

# 边界网关协议 4 (BGP-4)

## RFC 4271

陶志豪

[zhihao.tao@outlook.com](mailto:zhihao.tao@outlook.com)

<https://github.com/netwiki/share-doc>

目录

目录 ..... 1

备忘录状态..... 4

版权声明 ..... 4

概述 ..... 4

1. 介绍..... 5

    1.1 常用术语定义..... 5

    1.2 要求规范..... 6

2. 致谢 ..... 6

3. 操作摘要..... 7

    3.1 路由：公告和存储..... 8

    3.2 路由信息库..... 9

4. 消息格式..... 9

    4.1 消息头部格式..... 10

    4.2 OPEN 消息格式..... 10

    4.3 UPDATE 消息格式..... 12

    4.4 KEEPALIVE 消息格式..... 16

    4.5 NOTIFICATION 信息格式..... 17

5. 路径属性..... 18

    5.1 路径属性使用..... 19

        5.1.1 ORIGIN..... 19

        5.1.2 AS\_PATH..... 19

        5.1.3 NEXT\_HOP..... 20

        5.1.4 MULTI\_EXIT\_DISC..... 21

        5.1.5 LOCAL\_PREF..... 22

        5.1.6 ATOMIC\_AGGREGATE..... 22

        5.1.7 AGGREGATOR..... 22

6. BGP 错误处理..... 23

    6.1 消息头错误处理..... 23

    6.2 OPEN 消息错误处理..... 23

6.3 UPDATE 消息错误处理.....	24
6.4 NOTIFICATION 消息错误处理.....	25
6.5 HOLD 定时器超时错误处理.....	25
6.6 有限状态机错误处理.....	25
6.7 终止 .....	26
6.8 BGP 连接冲突检测.....	26
7. BGP 版本协商.....	27
8. BGP 有限状态机 (FSM) .....	27
8.1 BGP FSM 的事件.....	28
8.1.1 可选会话属性相关联的可选事件.....	28
8.1.2 管理事件.....	31
8.1.3 计时器事件.....	33
8.1.4 基于 TCP 连接的事件.....	34
8.1.5 基于 BGP 消息的事件.....	36
8.2 FSM 介绍.....	37
8.2.1 FSM 定义.....	37
8.2.2 有限状态机.....	39
9. UPDATE 消息处理.....	52
9.1 决策过程.....	53
9.1.1 阶段 1: 计算偏好度.....	53
9.1.2 阶段 2: 选路.....	54
9.1.3 阶段 3: 路由传播.....	57
9.1.4 重叠路由.....	57
9.2 UPDATE 发送过程.....	58
9.2.1 控制路由业务开销.....	58
9.2.2 路由信息的有效组织.....	59
9.3 路由选择标准.....	61
9.4 始发 BGP 路由.....	61
10. BGP 定时器.....	62
附录 A. 与 RFC 1771 的比较.....	63
附录 B. 与 RFC 1267 的比较.....	63
附录 C. 与 RFC 1163 的比较.....	63

附录 D. 与 RFC 1105 的比较.....	64
附录 E. 可能与 BGP 一起使用的 TCP 选项.....	64
附录 F. 实现建议.....	64
附录 F.1 每个消息的多个网络.....	64
附录 F.2 减少路线震荡.....	65
附录 F.3 路径属性排序.....	65
附录 F.4 AS_SET 排序.....	65
附录 F.5 控制版本谈判.....	65
附录 F.6 复杂的 AS_PATH 聚合.....	65
安全注意事项.....	66
IANA 注意事项.....	67

Network Working Group  
Request for Comments: 4271  
Obsoletes: 1771  
Category: Standards Track

Y. Rekhter, Ed.  
T. Li, Ed.  
S. Hares, Ed.  
January 2006

Thank Zhihao Tao for your hard work in Translation. The translator spent countless nights and weekends, using his hard work to make it convenient for everyone.

If you have any questions, please send a email to zhihao.tao@outlook.com

## 边界网关协议 4 (BGP-4)

### 备忘录状态

本文档为互联网社区规定的互联网标准化协议，并请讨论和建议来改进。

请参考当前版本的“互联网官方协议标准”（STD 1）和该协议的状态。此备忘录的传播不受限制。

### 版权声明

版权所有 (C) 互联网协会 (2006)。

### 概述

本文讨论了一种自治系统路由协议，即边界网关协议 (BGP)。

BGP 宣告系统的主要功能是与其他 BGP 系统交换网络可达性信息。这个网络可达性信息包括历经的一系列自治系统 (ASes) 可达信息。该信息足以构建 AS 可达性连通图，对路由环路进行修剪，在 AS 级别，可能会执行一些策略决定。

BGP-4 提供了一套支持无类别域间路由 (CIDR)。这些机制包括支持公告一组目的地作为 IP 前缀，并消除 BGP 内网络“类别”的概念。BGP-4 也介绍路由聚合机制，包括 AS 路径的聚合。

本文档作废 RFC 1771。

## 1. 介绍

BGP 宣告系统的主要功能是与其他 BGP 系统交换网络可达性信息。这个网络可达性信息包括历经的一系列自治系统 (ASes) 可达信息。该信息足以构建 AS 可达性连通图, 对路由环路进行修剪, 在 AS 级别, 可能会执行一些策略决定。

BGP-4 提供了一套支持无类别域间路由 (CIDR)。这些机制包括支持公告一组目的地作为 IP 前缀, 并消除 BGP 内网络“类别”的概念。BGP-4 也介绍路由聚合机制, 包括 AS 路径的聚合。

通过 BGP 交换的路由信息只支持基于目的地转发范例, 其认为路由器仅转发基于 IP 数据包的头中携带的目的地址的数据包。相反, 这又反映了一套可以 (和禁止) 使用的 BGP 决策。BGP 可以仅支持基于目的地转发范例的策略。

### 1.1 常用术语定义

本节提供了 BGP 协议具体术语的定义, 并且在整个文本中使用。

Adj-RIB-IN

Adj-RIBs-In 包含被 Peer 发布给了本地的 BGP Speaker 的未处理的路由信息。

Adj-RIB-OUT

Adj-RIBs-Out 包含本地 Speaker 公告给特定 Peer 的 UPDATE 消息中包含的路由。

自治系统 (AS)

自治系统的经典定义是一组在一个技术管理商下的路由器, 使用内部网关协议 (IGP) 和通用度量来确定如何路由 AS 内的数据包, 并使用 AS 间路由协议来确定如何将数据包路由到其他 ASes。自此经典度量定义的发展, 一个单一的 AS 可以使用几个 IGP 已经变得很通用, 有时在一个内部使用几组度量。自治系统术语的使用强调的是, 即使是多个 IGP 和度量被使用, AS 管理机构对其他 AS 显示有一个单一的连贯的内部路线设计, 并呈现一个通过其可达目的地的始终一致的路线图。

BGP 标识符

4 字节无符号整数, 表示 BGP 消息发送方的 BGP 标识符。给定的 BGP speaker 设置其 IP 地址为其标识符。BGP 标识符的值在启动时确定, 对于每个本地接口和 BGP 对等体都是一样的。

BGP speaker

实现 BGP 的路由器。

EBGP

外部 BGP (外部对等体之间的 BGP 连接)。

外部对等体

与本地系统属于不同的 AS 的 peer。

可行路由

接收方可以使用的公告路由。

IBGP

内部 BGP（内部对等体之间的 BGP 连接）。

内部对等体

与本地系统在同一个 AS 中的 peer。

IGP

内部网关协议 - 在单个自治区内，用于交换路由器间路由信息的路由协议。

LOC-RIB

Loc-RIB 包含由本地 BGP Speaker 决策过程选择出的路由信息。

NLRI

网络层可达性信息。

路由

一组目的地与到达这些目的地的路径的属性配对的信息。目的地是其 IP 地址，在 UPDATE 消息的 NLRI 信息字段携带的 IP 地址前缀中。路径是包含在同一 UPDATE 消息的路径属性字段中。

RIB

路由信息库。

不可行的路线

以前发布的可行路线不再可用。

## 1.2 要求规范

关键词“必须”，“不得”，“所需”，“已”，“不”，“应该”，“不应该”，“推荐”，“可能”和“可选”在文档[RFC2119]中所述。

## 2. 致谢

本文件最初于 1991 年 10 月发表为[RFC1267]由 Kirk Lougheed 和 Yakov Rekhter 共同创作。

我们要感谢 Guy Almes, Len Bosack 和 Jeffrey C. Honig 对早期版本的贡献（BGP-1）。

我们要特别承认丹尼斯·弗格森在本文档的早期版本做出许多贡献。

我们要明确地感谢 Bob Braden 的对于本文档的早期版本（BGP-2）观点，以及他的建性和宝贵意见。

我们也要感谢 Bob Hinden，对互联网工程指导小组（路由）的指导，和他的团队对本文档的早期版本（BGP-2）的审阅。这个团队由 Deborah Estrin, Milo Medin, John Moy, Radia Perlman, Martha Steenstrup, Mike St. Johns, and Paul Tsuchiya 组成，坚韧，专业，强大。

文件的某些部分从 IDRP 大量借用[IS10747]，它是 BGP 的 OSI 对等体。为此，从 ANSI X3S3.3 小组的 Lyman Chapin and to Charles Kunzinger（组内的 IDRP 编辑）获得授权。

感谢所有人，特别感谢陶志豪先生辛勤的翻译工作。

### 3. 操作摘要

边界网关协议（BGP）是一种跨自治系统路由协议 它是建立在 EGP 先前的经验（在[RFC904]中定义）和 NSFNET 骨干网中的 EGP 使用（在[RFC1092]和[RFC1093]中描述）。更多 BGP 相关信息，请参见[RFC1772]，[RFC1930]，[RFC1997]和[RFC2858]。

BGP 宣告系统的主要功能是与其它 BGP 系统交换网络可达性信息。这个网络可达性信息包括历经的一系列自治系统（ASes）可达信息。该信息足以构建 AS 可达性连通图，对路由环路进行修剪，在 AS 级别，可能会执行一些策略决定。

在本文的上下文中，我们假设一个 BGP Speaker 只向同行发布其自身使用的路由（在这种情况下，BGP speaker 被称为“使用”BGP 路由，如果是最优的 BGP 路由被用于转发）。所有其他情况不在本文档的范围之内。

在本文档的上下文中，术语“IP 地址”是指 IP 版本 4 地址[RFC791]。

通过 BGP 交换的路由信息只支持基于目的地转发范例，其认为路由器仅转发基于 IP 数据包的头中携带的目的地址的数据包。相反，这又反映了一套可以（和禁止）使用的 BGP 决策。注意一些策略不被基于目的地转发范例所支持，因此需要强制执行诸如源路由（也称为显式路由）的技术。一些政策也无法使用 BGP 强制执行。例如，BGP 不能使能一个 AS 发送流量到相邻 AS 而到达某个目的地（可通过但要）超越邻近的 AS，意图流量采取的路由，即源自相邻的 AS 的此流量路径（对于同一个目的地）。另一方面，BGP 可以支持任何遵循基于目的地的转发的策略。

BGP-4 提供了一套支持无类别域间路由（CIDR）[RFC1518, RFC1519]。这些机制包括支持公告一组目的地作为 IP 前缀，并消除 BGP 内网络“类别”的概念。BGP-4 也介绍路由聚合机制，包括 AS 路径的聚合。

本文档使用术语“自治系统”（AS）。自治系统的经典定义是一组在一个技术管理商下的路由器，使用内部网关协议（IGP）和通用度量来确定如何路由 AS 内的数据包，并使用 AS 间路由协议来确定如何将数据包路由到其他 ASes。自此经典度量定义的发展，一个



单一的 AS 可以使用几个 IGP 已经变得很通用，有时在一个内部使用几组度量。自治系统术语的使用强调的是，即使是多个 IGP 和度量被使用，AS 管理机构对其他 AS 显示有一个单一的连贯的内部路线设计，并呈现一个通过其可达目的地的始终一致的路线图。

BGP 使用 TCP [RFC793] 作为其传输协议。这消除了需要实现显式 UPDATE 分片，重传，确认和排序。BGP 在 TCP 端口 179 上侦听。BGP 中应用错误通知机制，假定 TCP 支持“优雅”关闭（即，在连接关闭之前，所有未完成的数据将被交付）。

在两个系统之间形成 TCP 连接。他们交流消息打开并确认连接参数。

初始数据流是 BGP 路由表的部分（被 export 策略认可），称为“Adj-Ribs-Out”（见 3.2）。随着路由表的变化，会发送增量更新。BGP 不需要定期刷新路由表。允许本地策略更改以获得正确的效果，而不必重置任何 BGP 连接，BGP speaker 应该（a）在连接的持续时间内保留所有同行通告给它的当前版本的路由，或（b）使用路由刷新扩展 [RFC2918]。

可以定期发送 KEEPALIVE 消息，以确保连接是活的。NOTIFICATION 消息作为错误或特殊条件响应发送。如果连接遇到错误条件，发送 NOTIFICATION 消息并且关闭连接。

不同 AS 中的对等体被称为外部对等体，而同一个 AS 中的对等体被称为内部对等体。内部 BGP 和外部 BGP 通常缩写为 IBGP 和 EBGP。

如果一个特定的 AS 有多个 BGP speakers，并为其他 ASes 提供中转服务，则必须小心确保 AS 内路由的一致性。由 AS 内使用的 IGP 提供一致性的 AS 内部路由。为了本文档的目的，假设所有对 AS 外的路由都通过所有 AS 内 BGP speaker 都互相保持 IBGP 来保持一致性。

本文档规定了 BGP 协议的基本行为。这个行为可以由扩展规范修改。什么时候该协议被扩展，新的行为被完全记录在了扩展规范。

### 3.1 路由：公告和存储

为了本协议的目的，路由被定义为一组目的地与到达这些目的地的路径的属性配对的信息。目的地是其 IP 地址，在 UPDATE 消息的 NLRI 信息字段携带的 IP 地址前缀中。路径是包含在同一 UPDATE 消息的路径属性字段中。

路由通过 UPDATE 消息在 BGP speaker 间通告。在单个 UPDATE 消息中可以发布具有相同路径属性的多个路由，通过在 UPDATE 消息 NLRI 中包含多个前缀。

路由存储在路由信息库（RIBs）中：即 Adj-RIBs-In，Loc-RIB 和 Adj-RIBs-Out，如第 3.2 节。

如果一个 BGP speaker 选择通告一个以前接收到的路由，它向 peer 公告前可以添加或修改路由的路径属性。

BGP 提供了 BGP speaker 可以通知对等体以前公告的路由不再可供使用的机制。给定的 BGP speaker 可以通过三种方法指示路由从服务中撤销：

- a) 以前公告目的地路由的 IP 前缀，可以在 UPDATE 消息中“撤销路由”字段发布，从而标记相关联的路由不再可用，
- b) 可以公布具有相同 NLRI 的替换路由，或者
- c) 可以关闭 BGP speaker 连接，这是隐式删除了这对 speaker 向每个人发布的所有路由。

改变路由的属性是通过公告一条替换路由来实现的。替换路由携带新（已更改）属性，并具有与原始路由相同的地址前缀。

### 3.2 路由信息库

BGP speaker 中的路由信息库（RIB）由三个不同的部分组成：

- a) Adj-RIBs-In：存储从其他 BGP speakers 收到的 UPDATE 消息中所学习到的路由信息。他们的内容作为决策过程输入的可用路由。
- b) Loc-RIB：Loc-RIB 包含本地路由信息，BGP speakers 应用本地策略筛选 Adj-RIBs-In 中的路由。这些是由本地 BGP speaker 使用的路由。每一个路由的下一跳必须通过本地 BGP speaker 的路由表是可解决的/resolvable。
- c) Adj-RIBs-Out：存储本地 BGP speaker 选择公告给其 peers 的路由信息。存储在 Adj-RIBs-Out 中的路由信息将被载入本地 BGP speaker 的 UPDATE 消息并通告给其 peers。

总而言之，Adj-RIBs-In 包含未处理的路由信息（已被同行发布给本地 BGP speaker）；Loc-RIB 包含被本地 BGP speaker 被本地决策过程选择过的路由；Adj-RIBs-Out 组织向特定 peer 发送的路由。

虽然概念模型区分了 Adj-RIBs-In，Loc-RIB 和 Adj-RIBs-Out，这既不意味着也不要求必须保持三个独立副本路由信息的实现。实现选择（例如，3 份信息 vs 1 复制与指针）不受协议的约束。

路由表中维护 BGP speaker 用于转发数据包的路由信息（或构建用于转发数据包的转发表）。路由表聚集来自直连网络、静态路由、从 IGP 协议和从 BGP 学到的路由。特定的 BGP 路由是否应该被安装在路由表中，或到同一目的地的 BGP 路由是否应该覆盖另一个来源的路由，是由本地策略决定，不在本文档中描述。除了真实的数据转发外，路由表用于在 BGP 更新中指定下一跳地址的解析（第 5.1.3 节）。

