Network Working Group
Request for Comments: 879

J. Postel
ISI

November 1983

The TCP Maximum Segment Size and Related Topics

This memo discusses the TCP Maximum Segment Size Option and related topics. The purposes is to clarify some aspects of TCP and its interaction with IP. This memo is a clarification to the TCP specification, and contains information that may be considered as "advice to implementers".

1. Introduction

This memo discusses the TCP Maximum Segment Size and its relation to the IP Maximum Datagram Size. TCP is specified in reference [1]. IP is specified in references [2,3].

This discussion is necessary because the current specification of this TCP option is ambiguous.

Much of the difficulty with understanding these sizes and their relationship has been due to the variable size of the IP and TCP headers.

There have been some assumptions made about using other than the default size for datagrams with some unfortunate results.

> HOSTS MUST NOT SEND DATAGRAMS LARGER THAN 576 OCTETS UNLESS THEY HAVE SPECIFIC KNOWLEDGE THAT THE DESTINATION HOST IS PREPARED TO ACCEPT LARGER DATAGRAMS.
>
> This is a long established rule.

To resolve the ambiguity in the TCP Maximum Segment Size option definition the following rule is established:

> THE TCP MAXIMUM SEGMENT SIZE IS THE IP MAXIMUM DATAGRAM SIZE MINUS FORTY.
>
> The default IP Maximum Datagram Size is 576.
> The default TCP Maximum Segment Size is 536.

2. The IP Maximum Datagram Size

Hosts are not required to reassemble infinitely large IP datagrams. The maximum size datagram that all hosts are required to accept or reassemble from fragments is 576 octets. The maximum size reassembly buffer every host must have is 576 octets. Hosts are allowed to accept larger datagrams and assemble fragments into larger datagrams, hosts may have buffers as large as they please.

Hosts must not send datagrams larger than 576 octets unless they have specific knowledge that the destination host is prepared to accept larger datagrams.

3. The TCP Maximum Segment Size Option

TCP provides an option that may be used at the time a connection is established (only) to indicate the maximum size TCP segment that can be accepted on that connection. This Maximum Segment Size (MSS) announcement (often mistakenly called a negotiation) is sent from the data receiver to the data sender and says "I can accept TCP segments up to size X". The size (X) may be larger or smaller than the default. The MSS can be used completely independently in each direction of data flow. The result may be quite different maximum sizes in the two directions.

The MSS counts only data octets in the segment, it does not count the TCP header or the IP header.

A footnote: The MSS value counts only data octets, thus it does not count the TCP SYN and FIN control bits even though SYN and FIN do consume TCP sequence numbers.

4. The Relationship of TCP Segments and IP Datagrams

TCP segment are transmitted as the data in IP datagrams. The correspondence between TCP segments and IP datagrams must be one to one. This is because TCP expects to find exactly one complete TCP segment in each block of data turned over to it by IP, and IP must turn over a block of data for each datagram received (or completely reassembled).

5. Layering and Modularity

TCP is an end to end reliable data stream protocol with error control, flow control, etc. TCP remembers many things about the state of a connection.

IP is a one shot datagram protocol. IP has no memory of the datagrams transmitted. It is not appropriate for IP to keep any information about the maximum datagram size a particular destination host might be capable of accepting.

TCP and IP are distinct layers in the protocol architecture, and are often implemented in distinct program modules.

Some people seem to think that there must be no communication between protocol layers or program modules. There must be communication between layers and modules, but it should be carefully specified and controlled. One problem in understanding the correct view of communication between protocol layers or program modules in general, or between TCP and IP in particular is that the documents on protocols are not very clear about it. This is often because the documents are about the protocol exchanges between machines, not the program architecture within a machine, and the desire to allow many program architectures with different organization of tasks into modules.

6. IP Information Requirements

There is no general requirement that IP keep information on a per host basis.

IP must make a decision about which directly attached network address to send each datagram to. This is simply mapping an IP address into a directly attached network address.

There are two cases to consider: the destination is on the same network, and the destination is on a different network.

Same Network

For some networks the the directly attached network address can be computed from the IP address for destination hosts on the directly attached network.

For other networks the mapping must be done by table look up (however the table is initialized and maintained, for example, [4]).

Different Network

The IP address must be mapped to the directly attached network address of a gateway. For networks with one gateway to the rest of the Internet the host need only determine and remember the gateway address and use it for sending all datagrams to other networks.

For networks with multiple gateways to the rest of the Internet, the host must decide which gateway to use for each datagram sent. It need only check the destination network of the IP address and keep information on which gateway to use for each network.

The IP does, in some cases, keep per host routing information for other hosts on the directly attached network. The IP does, in some cases, keep per network routing information.

A Special Case

There are two ICMP messages that convey information about particular hosts. These are subtypes of the Destination Unreachable and the Redirect ICMP messages. These messages are expected only in very unusual circumstances. To make effective use of these messages the receiving host would have to keep information about the specific hosts reported on. Because these messages are quite rare it is strongly recommended that this be done through an exception mechanism rather than having the IP keep per host tables for all hosts.

