

ICS 33.040.40

M 32

YD

中华人民共和国通信行业标准

YD/T 2169-2010

IPv6 技术要求 ——IPv6 路径最大传输单元发现协议

Technique requirement of IPv6: Path MTU discovery for IPv6

(IETF RFC 1981: 1996, Path MTU Discovery for IPv6, NEQ)

2010-12-29 发布

2011-01-01 实施

中华人民共和国工业和信息化部 发布

目 次

前 言..... II

1 范围.....1

2 规范性引用文件.....1

3 术语、定义和缩略语.....1

 3.1 术语和定义.....1

 3.2 缩略语.....1

4 概述.....2

5 协议的要求.....3

6 协议实现.....3

 6.1 分层.....3

 6.2 存储 PMTU 信息.....4

 6.3 清除过期的 PMTU 信息.....5

 6.4 TCP 层的行为.....5

 6.5 其他传输协议的问题.....6

 6.6 管理接口.....7

7 安全问题.....7

附录 A（资料性附录） IETF RFC1981 与 IETF RFC1191 的对比.....8

前 言

本标准是“IPv6 协议”系列标准之一。该系列标准预计的结构及名称如下：

1. YD/T 1341-2005 IPv6 基本协议——IPv6 协议 (IETF RFC2460:1998, MOD)
2. YD/T 1915-2009 IPv6 技术要求——移动 IPv6 快速切换
3. YD/T 2168-2010 IPv6 技术要求——IPv6 反向邻居发现协议 (IETF RFC3122:2001, NEQ)
4. YD/T 2169-2010 IPv6 技术要求——IPv6 路径最大传输单元发现协议 (IETF RFC1981:1996, NEQ)
5. YD/T 2170-2010 IPv6 技术要求——IPv6 路由器重编号协议 (IETF RFC2984:2000, NEQ)
6. IPv6 技术要求——IPv6 动态主机配置协议
7. IPv6 技术要求——支持计算机移动部分
8. YD/T 1442-2006 IPv6 网络技术要求——地址、过渡及服务质量
9. YD/T 1343-2005 IPv6 邻居发现协议——基于 IPv6 的邻居发现协议 (IETF RFC2461:1998, MOD)
10. IPv6 邻居发现安全性技术要求
11. YD/T 1344-2005 IPv6 地址结构协议——IPv6 无状态地址自动配置
12. YD/T 1612-2007 IPv4 网络向 IPv6 网络过渡中的互联互通技术要求
13. YD/T 2029-2009 基于软线技术的互联网 IPv6 过渡技术框架
14. YD/T 1635-2007 IPv6 网络技术要求——面向网络地址翻译 (NAT) 用户的 IPv6 隧道技术
15. YD/T 1656-2007 采用边界网关协议多协议扩展 (BGP-MP) 的基于 IPv6 骨干网的 IPv4 网络互联 (4 over 6) 技术要求

本标准对应于 IETF RFC1981 (1996 年英文版)《IPv6 路径 MTU 发现协议》。本标准与 IETF RFC1981 (1996 年英文版)的一致性程度为非等效。主要差异如下：

——根据 GB/T1 系列要求和汉语习惯进行了结构和格式的修改；

——本标准的第 4 章、第 5 章、第 6 章、第 7 章、附录 A 分别与 IETF RFC1981 (1996) 的第 3 章、第 4 章、第 5 章、第 6 章、附录 A 保持一致。

本标准的附录 A 为资料性附录。

本标准由中国通信标准化协会提出并归口。

本标准起草单位：工业和信息化部电信研究院、华为技术有限公司、杭州华三通信技术有限公司。

本标准主要起草人：毕立波、卜哲、陈国义、万晓兰。

IPv6 技术要求——IPv6 路径最大传输单元发现协议

1 范围

本标准规定了用于 IPv6 的路径最大传输单元 (PMTU) 发现协议, 主要包括协议概述、协议的要求以及协议实现过程等方面的内容。

本标准适用于支持 IPv6 协议的网络设备。

2 规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件, 其随后所有的修改单 (不包括勘误的内容) 或修订版均不适用于本标准。然而, 鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件, 其最新版本适用于本标准。

YD/T 1341-2005 IPv6 基本协议——IPv6 协议 (IETF RFC2460:1998, MOD)

YD/T 1343-2005 IPv6 邻居发现协议——基于 IPv6 的邻居发现协议 (IETF RFC2461:1998, MOD)