### 4. 消息格式

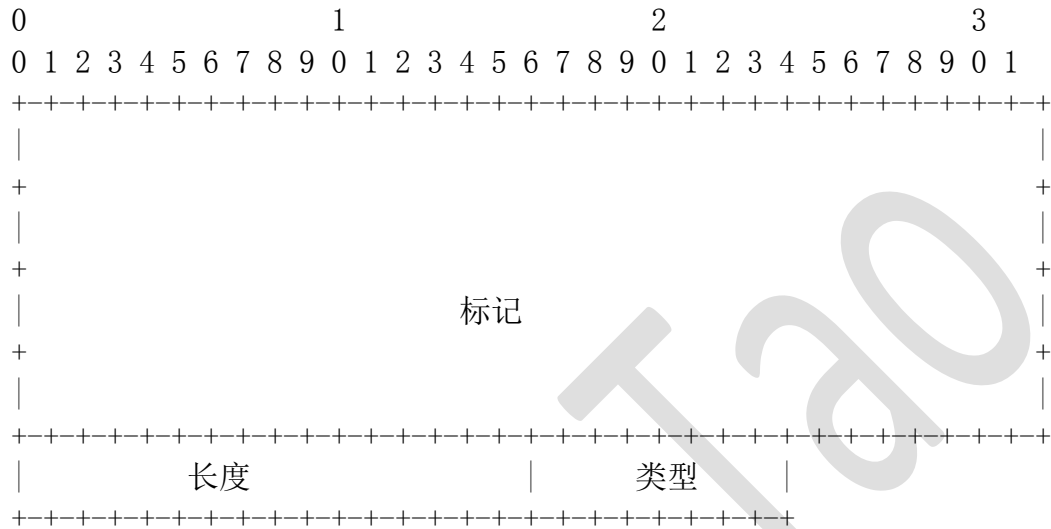
本节介绍 BGP 使用的消息格式。

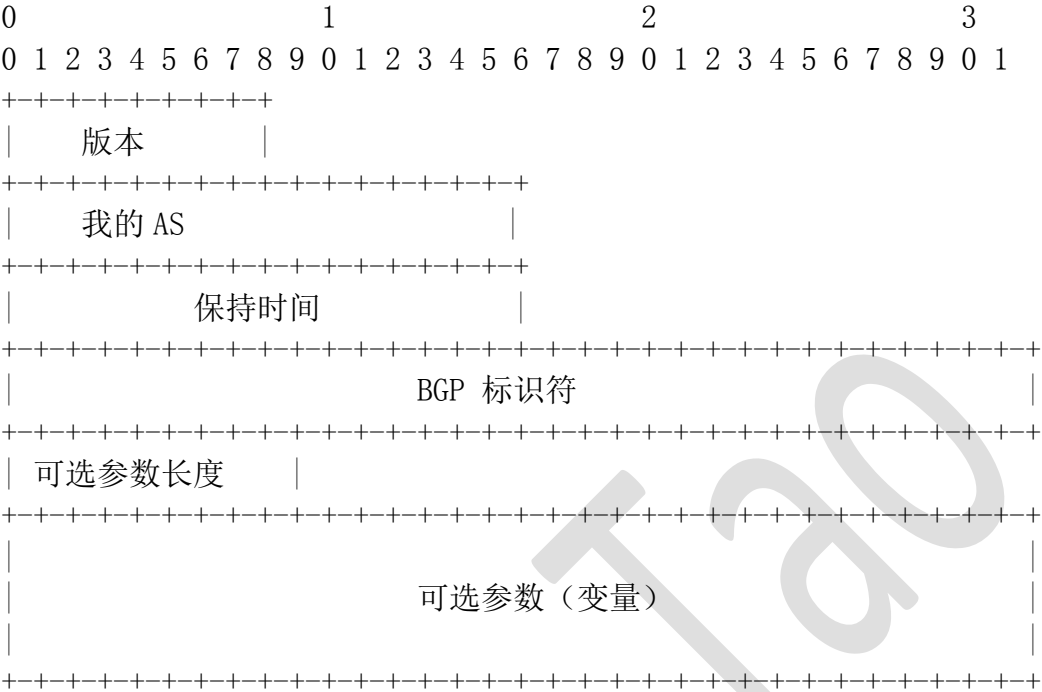
BGP 消息通过 TCP 连接发送。只有在完全收到之后才处理消息。最大消息大小为 4096 字节。所有实现都需要支持这个最大值。可能发送的最小消息包括没有数据部分的 BGP 报头（19 个字节）。

所有多字节字段都是网络字节顺序。

4.1 消息头部格式

每个消息都有一个固定大小的头。头部后面可能有也可能没有数据的部分，取决于消息类型。该这些字段的布局如下所示：





版本：该 1-octet 无符号整数表示协议版本消息号码。当前 BGP 版本号为 4。

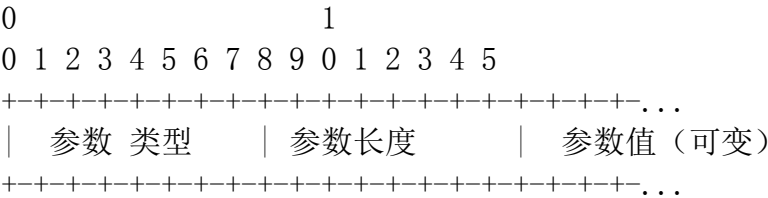
我的 AS：这个 2 字节无符号整数表示发送方 AS 号码。

保持时间：这个 2 字节无符号整数表示发送方建议保持定时器的值（秒数）。收到一个 OPEN 消息，一个 BGP 发言者必须计算保持定时器的值，使用其配置保持定时器的值和 OPEN 消息中接收到保持时间的较小值。该保持时间必须为零或至少三秒钟。一种实现以基于 Hold time 来拒绝连接。计算值表示连续收到来自发送方之间的 KEEPALIVE 和/或 UPDATE 消息可能经过的最大秒数。

BGP 标识符：该 4 字节无符号整数表示发送方的 BGP 标识符。给定的 BGP speaker 设置其 BGP 的标识符值为该 BGP speaker 的 IP 地址。BGP 标识符的值在初始化时便被确定，对每个本地接口和 BGP 对等体来说都是相同的。

可选参数长度：该 1-octet 无符号整数表示的总长度。如果这个值字段为零，不存在可选参数。

可选参数：此字段包含可选参数的列表，其中每个参数被编码为<参数类型，参数长度，参数值>三元组。



参数类型是明确的一个 1 字节字段用来识别各个参数。参数长度是一个包含参数值长度的 1 字节字段。参数值是一个可变长度字段这是根据参数类型来解析此字段。

[RFC3392]定义了能力可选参数。

OPEN 消息的最小长度为 29 个字节（包括消息头）。

4.3 UPDATE 消息格式

UPDATE 消息用于在 BGP peer 之间传递路由信息。可以使用 UPDATE 消息中的信息构建描述各种 AS 关系图。通过应用讨论的规则，路由可能会检测到路由环路和其他异常情况，并且从跨域路由中删除。

UPDATE 消息用于通告可共享通用路径属性的可行路由到 Peer，或撤销多个失效路由（见 3.1）。UPDATE 消息可以从服务中同时宣告一条可行路由，撤出多条不可行的路由。UPDATE 消息总是包含固定大小的 BGP 头，还包括其他字段，如下所示（注意，一些显示的字段可能不存在于某些 UPDATE 消息中）：

+-----+-----+
撤销路由长度（2 个八位字节）
+-----+-----+
撤销路由（可变）
+-----+-----+
总路径属性长度（2 个八位字节）
+-----+-----+
路径属性（变量）
+-----+-----+
网络层可达性信息（可变）
+-----+-----+

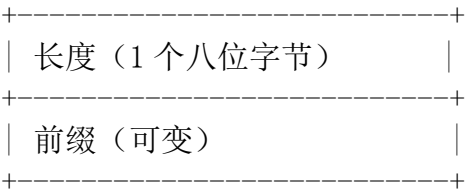
撤销路由长度：

这个 2 字节无符号整数表示撤销路由的总长度。其值由网络层可达性信息字段的长度确定，如下所述。

值为 0 表示没有路由正在从服务中撤销，并且“撤销路由”字段不存在这个 UPDATE 消息。

撤销路由：

这是一个包含一系列要从服务中撤销的路由 IP 地址前缀列表的可变长度字段。每个 IP 地址前缀被编码为 2 元组<length, prefix>，其字段描述如下：



这些领域的用途和含义如下：

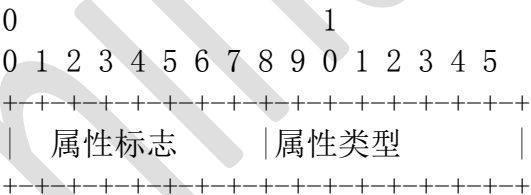
- a) 长度：长度字段指示 IP 地址前缀的位长度。长度为零表示前缀匹配所有 IP 地址（具有前缀，本身为零字节）。
- b) 前缀：前缀字段包含 IP 地址前缀，最后是用来表示结束的尾随位。注意尾随位的值是无关紧要的。

总路径属性长度：

这个 2 字节无符号整数表示路径属性字段的总长度。其值由网络层可达性信息字段的长度确定。在下面指定。  
值为 0 表示既不存在网络层可达性信息字段和路径属性字段是在此 UPDATE 消息中。

路径属性：

路径属性的可变长度序列存在于每个 UPDATE 中（不仅仅携带的撤销路由）。每个路径属性都是可变长度的三元组<属性类型，属性长度，属性值>。  
属性类型是由两个字节组成的字段，属性标志和属性类型。



属性标志字节的高位（bit 0）是可选位。是可选属性（如果设置为 1）或公认属性（如果设置为 0）。  
属性标志字节的第二个高位（bit 1）是传递位。可选属性是传递的（如果设置为 1）或非传递性（如果设置为 0）。  
对于公认的属性，传递位必须设置为 1。（有关传递属性的讨论，请参见第 5 节。）  
属性标志字节的第三个高位（bit 2）是部分位。可选传递属性中的信息是部分的（如果设置为 1）或完全的（如果设置为 0）。对于公认属性和可选的非传递属性，部分位必须设置为 0。

属性标志字节的第四个高位（位 3）是扩展长度位。它定义属性长度是一个字节（如果设置为 0）或两个字节（如果设置为 1）。

属性标志字节的低 4 位是没用过。它们在发送时必须为零，当必须被忽略接收时。

属性类型包含属性类型代码。第 5 部分中讨论了当前定义的属性类型。

如果设置了属性标志中扩展长度位被设置为 0，路径属性的第三个字节包含属性数据的字节长度。

如果设置了属性标志字节的扩展长度位被设置 1，路径属性的第三个和第四个字节包含属性数据的字节长度。

路径属性的剩余字节表示属性值，并根据属性标志和属性类型代码进行解释。支持的属性类型代码及其属性值和用途如下：

a) ORIGIN（类型代码 1）：

ORIGIN 是一个公认的强制属性，用于定义路径信息的起源。

数值	意义
0	IGP - AS 内产生的网络层可达性信息
1	EGP - 通过 EGP 协议学习的网络层可达性信息[RFC904]
2	INCOMPLETE - 网络层可达性通过其他方式学习的信息

此属性的用法在 5.1.1 中定义。

b) AS\_PATH（类型代码 2）：

AS\_PATH 是一个公认强制属性的 AS 路径段的序列。每个 AS 路径段是由三个<路径段类型，路径段表示长度，路径段值>。

路径段类型是 1 字节长度的字段定义如下：

价值段	类型
1	AS_SET: UPDATE 消息中已经遍历的无序 ASes 路由集合
2	AS_SEQUENCE: UPDATE 消息中已经遍历的有序 ASes 路由集合

路径段长度是 1 字节长度的字段，包含 AS 的数量（不是 AS 占用的字节数）路径字段值。

路径段值字段包含一个或多个 ASes 号码，每个编码为 2 字节长度字段。

此属性的用法在 5.1.2 中定义。

c) NEXT\_HOP（类型代码 3）：

这是一个公认强制属性，定义了 UPDATE 消息中 NLRI 字段中列出的用作路由器去往目的地下一跳的（单播）IP 地址。

此属性的用法在 5.1.3 中定义。

d) MULTI\_EXIT\_DISC（类型代码 4）：

这是一个可选的非传递属性，它是一个四字节无符号整数。在一个相邻 AS 存在多个入口点时，BGP speaker 进行决策过程时会使用该属性的值。

此属性的用法在 5.1.4 中定义。

e) LOCAL\_PREF（类型代码 5）：

LOCAL\_PREF 是一个四位字节无符号整数的公认属性。一个 BGP speaker 用来通知它的另一个内部 Peer，其对公告路由的偏好程度。

此属性的用法在 5.1.5 中定义。

f) ATOMIC\_AGGREGATE（类型代码 6）

ATOMIC\_AGGREGATE 是一个公认任意属性，长度为 0。

此属性的用法在 5.1.6 中定义。

g) AGGREGATOR（类型代码 7）

AGGREGATOR 是长度为 6 的可选传递属性。该属性包含形成聚合路由的最后一个 AS 号（用 2 个字节编码），后跟形成聚合路由的 BGP speaker 的 IP 地址（编码为 4 个八位字节）。应该与用于 speaker 的 BGP 标识符具有相同的地址。

此属性的用法在 5.1.7 中定义。

网络层可达性信息：

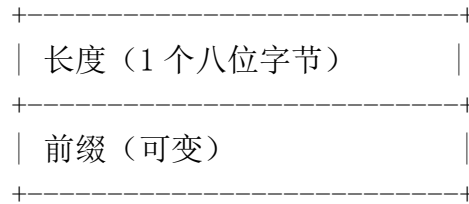
该可变长度字段包含 IP 地址列表前缀。NLRI 长度（以字节为单位）不是明确编码，而是可以计算：

UPDATE 消息长度 - 23 - 总路径属性长度 - 撤销路由长度

其中 UPDATE 消息长度是固定大小的 BGP 头，总路径属性长度和撤销路由长度。固定 BGP 头，总路径属性长度字段和撤回路线长度字段的总长度为 23。

可达性信息被编码为一个或多个 2 元组形式<length, prefix>，其字段如下所述：





这些字段的用途和含义如下：

a) 长度：

长度字段指示 IP 地址前缀的位长度。长度为零表示前缀匹配所有 IP 地址。

b) 前缀：

前缀字段包含 IP 地址前缀，后跟足够的尾随位，标识字段的末尾边界。请注意，尾随位的值是无关紧要的。

UPDATE 消息的最小长度是 23 个字节 = 19 个字节的固定头 + 2 个字节的撤销路由长度 + 2 个字节总路径属性长度（撤销的值路径长度为 0，总路径属性长度的值为 0）。

UPDATE 消息最多可以公告一组路径属性，但是可以有多个目的地，只要目的地共享这些目的地属性。包含一个给定 UPDATE 消息中的所有路径属性适用于 UPDATE 消息的 NLRI 字段中携带的所有目的地。

UPDATE 消息可以列出多个要从服务中撤销的路由。每个路由被其目的地标识（为 IP 前缀），其明确地标识 BGP speaker 上下文中的路由 - BGP speaker 关联之前已经被通告的路由。

UPDATE 消息可能只会公告将要从服务中撤销的路由，在这种情况下，消息将不包括在路径属性或 NLRI。相反，它可以只通告一条可行的路由，在这种情况下，“撤销路由”字段不需要存在。

UPDATE 消息不应该在撤销路由和网络层可达性信息字段包含相同的地址前缀。但是，BGP speaker 必须能够处理这种形式的 UPDATE 消息。BGP speaker 应该对待这种形式的 UPDATE 消息，好似撤销路由不包含该地址前缀。

#### 4.4 KEEPALIVE 消息格式

BGP 不使用任何基于 TCP 的保活机制，来确定是否同行可达。相反，在保活定时器过期前与 peer 交换 KEEPALIVE 消息。KEEPALIVE 消息之间最大的合理时间将是保持时间间隔的三分之一。KEEPALIVE 消息不得以短于一秒的间隔发送。实现可以根据保持时间调整 KEEPALIVE 消息的发送速率。

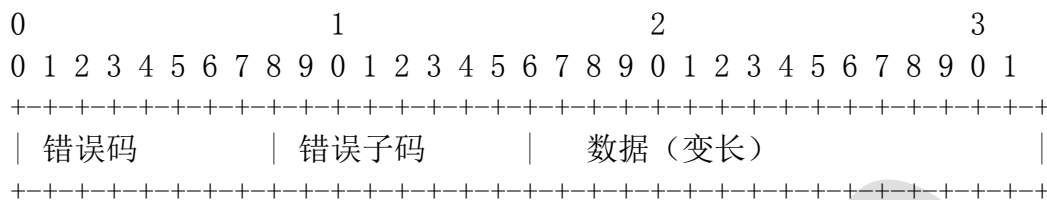
如果协商的保持时间间隔为零，则不得发送定期 KEEPALIVE 消息。

KEEPALIVE 消息仅包含 19 字节的消息头。

#### 4.5 NOTIFICATION 信息格式

当检测到错误条件时，发送 NOTIFICATION 消息。BGP 连接在发送后立即关闭。

除了固定大小的 BGP 头，NOTIFICATION 消息包含以下字段：



错误码:

该 1 字节无符号整数表示 NOTIFICATION 的类型。已经定义了以下错误代码：

错误码	符号名称	参考
1	消息头错误	第 6.1 节
2	打开消息错误	第 6.2 节
3	更新消息错误	第 6.3 节
4	保持定时器已过期	第 6.5 节
5	有限状态机错误	第 6.6 节
6	停止	第 6.7 节

错误子码:

这个 1 字节无符号整数提供关于报告错误性质更具体的信息。每个错误代码可能有一个或多个与之关联的错误子代码。如果没有定义适当的错误子代码，则为零（非特定）值用于错误子代码字段。

消息头错误子码:

- 1 - 连接未同步。
- 2 - 错误消息长度。
- 3 - 错误消息类型。

### OPEN 消息错误子码:

- 1 - 不支持的版本号。
- 2 - 坏对等 AS。
- 3 - 坏 BGP 标识符。
- 4 - 不支持的可选参数。
- 5 - [已弃用 - 见附录 A]。
- 6 - 不可接受的保持时间。

UPDATE 消息错误子码:

- 1 - 属性列表格式错误。
- 2 - 无法识别的公认属性

- 3 - 缺少公认的属性。
- 4 - 属性标志错误。
- 5 - 属性长度错误。
- 6 - ORIGIN 属性无效。
- 7 - [已弃用 - 见附录 A]。
- 8 - NEXT\_HOP 属性无效。
- 9 - 可选属性错误。
- 10 - 网络字段无效。
- 11 - 格式不正确的 AS\_PATH。

