

组织: 中国互动出版网 (<http://www.china-pub.com/>)

RFC 文档中文翻译计划 (<http://www.china-pub.com/compters/emook/aboutemook.htm>)

E-mail: ouyang@china-pub.com

译者: Hlp(hlp,huangliuji@hotmail.com)

译文发布时间: 2001-5-23

版权: 本中文翻译文档版权归中国互动出版网所有。可以用于非商业用途自由转载, 但必须保留本文档的翻译及版权信息。

Network Working Group
Request for Comment: 2003
Category: Standards Track

C. Perkins
IBM
October 1996

在 IP 内封装 IP

(RFC2003 IP Encapsulation within IP)

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

摘要

本文档描述了一种可在 IP 数据报中封装另一个 IP 数据包 (作为净负载) 的方法. 封装通过把路由信息送往某个中间目的地 (不是由原 IP 头部的 IP Destination Address 域) 把正常的 IP 路由变为数据报. 封装可用于多方面, 例如使用移动 IP 把数据报传送到某个移动节点.

1.简介

本文档描述了一种可在 IP 数据报中封装另一个 IP 数据包 (作为净负载) 的方法. 封装通过把路由信息送往某个中间目的地 (不是由原 IP 头部的 IP Destination Address 域) 把正常的 IP 路由变为数据报. 一旦封装后的数据报到达该中间目的地节点, 就被拆分, 得到原

IP 数据报, 然后原数据报被送到目的地址 (由原 Destination Address 域决定). 封装与拆分数据报的过程通常称为数据报“隧道”(“tunneling”), 封装方和拆分方分别为隧道的端点 (“endpoints”); 封装方称为隧道的“入口点”(“entry point”), 拆分方称为隧道的出口点 (“exit point”).

在最常见的隧道中我们有

```
source ---> encapsulator -----> decapsulator ---> destination
```

其中 source, encapsulator, decapsulator 和 destination 是独立的节点。encapsulator 节点称为隧道的“入口点”而 decapsulator 节点称为隧道的“出口点”。在封装与拆分的过程中同一个隧道可能有多个 source-destination 对。

2. 动机

移动 IP 工作组指定封装作为移动 IP 工作组已经规定把封装作为从移动节点的“家乡网络”(“home network”) 向代理 (agent) 传送数据包的方法, 该代理能够以传统方式在移动节点在当前异于家乡的位置“本地”地传送数据包 (参见参考文献[8])。封装的使用也可表明在 IP 数据报的源地址 (或者中间路由器) 必须影响数据报送达最终目的地所经过的路由。封装的其他的应用包括多播, 预付费, 安全属性选择路由, 总的 (general) 路由选择策略。

封装与松散的 IP 源路由选择 (IP loose source routing option, 参考文献[10]) 可以相似方式影响数据报的路由, 但由几个技术上的原因使得愿意选择:

- 松散的 IP 源路由还有尚未解决的安全问题
- 当前 Internet 路由器在转发包括 IP 选项的数据报 (包括 IP 远路由选择) 时暴露出性能问题。
- 很多 Internet 节点在处理 IP 源路由选择时出错。
- 防火墙 (firewalls) 可能把 IP 远路由数据报拒之门外。
- 插入 IP 远路由选择可能使数据报的源地址和/或目的地址的认证信息的处理变得复杂化, 取决于认证如何进行。
- 中间路由器不应 (it' s impolite) 改变不是由它产生的数据报。

使用封装时必须权衡封装的优缺点:

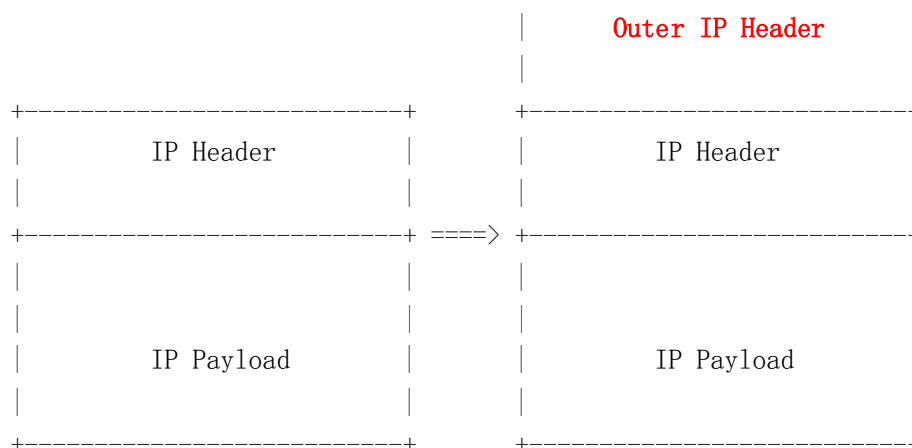
- 封装后的数据报一般比使用源路由算法的数据报大。
- 封装必须在事先知道隧道的出口点能够拆分数据报。
-

