(51) International Patent Classification: H04L 12/56 (2006.01)


(22) International Filing Date: 13 December 2007 (13.12.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/619,676 4 January 2007 (04.01.2007) US


(71) Applicant (for MG only): IBM UNITED KINGDOM LIMITED [GB/GB]; PO Box 41, North Harbour, Portsmouth Hampshire PO6 3AU (GB).

(72) Inventors: and

(75) Inventors/Applicants (for US only): DAS, Ranadip

(54) Title: PATH MTU DISCOVERY IN NETWORK SYSTEM

(57) Abstract: A method, computer program product, and data processing system for efficiently discovering and storing path MTU information in a sending host are disclosed. In a preferred embodiment, two path MTU tables are maintained. One path MTU table contains MTU values corresponding to the first-hop routers associated with the sending host. The other path MTU table contains MTU values corresponding to individual destination hosts. When the sending host needs to send information to a destination, it first consults the MTU table associated with individual destination hosts. If an entry for that destination host is found in the table, the sending host uses that MTU value. If not, the sending host consults the MTU table for the first-hop router on the path to the destination host and uses that MTU value. If that MTU value is too high, a new entry is made in the host-specific MTU table for the destination host.
Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published: with international search report
PATH MTU DISCOVERY IN NETWORK SYSTEM

Technical Field of the Invention

The present invention relates generally to computer networking.

Background of the Invention

Modern telecommunications depends on networks. A network is a set of interconnected machines (network elements) that allow data to be relayed across the set of machines from a source to a destination. Networks may be classified according to the geographic area they occupy. A local area network (LAN) is usually defined as a network that is physically limited to a relatively small area, such as a building or group of buildings. A wide area network (WAN) is a general term for a network of larger size.

An internetwork, or internet, is a collection of networks interconnected by routers. Routers are network elements that relay (route) data between networks. Most WANs are internets composed of multiple interconnected LANs. Thus, the term WAN is often used to refer to an internet, while the term LAN is often used to denote a constituent network of an internet or WAN. In this document, the terms WAN and LAN are used in this "internetworking" sense, with the caveat that in a significant amount of computing and telecommunications literature the terms LAN and WAN is also used in the previously mentioned "geographical" sense. The "worldwide Internet" or simply "Internet" (uppercase), which provides the backbone for the World Wide Web, is perhaps the best known internet (lowercase), and the protocols and standards defining the Internet define the basic model for most of current networking technology. Thus, in general, technology that applies to the Internet may also find application in other networks, as well.

The Internet is divided into a number of different "autonomous systems" (ASs), each of which contains one or more routers and/or LANs under the control of a single entity, such a university or business. Routers (also sometimes referred to as "gateways" in older literature) are network elements that relay (route) data between networks. Routers are connected to
other routers via physical or sometimes wireless links. Data is routed through an internetwork by being forwarded from router to router over physical links until the proper destination network is reached. To forward information properly over a network, routers maintain "routing tables," which give the router guidance as to which link a given piece of information should be forwarded on. In actuality, both routers and non-router network elements (hosts) maintain routing tables, but routers are distinguished from other network elements by the fact that routers are programmed to forward data, while hosts are generally programmed to discard any data not addressed to that host.

Networking protocols, which define the rules for communication between network elements, are typically designed to work in layers, where each layer performs a slightly different role in data transmission. TCP/IP (Transmission Control Protocol/Internet Protocol) is a collection of protocols (called a protocol suite) that forms the basis for the Internet and many other networks. TCP/IP is typically used to transmit data across a wide area network in the form of relatively small chunks, alternatively known as packets or datagrams. TCP/IP is generally considered to follow a four-layer protocol model. The lowest layer of the TCP/IP protocol suite is referred to as the "Link Layer" and it represents the physical interface for supporting a connection to a physical network media, such as a cable or wireless link. The Network Layer, the next highest layer in the four-layer model, handles the movement of data packets around the network. Above the Network Layer is the Transport Layer, which controls the manner in which network packets are organized and used at the sending and receiving host computers themselves. The top layer of a typical TCP/IP protocol stack is the Application Layer, which represents the functionality for supporting a particular network application, such as E-mail (via Simple Mail Transfer Protocol, or "SMTP") or World Wide Web access (via HyperText Transfer Protocol, or "HTTP").