数据：

该可变长度字段用于 NOTIFICATION 原因诊断。数据字段的内容取决于错误代码和错误子代码。请参阅第 6 节了解更多。

请注意，数据字段的长度可以从消息长度字段确定，由公式：

$$\text{消息长度} = 21 + \text{数据长度}$$

NOTIFICATION 消息的最小长度是 21 个八位字节（包括消息头）。

## 5. 路径属性

本节讨论 UPDATE 消息的路径属性。

路径属性分为四类：

- 1. 公认必须
- 2. 公认可选
- 3. 可选传递
- 4. 可选非传递

BGP 实现必须识别所有公认属性。一些属性是强制性的，必须包含在每个 UPDATE 消息的 NLRI 属性中。其他的是可以自行决定的，能或不能在特定的 UPDATE 消息中发送。

一旦 BGP peer 更新了任何公认属性，它必须在其发送 UPDATE 中传递这些属性给它的 peers。

除公认属性，每个路径可以包含一个或多个的可选属性。这并不是必需或期望所有 BGP 的实现都支持所有可选属性。处理无法识别的可选属性由属性中的传递位设置决定。路径携带不可识别的传递可选属性应该被接受。如果收到一个不识别的传递可选属性的路径应该接受它，并传递给其他 BGP peers，然后该无法识别的路径传递可选属性必须与路径一起被传递，传递给其他 BGP peers 时部分属性位应该置为 1。如果路径被识别，传递可选属性的路径被接受，并要传递给其他 BGP peers，其部分属性标志位被之前的 AS 设置为 1，它禁止被当前 AS 设置为 0。无法识别的非传递性可选属性必须静默忽略，不能传递给其他 BGP peers。

最近，可选传递属性可以被发起者或路径中任何其他 BGP speaker 附加到路径上。如果他们没有被发起者附加，属性中的局部位应该被设置为 1。附加新的非传递可选属性的规则取决于具体属性的性质。每个新的非传递可选属性的文档资料将包含这些规则（给出一个例子，如描述 MULTI\_EXIT\_DISC 属性）。全部可选属性（传递和非传递）都可以被路径中的 BGP speaker 更新（如果适当）。

UPDATE 消息的发送者应该要求 UPDATE 消息中的路径按属性以升序排列。该 UPDATE 消息的接收者必须准备好处理 UPDATE 消息中乱序的路径属性。

在一个特定 UPDATE 的“路径属性”字段，相同的属性（具有相同类型的属性）不能出现多次。

如果 IBGP 和 EBGP 交换的 UPDATE 报文中都有 NLRI，强制类别的属性必须存在。把属性归类为可选是为了，协议拓展机制在某些情况下，完全自由决定的，必须或禁止。

属性	EBGP	IBGP
ORIGIN	强制性	强制性
AS_PATH	强制性	强制性
NEXT_HOP	强制性	强制性
MULTI_EXIT_DISC	自由裁量	自由裁量
LOCAL_PREF	请参阅第 5.1.5 节	必要
ATOMIC_AGGREGATE	请参见第 5.1.6 节 和第 9.1.4 节	
AGGREGATOR	自由裁量	自由裁量

5.1 路径属性使用

下面介绍每个 BGP 路径属性的使用情况。

5.1.1 ORIGIN

ORIGIN 是一个公认强制属性。ORIGIN 属性是由 speaker 生成关于路由起源的信息。它的值不应该被任何其他 speakers 改变。

5.1.2 AS\_PATH

AS\_PATH 是一个公认强制属性。这个属性通过 UPDATE 消息中的路由信息识别已经过的自治系统。这个的组成部分列表可以是 AS\_SET 或 AS\_SEQUENCE。

当 BGP speaker 传播从另一个 BGP speaker 的 UPDATE 消息中学到的路由时，它基于路由要发送到的 BGP Speaker 的位置，修改路由的 AS\_PATH 属性：

- a) 当给定的 BGP speaker 将路由通告给内部 peer 时，公告的 speaker 不要修改与路由关联 AS\_PATH 属性。
- b) 当给定的 BGP speaker 将路由通告给外部 peer 时，公告的 speaker 将 AS\_PATH 属性更新为如下：

- 1) 如果 AS\_PATH 的第一个路径段是 AS\_SEQUENCE 类型，本地系统预先添加自己的 AS 号序列的最后一个元素（将其放在相对于协议消息字节位置的最左边）。如果前面的行为导致 AS\_PATH 段中溢出（即超过 255 个 ASes），它应该添加一个类型为 AS\_SEQUENCE 的新段并且将自己的 AS 号码添加到此新的段。
- 2) 如果 AS\_PATH 的第一个路径段为 AS\_SET 类型，本地系统预先添加一个新的 AS\_SEQUENCE 类型路径段到 AS\_PATH，这个段中包括自的 AS 号。
- 3) 如果 AS\_PATH 为空，则本地系统创建 AS\_SEQUENCE 类型的路径段，将自己的 AS 放入该段，并将该分段放入 AS\_PATH。

当 BGP speaker 发起一条路由时：

- a) 始发 speaker 在所有发送到外部 peer 的 UPDATE 消息 AS\_PATH 属性中包含自己 AS\_SEQUENCE 类型的 AS 号，。在这种情况下，初始 speaker 的 AS 号将是路径段的唯一条目，此路径段将是 AS\_PATH 属性中唯一的段。
- b) 始发 speaker 在所有发往内部 peer 的 UPDATE 消息中包含一个空的 AS\_PATH 属性。（空的 AS\_PATH 属性是其长度字段值为零）。

每当修改 AS\_PATH 属性时，都会要求包含或在头部添加本地系统的 AS 号，本地系统可能包含/预先考虑在头部安装它自己多个实例的 AS 号码在 AS\_PATH 属性中。这是通过本地的配置来控制的。

### 5.1.3 NEXT\_HOP

NEXT\_HOP 是一个公认强制属性，定义了 UPDATE 消息中列出用于到达路由器目的地下一跳的 IP 地址。NEXT\_HOP 属性是计算如下：

- 1) 向内部 speaker 发送消息时，如果路由不是由本地发起的，BGP speaker 不应该修改 NEXT\_HOP 属性，除非已被明确配置宣布自己的 IP 地址作为 NEXT\_HOP。当宣告一个本地产生的路由到内部 peer，BGP speaker 应该使用路由器的（speaker 宣告可达网络所通过的）接口地址作为下一跳。如果路由与 speaker 直接相连，或者如果路由器的（speaker 宣告可达网络所通过的）接口地址是内部 peer 的地址，那么 BGP speaker 应该使用自己的 IP 地址为 NEXT\_HOP 属性（用于到达 peer 的接口地址）。
- 2) 向外部 peer X 发送消息时，peer X 与 speaker 是一跳连接：
  - 如果被宣布的路由是从内部 peer 学习或本地发起的，BGP speaker 可以使用内部 peer 路由器（或内部路由器）的（speaker 宣告可达网络所通过的）接口地址为 NEXT\_HOP 属性，假如 peer X 与这个地址共享一个公共子网的话。这是“第三方”NEXT\_HOP 属性的形式。

- 否则，如果宣告的路由是从一个外部 peer 学习的，speaker 可以使用任何自己用于本地路由计算的邻接路由器 IP 地址（从接收的 NEXT\_HOP 属性中得知）作为 NEXT\_HOP 属性，假如 peer X 与这个子网共享一个公共子网的话。这是“第三方”NEXT\_HOP 属性第二种形式。
- 否则，如果宣告路由到达的外部 peer 与宣告此路由 BGP speaker 的某个接口共享一个公共子网，speaker 可以使用与这些接口相关联的 IP 地址作为 NEXT\_HOP 属性。这被称为“第一方”NEXT\_HOP 属性。
- 默认情况下（如果上述条件不适用），BGP speaker 应该使用 speaker 用于与 peer X 建立 BGP 连接的接口 IP 地址作为 NEXT\_HOP 属性。

3) 向外部 peer X 发送消息时，peer X 与 speaker 是多跳 EBGp 连接：

- 可以将 speaker 配置为传播 NEXT\_HOP 属性。在这种情况下，当 speaker 宣告从其 peers 学习到的路由时，宣告路由的 NEXT\_HOP 属性与学习到的 NEXT\_HOP 完全属性相同（speaker 不修改 NEXT\_HOP 属性）。
- 默认情况下（如果上述条件不适用），BGP speaker 应该使用 speaker 用于与 peer X 建立 BGP 连接的接口 IP 地址作为 NEXT\_HOP 属性。

通常情况下，最短可用路径的 NEXT\_HOP 属性将被选择。BGP speaker 必须能够支持禁用第三方 NEXT\_HOP 属性的宣告，来处理不完善的桥接媒介。

由 BGP speaker 发起的路由不应该通告一条 NEXT\_HOP 为 peer 地址的路由到 peer。BGP speaker 不应该安装自己作为下一跳的路由。

BGP speaker 使用 NEXT\_HOP 属性来确定实际出接口和直接用于转发数据包到相关联的目的地的下一跳地址。

通过使用路由表的内容，对下一跳属性中的 IP 地址执行路由递归查找操作来决定直连下一跳地址，选择一个条目即使存在多个 cost 相等的条目。路由表条目通过解析 NEXT\_HOP 属性中的 IP 地址来指定出接口。如果条目指定了附加的子网，但没有指定下一跳地址，则 NEXT\_HOP 属性中的地址应该用作直连下一跳地址。如果该条目也指定了下一跳地址，该地址应该用作报文转发的直连下一跳地址。

#### 5.1.4 MULTI\_EXIT\_DISC

MULTI\_EXIT\_DISC 是可选的非传递属性，旨在用于外部（跨 AS）链路来区分在同一邻接 AS 的多个进或出接口。该 MULTI\_EXIT\_DISC 属性的值是一个四字节无符号整数，称为度量。所有其他因素相同，较低的度量出接口应该被优选。如果通过 EBGp 接收，MULTI\_EXIT\_DISC 属性可以通过 IBGP 传播到同一 AS 内的其他 BGP speaker（见 9.1.2.2）。从相邻的 AS 接收到的 MULTI\_EXIT\_DISC 属性不能传播到其他相邻的 AS。



BGP speaker 必须实现一个允许 MULTI\_EXIT\_DISC 属性从路由中移除机制（基于本地的配置），如果 BGP speaker 配置从一个路由删除 MULTI\_EXIT\_DISC 属性，那么这个删除必须是在确定路由的偏好程度和在执行路由选择之前完成（决策过程阶段 1 和 2）。

实现也可以（基于本地配置）改变通过 EBGp 接收的 MULTI\_EXIT\_DISC 属性的值。如果是 BGP speaker 配置为更改 EBGp 接收的 MULTI\_EXIT\_DISC 属性的值，然后更改值必须在确定路由的偏好程度和执行路线选择之前完成（决策过程阶段 1 和 2）。对此必要的限制详见第 9.1.2.2 节。

### 5.1.5 LOCAL\_PREF

LOCAL\_PREF 是一个公认属性，包含在给定 BGP speaker 发送给其他内部 peer 的全部 UPDATE 消息中。BGP speaker 应该基于本地配置的策略计算出对每个外部路由偏好程度，及向其内部 peer 发布路由时包含偏好程度。较高的偏好度必须被优选。BGP speaker 在其决策过程使用通过 LOCAL\_PREF 学习的偏好程度（见第 9.1.1 节）。

BGP speaker 不得在其发送到外部 peer 的 UPDATE 消息中包含此属性，除了在 BGP 联盟的情况下[RFC3065]。如果它包含在从外部 peer 接收的 UPDATE 消息中，那么该属性必须被接收 speaker 忽略，除了 BGP 联盟的情况[RFC3065]。

### 5.1.6 ATOMIC\_AGGREGATE

ATOMIC\_AGGREGATE 是一个公认可选属性。

当一个 BGP speaker 为了向特定 peer 宣告的目的聚合了几条路由，聚合路由的 AS\_PATH 通常包括一个 AS\_SET，其由形成聚合的 ASes 集组成的。在很多情况下，网络管理员可以判定聚合是否可以在没有 AS\_SET 的情况下安全地被通告，且不会形成路由环路。

如果聚合逐出至少部分存在于聚合路由 AS\_PATH 中的 AS 号，作为结果抛弃 AS\_SET，当聚合路由通告给 peer 时，应该包含 ATOMIC\_AGGREGATE 属性。

接收具有 ATOMIC\_AGGREGATE 属性路由的 BGP speaker 在传播路由给其它 speaker 时不应删除该属性。

接收具有 ATOMIC\_AGGREGATE 属性路由的 BGP speaker 在将此路由通告给其他 BGP speaker 时不得使该路由的任何 NLRI 更特殊（在 9.1.4 中定义）。

接收具有 ATOMIC\_AGGREGATE 属性路由的 BGP speaker 需要明白一个事实，当实际到达的目的地路径，正如路由的 NLRI 所描述的，同时拥有无环路的属性时，可能不是路由的 AS\_PATH 属性中指定的路径。

### 5.1.7 AGGREGATOR

AGGREGATOR 是一个可选传递属性，可能包含在通过聚合形成的 UPDATE 中（见第 9.2.2.2 节）。一个执行路由聚合的 BGP speaker 可以添加 AGGREGATOR 属性，其中应该包含自己的 AS 号和 IP 地址。该 IP 地址应与 BGP speaker 标识符相同。

## 6. BGP 错误处理

本节介绍处理 BGP 消息时检测到错误时要采取的操作。

当检测到这里描述的任何条件时，一条指定错误码、错误子码、数据段的 NOTIFICATION 报文应该被发送，且关闭 BGP 连接（除非明确声明不会发送 NOTIFICATION 消息，BGP 连接不要关闭）。如果没有错误子代码指定，则必须使用零。

短语“BGP 连接已关闭”表示 TCP 连接已关闭，相关的 Adj-RIB-In 已被清除，该 BGP 连接的全部资源已被释放。与远程 peer 相关联的 Loc-RIB 条目被标记为无效。本地系统重新计算被标记为无效路由的目的地的最佳路由。无效路由被从系统中删除之前，它宣告给 peer，撤销标记为无效的路由，或在无效的路由将从系统中删除前生成新的最佳路由。

除非明确指定，否则指示错误的消息发送的 NOTIFICATION 的数据字段为空。

### 6.1 消息头错误处理

处理消息头时检测到的所有错误，必须通过发送带有消息头错误代码的 NOTIFICATION 消息来指示。

错误子代码详细阐述了具体错误内容的性质。

消息头的标记字段的期望值全部为 1。如果消息头的标记字段不如预期的那样，那么发生同步错误，并且错误子代码必须设置为连接不同步。

如果下面至少有一个是真的：

- 如果消息头的长度字段小于 19 或大于 4096，或
- 如果 OPEN 消息的长度字段小于最小值 OPEN 消息的长度，或
- 如果 UPDATE 消息的 Length 字段小于 UPDATE 消息的最小长度，或
- 如果 KEEPALIVE 消息的 Length 字段不等于 19，或
- 如果 NOTIFICATION 消息的 Length 字段小于 NOTIFICATION 消息的最小长度，

那么错误子代码必须设置为消息长度错误。数据字段必须包含错误的长度字段。

如果消息头的类型字段无法识别，则错误子代码必须设置为消息类型错误。数据字段必须包含错误的类型字段。

### 6.2 OPEN 消息错误处理

处理 OPEN 消息时检测到的所有错误，必须通过发送带有 OPEN 错误代码的 NOTIFICATION 消息指示 OPEN 消息错误。错误子代码详细阐述了具体错误内容的性质。

如果接收到的 OPEN 消息的版本字段中版本号不受支持，则必须将错误子代码设置为不支持的版本号。数据字段是无符号的 2 字节整数，如果是较大值，表示本地支持的版本小于从远端 BGP peer 接收到的 OPEN 消息标识的版本号，如果是较小值，本地支持的版本号大于远端 BGP peer 的版本号，那么最小的本地支持的版本号。



如果 OPEN 消息的 AS 字段不可接受，那么错误子代码必须设置为 Bad Peer AS。判定可接受的 AS 号超出了协议范围。

如果 OPEN 消息的 HOLD 时间字段是不可接受的，那么错误子代码必须设置为不可接受的 HOLD 时间。一种实现是必须拒绝一秒或两秒的 HOLD 时间值。一种实现可能是拒绝任何建议的 HOLD 时间。一个实现可能是接收协商的 HOLD 时间。

如果 OPEN 消息的 BGP 标识符字段使用不正确的语法，错误子代码必须设置为 Bad BGP Identifier。语法正确性意味着 BGP 标识符字段表示一个有效的单播 IP 主机地址。

如果 OPEN 消息中的可选参数之一不识别，则错误子代码必须设置为不支持可选参数。

如果 OPEN 消息中的可选参数之一被识别，但是格式错误，那么错误代码必须设置为 0（非特异性）。

### 6.3 UPDATE 消息错误处理

处理 UPDATE 消息时检测到的所有错误，必须通过发送带有 UPDATE 错误代码的 NOTIFICATION 消息指示 OPEN 消息错误。错误子代码详细阐述了具体错误内容的性质。

UPDATE 消息的错误检查始于路径属性检查。如果撤销路由长度或总属性长度太大（即，如果撤回路由长度+总属性长度+23，超过了消息总长度），则错误子代码必须设置为畸形属性列表。