既然现在大多数的 Internet 节点在使用 IP 松散源路由选择时性能不够好, 封装的第二个技术缺点不像起初想象的那么严重。

3. 在 IP 中封装 IP

为了使 IP-in-IP 来封装 IP 数据报, 在现存 IP 头部前面插入外层的 IP 头(参考文献[10]), 如下所示:

```
+-----+
```



外层 IP 头部中的 Source Address 和 Destination Address 标识了隧道的“端口”。内层 IP 头部的中的 Source Address 和 Destination Addresses 标识了数据报的原（最初，original）发送方和接收方。内层 IP 头部不能被封装方修改（并在向隧道出口传输的过程中保持不变），除非按下面的方法递减 TTL。封装后的数据报在隧道传输的过程中 IP 选项不做任何修改。如果要修改，则在内外层 IP 头部间插入其他协议头部，如 IP 认证头部（Authentication header，参考文献[1]）。注意内层 IP 头部的安全选项可能影响正在封装的（外层）IP 头部的安全选项。

3.1.IP 头部各域及管理

外层 IP 头部由封装方按下面设置：

Version

4

IHL

因特网头部长度的外部 IP 头部的长度，用 32 位的字表示（参考文献[10]）。

TOS

服务类型(TOS)从内层 IP 头部拷贝。

Total Length

Total Length 为整个封装后 IP 数据报的长度，包括外层 IP 头部，内层 IP 头部，及其净载数据。

Identification, Flags, Fragment Offset

这三个域按参考文献[10]进行设置。但是如果在 IP 头部设置了“Don't Fragment”位，**必须**在外部 IP 头部中设置该位；如果内部 IP 头部没有设置“Don't Fragment”位，在外层 IP 头部中**可能**（以）设置该位，见 5.1。

Time to Live

外层 IP 头部的生存期(TTL)域设置为封装后数据报传输到隧道出口点所经历的大致时间。

Protocol

4

Header Checksum

为外层 IP 头部的“Internet 头部检验和”（参考文献[10]）。

Source Address

封装方的 IP 地址，即隧道的入口点。

Destination Address

拆封方的 IP 地址, 即隧道的出口点。

Options

内部 IP 头部中出现的选项通常不出现在外层 IP 头部中。但是**可能（以）**增加隧道自定义的选项。特别地, 内层 IP 头部支持的安全选项**可能**影响到外层的头部。不应该 (not expected) 在这些选项到隧道的选项或安全头部之间建立一对一的映射。

在封装数据报时, 如果隧道作为转发数据报的一部分, 内层 IP 头部的 TTL 将减 1; 否则, 在封装的过程中内层 TTL 保持不变。如果得到的内层 IP 头部的 TTL 为 0, 数据报被丢弃并**应该**向发送者产生一个 Time Exceeded 的 ICMP 信息。不允许封装方对 TTL=0 的数据报进行封装。内层 IP 头部中的 TTL 在拆分的过程中保持不变。拆分后, 如果内层数据报 TTL=0, 拆分方**必须**丢弃该数据报。拆分后, 如果拆分方转发该数据报到它的一个网络接口, 它像正常转发 IP 数据报那样递减 TTL。见 4.4。

封装方可以使用现存适合的 IP 机制来把封装后的净载数据传送到隧道的出口点。特别地, 允许使用 IP 选项, 还可以允许分片, 除非内层 IP 头部中设置了“Don't Fragment”位。使用该分片限制是为了使使用路径 MTU 发现 (参考文献[7]) 的节点能够得到他们所要寻找的信息。

3.2. 路由失败

在隧道内部的路由环回 (Routing loops) 特别危险, 它们使数据报再次回到封装方。假设一个数据报到达路由器等待转发, 而该路由器认为该数据报在传送之前必须封装, 那么:

- 如果该数据报的 **Source Address** 与路由器自己的任一个网络接口的 IP 地址匹配, 该路由器不允许为该数据报建立隧道; 相反, 该数据报应该被丢弃。
- 如果该数据报的 **Source Address** 与隧道的目的 IP 地址匹配 (隧道出口点一般由路由器根据数据报的 IP 头部的 **Destination Address** 选择), 路由器不允许为该数据报建立隧道, 相反, 该数据报应该被丢弃。

