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.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | sales@docketalarm.com | 1-866-77-FASTCASE

fastcase® Smarter legal research.