如果任何已识别的属性具有与属性类型代码相冲突的属性标志，则必须将错误子代码设置为属性标志错误。数据字段必须包含错误的属性（类型，长度和值）。

如果任何已识别属性的属性长度与预期长度（基于属性类型代码）有冲突，错误子码必须设置为属性长度错误。数据字段必须包含错误的属性（类型，长度和值）。

如果任何公认强制属性不存在，错误子码必须设置为缺失强制属性。数据字段必须包含丢失公认强制属性类型代码。

如果公认强制属性不被识别，错误子代码必须设置为无法识别的属性。数据字段必须包含无法识别的属性（类型，长度和值）。

如果 ORIGIN 属性具有未定义的值，则错误子代码必须设置为无效的 ORIGIN 属性。数据字段必须包含未识别的属性（类型，长度和值）。

如果 NEXT\_HOP 属性字段在语法不正确，那么错误子码必须设置为 NEXT\_HOP 无效属性。数据字段必须包含不正确的属性（类型，长度和值）。语法正确性意味着 NEXT\_HOP 属性表示有效的 IP 主机地址。

NEXT\_HOP 中的 IP 地址考虑语义的正确性时必须满足以下条件：

- a) 它不能是接收 speaker 的 IP 地址。

- b) 在 EBGp 的情况下，发送者和接收者是一跳连接，不仅 NEXT\_HOP 的 IP 地址必须是发送方用于建立 BGP 连接的 IP 地址，而且与之关联接口的 NEXT\_HOP 的 IP 地址必须与接收方的 BGP speaker 共享一个子网。

如果 NEXT\_HOP 属性在语义上不正确，那么错误应该是被记录，路由应该被忽略。在这种情况下，不应该发送 NOTIFICATION 消息，连接不应该被关闭。

检查 AS\_PATH 属性的句法正确性。如果路径在语法上不正确，则必须设置错误子代码为畸形的 AS\_PATH。

如果从外部 peer 接收到 UPDATE 消息，则本地系统可以检查 AS\_PATH 属性中的最左边 AS（相对于协议消息中字节序）是否等于发送消息的 peer 的 AS 号。如果检查确定不是这种情况，则错误子码必须设置为 AS\_PATH 格式不正确。

如果可选属性被识别，那么必须检查这个属性的值。如果检测到错误，属性必须丢弃，并且错误子代码必须设置为可选属性错误。数据字段必须包含这个属性（类型，长度和值）。

如果任何属性在 UPDATE 消息中出现多次，那么错误子代码必须设置为属性列表的格式不正确。

检查 UPDATE 消息中的 NLRI 字段的语法有效性。如果该字段在语法上不正确，那么该错误子码必须设置为无效的网络字段。

如果 NLRI 字段中的前缀在语义上是不正确的（例如，不可知的组播 IP 地址），本地应该记录一个错误，该前缀应该被忽略。

包含正确路径属性但不包含 NLRI 的 UPDATE 消息，应被视为有效的 UPDATE 消息。

#### 6.4 NOTIFICATION 消息错误处理

如果一个 peer 发送一个 NOTIFICATION 消息，并且该消息的接收者检测到该消息中有错误，接收方不能使用 NOTIFICATION 消息将此错误报告回 peer。任何这样的错误（例如，无法识别的错误代码或错误子代码）应该被注意，在本地记录，并提醒 peer 的管理机构注意。然而，这样做的手段在于超出本文件的范围。

#### 6.5 HOLD 定时器超时错误处理

如果系统在 OPEN 消息 HOLD 时间字段指定的周期内没有收到连续的 KEEPALIVE，UPDATE 和/或 NOTIFICATION 消息，然后发送错误代码设置为 HOLD 定时器过期 NOTIFICATION 消息，并关闭 BGP 连接。

#### 6.6 有限状态机错误处理

BGP 有限状态机检测到的任何错误（例如，意外事件）通过发送错误代码设置为有限状态机错误的 NOTIFICATION 消息来指示。

## 6.7 终止

在没有任何致命错误（这章指出的），BGP speaker 可以选择在任何给定时间，发送设置错误代码为 Cease 的 NOTIFICATION 消息关闭 BGP 连接。然而，存在本章所指的致命错误时，不允许发送错误代码为 Cease 的 NOTIFICATION 消息。

BGP speaker 可以支持强加一个从邻居接受地址前缀数量上限能力的本地配置。当达到上限时，speaker 在本地配置的控制下，（a）丢弃来自邻居的新地址前缀（同时维护 BGP 与邻居的连接），或（b）终止 BGP 连接与邻居。如果 BGP speaker 决定终止其与邻居的 BGP 连接（因为从邻居接收的地址前缀的数量超过本地配置的上限），那么 speaker 必须向邻居发送一个错误代码为 Cease 的 NOTIFICATION 消息。speaker 也可以本地记录 log。

## 6.8 BGP 连接冲突检测

如果一对 BGP speaker 同时试图彼此间建立 BGP 连接，同时形成两个并行的连接。如果这些连接中一个连接使用的源 IP 地址与另一个连接使用的目的 IP 地址相同，第一个连接使用的目的 IP 地址与另一方使用的源 IP 地址相同，会发生连接冲突。在里面连接冲突事件中，一方连接必须被关闭。

根据 BGP 标识符的值建立一个约定，在 BGP 连接冲突发生时，将用于检测哪个 BGP 连接将被保留。这个约定是比较冲突涉及的两个 BGP peers 的标识符，只保留由具有较大 BGP 标识符的 BGP speaker 发起的连接。

接收到 OPEN 消息后，本地系统必须检查其所有处于 OpenConfirm 状态连接。如果 BGP speaker 通过协议之外的方式知道 peer 的标识，则要检查处于 OpenSent 状态下的连接。如果，这些连接中存在一条与远端 BGP speaker（其 BGP 标识符等于 OPEN 消息中的某个）的连接，与收到的 OPEN 消息的连接出现冲突，则本地系统执行以下冲突解决流程：

- 1) 将本地系统的 BGP 标识符与远端系统的 BGP 标识符（在 OPEN 消息中指定）进行比较。把 BGP 标识符转换为主机序，并将它们视为 4 字节无符号整数来完成比较。
- 2) 如果本地 BGP 标识符的值小于远程 BGP 标识符的值，本地系统关闭已存在的 BGP 连接（已经在 OpenConfirm 状态中的那个），并接受远端系统发起的 BGP 连接。
- 3) 否则，本地系统关闭新创建的 BGP 连接（与新接收的 OPEN 消息相关的连接），并继续使用现有的连接（一个已经在 OpenConfirm 状态的）。

除非配置允许，一个处于 Established 状态的现有 BGP 连接产生一个连接冲突，会关闭新创建的连接。

请注意，在连接处于 IDLE 状态，CONNECT 状态，ACTIVE 状态时，连接冲突不会被检测到。

关闭 BGP 连接（解决冲突）是通过发送错误代码为 Cease 的 NOTIFICATION 消息完成的。

## 7. BGP 版本协商

BGP speaker 可以通过多次尝试打开 BGP 连接的方式来协商协议版本，从每个 BGP speaker 支持的最高版本号开始。如果尝试 OPEN 失败，错误代码为 OPEN 消息错误，错误子代码为不支持的版本号，则 BGP speaker 尝试由其 peer 通过 NOTIFICATION 消息中传递来的可用的版本号。如果两个 peer 支持一个或多个共同版本，那么将允许他们快速确定最高的共同版本。为了支持 BGP 版本协商，未来版本的 BGP 必须保留 OPEN 和 NOTIFICATION 格式的消息。

## 8. BGP 有限状态机(FSM)

本文档对数据结构和 FSM 进行概念性描述，并不必像这里描述的那样精确地实现，只有实现支持描述的功能，而且对外部展现的行为相同。

本节详细说明了 BGP 有限状态机（FSM）的操作组成。该部分由两部分组成：

- 1) 事件状态机的描述（第 8.1 节）
- 2) 有限状态机描述（第 8.2 节）

每个连接所需的会话属性（强制性）包括：

- 1) State/状态
- 2) ConnectRetryCounter/重连计数器
- 3) ConnectRetryTimer/重连定时器
- 4) ConnectRetryTime/重连时间
- 5) HoldTimer/HOLD 定时器
- 6) HoldTime/HOLD 时间
- 7) KeepaliveTimer/保活定时器
- 8) KeepaliveTime/保活时间

State/状态会话属性表示 BGP 的 FSM 当前状态。

ConnectRetryCounter/重连计数器表示 BGP peer 尝试建立 peer 会话的次数。

与定时器相关的强制属性将在第 10 章节中介绍。每个定时器都有一个“定时器”和一个“时间”（初始值）。

可选会话属性如下所示。这些可选项可以被每个连接或每个本地系统支持属性：

- 1) AcceptConnectionsUnconfiguredPeers/接受非配置的 Peer 连接
- 2) AllowAutomaticStart/支持自动开始
- 3) AllowAutomaticStop/支持自动结束
- 4) CollisionDetectEstablishedState/建立状态的冲突检测
- 5) DampPeerOscillations/Peer 震荡抑制

- 6) DelayOpen/延时 OPEN
- 7) DelayOpenTime/延时 OPEN 时间
- 8) DelayOpenTimer/延时 OPEN 定时器
- 9) IdleHoldTime/空 HOLD 时间
- 10) IdleHoldTimer/空 HOLD 定时器
- 11) PassiveTcpEstablishment/被动 TCP 建立
- 12) SendNOTIFICATIONwithoutOPEN/没有 OPEN 时发送 NOTIFICATION
- 13) TrackTcpState/TCP 状态追踪

可选会话属性支持不同 BGP 功能特性，影响 BGP FSM 状态转换。两组与计时器有关的属性是：

组 1： DelayOpen, DelayOpenTime, DelayOpenTimer

组 2： DampPeerOscillations, IdleHoldTime, IdleHoldTimer

第一个参数 (DelayOpen, DampPeerOscillations) 是可选属性，表示定时器功能有效。该“时间”值表示“定时器” (DelayOpenTime, IdleHoldTime) 的初始值。“定时器”表示实际计时器。

有关交互的说明，请参阅 8.1.1 节对这些可选属性和状态机的事件之间描述。第 8.2.1.3 节还简要概述了不同类型的可选属性（标志或定时器）。

## 8.1 BGP FSM 的事件

### 8.1.1 可选会话属性相关联的可选事件

BGP FSM 的输入是事件。事件可以强制或者可选。一些可选事件与可选会话属性相关联。可选会话属性启用多组 FSM 功能。

FSM 功能，事件和可选会话属性之间的联系如下所述。

#### *组 1： 自动管理的事件（开始/停止）*

可选会话属性： AllowAutomaticStart,  
AllowAutomaticStop,  
DampPeerOscillations,  
IdleHoldTime, IdleHoldTimer

选项 1: AllowAutomaticStart

描述： 可以管理控制 BGP Peer 的启动和停止。这个管理控制既可以是手动的，基于操作者干预，或者是在特定 BGP 实现逻辑的控制之下。术语“自动”是指，当这些逻辑判定 BGP peer 连接应该重新启动时，一个“开始”事件被发布到 BGP peer 连接 FSM。AllowAutomaticStart 属性说明此 BGP 连接支持自动 BGP 连接启动。

如果 BGP 实现支持 AllowAutomaticStart, peer 可能会重复重启。另外三个选项控制发生自动重启的速率: DampPeerOscillations, IdleHoldTime 和 IdleHoldTimer。

DampPeerOscillations/Peer 震荡抑制选项表示带来额外的实现逻辑来抑制 BGP peer 连续的自动启动和自动停止的振荡。IdleHoldTime 表示 BGP peer 允许在接下来的自动重启前, 保持空闲状态的时间长度。IdleHoldTimer 是保持处于空闲状态的 peer 的定时器。

一个 DampPeerOscillations 逻辑的例子, 如果 BGP peer 在一段时间内重复振荡连接(连接/断开), 则增加 IdleHoldTime 值。为保证这个逻辑, 一个 peer 可以在 5 分钟内连接和断开 10 次。IdleHoldTime 值从 0 到 120 秒重置。

值 : TRUE 或 FALSE

选项 2 : AllowAutomaticStop

描述 : 这个 BGP peer 会话可选属性表示 BGP 连接允许“自动”停止其连接。“自动”停止是指特定实现逻辑控制下的停止。该特定实现逻辑超出本文的范围。

值 : TRUE 或 FALSE

选项 3 : DampPeerOscillations

描述 : DampPeerOscillations 可选会话属性表示 BGP 连接使用抑制 BGP peer 在空闲状态振荡的逻辑。

值 : TRUE 或 FALSE

选项 4 : IdleHoldTime

描述 : IdleHoldTime 是在 IdleHoldTimer 中设置的值。

值 : 以秒为单位的时间

选项 5 : IdleHoldTimer

描述 : IdleHoldTimer 有助于控制 BGP peer 振荡。IdleHoldTimer 用于保持 BGP peer 在空闲状态持续一段时间。IdleHoldTimer\_Expires 事件在第 8.1.3 节介绍。

值 : 以秒为单位的时间

## 组 2 : 未配置的 peer

可选会话属性 : AcceptConnectionsUnconfiguredPeers

选项 1 : AcceptConnectionsUnconfiguredPeers



描述：BGP FSM 可选的允许接受来自非预配置邻居 BGP peer 的连接。该“AcceptConnectionsUnconfiguredPeers”可选会话属性允许支持具有接受或拒绝未配置 peer 的状态转换的 FSM 实现。

AcceptConnectionsUnconfiguredPeers 有安全隐患 请参考 BGP 缺陷文件[RFC4272]中的详细信息。

值：真或假

### 组 3：TCP 处理

可选会话属性：PassiveTcpEstablishment, TrackTcpState

选项 1：PassiveTcpEstablishment/被动 TCP 建立

描述：此选项表示 BGP FSM 将会被动等待远端 BGP peer 建立 BGP TCP 连接。

值：TRUE 或 FALSE

选项 2：TrackTcpState

描述：BGP FSM 通常会追踪一个 TCP 连接尝试的最终结果，而不是单个 TCP 消息。视情况而定，BGP FSM 可以支持与 TCP 连接谈判的额外交互。与 TCP 事件的交互可能会增加 BGP peer 连接需求的记录数量和 BGP FSM 的变化数量。

值：TRUE 或 FALSE

### 组 4：BGP 消息处理

可选会话属性：DelayOpen, DelayOpenTime, DelayOpenTimer,  
SendNOTIFICATIONwithoutOPEN,  
CollisionDetectEstablishedState

选项 1：DelayOpen

描述：DelayOpen 可选会话属性允许，配置 OPEN 消息延迟特定时间段后（DelayOpenTime）发送的实现。延迟可让远程 BGP peer 先发送第一个 OPEN 消息。

值：TRUE 或 FALSE

选项 2：DelayOpenTime

描述：DelayOpenTime 是 DelayOpenTimer 设置的初始值。

值：以秒为单位的时间

选项 3：DelayOpenTimer

描述 : DelayOpenTimer 可选会话属性用于在一个连接中延迟发送 OPEN 消息。  
DelayOpenTimer\_Expires 事件 (事件 12) 在 8.1.3 节中描述。

值 : 时间 (秒)

选项 4 : SendNOTIFICATIONwithoutOPEN

描述 : SendNOTIFICATIONwithoutOPEN 允许 Peer 发送 NOTIFICATION 在没有先发送 OPEN 信息。没有这个可选会话属性, BGP 连接假定 OPEN 消息必须在 peer 发送 NOTIFICATION 消息之前被发送。

值 : 真或假

选项 5 : CollisionDetectEstablishedState

描述 : 通常, 冲突检测 (见第 6.8 节) 将在 Established 状态中被忽略。这个可选会话属性表示此 BGP 连接处理在 Established 中发生冲突。

值 : 真或假

注意: 可选会话属性明确了 BGP FSM 现有特性实现的描述。可选的会话属性可能会为其实现预先定义, 并且不能通过管理接口读取用于现有的正确实现。作为较新的 BGP MIB (版本 2 及更高版本), 这些字段可通过管理界面访问。

### 8.1.2 管理事件

管理事件是一种用于用户界面和 BGP 策略引擎通知 BGP FSM 开始或停止 BGP 状态机的事件。可选的连接属性增强了基本的开始和停止标志, 用一种确切类型开始或停止机制通知 BGP FSM。符合这种例子是事件 5, AutomaticStart\_with\_PassiveTcpEstablishment。BGP 通过这个事件实现了向 FSM 发送信号, 该实现使用自动启动及被动 TCP 建立选项。被动 TCP 建立通知该 BGP FSM 等待远端启动 TCP 建立。

请注意, 只有事件 1 (ManualStart) 和事件 2 (ManualStop) 是强制管理事件。所有其他管理事件都是可选 (事件 3-8)。以下每个事件都有一个名称, 定义, 状态 (强制或可选) 和在每个阶段都需要被设置的可选的会话属性。当通过事件 8 为 BGP FSM 生成事件 1 时, “可选属性状态” 中的条件应该被验证。如果对这些条件有任何不满足, 那么本地系统应该记录一个 FSM 错误。

某些可选会话属性的设置可能是隐式实现的, 因此可能不会被外部操作员行为显式设置。8.2.1.5 节描述了这些隐式可选会话属性的设置。下面描述的管理状态在一些实现中也可能是隐式的, 不能由外部操作员直接配置。

#### 事件 1 : ManualStart

定义 : 本地系统管理员手动启动对 peer 连接。

状态 : 强制



可选属性状态：PassiveTcpEstablishment 属性应该被设置为 FALSE。

#### 事件 2：ManualStop

定义：本地系统管理员手动停止对 peer 连接。

状态：强制

可选属性状态：不与任何可选属性进行交互。

#### 事件 3：AutomaticStart

定义：本地系统自动启动 BGP 连接。

状态：可选，具体取决于本地系统

可选属性状态：

- 1) 如果发生此事件，应该设置 AllowAutomaticStart 属性为 TRUE。
- 2) 如果支持 PassiveTcpEstablishment 可选会话属性，它应该被设置为 FALSE。
- 3) 当事件发生时，如果支持 DampPeerOscillations，应设置为 FALSE。

#### 事件 4：ManualStart\_with\_PassiveTcpEstablishment

定义：本地系统管理员手动启动对 peer 连接，但是 PassiveTcpEstablishment 使能。PassiveTcpEstablishment 可选属性表示 peer 将先建立连接。