参见 4.4。

4. 隧道内部的 ICMP 信息

封装后的数据报被发送后, 封装方可能从该隧道内的任一中间路由器而不是隧道出口接收到一条 ICMP 信息 (参考文献[9])。封装方采取的动作取决于所收到的 ICMP 信息的类型。当收到的信息包含足够信息时, 封装方**可能**使用收到的信息产生一个相似的 ICMP 信息, 发送给产生未封装 IP 数据报的构建者 (原始发送方)。该过程称为中继 (“relaying”) 来自隧道的 ICMP 信息。

ICMP 信息表明处理数据报的过程中产生一个错误, 它包含引起错误的数据报的 (一部分) 的一个拷贝。中继一个 ICMP 信息要求封装方从该返回的数据报中剥去外层 IP 头部。对收到不包含足够信息的 ICMP 信息的情况, 见 5。

4.1. 目标不可达 Destination Unreachable (Type 3)

ICMP 目标不可达信息由封装方根据它们的 Code 域进行处理。这里给出的模型允许隧道扩

展 (“extend”) 到一个包括非本地节点 (如移动节点) 的网络。这样, 如果未封装数据报中的目标地址与封装者处在同一个网络, 可以修改 **Destination Unreachable Code** 的值使之与给定模型一致。

网络不可达 Network Unreachable (Code 0)

一条目标不可达 ICMP 信息**应该**返回给原始发送方。如果未封装数据报的目的地址与封装者处在同一个网络上, 封装者新产生的目标不可达信息应该为 Code=1 (Host Unreachable), 因为推测数据报到达了正确的网络而且封装方把最初的目的地址视为该网络的本地地址, 即使事实并非如此。否则(目的地址与封装者处在不同的网络上), 如果封装者返回目标不可达信息, Code 域**必须**设置为 0 (Network Unreachable)。

主机不可达 Host Unreachable (Code 1)

封装者应该尽可能把该主机不可达信息中继到未封装数据报的发送者。

协议不可达 Protocol Unreachable (Code 2)

当收到协议不可达 ICMP, 封装方应该向为封装数据报的发送方发送一个 Code 域为 0 或 1 的目标不可达 信息。(见 Code 为 0 部分)。因为原始发送方没有使用协议号为 4 来发送该数据报, 将向该发送方返回 Code 2。

端口不可达 Port Unreachable (Code 3)

该代号应该从不被封装方接收, 因位外层 IP 头部不指定任何端口号。不允许把该代号发送给未封装数据报的发送方。

数据报太大 Datagram Too Big (Code 4)

封装方**必须**把数据报太大 ICMP 中继给未封装数据报的发送方。

源路由失败 Source Route Failed (Code 5)

该代号应该由封装方自己处理。**不允许**把它中继给位封装数据报的发送方。

4.2.源淹没 Source Quench (Type 4)

封装方**不应该**把源淹没信息中继给未封装数据报的发送方, 但应该激活所使用的拥塞控制机制以帮助减轻隧道内部所检测到的拥塞。

4.3.重定向 Redirect (Type 5)

封装方可能自己处理重定向 ICMP 信息。不允许把重定向中继到为封装数据报的发送方。4。

4.4.超时 (Type 11)

超时 ICMP 信息在隧道自身内部报告(推测)路由环回。封装方收到超时信息**必须**把该超时信息作为主机不可达 (Type 3, Code 1) 信息向未封装数据报的发送方报告。主机不可达与网络不可达更优越; 因为数据报由封装方处理, 封装方通常被视为未封装数据报的目的地址且位于相同的网络上, 数据报被视为到达正确的网络, 但错误的目标节点。

4.5. 参数问题 Parameter Problem (Type 12)

如果参数问题指向从未封装数据报中拷贝而来的某个域, 封装方可能把该 ICMP 信息中继给未封装数据报的发送方; 否则, 如果问题是由封装方插入的 IP 选项引起, 封装方不允许把

该 ICMP 信息中继给发送方。注意遵循实际情况的封装方永不会把 IP 选项插入到封装的数据报中，除非出于安全原因。

4.6. 其他 ICMP 信息

其他 ICMP 信息与本协议规范中的封装无关，封装方应该遵循按参考文献[9]中所定义规范。

5. 隧道管理

不幸的是，ICMP 仅要求 IP 路由器返回 IP 头部之外的 8 个字节(64bits)。这不足以包括一个封装后(内层)IP 头部的一个拷贝，所以封装方不总是能把隧道内部的 ICMP 信息中继给原发送方。但是，通过仔细维护隧道的“软状态”(“soft state”)，封装方可在大多数情况下把精确的 ICMP 信息返回给发送者，封装方应该至少维护每一个隧道的下述软状态信息：