IETF RFC905 (1984) ISO 传输协议说明 ISO DP 8073

3 术语、定义和缩略语

3.1 术语和定义

下列术语和定义适用于本标准。

3.1.1

链路 MTU link MTU

最大的传输单元, 也就是, 可以在链路上传递的无需分片的最大包字节尺寸。

3.1.2

路径 Path

在一个源节点以及一个目的节点之间, 一个包穿越的链路组。

3.1.3

路径 MTU PMTU

在一个源节点和一个目的节点之间, 一条路径中所有链路的最小链路 MTU。

3.1.4

路径 MTU 发现 Path MTU Discovery

一个节点用来学习一条路径 PMTU 的过程。

3.1.5

流 Id Flow Id

源地址和一个非零流标签的组合。

3.2 缩略语

下列缩略语适用于本标准。

DF	Don't Fragment	不分片
DOS	Denial of Service	拒绝服务
FDDI	Fiber Distributed Data Interface	光纤分布式数据接口
ICMP	Internet Control Message Protocol	Internet 控制消息协议
IPv6	Internet Protocol Version 6	互联网协议第 6 版本
MMS_S	Maximum Send Transport-Message Size	最大能发送的传送消息大小
MTU	Maximum Transmission Unit	最大传输单元
NFS	Sun Network File System	Sun 网络文件系统
PMTU	Path Maximum Transmission Unit	路径最大传输单元
ROM	Read-Only Memory	只读内存
RPC	Remote Procedure Call	远程过程调用
TCP	Transmission Control Protocol	传输控制协议
UDP	User Datagram Protocol	用户数据报协议

4 概述

本标准规定了一个动态发现 PMTU 的协议，具体方法是：一个源节点最初假定一条路径的 PMTU 是这条路径上已知的第一跳的 MTU，如果在这条路径上传递的一个分组的长度大于其中某条链路的 MTU，那么与这条链路有关的节点就无法转发这个分组，同时这个节点会丢弃这些分组并且返回 PacketTooBig 类型的 ICMPv6 消息，接收到这个 PacketTooBig 消息后，源节点会根据消息通告的瓶颈链路的 MTU，降低事先假定的整条路径 PMTU 数值。

当源节点估计 PMTU 小于或者等于实际 PMTU 的时候，结束 PMTU 发现过程。注意在结束路径 MTU 发现程序之前，会反复出现调整发送分组 PMTU、接收到 PacketTooBig 消息类型情况，因为整条路径中有可能存在离源节点距离更远并且具有更小 MTU 值的瓶颈链路。

节点能够可选通过终止发送大于 IPv6 最小链路 MTU 的分组方式来结束发现过程。

若路由拓扑发生变化，一条路径的 PMTU 可能会发生变化。PacketTooBig 消息会检测到 PMTU 的减小。为了检测到一条路径 PMTU 的增加，一个源节点会周期性地增大目前的 PMTU 数值。通常这样会导致分组被丢弃并且产生 PacketTooBig 消息报文，因为在大部分情况下，路径的 PMTU 都不会变化。因此，不能频繁检测一条路径 PMTU 的增加。

PMTU 发现同样可应用在组播环境中。在组播地址情况下，一个分组的副本可能会通过不同的路径到达不同的节点。每条路径都可能具有一个不同的 PMTU，而且一个组播分组可能会导致接收到网络中不同路径的多个节点产生的 PacketTooBig 消息类型报文，而且每个 PacketTooBig 消息都可能会报告不同大小的 MTU 值，其中最小的 MTU 值就是组播 PMTU 值。

需要注意的是动态发现路径 PMTU 的使用要求目的节点与中间节点必须具有相同方式的连接。在某些情况下，如一个邻居节点为多个目的节点代理时，表面上目的节点是以直连的方式出现，但实际上并不是这样，到目的节点的距离可能会大于一跳。

IPv6 节点应执行 PMTU 发现，从而发现并且利用具有高于 IPv6 最小链路 MTU 的 PMTU 的路径。但一个最简单功能的 IPv6 实现（例如在启动 ROM 中）可以不执行 PMTU 发现。不执行 PMTU 发现的 IPv6

节点可以利用在 YD/T 1341-2005 中定义的 IPv6 最小链路 MTU 作为 IPv6 报文的最大长度。在大多数情况下，没有必要采用这种方式，因为大部分路径都具有一个大于 IPv6 最小链路 MTU 的 PMTU。如果不支持 PMTU 发现而采用将 IPv6 最小链路 MTU 作为 IPv6 报文的最大长度的方法，不仅浪费网络资源而且可能无法达到最优的吞吐量。