状态：可选，具体取决于本地系统

可选属性状态：

- 1) 如果发生此事件，PassiveTcpEstablishment 属性应该被设置为 TRUE。
- 2) 发生此事件时，DampPeerOscillations 属性应该被设置为 FALSE。

#### 事件 5：AutomaticStart\_with\_PassiveTcpEstablishment

定义：本地系统自动启动 BGP 连接并使能 PassiveTcpEstablishment。PassiveTcpEstablishment 可选属性表示 peer 将先建立连接。

状态：可选，具体取决于本地系统

可选属性状态：

- 1) 应该设置 AllowAutomaticStart 属性为 TRUE。
- 2) PassiveTcpEstablishment 属性应该设置为 TRUE。

- 3) 如果支持 DampPeerOscillations 属性, DampPeerOscillations 应该设置为 FALSE。

#### 事件 6 : *AutomaticStart\_with\_DampPeerOscillations*

定义 : 本地系统自动启动 BGP peer 连接, 并使能 peer 冲突抑制。对持续的 peer 抑制的确切方法由实现决定, 并超出本文档的范围。

状态 : 可选, 具体取决于本地系统。

可选属性状态 :

- 1) 应该设置 AllowAutomaticStart 属性为 TRUE。
- 2) DampPeerOscillations 属性应该被设置为 TRUE。
- 3) PassiveTcpEstablishment 属性应该设置为 FALSE。

#### 事件 7 : *AutomaticStart\_with\_DampPeerOscillations\_and\_PassiveTcpEstablishment*

定义 : 本地系统自动启动 BGP peer 连接, 并使能 peer 振荡抑制和被动 TCP 建立。对持续的 peer 抑制的确切方法由实现决定, 并超出本文档的范围。

状态 : 可选, 具体取决于本地系统

可选属性状态 :

- 1) 应该设置 AllowAutomaticStart 属性为 TRUE。
- 2) DampPeerOscillations 属性应该被设置为 TRUE。
- 3) PassiveTcpEstablishment 属性应该被设置为 TRUE。

#### 事件 8 : *AutomaticStop*

定义 : 本地系统自动停止 BGP 连接。一个自动停止事件的示例是, 超过了给定 peer 的前缀数量, 本地系统自动断开与 peer 的连接。

状态 : 可选, 具体取决于本地系统

可选属性状态 :

- 1) AllowAutomaticStop 属性应该设置为 TRUE。

### 8.1.3 计时器事件

#### 事件 9 : *ConnectRetryTimer\_Expires*

定义 : 当 ConnectRetryTimer 到期生成的事件。

状态 : 强制

事件 10 : *HoldTimer\_Expires*

定义 : 当 HoldTimer 到期时生成的事件。

状态 : 强制

事件 11 : *KeepaliveTimer\_Expires*

定义 : 当 KeepaliveTimer 到期时生成的事件。

状态 : 强制

事件 12 : *DelayOpenTimer\_Expires*

定义 : DelayOpenTimer 到期时生成的事件。

状态: 可选

可选属性状态: 如果发生此事件,

- 1) DelayOpen 属性应该设置为 TRUE,
- 2) DelayOpenTime 属性应该被支持,
- 3) DelayOpenTimer 应该被支持。

事件 13 : *IdleHoldTimer\_Expires*

定义 : 当 IdleHoldTimer 到期时生成的事件, 表示 BGP 连接已经完成等待退避时间, 来阻止 BGP peer 振荡。

IdleHoldTimer 只用于, DampPeerOscillations 可选属性设置为 TRUE, 即持续 peer 振荡抑制功能使能时。

不支持持续 peer 振荡抑制功能的实现可能没有 IdleHoldTimer。

状态 : 可选

可选属性状态 : 如果发生此事件:

- 1) DampPeerOscillations 属性应该设置为 TRUE。
- 2) IdleHoldTimer 应该刚刚过期。

#### 8.1.4 基于 TCP 连接的事件

事件 14 : *TcpConnection\_Valid*

定义 : 表示本地系统接收具有有效源 IP 地址、TCP 端口、目标 IP 地址和 TCP 端口号的 TCP 连接请求。无效源 IP 地址和无效目标 IP 地址的定义由实现决定。

BGP 的目的端口应为端口 179, 由 IANA 定义。

Painting the night with sun, Lit in another time and place.

本地系统接收 TCP SYN 表示 TCP 连接请求。

状态： 可选

可选属性状态：

1) 如果发生此事件，TrackTcpState 属性应该设置为 TRUE。

#### 事件 15：*Tcp\_CR\_Invalid*

定义： 表示本地系统接收到具有无效源地址或端口号，无效目的地址或端口号的 TCP 连接请求。

BGP 目的端口号应为 179，由 IANA 定义。

TCP 连接请求发生在本地系统接收到 TCP SYN 时。

状态： 可选

可选属性状态：

1) 如果发生此事件，TrackTcpState 属性应设置为 TRUE。

#### 事件 16：*Tcp\_CR\_Acked*

定义： 表示本地系统请求与远程 peer 建立 TCP 连接。

本地系统的 TCP 连接发送 TCP SYN，接收到 TCP SYN / ACK 消息，并发送 TCP ACK。

状态： 强制

#### 事件 17：*TcpConnectionConfirmed*

定义： 表示本地系统收到远程站点 TCP 连接已经建立的确认消息。

远程 peer 的 TCP 引擎发送 TCP SYN。本地 peer 的 TCP 引擎发送了一个 SYN, ACK 消息，已经收到一个最后的 ACK。

状态： 强制

#### 事件 18：*TcpConnectionFails*

定义： 表示本地系统收到 TCP 连接失败的通知。

远程 BGP peer 的 TCP 引擎可能已经发送了一个 FIN。本地 peer 将以 FIN-ACK 进行响应。另一种可能性是本地 peer 在 TCP 指示连接超时，关闭连接。

状态： 强制

### 8.1.5 基于 BGP 消息的事件

#### 事件 19 : *BGPOpen*

定义 : 当有效的 OPEN 消息被收到时, 生成该事件。

状态 : 强制

可选属性状态 :

- 1) DelayOpen 可选属性应该被设置为 FALSE。
- 2) DelayOpenTimer 不应该运行。

#### 事件 20 : *DelayOpenTimer* 运行中的 *BGPOpen*

定义 : 当有效的 OPEN 消息已从建立传输连接 peer 接收成功, 而且目前正推迟发送 BGP OPEN 消息。

状态 : 可选

可选属性状态 :

- 1) DelayOpen 属性应该设置为 TRUE。
- 2) DelayOpenTimer 应该运行。

#### 事件 21 : *BGPHeaderErr*

定义 : 收到存在无效头部的 BGP 消息时, 会生成该事件。

状态 : 强制

#### 事件 22 : *BGPOpenMsgErr*

定义 : 收到错误的 OPEN 消息时生成该事件。

状态 : 强制

#### 事件 23 : *OpenCollisionDump*

定义 : 处理传入的 OPEN 消息时检测到连接冲突时产生此事件, 此连接已经选择断开连接。有关冲突检测的更多信息, 请参见第 6.8 节。

事件 23 是由管理行为生成的, 基于第 6.8 节中每个规则确定是否删除连接的逻辑。如果 FSM 以两台串联状态机实现的, 则这个事件可能会发生。

状态 : 可选

可选属性状态 : 如果处于 Established 状态的状态机要处理此事件,

- 1) CollisionDetectEstablishedState 可选属性应该设置为 TRUE。

Painting the night with sun, Lit in another time and place.

请注意：OpenCollisionDump 事件可能在 Idle, Connect, Active, OpenSent, 和 OpenConfirm 状态，而且没有设置任何可选属性。

*事件 24：NotifMsgVerErr*

定义：当收到“版本错误”的 NOTIFICATION 消息时产生事件。

状态：强制

*事件 25：NotifMsg*

定义：当收到非“版本错误”的错误 NOTIFICATION 消息时产生事件。

状态：强制

*事件 26：KeepAliveMsg*

定义：当接收到 KEEPALIVE 消息时，会生成该事件。

状态：强制

*事件 27：UpdateMsg*

定义：当收到有效的 UPDATE 消息时生成事件

状态：强制

*事件 28：UpdateMsgErr*

定义：收到无效 UPDATE 消息时会生成该事件。

状态：强制

## 8.2 FSM 介绍

### 8.2.1 FSM 定义

BGP 必须为配置的每个 peer 维护一个单独的 FSM。每个潜在连接的 BGP peer 对将尝试与彼此连接，除非配置为保持 IDLE 状态或配置为保持被动状态。为了探讨的目的，TCP 连接中主动或进行连接的一端（发送第一个 TCP SYN 包的 TCP 连接侧）被称为传出 /outgoing。被动或侦听端（第一个 SYN/ACK 的发送者）称为传入连接 /incoming connection。（有关主动或被动的术语信息，请参见第 8.2.1.1 节）

BGP 的实现必须在 TCP 端口 179 连接并侦听传入连接，除尝试连接到 peer 外。对于每个传入连接，一个状态机必须被实例化。传入连接端对于另一端 peer 身份的验证存在一个已知时段，但是 BGP 标识符不是已知。在这段时间内，传入和传出的连接可能存在配置相同的 peer。这被称为连接冲突（见第 6.8 节）。

每个配置的 BGP peer 的实现最多只能有一个 FSM，外加为每个传入的尚未被验证 peer 的 TCP 连接的一个 FSM。每个 FSM 恰好对应一个 TCP 连接。

如果连接被配置为使用一对不同的 IP 地址，一对 peer 间可能存在有多个连接。这被称为相同 peer 的多个“配置 peer”。

#### 8.2.1.1 术语“主动”和“被动”

主动和被动这对术语已经存在于互联网运营商中近十年了，并被证明是有用的。当应用于 TCP 连接和一个 peer 时，主动和被动的含义略有不同。按照上面的定义和下面的状态机，在任何一个 TCP 连接中只有一个主动端和一个被动端。当 BGP speaker 配置为主动时，最后的建立可能会在连接的主动方或被动方结束。一旦 TCP 连接已经完成了，哪一方是主动的，哪一方是被动的，都无关紧要。唯一的区别是 TCP 连接的哪一边有端口号 179。

#### 8.2.1.2 FSM 和冲突检测

每个 BGP 连接有一个 FSM。在确定连接相关联的 peer 之前，当连接冲突发生，一个 peer 可能有两个连接。连接冲突被解决（见第 6.8 节）之后，关闭的连接对应的 FSM 应该被处理。

#### 8.2.1.3 FSM 和可选会话属性

可选会话属性详细描述了属性作为标志（TRUE 或 FALSE）或可选定时器。对于作为标志的可选属性，如果可选会话属性在系统上被设置为 TRUE，相应的 BGP FSM 操作必须被支持的。例如，如果在 BGP 中可以设置以下选项实现：AutoStart 和 PassiveTcpEstablishment，然后必须支持事件 3，4 和 5。如果可选会话属性无法设置为 TRUE，支持该组选项的事件不必被支持。

每个可选定时器（DelayOpenTimer 和 IdleHoldTimer）都有一个一组属性是：

- 表示支持标志，
- 定时器中设置的时间
- 定时器。

两个可选定时器显示以下格式：

DelayOpenTimer : DelayOpen, DelayOpenTime, DelayOpenTimer

IdleHoldTimer : DampPeerOscillations, IdleHoldTime, IdleHoldTimer

如果可选的定时器（DelayOpen 或 DampPeerOscillations）支持标志（用于指示支持与否则的）无法设置为 TRUE，则计时器和事件不需要支持该选项。

#### 8.2.1.4 FSM 事件编号

在这状态机描述中使用的事件号（1-28）有助于表示 BGP 状态机的行为。这些号码应用实现可以用来提供网络管理信息。FSM 或 FSM 事件的确切形式具体实现具体分析。

#### 8.2.1.5 FSM 操作实现依赖

某些观点，BGP FSM 详细说明 BGP 初始化发生或者 BGP 资源的删除。BGP FSM 和相关资源初始化取决于 BGP 策略部分的实现。这些超出本文范围不再介绍。



### 8.2.2 有限状态机

#### *IDLE/空闲状态：*

最初，BGP peer FSM 处于空闲状态。后文，BGP peer FSM 将简称为 BGP FSM。

在这种状态下，BGP FSM 拒绝 peer 所有传入的 BGP 连接。没有资源分配给 peer。作为对一个 ManualStart 事件（Event 1）或一个 AutomaticStart 事件（Event3）的响应，本地系统：

- 初始化 peer 连接的所有 BGP 资源，
- 将 ConnectRetryCounter 设置为零，
- 使用初始值启动 ConnectRetryTimer，
- 发起到另一个 BGP 对等体的 TCP 连接，
- 侦听可能启动的远程 BGP peer 连接，
- 将其状态更改为 Connect。

在空闲状态下，ManualStop 事件（Event 2）和 AutomaticStop（Event 8）事件被忽略。

作为 ManualStart\_with\_PassiveTcpEstablishment 事件（事件 4）或 AutomaticStart\_with\_PassiveTcpEstablishment 事件（事件 5）的响应，本地系统：

- 初始化所有 BGP 资源，
- 将 ConnectRetryCounter 设置为零，
- 使用初始值启动 ConnectRetryTimer，
- 侦听可能启动的远程 peer 连接，
- 将其状态更改为“活动”。

ConnectRetryTimer 的确切值是 local/本地的，但它应该足够大，以允许 TCP 初始化。

如果 DampPeerOscillations 属性设置为 TRUE，则以下三个可能会发生在 IDLE 状态的额外事件：

- AutomaticStart\_with\_DampPeerOscillations（事件 6），
- AutomaticStart\_with\_DampPeerOscillations\_and\_PassiveTcpEstablishment（Event 7），
- IdleHoldTimer\_Expires（事件 13）。

在收到这 3 个事件后，本地系统将使用这些事件防止 peer 振荡。持续的 peer 振荡的预防方法超出了本文档的范围。

在空闲状态下收到的任何其他事件（事件 9-12, 15-28）不会导致本地系统状态的变化。

#### *CONNECT/连接状态：*

在这种状态下，BGP FSM 正在等待 TCP 完成连接。



在连接状态下忽略 START/开始事件（事件 1，3-7）。

为了响应 ManualStop 事件（事件 2），本地系统：

- 删除 TCP 连接，
- 释放所有 BGP 资源，
- 将 ConnectRetryCounter 设置为零，
- 停止 ConnectRetryTimer 并将 ConnectRetryTimer 设置为零，
- 将其状态更改为空闲。

响应 ConnectRetryTimer\_Expires 事件（事件 9），本地系统：

- 删除 TCP 连接，
- 重新启动 ConnectRetryTimer，
- 停止 DelayOpenTimer 并将定时器重置为零，
- 发起到另一个 BGP peer 的 TCP 连接，
- 继续监听远程 BGP peer 可能发起的连接，以及
- 保持在连接状态。

如果在 CONNECT 状态发生 DelayOpenTimer\_Expires 事件（事件 12），本地系统：

- 向对等体发送 OPEN 消息，
- 将 HoldTimer 设置为较大的值，
- 将其状态更改为

如果 BGP FSM 收到 TcpConnection\_Valid 事件（事件 14），TCP 连接被处理，并且连接保持在连接状态。

如果 BGP FSM 收到一个 Tcp\_CR\_Invalid 事件（事件 15），那么本地系统拒绝 TCP 连接和连接保持在连接状态。

如果 TCP 连接成功（事件 16 或事件 17），则本地系统在处理之前先检查 DelayOpen 属性。如果 DelayOpen 属性设置为 TRUE，本地系统：

- 停止 ConnectRetryTimer（如果运行）并设置 ConnectRetryTimer 为零，
- 将 DelayOpenTimer 设置为初始值，
- 保持在连接状态。

如果 DelayOpen 属性设置为 FALSE，则本地系统：

- 停止 ConnectRetryTimer（如果运行）并设置 ConnectRetryTimer 为零，
- 完成 BGP 初始化
- 向 peer 发送 OPEN 消息，
- 将 HoldTimer 设置为较大的值，
- 将其状态更改为 OpenSent。

HoldTimer 建议值为 4 分钟。

如果 TCP 连接失败（事件 18），本地系统检查 DelayOpenTimer。如果 DelayOpenTimer 正在运行，那么本地系统：

- 使用初始值重新启动 ConnectRetryTimer，
- 停止 DelayOpenTimer 并将其值重置为零，
- 继续收听远程 BGP peer 可能发起的连接，以及
- 将其状态更改为“活动”。

如果 DelayOpenTimer 未运行，则本地系统：

- 停止 ConnectRetryTimer 为零，
- 删除 TCP 连接，
- 释放所有的 BGP 资源，
- 将其状态更改为空闲。

如果在 DelayOpenTimer 运行时接收到 OPEN 消息（事件 20），本地系统：

- 停止 ConnectRetryTimer（如果运行）并设置 ConnectRetryTimer 为零，
- 完成 BGP 初始化，
- 停止并清除 DelayOpenTimer（将值设置为零），
- 发送 OPEN 消息，
- 发送 KEEPALIVE 消息，
- 如果 HoldTimer 初始值不为零，
- 启动具有初始值的 KeepaliveTimer
- 将 HoldTimer 重置为协商值，

否则，如果 HoldTimer 初始值为零，

- 重置 KeepaliveTimer 和
- 将 HoldTimer 值重置为零，
- 并将其状态更改为 OpenConfirm。

如果自治系统字段的值与本地自治系统号相同，将连接状态设置为内部连接，否则将是“外部”。

如果 BGP 消息头检查（Event 21）或 OPEN 消息检查检测到错误（事件 22）（见第 6.2 节），本地系统：

- （可选）如果 SendNOTIFICATIONwithoutOPEN 属性设置为 TRUE，则本地系统首先发送一个带有相应的错误代码 OTIFICATION 消息，然后
- 停止 ConnectRetryTimer（如果运行）并设置 ConnectRetryTimer 为零，
- 释放所有 BGP 资源，
- 删除 TCP 连接，
- 将 ConnectRetryCounter 递增 1，