7. The Relationship between IP Datagram and TCP Segment Sizes

The relationship between the value of the maximum IP datagram size and the maximum TCP segment size is obscure. The problem is that both the IP header and the TCP header may vary in length. The TCP Maximum Segment Size option (MSS) is defined to specify the maximum number of data octets in a TCP segment exclusive of TCP (or IP) header.

To notify the data sender of the largest TCP segment it is possible to receive the calculation of the MSS value to send is:

```
MSS = MTU - sizeof(TCPHDR) - sizeof(IPHDR)
```

On receipt of the MSS option the calculation of the size of segment that can be sent is:

```
SndMaxSegSiz = MIN((MTU - sizeof(TCPHDR) - sizeof(IPHDR)), MSS)
```

where MSS is the value in the option, and MTU is the Maximum Transmission Unit (or the maximum packet size) allowed on the directly attached network.

This begs the question, though. What value should be used for the "sizeof(TCPHDR)" and for the "sizeof(IPHDR)"?

There are three reasonable positions to take: the conservative, the moderate, and the liberal.

The conservative or pessimistic position assumes the worst -- that both the IP header and the TCP header are maximum size, that is, 60 octets each.

MSS = MTU - 60 - 60 = MTU - 120

If MTU is 576 then MSS = 456

The moderate position assumes the that the IP is maximum size (60 octets) and the TCP header is minimum size (20 octets), because there are no TCP header options currently defined that would normally be sent at the same time as data segments.

MSS = MTU - 60 - 20 = MTU - 80

If MTU is 576 then MSS = 496

The liberal or optimistic position assumes the best -- that both the IP header and the TCP header are minimum size, that is, 20 octets each.

MSS = MTU - 20 - 20 = MTU - 40

If MTU is 576 then MSS = 536

If nothing is said about MSS, the data sender may cram as much as possible into a 576 octet datagram, and if the datagram has minimum headers (which is most likely), the result will be 536 data octets in the TCP segment. The rule relating MSS to the maximum datagram size ought to be consistent with this.

A practical point is raised in favor of the liberal position too. Since the use of minimum IP and TCP headers is very likely in the very large percentage of cases, it seems wasteful to limit the TCP segment data to so much less than could be transmitted at once, especially since it is less that 512 octets.

For comparison: 536/576 is 93% data, 496/576 is 86% data, 456/576 is 79% data.

8. Maximum Packet Size

Each network has some maximum packet size, or maximum transmission unit (MTU). Ultimately there is some limit imposed by the technology, but often the limit is an engineering choice or even an administrative choice. Different installations of the same network product do not have to use the same maximum packet size. Even within one installation not all host must use the same packet size (this way lies madness, though).

Some IP implementers have assumed that all hosts on the directly attached network will be the same or at least run the same implementation. This is a dangerous assumption. It has often developed that after a small homogeneous set of host have become operational additional hosts of different types are introduced into the environment. And it has often developed that it is desired to use a copy of the implementation in a different inhomogeneous environment.

Designers of gateways should be prepared for the fact that successful gateways will be copied and used in other situation and installations. Gateways must be prepared to accept datagrams as large as can be sent in the maximum packets of the directly attached networks. Gateway implementations should be easily configured for installation in different circumstances.

A footnote: The MTUs of some popular networks (note that the actual limit in some installations may be set lower by administrative policy):

ARPANET, MILNET = 1007
Ethernet (10Mb) = 1500
Proteon PRONET = 2046

9. Source Fragmentation

A source host would not normally create datagram fragments. Under normal circumstances datagram fragments only arise when a gateway must send a datagram into a network with a smaller maximum packet size than the datagram. In this case the gateway must fragment the datagram (unless it is marked "don't fragment" in which case it is discarded, with the option of sending an ICMP message to the source reporting the problem).

It might be desirable for the source host to send datagram fragments

if the maximum segment size (default or negotiated) allowed by the data receiver were larger than the maximum packet size allowed by the directly attached network. However, such datagram fragments must not combine to a size larger than allowed by the destination host.

For example, if the receiving TCP announced that it would accept segments up to 5000 octets (in cooperation with the receiving IP) then the sending TCP could give such a large segment to the sending IP provided the sending IP would send it in datagram fragments that fit in the packets of the directly attached network.

There are some conditions where source host fragmentation would be necessary.

If the host is attached to a network with a small packet size (for example 256 octets), and it supports an application defined to send fixed sized messages larger than that packet size (for example TFTP [5]).

If the host receives ICMP Echo messages with data it is required to send an ICMP Echo-Reply message with the same data. If the amount of data in the Echo were larger than the packet size of the directly attached network the following steps might be required: (1) receive the fragments, (2) reassemble the datagram, (3) interpret the Echo, (4) create an Echo-Reply, (5) fragment it, and (6) send the fragments.

10. Gateway Fragmentation

Gateways must be prepared to do fragmentation. It is not an optional feature for a gateway.