5 协议的要求

本标准不要求 IPv6 节点必须支持 PMTU 发现协议，本章所规定的协议只适用那些支持本协议的 IPv6 节点。

当一个节点接收到一个 PacketTooBig 消息报文时，它必须根据该消息通告的瓶颈链路 MTU 值来降低相关路径 PMTU 值。本标准不规定在这种情况下节点的详细行为，因为不同的应用可能有不同的需求，而且不同的实现结构的侧重点不同。

在接收到 PacketTooBig ICMPv6 消息后，一个节点必须试图避免在近一段时间内引发更多的这个消息。节点必须降低它在路径上发送的分组的长度。采用一个可能大于 IPv6 最小链路 MTU 的 PMTU 也可能会持续引发 PacketTooBig 消息。由于被丢弃的超出瓶颈链路 MTU 的报文以及由此产生的 ICMPv6 消息报文会消耗网络的资源，因此节点必须强制结束 PMTU 发现程序。

采用 PMTU 发现的节点必须尽可能快地检测到 PMTU 的减少。节点可以检测到 PMTU 的增加，但是会要求节点发送大于当前估值的 PMTU，同时 PMTU 可能没有增加，因此，不能在较短的间隔内频繁地检测路径 PMTU 的增加。在已经接收到某条路径上的 PacketTooBig 消息情况下，至少 5min 后，节点才能去检测一个可能增加的 PMTU 值，该计时器的建议设置为 10min，即是最小间隔 5min 的两倍。

一个节点不能把相关 PMTU 估值降低到 IPv6 最低链路 MTU 以下。

需要注意的是：一个节点可能会接收到一个 PacketTooBig 消息，通告一个低于 IPv6 最小链路 MTU 的下一跳 MTU。在这种情况下，不要求这个节点把发送到该路径上的后续分组长度降低到所通告的 MTU 上，但是必须在这些分组中包括一个 Fragment header 字段，该字段在 YD/T 1341-2005 中规定。

一个节点一定不能根据接收到的 PacketTooBig 消息而去加大 PMTU 的数值。通告瓶颈链路 MTU 数值大于目前路径 PMTU 值的 PacketTooBig 消息，可能是网络中无序传输的失效报文，也可能是 DOS 攻击的伪造报文，还可能是到目的节点存在多条路径，每条路径的 PMTU 不同造成的。

注：本标准与 IETF RFC1981 的一致性程度为非等效，而 IETF RFC1981 主要来自于 IETF RFC1191，两者之间的区别参见附录 A。

6 协议实现

本章规定的是 PMTU 发现协议实现时所涉及到的一些细节，内容包括：

- a) 涉及到执行 PMTU 发现的协议层；
- b) 存储 PMTU 信息的方法；
- c) 删除失效的 PMTU 信息的方法；
- d) 传输或面向应用的协议层（在 TCP/IP 分层模型中传输层以上的协议层即应用层，在 OSI/RM 模型中传输层以上的协议层，包括会话层、表示层、应用层）的行为。

6.1 分层

在 IP 结构中，选择发送什么长度的分组由 IP 协议层之上的协议来完成。本标准把这样的协议称作是“分组协议”。分组协议通常是传输协议（例如，TCP），也可以是面向应用的协议层协议。

在分组协议上执行 PMTU 发现简化了一些涉及到跨层的问题，但是有几个缺陷：需要在不同的每个分组协议上重复实现；不同的分组协议之间共享 PMTU 信息较为困难；一些分组协议维护的面向连接的状态可能不易扩展从而来较长时间保存 PMTU 信息。

因此要求 IP 层存储 PMTU 信息，ICMP 层处理接收到的 PacketTooBig 通告消息。基于分组的协议层可以根据变化后的 PMTU 值来改变该层协议报文的长度。为了支持这种功能，基于分组的协议层需要通过一种方式来学习可能变化的 MMS_S 数值。MMS_S 值可以是 PMTU 值减去 IPv6 协议报文长度以及可能存在的 IP 层预留字段的长度。

一个分组协议如内核外 UDP 应用，可能无法改变它发送消息的长度，这可能导致分组长度大于 PMTU。为了解决这个问题，IPv6 规定了一个机制，该机制允许把较大的净荷进行分片，每个分片都通过一个分组来发送。YD/T 1341-2005 中对此机制进行了规定。尽管如此，在实现中应该尽可能避免分组协议发送需要分片的消息。