Internet Protocol (IP) is the primary Network Layer protocol of the TCP/IP protocol suite. There are two main versions of IP currently in use, version 4 (IPv4), which is defined in RFC 791, and version 6 (IPv6), which is defined in RFC 1883. IP allows packets of data to be sent from a numerical source address in the network to a numerical destination address specified in the packet's header. Typically, these packets are "encapsulated" in the packets
of whatever Link Layer protocol is involved. This means that the IP packets are carried as data within the packets generated by a Link Layer protocol, such as Ethernet.

These numerical addresses in the TCP/IP protocol suite are therefore generally referred to as "IP addresses," although the generic, non-IP-specific term is "network addresses." Network addresses are different from hardware addresses, because network addresses are used to identify a network element over an entire WAN (e.g., the Internet), rather than to identify an NA among NAs on the same LAN. Thus, a given network element will have a hardware address corresponding to its NA and one or more network addresses that identify the network element over the WAN. IPv4 supports 32-bit IP addresses, while IPv6 supports 128-bit IP addresses, to accommodate the explosive growth in Internet-connected hosts.

Other network layer protocols, such as Internet Control Message Protocol version 4 (ICMPv4) (RFC 792) and Internet Group Management Protocol (IGMP) (RFC 1112) are used for sending control and error messages and for the grouping of IP addresses for multicasting of individual packets to multiple destinations in the group, respectively. As ICMPv4 was designed to be used in conjunction with IPv4, a new version of the protocol, ICMPv\textbeta\textsuperscript{6} (RFC 1885), is required to be used in conjunction with IPv6.

One of the tasks typically performed by ICMP is known as "Path MTU discovery." The term "MTU" stands for "maximum transport unit," and it refers to the maximum allowable packet size between two nodes in an IP network. When packets are transmitted between any two nodes in an IP network, there is an MTU associated with that link. A typical IP packet will be routed through multiple routers (and, hence, multiple links) on its way to its destination host. Each link has an MTU value associated with it, which is typically a function of the link-layer protocol used for transport over that link (so that each IP packet can be encapsulated into a single link-layer packet, the maximum link-layer packet size will typically determine the MTU for the link). Because each link has an MTU associated with it, if the packet is to reach its intended destination, the packet must not exceed the MTU for any of the links along the path from the source host to the destination host. Hence, the minimum MTU over all of the links in a given path is referred to as the "path MTU" for that
path. The sending host must generate IP packets that are no greater than the path MTU for
the path used to reach the destination host.

The standard method for path MTU discovery is described in RFC 1191. Since a sending
host will generally not be aware of the actual path taken by the packet to reach the
destination host, what is discovered is actually the "path MTU" associated with each
destination host. The path MTU for a given path is initially discovered using what might be
characterized as a trial and error process using ICMP. If a router receives a packet that is
larger than the MTU for the next link in the path, the router discards the packet and returns a
datagram to the sending host containing the ICMP message "datagram too big," which also
contains the MTU value for the next link in the path. The sending host adjusts its packet size
(i.e., its estimate of "path MTU" for the particular destination host) to accommodate the
MTU value returned and tries again. Eventually, the sending host reaches a path MTU value
that works for sending packets to the destination host, and the sending host caches that path
MTU for future use, so that the trial-and-error process does not have to be repeated (for as
long as the path MTU continues to reside in the cache, anyway).

Under this path MTU discovery mechanism, a large, frequently used host (such as a World
Wide Web search engine, for example), must either cache a very large number of path MTU
values or it will suffer significant performance degradation when it must repeatedly calculate
path MTU values for reoccurring destination hosts. Under either of these scenarios,
estensive computing resources are required (be they network bandwidth, computing time, or
storage cost).

What is needed, therefore, is a more efficient way for a frequently used network host to keep
track of the correct path MTU to use for particular destination hosts. The present invention
addresses this and/or other problems, and offers advantages over previous solutions.

Summary of the Invention

Accordingly, the present invention provides a method, computer program product, and data
processing system for efficiently discovering and storing path MTU information in a sending
host. In a preferred embodiment, two path MTU tables are maintained. One path MTU table contains MTU values corresponding to the first-hop routers associated with the sending host. The other path MTU table contains MTU values corresponding to individual destination hosts for which the path MTU is lower than the MTU for the first-hop router used to reach that destination host. When the sending host needs to send information to a destination, it first consults the MTU table associated with individual destination hosts. If an entry for that destination host is found in the table, the sending host uses that MTU value. If not, the sending host consults the MTU table for the first-hop router on the path to the destination host and uses that MTU value. If that MTU value is too high to allow a packet to reach the intended destination, a new entry is made in the host-specific MTU table for the destination host.

