe fails, followed by successively finer scans between the largest previously successful and unsuccess ful probes.

Table 1: Recommended MSS scanning sequence
(Course scan down column 1, fine scan across each row)
512, [Use only after repeated timeouts]
1024, 1492, 2002
2048
4096, 4352
8192, 9000
16384, 17914
32768
64512
((Additional values needed))

During the scan it is recommended that the MSS not be raised if cwnd is too small as determined by a heuristic. For the time being the recommended heuristic is that the MSS is only raised when the cwnd is larger than 20 segments.

Once the scan has has found an appropriate MSS, the probe strategy enters the monitor state, where it re-probes the most recent failed MTU, once every MONITOR_INTERVAL seconds. If the probe fails, it remains in the monitor state. If it succeeds, it enters the scanning state.

If the network becomes too congested during either the scan or moni tor states it is recommended that the MSS be reduced to smaller size as determined by a heuristic. The recommended heuristic is to reduce the MSS if ssthresh is reduced to 5 segments or smaller. The recom mended reduction is to the next smaller major MSS step in table 1.

When there are repeated timeouts (MAX_TIMO or more retransmissions, w/o any received ACKs), it is presumed that the connection was re- routed onto a link with a smaller MSS, and that ICMP messages are not being delivered. The MSS probing algorithms is reset by pulling back the MSS to 1024 Bytes, rescaling the congestion control variables and reentering the search state.

If there is a timeout and cwnd prior to the timeout was smaller than 6 packets, then the probe strategy can enter the suspended phase and set the MSS to 512 (1280) Bytes. This has the effect of reducing the minimum data rate that TCP can stably manage.

6. Shared state

The common implementations of RFC1191 keep the discovered MTU in a route structure in the IP layer, because that is really the proper place to process ICMP messages. Path MSS discovery can most easily be added to a current pMTUd implementation by keeping most of the state variables for MSS probing in the same route structure.

The following state should be keep in the IP layer per peer address: Most recent successful IP message size (MSS+full TCP/IP header size), most recent failed IP message size, Probe strategy state, indication if there is currently a probe in progress, and the probing TCP con nection, if so.

TCP should keep the following state: indication if currently probing, sequence of the most recent probe gap, TCP/IP header size.

[[Note, we really need to take all of the relevant parts of RFC1191 as well as various lessons learned and fold all of them into one new document]]

7. Probing intervals

COUNTDOWN_SCALE 2 - The scale factor applied to the window squared in packets to compute the the smallest number of non-probe packets required before the next probe.

MONITOR_INTERVAL 600 - The interval in seconds between attempts to probe for larger MSS when in the monitor state.

MAX_TIMO 2 - The number of repeated timeouts needed to trigger

8. Normative references

[RFC1191] Path MTU discovery. J.C. Mogul, S.E. Deering. Nov-01-1990. (Format: TXT=47936 bytes) (Obsoletes RFC1063) (Status: DRAFT STANDARD)

[RFC1435] IESG Advice from Experience with Path MTU Discovery. S. Knowles. March 1993. (Format: TXT=2708 bytes) (Status: INFORMATIONAL)

[RFC1981] Path MTU Discovery for IP version 6. J. McCann, S. Deering, J. Mogul. August 1996. (Format: TXT=34088 bytes) (Status: PROPOSED STANDARD)

[RFC2923] TCP Problems with Path MTU Discovery. K. Lahey. September 2000. (Format: TXT=30976 bytes) (Status: INFORMATIONAL)

9. Informative references

[RFC1063] IP MTU discovery options. J.C. Mogul, C.A. Kent, C. Par tridge, K. McCloghrie. Jul-01-1988. (Format: TXT=27121 bytes) (Obsoleted by RFC1191)

[RFC1626] Default IP MTU for use over ATM AAL5. R. Atkinson. May 1994. (Format: TXT=11841 bytes) (Obsoleted by RFC2225) (Status: PROPOSED STANDARD)

[RFC1791] TCP And UDP Over IPX Networks With Fixed Path MTU. T. Sung. April 1995. (Format: TXT=22347 bytes) (Status: EXPERIMENTAL)

10. Security considerations

Since the MTU reported in the ICMP messages is constrained to be between the old MTU and the candidate MTU, this algorithm is more difficult to attack through fraudulent ICMP messaged.

Furthermore, since this algorithm can function properly without ICMP messages that part of the algorithm can be disabled for additional robustness in hostile environments.

11. IANA considerations

12. Contributors

13. Acknowledgements

Matt Mathis and John Heffner are supported by a grant from Cisco Sys tems, Inc.

14. Authors' addresses

Please send comments and suggestions to mtu@psc.edu.

Matt Mathis and John Heffner
Pittsburgh Supercomputing Center
4400 Fifth Ave.
Pittsburgh, PA 15213
mathis@psc.edu
jheffner@psc.edu

Kevin Lahey
Freelance
kml@patheticgeek.net

15. Intellectual Property

any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

16. Full copyright statement