6.2 存储 PMTU 信息

理想情况下，一个 PMTU 值应该和一条特定的、在源和目的节点之间进行分组交换的路径相关联。但是，在大部分情况下，一个节点可能不具备足够的信息来完全准确地标识路径。因此，一个节点必须把一个 PMTU 值和一条路径的某些本地标识特征相关联。在节点实现时就需要根据情况确定一种基于本地特征的路径标识方法。

在组播目的地址的环境下，一个分组的副本可能会穿越不同的路径到达多个不同的节点。因此要求需要确定的基于本地特征的路径标识方法应该能够表示一个元素数量巨大的路径集合。

最低限度：一个实现本协议的节点能够为与该节点相关的所有路径仅仅维护一个 PMTU。这个 PMTU 就是 PMTU 集合中最小的 PMTU。这种方式可能导致所采用的 PMTU 值小于多数路径的 PMTU 值。

一种实现方式是把目的地址作为一条路径的本地表示法。一个目的节点相关的 PMTU 值可能是在到达那个目的节点所有可能路径上的最小 PMTU 值。可能的路径数量一般较少，大部分情况下只有一条。这中基于目的节点的方法可以针对每个目的节点采用最优长度的分组。这种方式和在 YD/T 1343-2005 中定义的一个主机的概念模型可以很好地结合：一个 PMTU 值与对应的出口存储在目的缓存中。

如果采用 YD/T 1341-2005 中规定的流的概念，一种实现方法是采用流 ID 作为一条路径的本地表示法。发送到特定目的地但是属于不同流的分组可以采用不同的路径，根据流 ID 来选择路径。用流 ID 来标识路径比用目的地址来标识路径更为精确。

对于源路由的分组（包括一个 YD/T 1341-2005 定义的 IPv6 路由头的分组），源路由信息可以被用来进一步标识一条路径，因此一种实现方法就是利用源路由信息作为本地表示法。

需要注意的是，有些路径可能通过不同的安全类别来进行进一步的区分。这些类别的详细情况不在本标准的规定范围内。

最初，假定一条路径的 PMTU 值是第一跳链路的 MTU（已知）。

在接收到一个 PacketTooBig 消息的时候，节点根据消息的内容来决定把消息应用到哪条路径。例如，如果把目的地址用作一条路径的本地表示法，那么利用来自最初分组的地址来决定把消息应用到哪条路径上。

注：如果原始的分组包含一个 Routing Header 路由头字段，应该利用路由头来决定这个分组内的目的地址位置。如果 Segments Left 字段为 0，那么目的地址是 IPv6 头中包含的 Destination Address 字段地址；如果剩余分段 Segments Left 字段大于 0，那么目的地址是路由头中最后的地址。

于是，节点把 **PacketTooBig** 消息中的 MTU 字段中的值作为临时 PMTU 值，并且把临时 PMTU 值和现使用的 PMTU 值进行比较，若前者小于后者，则覆盖后者。

节点必须把修改后减少的 PMTU 情况通告给分组层。如果 PMTU 值降低了，那么任何正在利用该条路径的分组层协议（例如，一个 TCP 连接）都必须得到通知。

即使 **PacketTooBig** 消息中包含的原始报文是 UDP 的，使用该路径的 TCP 也应该收到这个消息。

节点还应该通知触发 **PacketTooBig** 消息的消息发送实例，它的分组已经被丢弃，以便于它可以重新传输已丢弃的数据。

一个执行可以通过推迟通知直到下一次试图发送一个大于 PMTU 估值的分组的方式来避免使用通告降低 PMTU 的异步机制。在这种方式下，当试图发送一个大于 PMTU 估值的分组时，发送功能应该失效而且返回一个适当的差错指示。这个方式更适合于一个无连接的分组层协议（例如使用 UDP 的分组层协议），因为这时很难从 ICMP 层（在某些执行情况下）得到通告。在这种情况下，可以利用常用的超时重传机制来重传被丢弃的分组。

应该注意区分通过感知 PMTU 改变而在分组层协议上产生的通告和报文丢弃而产生的通告。前者可以推迟到分组层协议发送者要构造一个新报文的时候再通告，而后者必须立即通告给发送者，便于发送者立即重传被丢弃的报文。前者重新传送的报文则是通过处理接收 **PacketTooBig** 消息的方式，来确认已被丢弃需要重新发送的分组。