- 隧道的 MTU (见 5.1)
- 隧道的 TTL (路径长度 path length)
- 隧道端点的可达性

封装方使用它收到的来自隧道内部的 ICMP 信息更新该隧道的软状态信息。可能从隧道中的路由器返回的 ICMP 错误包括：

- 数据报太大
- 超时
- 目标不可达
- 源淹没

当随后经过该隧道的数据报到达时，封装方(器)检查该隧道的软状态。如果该数据报与隧道的当前状态冲突(新数据报的 TTL 小于隧道的“软状态”TTL) 封装方向原始数据报的发送方送回一个 ICMP 错误信息，但还是封装该数据报并把它转交给隧道。

使用这种技术，用封装方发送的 ICMP 错误信息不会总是与隧道内部发生的错误一一匹配，但它们可以精确地反映网络的状态。

隧道软状态最初开发用于 IP 地址封装 (IP Address Encapsulation, IPAE)，见参考文献[4]。

5.1. 隧道 MTU 发现

如果源发送方设置了 **Don't Fragment** 位并被拷贝到外层 IP 头部中，可以通过报告给封装方的 **Datagram Too Big** (Type 3, Code 4) ICMP 信息得知隧道的 MTU。为支持使用路径 MTU 发现的发送节点，所有封装实现必须支持隧道内部“路径 MTU 发现”软状态(参考文献[5, 7])。在这种特殊应用中，有几个好处：

- 分片(由于封装头部的大小)将作为路径 MTU 发现的受益者，在封装后只执行一次。这将阻止对一个数据报进行多次分片，提高拆分方和隧道内部的处理效率。
- 如果未封装数据报的源正在做路径 MTU 发现，那么要求封装方知道隧道的 MTU。任何来

自隧道内部的 **Datagram Too Big** 信息被返回到封装方, 正如在 5 中所注的那样, 封装方不可能把所有 ICMP 信息中继给未封装数据报的发送方. 通过维护隧道 MTU 的软状态, 封装方可以把正确的 **Datagram Too Big** 信息返回给未封装数据报的发送方以支持它自己的路径 MTU 发现. 在这种情况下, 由封装方发送给原发送方的 MTU **应该是**隧道的 MTU 减去正封装的 IP 头部的大小. 这将避免最初 IP 数据报被封装方分片。

- 如果未封装数据报的源不在做路径 MTU 发现, 封装方仍然需要知道隧道的 MTU. 特别地, 在封装时对原始数据报进行分片比允许对封装后的数据报分片要好得多. 对原始数据报的分片可由封装方完成, 且不需要特殊缓冲要求, 也不需要再在拆分方保存重新装配的状态. 相比之下, 如果对封装后的数据报进行分片, 那么拆分方必须在拆分前重新组装分片 (封装后) 后的数据报, 这就要求在拆分方重新组装状态和缓冲空间。

这样, 封装方正常情况下应该做路径 MTU 发现, 要求封装方在所有送往隧道的数据报均在 IP 头部设置“**Don't Fragment**”位. 但是该方法带来几个问题. 当原始发送方设置“**Don't Fragment**”位时, 发送方能通过重传原始数据报来对返回的 **Datagram Too Big** ICMP 错误信息迅速做出反应. 另一方面, 假定封装方收到来自隧道内部的 **Datagram Too Big** ICMP 错误信息, 如果未封装数据报的发送者没有设置“**Don't Fragment**”位, 封装方将无法让原始发送方知道该错误. 封装方**可能**在试图递增隧道的 MTU 时保存已发送数据报的一份拷贝, 以允许它在收到 **Datagram Too Big** 响应时分片并重传该数据报。

另一种选择是在未封装数据报没有设置“**Don't Fragment**”位时, 封装方可能 (以) 设置某些类型的数据报不设置“**Don't Fragment**”位。

5.2. 拥塞

封装方可能收到来自隧道内部的拥塞的暗示, 例如, 收到隧道内部的源淹没 (Source Quench) ICMP 信息. 另外, 与 Internet 无关的链路层以及各种协议可能以 Congestion Experienced 标志位 (参考文献[6]) 的形式提供该暗示. 封装方**应该**在隧道的软状态中反映拥塞状态, 在随后向隧道转发数据报时, 封装方**应该**使用适当手段来对拥塞进行控制 (参考文献[3]); 但是, 封装方**不应该**向位封装数据报的发送方发送源淹没 (Source Quench) ICMP 信息。