Gateways have no information about the size of datagrams destination hosts are prepared to accept. It would be inappropriate for gateways to attempt to keep such information.

Gateways must be prepared to accept the largest datagrams that are allowed on each of the directly attached networks, even if it is larger than 576 octets.

Gateways must be prepared to fragment datagrams to fit into the packets of the next network, even if it smaller than 576 octets.

If a source host thought to take advantage of the local network’s ability to carry larger datagrams but doesn’t have the slightest idea if the destination host can accept larger than default datagrams and expects the gateway to fragment the datagram into default size

fragments, then the source host is misguided. If indeed, the destination host can't accept larger than default datagrams, it probably can't reassemble them either. If the gateway either passes on the large datagram whole or fragments into default size fragments the destination will not accept it. Thus, this mode of behavior by source hosts must be outlawed.

A larger than default datagram can only arrive at a gateway because the source host knows that the destination host can handle such large datagrams (probably because the destination host announced it to the source host in an TCP MSS option). Thus, the gateway should pass on this large datagram in one piece or in the largest fragments that fit into the next network.

An interesting footnote is that even though the gateways may know about know the 576 rule, it is irrelevant to them.

11. Inter-Layer Communication

The Network Driver (ND) or interface should know the Maximum Transmission Unit (MTU) of the directly attached network.

The IP should ask the Network Driver for the Maximum Transmission Unit.

The TCP should ask the IP for the Maximum Datagram Data Size (MDDS). This is the MTU minus the IP header length (MDDS = MTU - IPHdrLen).

When opening a connection TCP can send an MSS option with the value equal MDDS - TCPHdrLen.

TCP should determine the Maximum Segment Data Size (MSDS) from either the default or the received value of the MSS option.

TCP should determine if source fragmentation is possible (by asking the IP) and desirable.

If so TCP may hand to IP segments (including the TCP header) up to MSDS + TCPHdrLen.

If not TCP may hand to IP segments (including the TCP header) up to the lesser of (MSDS + TCPHdrLen) and MDDS.

IP checks the length of data passed to it by TCP. If the length is less than or equal MDDS, IP attached the IP header and hands it to the ND. Otherwise the IP must do source fragmentation.

12. What is the Default MSS ?

Another way of asking this question is "What transmitted value for MSS has exactly the same effect of not transmitting the option at all?".

In terms of the previous section:

The default assumption is that the Maximum Transmission Unit is 576 octets.

MTU = 576

The Maximum Datagram Data Size (MDDS) is the MTU minus the IP header length.

MDDS = MTU - IPHdrLen = 576 - 20 = 556

When opening a connection TCP can send an MSS option with the value equal MDDS - TCPHdrLen.

MSS = MDDS - TCPHdrLen = 556 - 20 = 536

TCP should determine the Maximum Segment Data Size (MSDS) from either the default or the received value of the MSS option.

Default MSS = 536, then MSDS = 536

TCP should determine if source fragmentation is possible and desirable.

If so TCP may hand to IP segments (including the TCP header) up to MSDS + TCPHdrLen (536 + 20 = 556).

If not TCP may hand to IP segments (including the TCP header) up to the lesser of (MSDS + TCPHdrLen (536 + 20 = 556)) and MDDS (556).

13. The Truth

The rule relating the maximum IP datagram size and the maximum TCP segment size is:

> TCP Maximum Segment Size = IP Maximum Datagram Size - 40

The rule must match the default case.

> If the TCP Maximum Segment Size option is not transmitted then the data sender is allowed to send IP datagrams of maximum size (576) with a minimum IP header (20) and a minimum TCP header (20) and thereby be able to stuff 536 octets of data into each TCP segment.

The definition of the MSS option can be stated:

> The maximum number of data octets that may be received by the sender of this TCP option in TCP segments with no TCP header options transmitted in IP datagrams with no IP header options.

14. The Consequences

When TCP is used in a situation when either the IP or TCP headers are not minimum and yet the maximum IP datagram that can be received remains 576 octets then the TCP Maximum Segment Size option must be used to reduce the limit on data octets allowed in a TCP segment.

> For example, if the IP Security option (11 octets) were in use and the IP maximum datagram size remained at 576 octets, then the TCP should send the MSS with a value of 525 (536-11).

15. References

[1] Postel, J., ed., "Transmission Control Protocol - DARPA Internet Program Protocol Specification", RFC 793, USC/Information Sciences Institute, September 1981.

[2] Postel, J., ed., "Internet Protocol - DARPA Internet Program Protocol Specification", RFC 791, USC/Information Sciences Institute, September 1981.

[3] Postel, J., "Internet Control Message Protocol - DARPA Internet Program Protocol Specification", RFC 792, USC/Information Sciences Institute, September 1981.

[4] Plummer, D., "An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48-bit Ethernet Addresses for Transmission on Ethernet Hardware", RFC 826, MIT/LCS, November 1982.

[5] Sollins, K., "The TFTP Protocol (Revision 2)", RFC 783, MIT/LCS, June 1981.