6.3 清除过期的 PMTU 信息

网络的拓扑是动态的，路由随时会发生变化。因此可能出现一条路径的本地表示法保持不变，而实际使用的路径已经发生变化的情况。因此，由一个节点存储的 PMTU 信息就会过期。

如果已过期的 PMTU 值过大，一旦在路径上发送一个分组，那么它立刻就可以被发现。目前还没有检测 PMTU 足够小的机制，因此实现时应该“老化”存储的 PMTU 值。当一个 PMTU 值在给定的时间内稳定（给定时间可以是 10min 的整数倍），应将 PMTU 估值重新设置为第一跳链路的 MTU，并通知分组层相关协议这个变化。这会导致再次运行完整的 PMTU 发现程序。

当提供这种更新机制的同时，应该提供给用户配置这个有效期时间的方法，并且用户也可以将这个有效期设置为无限长。

上层分组层协议一定不能根据 PMTU 的增加而重新发送数据，因为这个增加永远不会是对分组丢弃事件的响应。

实现这个超时机制的一种方法就是将 PMTU 值附带一个时间戳。最初的 PMTU（即下一跳链路的 MTU）对应的时间戳是一个特殊的保留值。之后如果由于 **PacketTooBig** 消息导致该 PMTU 被更新，那么就将其对应的时间戳置为当前的实际时间。

每隔 1min 检查一次所有缓存的 PMTU 信息，对于那些时间戳不是保留值的 PMTU，判断其时间是否已经超过设置的时间。如果超时，则执行下列操作：

- a) 将该 PMTU 设为初始的下一跳链路 MTU；
- b) 将其对应的时间戳设为保留值；
- c) 通告上层 PMTU 已经发生了变化。

6.4 TCP 层的行为

TCP 层必须跟踪一个 TCP 连接所使用的路径的 PMTU；它不应该发送可能会导致分组大于 PMTU 的数据段。一个实现方法就是在 TCP 层每次生成数据段时向 IP 层询问 PMTU，但这样做的效率比较低。而

且，TCP 使用了“慢启动”（Slow-start）的拥塞避免算法特别计算、缓存有关 PMTU 的数值。当 PMTU 改变时，根据接收到的异步通知，也能够更新 PMTU。

一个 TCP 实现也必须保存来自邻居的 MSS 值，两端发送的数据段长度不能超过 MSS 值。基于 4.xBSD 的实现中，这种方式可能需要为 TCP 状态记录增加一个附加的字段。

TCP MSS 值与 PMTU 无关。这个 MSS 值被 TCP 连接的对端使用，它可以是与 PMTU 没有关系的一个数值。关于确定 TCP MSS 选项数值在 YD/T 1341-2005 的分组尺寸（Packet Size Issues）以及最大上层净荷长度（Maximum Upper-Layer Payload Size）部分中有规定。

接收到一个 PacketTooBig 消息表明发送该消息的节点丢弃了一个报文。发送者也应按照其他丢弃数据段的情况来处理，即直到超时后重新发送该数据段。如果 PMTU 发现过程需要几次才能发现整个路径的 PMTU，这时可能会造成 TCP 连接的建立延迟数个 TCP 往返时间。

此外可以立即进行响应 PMTU 变化，重传丢弃报文，但是这只适用于采用 PacketTooBig 消息机制的特定连接。在重传中采用的分组尺寸不应该大于更新后的 PMTU。

分组层协议不能响应每个接收到 PacketTooBig 消息而重传报文，因为几个超长数据段的突发会引发几个这样的消息，从而会多次重传同样的数据段。如果更新后的 PMTU 值依然不准确，重复执行实现会造成重复发送的片段数量呈指数增长。

这意味着，TCP 层必须要能识别那些能够降低目前有效被使用 PMTU 值的 ICMPv6 消息，能够忽略其他消息。

许多 TCP 使用“拥塞避免”和“慢启动（slow-start）”算法以此提高性能。与 TCP 重传超时引起的重传机制不同，由 PacketTooBig 通告消息引起的重传不应改变拥塞窗口，同时应触发慢启动（slow-start）机制，即只重传一个报文，直到收到了相应的确认消息为止。

当 TCP 发送方的最大窗口大小不是数据段长度的整倍数时，将会引起 TCP 性能的下降。因此一般的实现都是使用了数据段长度的倍数值作为最大窗口大小。这里的最大窗口不是拥塞窗口，拥塞窗口的大小通常是数据段长度的倍数。在很多系统里，通常把片段尺寸设置为 1024byte，最大窗口长度（send space）通常是 1024byte 的倍数，因此缺省的情况下保持着正确的关系。如果采用 PMTU 发现，数据段长度有可能不是发送空间的因数，同时在连接中数据段长度还可能变化，这就意味 TCP 需要根据采用路径 MTU 发现协议而改变的 PMTU 值，改变发送窗口大小。发送窗口长度应该被设置为在保证小于发送方缓存条件下数据段长度的最大整数倍。