6. 安全方面的考虑

IP 封装潜在地降低了 Internet 的安全性, 所以在使用 IP 封装时应该注意. 例如 IP 封装使边沿路由器很难根据其头部对数据报进行过滤. 特别是, IP 头部的原始的 Source Address, Destination Address, 和 Protocol 各域, 以及数据报中传输层头部使用的端口号, 在封装后并不处在它们正常的位置. 因为任何 IP 数据报能被封装并通过隧道传输, 这样的过滤边沿路由器需要认真检查每一个数据报

6.1. 路由器方面的考虑

路由器需要知道 IP 封装的协议以便能够对传进来的数据报进行过滤。这样的过滤应该与 IP 身份认证（参考文献 1）集成在一起。在使用 IP 身份认证的地方，如果正在封装的（外层）数据包或者已经封装的（内层）数据包由一个经过认证的可信的源发送，则封装后的数据报可被允许进入某组织。不包含这些认证的封装后的数据包是一个极大的安全隐患。

封装和加密后的 IP 数据报(参考文献[2])也可能给过滤路由器带来问题。在这种情况下,路由器只能过滤那些共享了用于加密的安全联合的数据报。在所有数据包都需要过滤(或者至少说明)的环境中,为允许这种加密,接收节点必须采用一种机制来安全地把安全联合送到边沿路由器。对于传出的数据包也适用这种安全联合,但较少使用。

6.2.主机方面的考虑

能够接收封装后的 IP 数据报的主机应该只接受符合下面几种类型的一种或多种的数据报:

- 协议无害:不需要进行基于源地址的身份认证。
 - 正封装的(外层)数据报来自认证识别的可信的源,源的真实性建立于物理安全和边沿路由器的配置,但更可能来自 IP 身份认证头部(参考文献[1])。
 - 封装后的(内层)数据报包括一个 IP 身份认证头部
- 封装后的(内层)数据报送到属于拆分方的网络接口,或者拆分方已与之建立特殊关系以传输这些封装后数据报的节点。

这些检查的某些或全部在边沿路由器而不是接受节点进行,但如果边沿路由器检查作为备份而不是仅仅作为检察会更好。

7.致谢

3 和 5 节部分节选自移动 IP 因特网草案(Bill Simpson)的早期版本(参考文献[8])。6 节(安全考虑)的源文来自 Bob Smart。从 RFC 1853(参考文献[11],作者也是 Bill Simpson)中的到很多好主意,也感谢 Anders Klemets 发现草案中的错误并提出改进建议。最后感谢 David Johnson 对草案的非常细致的审阅,勘误,润色以及其他方面的。

参考文献

- [1] Atkinson, R., "IP Authentication Header", RFC 1826, August 1995.
- [2] Atkinson, R., "IP Encapsulating Security Payload", RFC 1827, August 1995.
- [3] Baker, F., Editor, "Requirements for IP Version 4 Routers", RFC 1812, June 1995.
- [4] Gilligan, R., Nordmark, E., and B. Hinden, "IPAE: The SIPP Interoperability and Transition Mechanism", Work in Progress.
- [5] Knowles, S., "IESG Advice from Experience with Path MTU Discovery", RFC 1435, March 1993.
- [6] Mankin, A., and K. Ramakrishnan, "Gateway Congestion Control Survey", RFC 1254, August 1991.
- [7] Mogul, J., and S. Deering, "Path MTU Discovery", RFC 1191, November 1990.

- [8] Perkins, C., Editor, "IP Mobility Support", RFC 2002, October 1996.
- [9] Postel, J., Editor, "Internet Control Message Protocol", STD 5, RFC 792, September 1981.
- [10] Postel, J., Editor, "Internet Protocol", STD 5, RFC 791, September 1981.
- [11] Simpson, W., "IP in IP Tunneling", RFC 1853, October 1995.

作者地址

关于本文档的问题可通过下述方式直接联系:

Charles Perkins
Room H3-D34
T. J. Watson Research Center
IBM Corporation
30 Saw Mill River Rd.
Hawthorne, NY 10532
Work: +1-914-784-7350
Fax: +1-914-784-6205
EMail: perk@watson.ibm.com

本工作组可以通过现任主席联系:

Jim Solomon
Motorola, Inc.
1301 E. Algonquin Rd.
Schaumburg, IL 60196

Work: +1-847-576-2753
EMail: solomon@comm.mot.com