- （可选）如果 DampPeerOscillations 属性设置为 TRUE，执行 peer 振荡抑制，和
- 将其状态更改为 IDLE。

如果收到带有版本错误（Event 24）的 NOTIFICATION 消息，本地系统检查 DelayOpenTimer。如果 DelayOpenTimer 正在运行，本地系统：

- 停止 ConnectRetryTimer（如果运行）并设置 ConnectRetryTimer 为零，
- 停止并重置 DelayOpenTimer（设置为零），
- 释放所有 BGP 资源，
- 删除 TCP 连接，和
- 将其状态更改为 IDLE。

如果 DelayOpenTimer 未运行，则本地系统：

- 停止 ConnectRetryTimer 并将 ConnectRetryTimer 设置为零，
- 释放所有 BGP 资源，
- 删除 TCP 连接，
- 将 ConnectRetryCounter 递增 1，
- 如果 DampPeerOscillations 属性设置为 True，则执行 peer 振荡抑制
- 将其状态更改为 IDLE。

对于任何其他事件（事件 8，10-11，13，19，23，25-28），本地系统：

- 如果 ConnectRetryTimer 正在运行，请停止并重置 ConnectRetryTimer（设置为零），
- 如果 DelayOpenTimer 运行，停止并重置 DelayOpenTimer（设置为零），
- 释放所有 BGP 资源，
- 删除 TCP 连接，
- 将 ConnectRetryCounter 递增 1，
- 如果 DampPeerOscillations 属性设置为 True，则执行 peer 振荡抑制
- 将其状态更改为 IDLE。

#### *ACTIVE/活动状态：*

在这种状态下，BGP FSM 正在尝试通过监听和接受 TCP 连接来获取 peer。

在 ACTIVE/活动状态下忽略 START/开始事件（事件 1，3-7）。

为了响应 ManualStop 事件（事件 2），本地系统：

- 如果 DelayOpenTimer 正在运行，且 SendNOTIFICATIONwithoutOPEN 会话属性被设置，本地系统发出 CEASE/停止 NOTIFICATION，
- 释放所有 BGP 资源，包括停止 DelayOpenTimer

- 删除 TCP 连接,
- 将 ConnectRetryCounter 设置为零,
- 停止 ConnectRetryTimer 并将 ConnectRetryTimer 设置为零,
- 将其状态更改为 IDLE。

为了响应 ConnectRetryTimer\_Expires 事件 (Event 9), 本地系统:

- 重新启动 ConnectRetryTimer (具有初始值),
- 发起到另一个 BGP peer 的 TCP 连接,
- 继续监听可能由远端 BGP peer 启动的 TCP 连接,
- 将其状态更改为 Connect。

如果本地系统收到一个 DelayOpenTimer\_Expires 事件 (Event 12), 本地系统:

- 将 ConnectRetryTimer 设置为零,
- 停止并清除 DelayOpenTimer (设置为零),
- 完成 BGP 初始化,
- 将 OPEN 消息发送到其远端 peer,
- 将 HOLD 定时器设置为一个较大的值,
- 将其状态更改为 OpenSent。

对于这种状态过渡, 建议 HoldTimer 值为 4 分钟。

如果本地系统收到一个 TcpConnection\_Valid 事件 (Event 14), 本地系统处理 TCP 连接标志, 并保留在 ACTIVE 状态。

如果本地系统收到一个 Tcp\_CR\_Invalid 事件 (事件 15), 本地系统拒绝 TCP 连接并停留在 ACTIVE 状态。

为了对 TCP 连接的成功 (事件 16 或事件 17) 做出响应, 本地系统在处理之前先检查 DelayOpen 可选属性。

如果 DelayOpen 属性设置为 TRUE, 则本地系统:

- 停止 ConnectRetryTimer 并设置 ConnectRetryTimer 为零,
- 将 DelayOpenTimer 设置为初始值 (DelayOpenTime),
- 保持在 ACTIVE 状态。

如果 DelayOpen 属性设置为 FALSE, 则本地系统:

- 将 ConnectRetryTimer 设置为零,
- 完成 BGP 初始化,
- 将 OPEN 消息发送给 peer,
- 将其 HoldTimer 设置为一个很大的值,
- 将其状态更改为 OpenSent。

建议 HoldTimer “大值” 为 4 分钟。

如果本地系统收到一个 TcpConnectionFails 事件 (Event 18)，本地系统：

- 重新启动 ConnectRetryTimer (具有初始值)，
- 停止并清除 DelayOpenTimer (将值设置为零)，
- 释放所有 BGP 资源，
- 将 ConnectRetryCounter 递增 1，
- 如果 DampPeerOscillations 属性设置为 TRUE，可选地执行 peer 振荡抑制，
- 将其状态更改为 IDLE。

如果接收到 OPEN 消息并且 DelayOpenTimer 正在运行 (事件 20)，本地系统：

- 停止 ConnectRetryTimer (如果运行) 并设置 ConnectRetryTimer 为零，
- 停止并清除 DelayOpenTimer (设置为零)，
- 完成 BGP 初始化，
- 发送 OPEN 消息，
- 发送 KEEPALIVE 消息，
- 如果 HoldTimer 值不为零，
  - 启动 KeepaliveTimer 到初始值，
  - 将 HoldTimer 重置为协商值，
- 否则如果 HoldTimer 为零
  - 重置 KeepaliveTimer (设置为零)，
  - 将 HoldTimer 重置为零，和
- 将其状态更改为 OpenConfirm。

如果 AS 字段的值与本地 AS 号相同，将连接状态设置为内部连接 否则会是外部的。

如果 BGP 消息头检查 (Event 21) 或 OPEN 消息检查检测到错误 (事件 22) (见第 6.2 节)，本地系统：

- (可选) 如果设置 SendNOTIFICATIONwithoutOPEN 属性为 TRUE，发送带有合适错误代码的 NOTIFICATION 消息，
- 将 ConnectRetryTimer 设置为零，
- 释放所有 BGP 资源，
- 删除 TCP 连接，
- 将 ConnectRetryCounter 递增 1，
- (可选) 如果 DampPeerOscillations 属性设置为 TRUE，执行 peer 振荡抑制，
- 将其状态更改为 IDLE。

如果收到带有版本错误 (Event 24) 的 NOTIFICATION 消息，本地系统检查 DelayOpenTimer。如果 DelayOpenTimer 正在运行，本地系统：

- 停止 ConnectRetryTimer (如果运行) 并设置 ConnectRetryTimer 为零，
- 停止并重置 DelayOpenTimer (设置为零)，
- 释放所有 BGP 资源，
- 删除 TCP 连接，

- 将其状态更改为 IDLE。

如果 DelayOpenTimer 未运行，则本地系统：

- 将 ConnectRetryTimer 设置为零，
- 释放所有 BGP 资源，
- 删除 TCP 连接，
- 将 ConnectRetryCounter 递增 1，
- （可选）如果 DampPeerOscillations 属性设置为 TRUE，执行 peer 振荡抑制，
- 将其状态更改为 IDLE。

为响应任何其他事件（活动 8，10-11，13，19，23，25-28），本地系统：

- 将 ConnectRetryTimer 设置为零，
- 释放所有 BGP 资源，
- 删除 TCP 连接，
- 将 ConnectRetryCounter 增加一个，
- （可选）如果 DampPeerOscillations 属性设置为 TRUE，执行 peer 振荡抑制，
- 将其状态更改为 IDLE。

#### *OpenSent 状态:*

在这种状态下，BGP FSM 等待来自其 peer 的 OPEN 消息。

START/开始事件（事件 1，3-7）在 OpenSent 状态中被忽略。

如果在 OpenSent 状态下发生 ManualStop 事件（事件 2），本地系统：

- 发出 CEASE NOTIFICATION 消息，
- 将 ConnectRetryTimer 设置为零，
- 释放所有 BGP 资源，
- 删除 TCP 连接，
- 将 ConnectRetryCounter 设置为零，以及
- 将其状态更改为 IDLE。

如果在 OpenSent 状态中发出 AutomaticStop 事件（Event 8），本地系统：

- 发出通知停止，
- 将 ConnectRetryTimer 设置为零，
- 释放所有的 BGP 资源，
- 删除 TCP 连接，
- 将 ConnectRetryCounter 递增 1，
- （可选）如果 DampPeerOscillations 属性设置为 TRUE，执行 peer 振荡抑制，
- 将其状态更改为 IDLE。

如果 HoldTimer\_Expires（Event 10），本地系统：

- 发送带有错误代码 HOLD 定时器过期的 NOTIFICATION 消息,
- 将 ConnectRetryTimer 设置为零,
- 释放所有 BGP 资源,
- 删除 TCP 连接,
- 增加 ConnectRetryCounter,
- (可选) 如果 DampPeerOscillations 属性设置为 TRUE, 执行 peer 振荡抑制,
- 将其状态更改为 IDLE。

如果 TcpConnection\_Valid (Event 14), Tcp\_CR\_Acked (Event 16) 或者接收到 TcpConnectionConfirmed 事件 (事件 17), 第二个 TCP 连接可能正在进行中。这个第二个 TCP 连接是追踪每个连接冲突的处理 (6.8 节), 直到接收到 OPEN 消息。

无效端口的 TCP 连接请求 (Tcp\_CR\_Invalid (事件 15)) 被忽略。

如果接收到 TcpConnectionFails 事件 (事件 18), 则本地系统:

- 关闭 BGP 连接,
- 重新启动 ConnectRetryTimer,
- 继续监听远程 BGP peer 可能发起的连接,
- 将其状态更改为 “ACTIVE”。

当接收到 OPEN 消息时, 将检查所有字段正确性。如果 OPEN 消息中没有错误 (事件 19), 本地系统:

- 将 DelayOpenTimer 重置为零,
- 将 BGP ConnectRetryTimer 设置为零,
- 发送一个 KEEPALIVE 消息,
- 设置一个 KeepaliveTimer (通过下面的文本)
- 根据协商值设置 HoldTimer (见第 4.2 节),
- 将其状态更改为 OpenConfirm。

如果协商的 HOLD 时间值为零, 则 HoldTimer 和 KeepaliveTimer 不启动 如果是 AS 字段的值与本地 AS 号相同, 那么连接是一个 “内部” 连接; 否则的话, 是一个 “外部” 连接。(这将影响下面描述的 UPDATE 处理。)

如果 BGP 消息头检查 (Event 21) 或 OPEN 消息检查检测到错误 (事件 22) (见第 6.2 节), 本地系统:

- 发送带有相应错误代码的 NOTIFICATION 消息,
- 将 ConnectRetryTimer 设置为零,
- 释放所有 BGP 资源,
- 删除 TCP 连接,
- 将 ConnectRetryCounter 递增 1,
- (可选) 如果 DampPeerOscillations 属性设置为 TRUE, 执行 peer 振荡抑制,



- 将其状态更改为 IDLE。

当接收到有效的 BGP OPEN 消息（事件 19 或事件 20）时，需要应用冲突检测机制（6.8 节）。有关详细的信息，请参见第 6.8 节。当 BGP 实现确定一个 CollisionDetectDump 事件发生时，对冲突检测超出本文档范围。

如果处于 OpenSent 状态下的连接，被确定为必须关闭的连接，发送 OpenCollisionDump（事件 23）信号给状态机。如果在 OpenSent 状态收到这样的事件，本地系统：

- 发出通知停止，
- 将 ConnectRetryTimer 设置为零，
- 释放所有 BGP 资源，
- 删除 TCP 连接，
- 将 ConnectRetryCounter 递增 1，
- （可选）如果 DampPeerOscillations 属性设置为 TRUE，执行 peer 振荡抑制，
- 将其状态更改为 IDLE。

如果收到带有版本错误（Event 24）的 NOTIFICATION 消息，本地系统：

- 将 ConnectRetryTimer 设置为零，
- 释放所有 BGP 资源，
- 删除 TCP 连接，
- 将其状态更改为 IDLE。

对于任何其他事件（事件 9，11-13，20，25-28）的响应，本地系统：

- 将通知发送到错误代码有限状态机错误，
- 将 ConnectRetryTimer 设置为零，
- 释放所有 BGP 资源，
- 删除 TCP 连接，
- 将 ConnectRetryCounter 递增 1，
- （可选）如果 DampPeerOscillations 属性设置为 TRUE，执行 peer 振荡抑制，
- 将其状态更改为 IDLE。

#### *OpenConfirm 状态:*

在这种状态下，BGP 等待 KEEPALIVE 或 NOTIFICATION 消息。

任何 START/开始事件（事件 1，3-7）在 OpenConfirm 状态中被忽略。

对于操作员发起的 ManualStop 事件（事件 2）响应，本地系统：

- 发送 CEASE NOTIFICATION 消息，
- 释放所有 BGP 资源，
- 删除 TCP 连接，

- 将 ConnectRetryCounter 设置为零,
- 将 ConnectRetryTimer 设置为零,
- 将其状态更改为 IDLE。

对于系统启动的 AutomaticStop 事件（事件 8），本地系统：

- 发送 CEASE NOTIFICATION 消息,
- 释放所有 BGP 资源,
- 删除 TCP 连接,
- 将 ConnectRetryCounter 递增 1,
- （可选）如果 DampPeerOscillations 属性设置为 TRUE，执行 peer 振荡抑制,
- 将其状态更改为 IDLE。

如果在收到 KEEPALIVE 消息之前发生 HoldTimer\_Expires 事件（Event 10），本地系统：

- 发送错误代码为 HOLD 定时器过期的 NOTIFICATION 消息,
- 将 ConnectRetryTimer 设置为零,
- 释放所有 BGP 资源,
- 删除 TCP 连接,
- 将 ConnectRetryCounter 递增 1,
- （可选）如果 DampPeerOscillations 属性设置为 TRUE，执行 peer 振荡抑制,
- 将其状态更改为 IDLE。

如果本地系统收到一个 KeepaliveTimer\_Expires 事件（Event 11），本地系统：

- 发送 KEEPALIVE 消息,
- 重新启动 KeepaliveTimer,
- 保持 OpenConfirmed 状态。

在 TcpConnection\_Valid 事件（事件 14）的情况下，或者在 OpenConfirm 状态 TCP 连接成功（事件 16 或事件 17），本地系统需要跟踪第二个连接。

如果 TCP 连接尝试使用无效端口（事件 15），本地系统将忽略第二次连接尝试。

如果本地系统从相关的 TCP 或 NOTIFICATION 消息（事件 25）收到一个 TcpConnectionFails 事件（事件 18），本地系统：

- 将 ConnectRetryTimer 设置为零,
- 释放所有 BGP 资源,
- 删除 TCP 连接,
- 将 ConnectRetryCounter 递增 1,
- （可选）如果 DampPeerOscillations 属性设置为 TRUE，执行 peer 振荡抑制,
- 将其状态更改为 IDLE。

如果本地系统收到带有版本错误 (NotifMsgVerErr (Event 24)) 的 NOTIFICATION 消息, 本地系统:

- 将 ConnectRetryTimer 设置为零,
- 释放所有 BGP 资源,
- 删除 TCP 连接,
- 将其状态更改为 IDLE。

如果本地系统收到一个有效的 OPEN 消息 (BGPOpen (Event 19)), 根据 6.8 节内容描述执行连接冲突检测功能。如果由于连接冲突而导致该连接掉线, 本地系统:

- 发出通知停止,
- 将 ConnectRetryTimer 设置为零,
- 释放所有 BGP 资源,
- 丢弃 TCP 连接 (发送 TCP FIN),
- 将 ConnectRetryCounter 递增 1,
- (可选) 如果 DampPeerOscillations 属性设置为 TRUE, 执行 peer 振荡抑制,
- 将其状态更改为 IDLE。

如果接收到 OPEN 消息, 则检查所有字段正确性。如果 BGP 消息头检查 (BGPHeaderErr (事件 21)) 或 OPEN 消息检查检测到错误 (参见 6.2 节) (BGPOpenMsgErr (Event 22)), 本地系统:

- 发送带有相应错误代码的 NOTIFICATION 消息,
- 将 ConnectRetryTimer 设置为零,
- 释放所有 BGP 资源,
- 删除 TCP 连接,
- 将 ConnectRetryCounter 递增 1,
- (可选) 如果 DampPeerOscillations 属性设置为 TRUE, 执行 peer 振荡抑制,
- 将其状态更改为 IDLE。

在处理另一个 OPEN 消息时, 如果 BGP 实现确定 (通过本文档范围以外的手段) 发生连接冲突, 此连接将被关闭, 本地系统会发出一个 OpenCollisionDump 事件 (事件 23)。本地系统收到 OpenCollisionDump 事件 (事件 23), 本地系统:

- 发出 CEASE NOTIFICATION 消息,
- 将 ConnectRetryTimer 设置为零,
- 释放所有 BGP 资源
- 删除 TCP 连接,
- 将 ConnectRetryCounter 递增 1,
- (可选) 如果 DampPeerOscillations 属性设置为 TRUE, 执行 peer 振荡抑制,
- 将其状态更改为 IDLE。

如果本地系统收到 KEEPALIVE 消息 (KeepAliveMsg (事件 26)), 本地系统:

- 重新启动 HoldTimer,

- 将其状态更改为 Established。

对于任何其他事件（事件 9，12-13，20，27-28）的响应，本地系统：

- 发送带有 FSM 错误代码的 NOTIFICATION 消息，
- 将 ConnectRetryTimer 设置为零，
- 释放所有 BGP 资源，
- 删除 TCP 连接，
- 将 ConnectRetryCounter 递增 1，