**Brief Description of the Drawings**

Embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

**Figure 1** is a diagram of a portion of a network in which a preferred embodiment of the present invention may be implemented;

**Figure 2** is a diagram of a dual MTU table in accordance with a preferred embodiment of the present invention;

**Figure 3** is a flowchart representation of a process of MTU information discovery and storage in accordance with a preferred embodiment of the present invention; and

**Figure 4** is a block diagram of a data processing system in which a preferred embodiment of the present invention may be implemented.

**Detailed Description of Example Embodiments**
**Figure 1** is a diagram of a portion 100 of a network in which a preferred embodiment of the present invention may be implemented. Network portion 100 is viewed from the perspective of a sending (source) host 102, which can data to any of a plurality of destination hosts 116, 118, and 120 through interconnected routers 104, 106, 108, 110, 112, and 114. One skilled in the art will recognize, however, that the terms "sending host" and "destination host" are understood to be relative to the roles played by two networked computers in a single data transmission. Any computer in a network that is sending data to another networked computer is a sending host for that data transmission; conversely, any computer in a network that is receiving data from another networked computer is a destination host for that data transmission.

As shown in **Figure 1**, each connection between two nodes in network portion 100 has an associated MTU value. For example, the MTU for the link between host 102 and router 104 is 1500. For each combination of sending and destination hosts, there is a "path MTU," which represents the largest packet size allowable on the network path taken from the sending host to the destination host. The "path MTU" is the minimum MTU over all connections on the network path from the sending host to the destination host. For example, the path MTU between sending host 102 and destination host 116 is 1500, while the path MTU between sending host 102 and destination host 118 is 1200 (because the minimum MTU on the path from sending host 102 to destination host 118 is the MTU of 1200, which occurs between router 108 and router 112).

From **Figure 1** it is apparent that in any case, the path MTU from a sending host to a destination host is no greater than the MTU on the connection between the sending host and the "first-hop router" on the path to the destination host. The "first-hop router" is the first router encountered on the path between the sending host and destination host. For any given sending host, there is a finite number of first-hop routers (because there can only be a finite number of physical connections from a given sending host). For example, in **Figure 1**, sending host 102 has two first-hop routers, router 104 and router 106. First-hop router 104 is the first-hop router on the paths to destination hosts 116 and 118, while first-hop router 106 is the first-hop router on the path to destination host 120. A preferred embodiment of the present invention reduces the amount of storage needed to store path MTU information in a
sending host by taking advantage of the fact that the MTU to a given first-hop router is an upper bound on path MTU.

Figure 2 is a diagram of a dual MTU table in accordance with a preferred embodiment of the present invention. In this preferred embodiment, a sending host maintains two path MTU tables (path MTU tables 200 and 202). Path MTU table 200 contains MTU values corresponding to the first-hop routers associated with the sending host. Path MTU table 202 contains MTU values corresponding to individual destination hosts for which the path MTU is lower than the MTU for the first-hop router used to reach that destination host. When the sending host needs to send information to a destination, it first consults MTU table 202. If an entry for that destination host is found in MTU table 202, the sending host uses that MTU value. If not, the sending host consults MTU table 200 to locate the MTU value associated with the first-hop router on the path to the destination host and uses that MTU value. If the first-hop router's MTU value turns out to be too high to allow a packet to reach the intended destination, a new entry is made in table 202 for the destination host with the actual path MTU value for that host.

For example, if no entry in table 202 exists for destination host 120 in Figure 1, sending host 102 will first attempt to send a packet to destination host 120 using an MTU of 1200 for first-hop router 106, which sending host 102 retrieves from table 200. Since the MTU for the link between router 106 and router 114 is 800, the first-hop router MTU of 1200 is too high for transmitting packets to destination host 120 and router 106 will return an ICMP message to sending host 102 to inform sending host 102 that it must send packets that do not exceed the MTU of 800 between router 106 and router 114. In this case, sending host 102 will make a new entry in table 202 for destination host 120 containing the adjusted MTU of 800. This technique ensures that host-specific MTU information is only stored when absolutely necessary by using the first-hop router's MTU whenever possible.