6.5 其他传输协议的问题

有些传输协议（例如 IETF RFC905 定义的 ISO TP4）在进行重传的时候，不允许重新打包。也就是说，一旦确定传输特定尺寸的数据段，不能把数据段内容分成更小的数据段用于重传。在重传的时候，可以通过 IP 层把最初的数据段进行分片处理。剩下的数据段在第一次传输的时候就不能超出 PMTU。

NFS 利用一个 RPC（远程过程调用，Remote Procedure Call）协议，当 RPC 在 UDP 上使用的时候，多数情况下，它会生成必须进行分片处理的净荷，即使是用于第一跳链路的净荷。某些情况下这样做会提高性能，但同时会在可靠性和性能方面存在问题，尤其是客户机和服务器之间存在路由器设备时。

因此在使用 NFS 的情况下，当路径必须经过路由器时，应该使用 PMTU 发现机制。大部分 NFS 的实现支持在挂载时间（mount-time）时改变 RPC 数据报文长度（通过改变有效的文件系统块长度来间接实现），但是后期可能需要通过修改来实现。

因为单一的 NFS 操作无法通过几个 UDP 数据报文分开，特定的操作（主要是文件名和目录的操作）

要求最小的净荷长度，这个最小的净荷长度如果由一个单独的报文发送，也会超出 PMTU。NFS 实现中不能将净荷大小设置为小于这个最小净荷长度。在这种情况下，IP 层会对净荷进行分片处理。

6.6 管理接口

实现时可以通过系统工具提供下列功能：

- a) 规定在一个给定的路径上不使能 PMTU 发现；
- b) 改变某条指定路径的 PMTU 值。

前一条可以通过标识给定路径方法实现。当在这条已标识的路径上发送报文时，IP 层不会发送大于 IPv6 最小链路 MTU 的分组。

上述功能可以应用到异常的网络环境中，或者被一个能得到 PMTU 值的路由协议使用。

实现还应该能够设置、改变 PMTU 老化时间。

7 安全问题

PMTU 发现机制将可能导致两种拒绝服务（Denial of Service）攻击。这些攻击都是基于发送虚假的 PacketTooBig 消息的方法。

第一种情况，就是发送一些 PacketTooBig 消息，并且通告的瓶颈链路 MTU 值远小于实际的 PMTU 值。这种攻击方式不会终止正常流量，因为 IPv6 规定不能将 MTU 设置为小于最小 MTU 的值，但这样会导致网络性能下降。

第二种情况与第一种相反，即 PacketTooBig 消息通告的瓶颈链路 MTU 值大于实际 PMTU 值。这种攻击方式将导致节点发出的超长报文被路由器丢弃。因此可能会造成暂时的阻塞。在一个往返时间内，节点会发现它的错误（从路由器接收到 PacketTooBig 消息），但是频繁受到这种类型的攻击会导致大量的分组被丢弃。但是节点不会因为接收 ICMPv6 消息而提高 PMTU 的数值，因此对此类攻击并不敏感。

如果恶意攻击者阻止一个正常节点接收到合法的 PacketTooBig 消息，这样也会导致拒绝服务。但是在这种情况下，可能有更为简单的攻击方式。

附 录 A
(资料性附录)

IETF RFC1981 与 IETF RFC1191 的对比

本标准与 IETF RFC1981 的一致性程度为非等效, 而 IETF RFC1981 主要来自于 IETF RFC1191, IETF RFC1191 规定了 IPv4 协议的 PMTU 发现。IETF RFC1191 的如下部分内容在本标准中并不需要:

- 路由器行为规范——IETF RFC1885 规定了 PacketTooBig 类型的 ICMPv6 消息以及相应的路由器行为。
 - DF 比特——在 IPv6 中不存在 DF 比特。
 - TCP MSS 讨论——在 YD/T 1341-2005 中规定了如何确定 TCP MSS 选项的数值。
 - old-style 消息——所有的 PacketTooBig 消息报告瓶颈链路的 MTU。
 - MTU 稳定表项——不需要, 因为本标准中没有 old-style 消息。
-