#### *Established 状态:*

在建立 Established 状态下，BGP FSM 可以与 peer 交换 UPDATE，NOTIFICATION 和 KEEPALIVE 消息。

任何 START/开始事件（事件 1，3-7）在 Established 状态中被忽略。

为了响应 ManualStop 事件（由操作员启动）（事件 2），本地系统：

- 发送 CEASE NOTIFICATION 消息，
- 将 ConnectRetryTimer 设置为零，
- 删除与此连接相关联的所有路由，
- 释放 BGP 资源，
- 删除 TCP 连接，
- 将 ConnectRetryCounter 设置为零，
- 将其状态更改为 IDLE。

响应 AutomaticStop 事件（Event 8），本地系统：

- 发送 CEASE NOTIFICATION 消息，
- 将 ConnectRetryTimer 设置为零，
- 删除与此连接相关联的所有路由，
- 释放所有 BGP 资源，
- 删除 TCP 连接，
- 将 ConnectRetryCounter 递增 1，
- （可选）如果 DampPeerOscillations 属性设置为 TRUE，执行 peer 振荡抑制，
- 将其状态更改为 IDLE。

AutomaticStop 事件的一个原因是：BGP 接收到 UPDATE 消息，其携带给定 peer 的多个前缀，这些收到的总前缀超过配置的前缀的最大数量。本地系统自动断开 peer 的连接。

如果发生 HoldTimer\_Expires 事件（事件 10），则本地系统：

- 发送一个错误代码为 HOLD 定时器到期 NOTIFICATION 消息，
- 将 ConnectRetryTimer 设置为零，
- 释放所有 BGP 资源，
- 删除 TCP 连接，
- 将 ConnectRetryCounter 递增 1，

- （可选）如果 DampPeerOscillations 属性设置为 TRUE，执行 peer 振荡抑制，
- 将其状态更改为 IDLE。

如果发生 KeepaliveTimer\_Expires 事件（事件 11），则本地系统：

- 发送一个 KEEPALIVE 消息，
- 重新启动其 KeepaliveTimer，除非协商的 HoldTime 值为零。

每次本地系统发送一个 KEEPALIVE 或 UPDATE 消息，它重新启动其 KeepaliveTimer，除非协商的 HoldTime 值是零。

将接收到一个有效端口的 TcpConnection\_Valid (Event 14) 导致第二个连接被跟踪。

TCP 连接无效 (Tcp\_CR\_Invalid 事件 (事件 15)) 将会被忽略。

作为 TCP 连接的指示成功建立 (事件 16 或事件 17) 的响应，第二个连接应该被跟踪，直到它发送一个 OPEN 消息。

如果接收到有效的 OPEN 消息 (BGPOpen (Event 19))，并且如果设置可选属性 CollisionDetectEstablishedState 为 TRUE，OPEN 消息将被检查，看看它是否与任何其他连接冲突 (6.8 节)。如果 BGP 实现确定此连接需要终止，它会进行 OpenCollisionDump 事件 (事件 23) 处理。如果这个连接需要终止，本地系统：

- 发出 CEASE NOTIFICATION 消息，
- 将 ConnectRetryTimer 设置为零，
- 删除与此连接相关联的所有路由，
- 释放所有 BGP 资源，
- 删除 TCP 连接，
- 将 ConnectRetryCounter 递增 1，
- （可选）如果 DampPeerOscillations 属性设置为 TRUE，执行 peer 振荡抑制，
- 将其状态更改为 IDLE。

如果本地系统从相关的 TCP 收到 NOTIFICATION 消息 (事件 24 或事件 25) 或 TcpConnectionFails (事件 18)，本地系统：

- 将 ConnectRetryTimer 设置为零，
- 删除与此连接相关联的所有路由，
- 释放所有的 BGP 资源，
- 删除 TCP 连接，
- 将 ConnectRetryCounter 递增 1，
- 将其状态更改为空闲。

如果本地系统收到一个 KEEPALIVE 消息 (事件 26)，则本地系统：

- 如果协商的 HoldTime 值是非零值，则重新启动其 HoldTimer，
- 保持 Established 状态。

如果本地系统收到一个 UPDATE 消息 (Event 27)，那么本地系统：

– 处理消息，

- 如果协商的 HoldTime 值是非零值，则重新启动其 HoldTimer，
- 保持 Established 状态。

如果本地系统收到 UPDATE 消息，和 UPDATE 消息处理过程（见 6.3 节）检测到错误（事件 28），本地系统：

- 发送带有 UPDATE 错误的 NOTIFICATION 消息，
- 将 ConnectRetryTimer 设置为零，
- 删除与此连接相关联的所有路由，
- 释放所有 BGP 资源，
- 删除 TCP 连接，
- 将 ConnectRetryCounter 递增 1，
- （可选）如果 DampPeerOscillations 属性设置为 TRUE，执行 peer 振荡抑制，
- 将其状态更改为 IDLE。

作为任何其他事件（事件 9，12-13，20-22）响应，本地系统：

- 发送带有错误代码为 FSM 错误的 NOTIFICATION 消息，
- 删除与此连接相关联的所有路由，
- 将 ConnectRetryTimer 设置为零，
- 释放所有 BGP 资源，
- 删除 TCP 连接，
- 将 ConnectRetryCounter 递增 1，
- （可选）如果 DampPeerOscillations 属性设置为 TRUE，执行 peer 振荡抑制，
- 将其状态更改为 IDLE。

## 9. UPDATE 消息处理

只有在 ESTABLISHED 状态下才可以接收 UPDATE 消息。在任何其他状态下接收 UPDATE 消息都是错误的。当一个 UPDATE 消息被接收，检查每个字段的有效性，像 6.3 节中规定。

如果可选的非传递属性无法识别，那么它默默忽略，如果可选的传递属性无法识别，属性标志字节的局部/Partial 位（第三高位）设置为 1，并保留属性传播到其他 BGP speakers。

如果可选属性被识别并具有有效值，那么根据可选属性的类型，它被本地处理、保留和更新，在必要时可以传播到其他 BGP speakers。

如果 UPDATE 消息包含非空的 WITHDRAWN ROUTES 字段，以前通告的路由的目的地（表示为 IP 前缀）包含在此字段中，应从 Adj-RIB-IN 中删除。此 BGP speaker 将运行其决策处理，因为以前发布的路由不再可用。

如果 UPDATE 消息包含可行路由，则 Adj-RIB-In 将会使用此路由做如下更新：如果 NLRI 的新路由与当前存储在 Adj-RIB-In 的某个路由相同，那么新路由将替换 Adj-RIB-In

中的老路由，这样隐式地撤销服务中的旧路由。否则，如果 Adj-RIB-In 没有与 NLRI 中新路由相同的路由，新路由应放置在 Adj-RIB-In 中。

一旦 BGP 扬声器更新了 Adj-RIB-In，speaker 就会运行其决策过程。

## 9.1 决策过程

决策过程为后续宣告选择路由，其通过对其存储在其 Adj-RIBs-In 中的路由应用本地策略信息库（PIB）的策略来选择路由。决策过程的输出是将向 peers 通告的路由集合；根据策略，该选定的路由将存储在本地 speaker 的 Adj-RIBs-Out 中。

这里描述的 BGP 决策过程是概念性的，而不必按照描述精确地实现，只要支持所描述的功能的实现并且它们对外部展示可见行为是相同的。

选择过程通过定义所需的函数来正式化，给定路由的属性作为其参数，并返回（a）一个非负整数表示路由的偏好程度，或（b）表示安装在 Loc-RIB 中不符合条件值的路由，并将其从下一阶段路由选择排除。

计算给定路由偏好程度的函数不得使用以下任何一个作为其输入：现存的其他路由，其他不存在的路由或其他路由的路径属性。那么，路由选择由每个可行路由的偏好度函数的单项应用组成，紧接着选择最高偏好度的那个。

对 Adj-RIBs-In 中包含的路由进行决策过程操作，并负责：

- 选择 speaker 本地使用的路由
- 选择要发布给其他 BGP speakers 的路由
- 路由聚合和减少路由信息

决策过程分三个阶段进行，由不同的事件触发：

- a) 第一阶段负责计算从 peer 接收的每个路由的偏好程度。
- b) 第二阶段在第一阶段完成时被调用。负责为每个不同的目的地从所有可用的路由中选择最佳路线，并安装每个选定的路由到 Loc-RIB。
- c) 第三阶段在 Loc-RIB 被修改之后被调用。它根据 PIB 中策略，负责将 Loc-RIB 中的路由公告给每个 peer。在这个阶段，路由聚合和路由信息减少可以可选地进行。

### 9.1.1 阶段 1：计算偏好度

无论本地 BGP speaker 何时从 peer 收到公告新路由、替代路由、或撤销路由的 UPDATE 消息，都会调用第一阶段的决策函数。

第一阶段的决策函数是一个分离的过程，当它没有进一步的工作要做时，完成处理。

第一阶段的决策函数锁定了一个 Adj-RIB-In 在操作其中包含的任何路由之前，并在操作完其中包含的所有新的或不可行的路由后解锁。

对于每个新接收或替换的可行路由，本地 BGP speaker 确定一个喜好程度如下：



如果从内部 peer 学到路由，则将 LOCAL\_PREF 属性的值作为偏好度，或者本地系统基于预配置的策略信息计算路由的优先级。注意后者可能导致形成持久路由环路。

如果从外部 peer 学到路由，则本地 BGP speaker 基于预先配置的策略信息来计算偏好度。如果返回值表示路由是不合格的，路由可能不会作为下一阶段路由选择的输入；否则返回值必须作为任何 IBGP 重新宣告中的 LOCAL\_PREF 值。

这个策略信息的确切性质和涉及计算，是一个本地的事情。

### 9.1.2 阶段 2：选路

第一阶段决策函数在第一阶段完成时被调用。第二阶段函数是一个分离的过程，当它没有进一步的工作要做时，完成处理。第二阶段的处理考虑 Adj-RIBs-In 中所有符合资格的路由。

当第三阶段的决策函数运行时，第二阶段的决策函数被阻塞无法运行。第二阶段函数开始其功能前，锁定所有 Adj-RIBs-In，并在完成后解锁它们。

如果 BGP 路由的 NEXT\_HOP 属性描述了一个未解决的地址，或者如果路由安装在路由表中的路由是未解决的，必须从第二阶段决策函数中移除该 BGP 路由。

如果 BGP 路由的 AS\_PATH 属性包含 AS 环，应该从第二阶段决策函数中移除该 BGP 路由。通过扫描完整的 AS 路径来检测 AS 环（如 AS\_PATH 属性），并检查本地系统的 AS 号不会出现在 AS 路径中。操作 BGP speaker 接受具有自己 AS 路径的路由配置，不在本文档介绍的范围之内。

在一个 AS 中的 BGP speaker 不对路由选择进行冲突检测是危险的，这会导致转发环路发生。

对于 Adj-RIBs-In 中每组目的地的可行路由，本地 BGP speaker 识别具有以下功能的路由：

- a) 到同一组目的地的任何路由的最高偏好度，或
- b) 是到目的地的唯一路由，或
- c) 作为打破在 9.1.2.2 节中规定的规则的第二阶段结果，被选择。

然后本地 speaker 应该将该路由安装在 Loc-RIB 中，替换 Loc-RIB 中同一目的地任何已经存在的路由。当新的 BGP 路由被安装在路由表，必须注意确保同一个目的地现有路由现在被认为是无效的，将被从路由表中删除。新 BGP 路由是否替换现有路由表中的非 BGP 路由取决于 BGP speaker 上配置的策略。

本地 speaker 必须从所选路由的 NEXT\_HOP 属性确定直接下一跳地址（见第 5.1.3 节）。如果 NEXT\_HOP 的直接下一跳或 IGP 代价（其中 NEXT\_HOP 通过 IGP 路由解决）发生更改，第二阶段的路由选择必须被再次执行。

请注意，虽然 BGP 路由不是必须安装具有直接下一跳的路由表，实现时必须注意，通过 BGP 路由转发任何报文之前，其关联的 NEXT\_HOP 地址被解析为直接（直接连接）下一跳地址，并且该地址（或多个地址）最终用于实际的报文转发。

未解决的路由应从 Loc-RIB 和路由表中删除。但是，应该保留 Adj-RIBs-In 中相应的未解决的路由（万一它们变得可解析）。

### 9.1.2.1 路由解决条件

如第 9.1.2 节所述，BGP speaker 应该移除第二阶段决定中无法解决的路由。这确保只有有效的路由安装在 Loc-RIB 和路由表中。

路由可解析性条件定义如下：

- 1) 如果路由表包含至少有一个与 Rte1 相匹配的可解析的路由 Rte2，其不是（直接或间接）通过 Rte1 递归解析出的。如果多个匹配路线可用，只有最长的匹配的路由应该被考虑。
- 2) 如果被关联的接口状态为 UP，并且如果在这个接口启用了 IP 处理，接口关联的路由（有或没有中间地址）被认为是可解析的。

BGP 路由与接口不相关，但可以通过路由表中的两种类型的路由来解析（那些指定接口或不指定接口）。IGP 路由和到直连网络路由被期待用于指定出接口。静态路由可以指定出接口，中继地址，或两者。

请注意，在 BGP speaker 的路由表不包含，与 BGP 匹配的路由的 NEXT\_HOP 的情况下，BGP 路由被认为是无法解析的。相互递归路由（路由相互解析或自己解决）也无法进行可解析性检查。

同样重要的是，如果安装路由表中存在将变得无法解析路由，实现时不应该考虑此可行路由，即使它们的 NEXT\_HOP 通过路由表的当前内容是可解析的（一个例子：相互递归的路由）。此检查可以确保 BGP speaker 不会在路由表中安装将被移除和未被 speaker 使用的路由。因此，除了提升本地路由表的稳定性，这个检查也改善了网络中协议的行为。

每当 BGP speaker 识别出路由可解析性检查因为相互递归失败时，出现错误信息应该被记录。

### 9.1.2.2 Tie-Break (阶段 2)

在其 Adj-RIBs-In 中，BGP speaker 可能有几条到同一个目的地且具有相同偏好度的路由。本地的 speaker 只能在相关的 Loc-RIB 中的这些路由中选择一个。本地 speaker 会考虑所有同等偏好度的路由，包括从内部 peer 收到的和从外部 peer 接收的。

以下 Tie-break 规程假定，对于每个候选路由，自治系统内的所有 BGP speaker 都可以确定一条到路由中的 NEXT\_HOP 属性描绘的地址的路线（内部距离）的成本，并遵循相同的路由选择算法。

Tie-break 算法首先关注到同一目的地一切优选路由，然后选择路由并从关注的路由中删除。算法终止时只关注一条路线。该标准必须是按照指定的顺序应用。

使用伪代码描述了几个标准。注意为了清楚起见，选择了所示的伪代码，而不是为了效率。其并不打算指定任何特定的实现。BGP 实现可以使用产生如这里描述的那样相同结果的任何算法。

- a) 从考虑的路由中，删除不具备最小的 AS 数量 (AS\_PATH 属性中) 的路由。注意，当计算这个数量时，AS\_SET 计数为 1，无论集合中有多少个 AS。
- b) 从考虑的路由中，删除不具备最小 Origin 值 (Origin 属性中) 的路由。
- c) 从考虑的路由中，删除次优选 MULTI\_EXIT\_DISC 的路由。MULTI\_EXIT\_DISC 只能比较从相邻邻居 AS (相邻 AS 由 AS\_PATH 属性确定) 学习到的路由。没有 MULTI\_EXIT\_DISC 属性的路由是被认为具有最低可能的 MULTI\_EXIT\_DISC 值。

也可以使用以下过程中描述：

```
for m = all routes still under consideration
  for n = all routes still under consideration
    if (neighborAS(m) == neighborAS(n)) and (MED(n) < MED(m))
      remove route m from consideration
```

在上面的伪代码中，MED (n) 是一个函数，其返回值为路由 n 的 MULTI\_EXIT\_DISC 属性值。如果路由没有 MULTI\_EXIT\_DISC 属性，函数返回最低可能的 MULTI\_EXIT\_DISC 值 (即 0 值)。

类似地，neighborAS (n) 是一个函数，其返回值为收到路由的邻居 AS 号。如果路由是通过 IBGP 学习且另一个 IBGP speaker 不是路由的起源地，它是另一个 IBGP speaker 从邻居 AS 学习路由。如果路由是通过 IBGP 学习路由，和另一个 IBGP speaker (a) 产生路由，或者 (b) 通过聚合创建路由，聚合路由的 AS\_PATH 属性是空的或以一个 AS\_SET 开头，其是本地 AS。

如果在向 IBGP 重新宣告一个路由之前，就删除了其 MULTI\_EXIT\_DISC 属性，然后根据接收到的 EBGp 的 MULTI\_EXIT\_DISC 属性进行比较的方式仍然可以执行。如果实现选择删除 MULTI\_EXIT\_DISC，然后可选的基于 MULTI\_EXIT\_DISC 比较，如果执行，必须是仅在 EBGp 学习的路由中执行。在移除 MULTI\_EXIT\_DISC 属性后，最好的 EBGp 学习的路由可能与 IBGP 学习的路由进行比较。如果从 EBGp 学习的路由的子集中删除 MULTI\_EXIT\_DISC，选择没有移除 MULTI\_EXIT\_DISC 的“最佳”EBGP 学习路由，则 MULTI\_EXIT\_DISC 必须用于与 IBGP 学习路由的比较。对于 IBGP 学习的路由，MULTI\_EXIT\_DISC 必须在路由比较中使用，在决策过程中进行到这一步骤时。包含 MULTI\_EXIT\_DISC 属性的从 EBGp 学习的路由与 IBGP 学习路由进行比较，然后删除 MULTI\_EXIT\_DISC 属性，并且宣告已经被证明导致路由环路的路由。