Figure 3 is a flowchart representation of a process of sending a single packet with MTU information discovery and storage in accordance with a preferred embodiment of the present invention. When a sending host has data to send to a destination host (block 300), a determination is first made as to whether the destination host has an entry in the host-specific
MTU table (e.g., table 202 in Figure 2) (block 302). If there is no host-specific entry for the destination host (block 302:No), then the first-hop router on the path to destination host is determined (block 304). An attempt is then made to transmit a packet of data using a packet size set by the MTU associated with this first-hop router (e.g., as determined from table 200 in Figure 2) (block 306). If it is discovered that this first-hop router MTU is larger than the actual path MTU to the destination host (e.g., because an "packet size exceeded" ICMP message has been received) (block 308:Yes), then an entry storing an adjusted host-specific MTU (e.g., in table 202 in Figure 2) is generated for the destination host (block 310). If no such "packet size exceeded" message is received (Le, the packet sent was not too big) (block 308:No), the process of sending a single packet is complete: the process is repeated for the sending of subsequent packets, as necessary.

If there is a host-specific MTU value in the host MTU table, either because it was determined to exist *apriori* (block 302:Yes) or because it was just generated (block 310), a packet is transmitted with a packet size determined by this host-specific MTU value (block 312). If this packet (having a host-specific size) turns out to be too big to reach the destination (block 314:Yes), then the host-specific value stored in the host MTU table is adjusted (e.g., in accordance with the "packet size exceeded" ICMP message received) to overcome the failure (block 316) and an attempt to transmit a packet of this new, smaller size is made (block 312). This process repeats until a packet of the proper size has been transmitted and no more packet-size-related failures occur (block 314:No).

One skilled in the art will recognize that blocks 312, 314, and 316, in isolation, are equivalent to the MTU discovery mechanism described in the Internet standards document RFC-1 191, which is incorporated herein by reference. Hence, one manner of implementing the present invention would be to augment/instrument existing MTU discovery code (implementing blocks 312, 314, and 316 as a "black box") with the functionality provided by blocks 300-310.

Figure 4 illustrates information handling system 401 which is a simplified example of a computer system/data processing system capable of performing the computing operations described herein with respect to a preferred embodiment of the present invention. Computer
system 401 includes processor 400 which is coupled to host bus 402. A level two (L2) cache memory 404 is also coupled to host bus 402. Host-to-PCI bridge 406 is coupled to main memory 408, includes cache memory and main memory control functions, and provides bus control to handle transfers among PCI bus 410, processor 400, L2 cache 404, main memory 408, and host bus 402. Main memory 408 is coupled to Host-to-PCI bridge 406 as well as host bus 402. Devices used solely by host processor(s) 400, such as LAN card 430, are coupled to PCI bus 410. Service Processor Interface and ISA Access Pass-through 412 provides an interface between PCI bus 410 and PCI bus 414. In this manner, PCI bus 414 is insulated from PCI bus 410. Devices, such as flash memory 418, are coupled to PCI bus 414. In one implementation, flash memory 418 includes BIOS code that incorporates the necessary processor executable code for a variety of low-level system functions and system boot functions.

PCI bus 414 provides an interface for a variety of devices that are shared by host processor(s) 400 and Service Processor 416 including, for example, flash memory 418. PCI-to-ISA bridge 435 provides bus control to handle transfers between PCI bus 414 and ISA bus 440, universal serial bus (USB) functionality 445, power management functionality 455, and can include other functional elements not shown, such as a real-time clock (RTC), DMA control, interrupt support, and system management bus support. Nonvolatile RAM 420 is attached to ISA Bus 440. Service Processor 416 includes JTAG and I2C buses 422 for communication with processor(s) 400 during initialization steps. JTAG/I2C buses 422 are also coupled to L2 cache 404, Host-to-PCI bridge 406, and main memory 408 providing a communications path between the processor, the Service Processor, the L2 cache, the Host-to-PCI bridge, and the main memory. Service Processor 416 also has access to system power resources for powering down information handling device 401.

Peripheral devices and input/output (I/O) devices can be attached to various interfaces (e.g., parallel interface 462, serial interface 464, keyboard interface 468, and mouse interface 470 coupled to ISA bus 440. Alternatively, many I/O devices can be accommodated by a super I/O controller (not shown) attached to ISA bus 440.
In order to attach computer system 401 to another computer system to copy files over a network, LAN card 430 is coupled to PCI bus 410. Similarly, to connect computer system 401 to an ISP to connect to the Internet using a telephone line connection, modem 475 is connected to serial port 464 and PCI-to-ISA Bridge 435.

While the computer system described in Figure 4 is capable of executing the processes described herein, this computer system is simply one example of a computer system. Those skilled in the art will appreciate that many other computer system designs are capable of performing the processes described herein.

One of the preferred implementations of the invention is a client application, namely, a set of instructions (program code) or other functional descriptive material in a code module that may, for example, be resident in the random access memory of the computer. Until required by the computer, the set of instructions may be stored in another computer memory, for example, in a hard disk drive, or in a removable memory such as an optical disk (for eventual use in a CD ROM) or floppy disk (for eventual use in a floppy disk drive), or downloaded via the Internet or other computer network. Thus, the present invention may be implemented as a computer program product for use in a computer. In addition, although the various methods described are conveniently implemented in a general purpose computer selectively activated or reconfigured by software, one of ordinary skill in the art would also recognize that such methods may be carried out in hardware, in firmware, or in more specialized apparatus constructed to perform the required method steps. Functional descriptive material is information that imparts functionality to a machine. Functional descriptive material includes, but is not limited to, computer programs, instructions, rules, facts, definitions of computable functions, objects, and data structures.

While particular embodiments of the present invention have been shown and described, it will be obvious to those skilled in the art that, based upon the teachings herein, changes and modifications may be made without departing from this invention and its broader aspects. Therefore, the appended claims are to encompass within their scope all such changes and modifications as are within the true spirit and scope of this invention. Furthermore, it is to be understood that the invention is solely defined by the appended claims. It will be
understood by those with skill in the art that if a specific number of an introduced claim element is intended, such intent will be explicitly recited in the claim, and in the absence of such recitation no such limitation is present. For non-limiting example, as an aid to understanding, the following appended claims contain usage of the introductory phrases "at least one" and "one or more" to introduce claim elements. However, the use of such phrases should not be construed to imply that the introduction of a claim element by the indefinite articles "a" or "an" limits any particular claim containing such introduced claim element to inventions containing only one such element, even when the same claim includes the introductory phrases "one or more" or "at least one" and indefinite articles such as "a" or "an;" the same holds true for the use in the claims of definite articles. Where the word "or" is used in the claims, it is used in an inclusive sense (i.e., "A and/or B," as opposed to "either A or B").
CLAIMS

1. A method comprising:
   storing a set of maximum transmission unit (MTU) values associated with one or more first-hop routers;
   identifying a first-hop router on a network path to a destination host;
   determining a first-hop router maximum transmission unit value corresponding to the identified first-hop router; and
   attempting to transmit a data packet to the destination host through the network path, wherein a size of the data packet is defined by said first-hop router maximum transmission unit value.

2. A method comprising:
   storing a first set of maximum transmission unit (MTU) values correlated with one or more first-hop routers;
   storing a second set of maximum transmission unit values correlated with one or more hosts;
   determining whether a particular destination host is one of said one or more hosts; in response to a determination that the particular destination host is one of said one or more hosts, retrieving a value from said second set for use as a path maximum transmission unit value;
   in response to a determination that the particular destination host is not one of said one or more hosts, retrieving a value from said first set for use as the path maximum transmission unit value; and
   transmitting a data packet having a size that is determined by the path maximum transmission unit value.

3. The method of claim 1 or claim 2, further comprising:
   determining whether an actual required path maximum transmission unit value for the network path is smaller than said determined maximum transmission unit value;
   in response to a determination that the actual required path maximum transmission unit value for the network path is smaller than said determined maximum transmission unit value
value, storing said actual required path maximum transmission unit value as being associated with the destination host.

4. The method of claim 3, further comprising:
   determining if the actual required path maximum transmission unit value was stored as being associated with the destination host; and
   in response to a determination that the actual required path maximum transmission unit value was stored as being associated with the destination host, transmitting a second data packet to the destination host, wherein the size of the data packet is defined by the stored actual required path maximum transmission unit value.

5. The method of claim 4, further comprising:
   in response to a determination that the actual required path maximum transmission unit value was not stored as being associated with the destination host, transmitting a second data packet to the destination host, wherein the size of the data packet is defined by said first-hop router maximum transmission unit value.

6. The method of claim 2 or claim 3, further comprising:
   detecting a failure of the data packet to reach the destination host because of excessive packet size;
   in response to detecting that the data packet failed to reach the destination host because of excessive packet size,
   adjusting the size of the data packet to overcome the failure; and
   attempting to retransmit the adjusted data packet to the destination host.

7. The method of claim 6, further comprising:
   determining the actual required path maximum transmission unit value for the network path from the size of the adjusted data packet.

8. A computer program product comprising functional descriptive material that, when executed by a computer, causes the computer to perform actions of any preceding claim.
9. A data processing system comprising:
   at least one processor;
   data storage accessible to the at least one processor; and
   a set of instructions in the data storage, wherein the at least one processor is operable
to execute the set of instructions to perform the actions of any of claims 1 to 7.
<table>
<thead>
<tr>
<th>1st Hop Router</th>
<th>MTU</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.54.29.1</td>
<td>1500</td>
</tr>
<tr>
<td>128.54.29.2</td>
<td>600</td>
</tr>
<tr>
<td>128.54.29.3</td>
<td>900</td>
</tr>
<tr>
<td>128.54.29.4</td>
<td>1200</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Destination Host</th>
<th>MTU</th>
</tr>
</thead>
<tbody>
<tr>
<td>196.2.78.4</td>
<td>200</td>
</tr>
<tr>
<td>24.1.7.8</td>
<td>600</td>
</tr>
<tr>
<td>154.65.3.20</td>
<td>800</td>
</tr>
<tr>
<td>121.54.46.75</td>
<td>1200</td>
</tr>
<tr>
<td>96.76.42.82</td>
<td>750</td>
</tr>
<tr>
<td>203.84.191.68</td>
<td>256</td>
</tr>
<tr>
<td>73.9.5.2</td>
<td>768</td>
</tr>
<tr>
<td>192.5.34.9</td>
<td>1400</td>
</tr>
</tbody>
</table>

Fig. 2
Figure 4
**INTERNATIONAL SEARCH REPORT**

**PCT/EP2007/063891**

**A. CLASSIFICATION OF SUBJECT MATTER**

INV. H04L12/56

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>MOGUL DECRWL S DEERING STANFORD UNIVERSITY</td>
<td>1-9</td>
</tr>
<tr>
<td></td>
<td>J: &quot;Path MTU Discovery; rfc1191.txt&quot; IETF STANDARD, INTERNET ENGINEERING TASK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FORCE, IETF, CH, November 1990 (1990-11), XP015006135</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ISSN: 0000-0003</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cited in the application</td>
<td></td>
</tr>
<tr>
<td></td>
<td>page 2, line 31 - line 37</td>
<td></td>
</tr>
<tr>
<td></td>
<td>page 3, line 21 - line 23</td>
<td></td>
</tr>
<tr>
<td></td>
<td>page 4, lines 19-21,28,29</td>
<td></td>
</tr>
<tr>
<td></td>
<td>page 10, lines 10-12,21-24,27-32,37-45</td>
<td></td>
</tr>
<tr>
<td></td>
<td>page 11, line 10 - line 13</td>
<td></td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents:
  * A: document defining the general state of the art which is not considered to be of particular relevance
  * E: earlier document but published on or after the international filing date
  * L: document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
  * O: document referring to an oral disclosure, use, exhibition or other means
  * P: document published prior to the international filing date but later than the priority date claimed

* T: later document published after the international filing date that cited to understand the principle or theory underlying the invention

* X: document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

* Y: document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

* S: document member of the same patent family

Date of the actual completion of the international search: 7 March 2008

Date of mailing of the international search report: 14/04/2008

Name and mailing address of the ISA/European Patent Office, P.B. 5818 Patentlaan 2 NL-2380 HV Rijswijk Tel. (+31-70) 340-2030, Tx 31 651 epo nl. Fax: (+31-70) 340-3016

Authorized officer: Engmann, Steffen
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>US 5 809 254 A (MATSUZONO MASAYA [JP]) 15 September 1998 (1998-09-15) abstract column 1, line 53 - line 57 column 1, line 66 - column 2, line 7 column 2, line 56 - column 3, line 5 column 4, line 11 - line 27 column 4, line 31 - line 34 column 5, line 3 - line 10 column 5, line 46 - column 6, line 15 figures 4,6,7 claims 1-5</td>
<td>1-9</td>
</tr>
</tbody>
</table>

Form PCT/ISA/210 (continuation of second sheet) (April 2005)
<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td>US 5809254 A</td>
<td>15-09-1998</td>
<td>NONE</td>
<td></td>
</tr>
</tbody>
</table>