- d) 如果至少一个候选路由是通过 EBGp 收到的，从关注的路由中移除所有从 IBGP 收到的。
- e) 从关注的路由中移除次优选的内部 cost 的路由。路线的内部 cost 取决于路由表中用于计算路由 NEXT\_HOP 的度量。如果路由的 NEXT\_HOP 可达，但不能确定 cost，那么这一步应该被跳过 (相当于考虑所有路线具有相等的成本)。

也可以使用以下过程中描述：

```
for m = all routes still under consideration
  for n = all routes in still under consideration
    if (cost (n) is lower than cost (m) )
      remove m from consideration
```

在上面的伪代码中，cost (n) 是一个函数，其返回值为到给定地址的路径（内部距离）cost，给定地址在路由的 NEXT\_HOP 属性中。

- f) 移除所有关注的路由，除了最小的 BGP ID 的 BGP speaker 发布的路由。
- g) 优选从最小 peer 地址接收到的路由。

### 9.1.3 阶段 3：路由传播

第三阶段决定函数在第二阶段完成时被调用，或者发生以下任何事件时：

- a) 当 Loc-RIB 中到本地目的地的路由发生变化时
- b) 通过 BGP 之外的手段学习的本地生成的路由已改变
- c) 建立新的 BGP speaker 连接时

第三阶段功能是一个单独的过程，当它没有进一步的工作要做，完成处理。在第二阶段决策功能处理时，第三阶段路由决策功能被阻止。

Loc-RIB 中的所有路由按照配置策略处理后送到 Adj-RIBs-Out。此策略可以拒绝把 Loc-RIB 中的路由安装在特定的 Adj-RIB-Out。路由不应该被安装在 Adj-Rib-Out 中，除非路由描述的目的地和 NEXT\_HOP 可以被路由表用于转发的。如果 Loc-RIB 中的路由被排除在特定的 Adj-RIB-Out 之外，Adj-RIB-Out 中先前通告的路由必须通过 UPDATE 消息从服务中撤销（见 9.2）。

路由聚合和信息减少技术（参见第 9.2.2.1）。

任何导致路由添加到 Adj-RIB-Out，但不会添加到本地 BGP speaker 的转发表中的本地策略，超出了本文档的范围。

当 Adj-RIBs-Out 和路由表更新完成后，本地 BGP speaker 运行 9.2 的 Update-Send 处理。

### 9.1.4 重叠路由

BGP speaker 可以传送具有重叠网络层可达信息（NLRI）的路由到另一个 BGP speaker。当多条不一致路由识别一组目的地时，NLRI 重叠会发生。因为 BGP 使用 IP 前缀编码 NLRI，所以重叠将始终展示子集关系。一条描述较小的目的地集合（较长的前缀）的路由，一般来说比描述更大的目的地集合（较短的前缀）的路由更特殊；类似地，描述较大集合的路由，比描述的更小目的地集合路由的具有更少的特殊性。

优先关系有效地分解不具体路由为两部分：

- 仅由较笼统的路由描述的一组目的地，
- 由较笼统和更具体的重叠路由描述的一组目的地

由重叠描述的目的地集合表示一部分非特定的路由是可行的，但目前不使用。如果一个更具体的路线后来被撤销，那么由重叠描述的笼统的目的地集合仍然可以使用。

如果 BGP speaker 收到重叠的路由，则决策处理必须基于配置接受策略考虑两条路由。如果一个笼统和一个具体的路由都被接受，那么决策过程必须在 Loc-RIB 中安装笼统和具体的路由，或聚合两条路由并安装聚合路由在 Loc-RIB 中，倘若两条路由都有相同的 NEXT\_HOP 属性值。

如果一个 BGP speaker 选择聚合，那么它应该在 AS\_SET 中包含所有用于聚合 ASes，或添加 ATOMIC\_AGGREGATE 属性到路由。此属性现在主要是信息。随着不支持无类路由 IP 路由协议的消失，并且不支持无类路由实现的路由器和主机消失，不再需要解聚合。路由应该不能去聚合。特别是一条载有 ATOMIC\_AGGREGATE 属性的路由绝不能解聚合。即这条路由的 NLRI 不能更具体化。沿着这样的方向转发路由不能保证 IP 包，实际上只穿越路由 AS\_PATH 属性中列出的 ASes。

## 9.2 UPDATE 发送过程

UPDATE 发送过程负责宣告 UPDATE 消息给所有 peers。例如，它分发通过决策过程选出的路由到其他 BGP speakers，其可以位于同一个自治系统或相邻的自治系统中。

当 BGP Speaker 接收到内部 peer 的 UPDATE 消息时，接收 BGP speaker 不应重新分发该 UPDATE 消息中包含的路由信息给其他内部 peer（除非 speaker 充当 BGP 路由反射器 [RFC2796]）。

作为路由选择过程的第三阶段的一部分，BGP speaker 已更新其 Adj-RIBs-Out。所有新安装的路由和全部没有替代路由的新的不可行路由，通过 UPDATE 消息通告给其 peers。

BGP speaker 不应该通告其 Adj-RIB-Out 给定的可行 BGP 路由，如果它会产生一个包含在以前发布的 UPDATE 消息中相同的 BGP 路由。

Loc-RIB 中标记为不可行的任何路由都应被移除。自己 AS 内可达目的地改变也应该在 UPDATE 消息中通告。

如果单个路由不能适配 UPDATE 消息，由于对 UPDATE 消息的最大长度的限制（见第 4 节），BGP speaker 不得向其 peers 发布路由，可以选择在本地记录错误。

### 9.2.1 控制路由业务开销

BGP 协议限制路由流量（即 UPDATE 消息），理解 UPDATE 中包含的信息，以便限制发布 UPDATE 消息所需的链路带宽和决策过程所需的处理能力。

#### 9.2.1.1 路由宣告频率

参数 MinRouteAdvertisementIntervalTimer 确定了 BGP speaker 宣告和/或撤销到特定目的地的路由给 peer，必须经过的最短间隔时间。该速率限制规程基于每个目的地应用，虽然 MinRouteAdvertisementIntervalTimer 的值基于每个 BGP peer 设置。

由 BGP speaker 发送给 peer 的，宣告可行路由和/或撤销不可行路由到共同目的地集合的两个 UPDATE 消息，必须至少间隔 MinRouteAdvertisementIntervalTimer。这只能通过为每个共同的目的地集合保留一个单独的定时器。这个将是没有必要的开销。任何技术



都要确保由 BGP speaker 发送给 peer 的，宣告可行路由和/或撤销不可行路由到共同目的地集合的两个 UPDATE 消息，至少间隔 MinRouteAdvertisementIntervalTimer，并且还要确保恒定上限的可接受的间隔。

由于 AS 中需要快速收敛 (a) MinRouteAdvertisementIntervalTimer 用于内部 peer 的应该比用于外部 peer 的短，或 (b) 本节中描述的处理不适用于发送给内部 peer 的路由。

此规程不限制路由选择的速率，而是仅限于路由宣告的速率。如果新路由在 MinRouteAdvertisementIntervalTimer 等待到期期间多次被选择，最后选择的路由应该在 MinRouteAdvertisementIntervalTimer 到期后宣告。

### 9.2.1.2 路由起始频率

参数 MinASOriginationIntervalTimer 确定了，在 BGP speaker 本身的 AS 中，报告变化的 UPDATE 消息的连续通告的最小间隔。

## 9.2.2 路由信息的有效组织

对于 BGP speaker 将通告的已经选择的路由信息，一个 BGP speaker 可能会利用几种方法来有效的方式组织信息。

### 9.2.2.1 信息简约

信息简约可能意味着策略控制粒度降低 - 信息崩溃后，相同的策略会应用于所有等价类中的目的地和路径。

决策过程可以可选地减少置于 Adj-RIBs-Out 中的信息量，通过以下任何一项方式：

#### a) 网络层可达性信息 (NLRI)：

目的 IP 地址可以表示为 IP 地址前缀。在这种情况下，地址结构和 AS 管理员控制的系统之间有一个对应关系，可以减少在 UPDATE 消息中携带的 NLRI 大小。

#### b) AS\_PATHs：

AS 路径信息可以表示为有序的 AS\_SEQUENCE 或无序 AS\_SET。在 9.2.2.2 节中描述的路由聚合算法中使用 AS\_SET。他们通过仅对每个 AS 号码列出一次来减小 AS\_PATH 信息的大小，不管它出现在聚合的可能多少次多个 AS\_PATH 中。

AS\_SET 意味着 NLRI 中列出的目的地，至少可以通过穿越部分 AS 组成的路径可达。AS\_SET 提供足够的信息来避免路由信息环路；然而，它们的使用可能删除潜在可行的路径，因为这样路径不再以 AS\_SEQUENCEs 形式单独列出。实际上这不是一个问题，因为一旦一个 IP 包到达一组 AS 的边缘，BGP speaker 可能会有更多的详细的路径信息，可以根据目的地来区分各个路径。

### 9.2.2.2 聚合路由信息

聚合是将几条不同特征的路由结合的过程，并以一条路由的方式进行宣告。聚合可以作为决策过程的一部分，来减少放置在 Adj-RIBs-Out 中的路由信息的数量。

聚合减少 BGP speaker 必须存储和与其它 BGP speaker 交换的信息量。路由可以通过应用以下过程来分别聚合相同类型路径属性和网络层可达性信息。

具有不同 MULTI\_EXIT\_DISC 属性的路由不应该被聚合。

如果聚合路由使用 AS\_SET 作为其 AS\_PATH 属性的第一个元素，然后发起路由的路由器不应该通告这条路由的 MULTI\_EXIT\_DISC 属性。

具有不同类型代码的路径属性无法聚合在一起。相同类型代码的路径属性可以被聚合，根据以下规则：

NEXT\_HOP：

聚合具有不同 NEXT\_HOP 属性的路由时，聚合路由的 NEXT\_HOP 属性应该识别 BGP speaker 上执行聚合的接口。

ORIGIN 属性：

如果聚合的路由中至少有一条路由的 ORIGIN 值为 INCOMPLETE，则聚合路由的 ORIGIN 属性值必须为 INCOMPLETE。否则，如果聚合路由中至少有一条路由的 ORIGIN 值为 EGP，则聚合路由的 ORIGIN 属性值必须为 EGP。在其他所有情况下，聚合路由的 ORIGIN 属性的值是 IGP。

AS\_PATH 属性：

如果要聚合的路由具有相同的 AS\_PATH 属性，那么聚合路由的 AS\_PATH 属性与每条路由的 AS\_PATH 属性相同。

为了聚合 AS\_PATH 属性，我们为 AS\_PATH 属性中的每个 AS 建立一个元组 <type, value>模型，其中“type”标识 AS 的路径段所属的类型（例如，AS\_SEQUENCE, AS\_SET），“值”标识 AS 号码。如果要聚合的路由有不同的 AS\_PATH 属性，则聚合的 AS\_PATH 属性应该满足以下所有条件：

- 所有聚合的 AS\_PATH 中类型为 AS\_SEQUENCE 元组应出现在要聚合的初始路由集合的所有 AS\_PATH 中。
- 所有聚合的 AS\_PATH 中类型为 AS\_SET 元组应出现在至少一个初始集合中的 AS\_PATH（它们可能显示为 AS\_SET 或 AS\_SEQUENCE 类型）。
- 对于任何聚合 AS\_PATH 中 AS\_SEQUENCE 类型的元组 X，其在聚合 AS\_PATH 中出现于元组 Y 之前，则在每个包含 Y 初始集合中的 AS\_PATH 中，X 要在 Y 之前，且不关心 Y 的类型。
- 相同 AS\_SET 类型值的元组，不应该在聚合的 AS\_PATH 中出现多次。
- 具有相同类型值为 AS\_SEQUENCE 的元组可能多次出现在聚合的 AS\_PATH 中，只有在与另一个相同类型和值的元组相邻时。

实现可以选择任何符合这些规则的算法。至少，一致的实现能够执行满足上述所有条件的下面算法：



- 确定最长的主导序列（定义如上）元组, 对所有要聚合的路由的 AS\_PATH 属性通用。使这个序列成为聚合 AS\_PATH 属性的主导序列。
- 设置其余的元组的类型从路由的 AS\_PATH 属性变为聚合路由的 AS\_SET 的属性, 且将它们附加到聚合 AS\_PATH 属性后面。
- 如果聚合的 AS\_PATH 具有多个相同的值元组（不管元组的类型）, 删除所有聚合 AS\_PATH 属性中的元组, 只留下一个其中 AS\_SET 类型。
- 对于聚合的 AS\_PATH 中的每对相邻元组, 如果两个元组的类型相同, 则将它们合并在一起, 只要这样做不会生成一个长度大于 255 的段。

附录 F, F.6 节提出了另一种算法, 满足条件并允许更复杂的复杂配置。

#### ATOMIC\_AGGREGATE:

如果至少一条要聚合的路由存在 ATOMIC\_AGGREGATE 路径属性, 则聚合路由也应该有这个属性。

#### AGGREGATOR:

要聚合的路由中禁止在聚合路由中包含的任何 AGGREGATOR 属性。BGP speaker 执行路由聚合可能会附加一个新的 AGGREGATOR 属性（见第 5.1.7 节）。

### 9.3 路由选择标准

一般来说, 在几种替代方案中比较路由的附加规则, 不属于本文档的范围。有两个例外:

- 如果考虑的新路由的 AS 路径中出现本地 AS 号, 这条新的路线不能被视为好的路由, 相比任何其他路由（只要 speaker 配置为接受这样的路线）。如果这样的路由被使用, 可能导致路由环路。
- 为了实现成功的分布式操作, 只可以选择具有稳定性的路由。因此, AS 应该避免使用不稳定的路由, 且它不应该快速地、自发地改变其选择的路由。术语“不稳定”和“快速”（上一句）的量化需要经验, 但原则是明确的。不稳定的路由可以被“惩罚”（例如, 通过使用[RFC2439]中描述的处理）。

### 9.4 始发 BGP 路由

BGP speaker 可以通过从一些其他方式（例如, 通过 IGP）获得引用的路由信息来始发 BGP 路由。发起 BGP 路由的 BGP speaker 为这些路由分配优先级（例如, 根据本地配置）, 通过决策过程（见第 9.1 节）。这些路由也可以分发给本地 AS 其他 BGP speaker, 作为 UPDATE 过程的一部分（参见第 9.2 节）。AS 内是否通过 BGP 分发非 BGP 获取的路由, 取决于 AS 内的环境（例如, IGP 类型）, 应通过配置进行控制。

## 10. BGP 定时器

BGP 使用五个定时器: ConnectRetryTimer (见第 8 节), HoldTimer (见第 4.2 节), KeepaliveTimer (见第 8 节), MinASOriginationIntervalTimer (见第 9.2.1.2 节) 和 MinRouteAdvertisementIntervalTimer (见第 9.2.1.1 节)。

可以支持两个可选的定时器: DelayOpenTimer, IdleHoldTimer (见第 8 节)。第 8 节描述它们的用途。这些可选定时器的完整操作不在此文档范围之内。

ConnectRetryTime 是一个强制性 FSM 属性, 存储 ConnectRetryTimer 的初始值。默认值建议为是 120 秒。

HoldTime 是一个强制 FSM 属性, 存储 HoldTimer 的初始值。HoldTime 的默认值建议为 90 秒。

在状态机的某些部分 (见第 8 节) 中, HoldTimer 被设置为一个很大的值。建议默认大值是 4 分钟。

KeepaliveTime 是一个强制性 FSM 属性, 用于存储 KeepaliveTimer 的初始值。默认值建议为 HoldTime 的 1/3。

MinASOriginationIntervalTimer 的建议默认值为 15 秒。

建议 EBGp 连接上的 MinRouteAdvertisementIntervalTimer 的默认值为 30 秒。

建议 IBGP 连接上的 MinRouteAdvertisementIntervalTimer 的默认值为 5 秒。

BGP 的一种实现是必须允许 HoldTimer 在每个同行的基础上是可配置的, 可以也允许其他计时器可配置。

为了使给定 BGP speaker 分发 BGP 消息的可能性峰值最小化, 抖动/JITTER 应该被应用到 MinASOriginationIntervalTimer, KeepaliveTimer, ConnectRetryTimer 和 MinRouteAdvertisementIntervalTimer。给定的 BGP speaker 可以每一个定时器应用相同的抖动量, 不管发送 UPDATE 的目的地;也就是说, 抖动量不需要在每个 peer 的基础上进行配置。

建议的默认抖动量应由适当的定时器的基值乘以随机因子决定, 随机因子均匀分布在 0.75 至 1.0 的范围内。每次定时器设置时应该选择一个新的随机值。该抖动的随机值的范围可以配置。

## 附录 A. 与 RFC 1771 的比较

与[RFC1771]相比，有很多的编辑变化（太多而不能列在这里）。

以下列出技术变更：

更改以反映 TCP MD5 等功能的使用[RFC2385]，BGP 路由反射器[RFC2796]，BGP 联邦[RFC3065]和 BGP 路由刷新[RFC2918]。

说明 AGGREGATOR 属性中使用 BGP 标识符。

对 BGP speaker 可以从 peer 接受前缀数量施加上限的处理。

为了跨 AS 流量工程/TE 的目的，在 AS\_PATH 属性中加入 BGP speaker 能够包含多个实例在其 AS 内的能力。

澄清各种类型的 NEXT\_HOPs。

澄清 ATOMIC\_AGGREGATE 属性的使用。

直接下一跳与 NEXT\_HOP 路径属性中指定的下一跳之间的关系。

澄清 tie-breaking 处理。

澄清路由宣告频率。

可选参数类型 1（认证信息）已被弃用。

更新消息错误子代码 7（AS 路由环路）已经弃用。

OPEN 消息错误子代码 5（认证失败）已经弃用。

使用标记字段进行身份验证已被弃用。

实现必须支持 TCP MD5 [RFC2385]进行身份验证。

说明 BGP FSM。

## 附录 B. 与 RFC 1267 的比较

附录 A 中列出的所有更改，以及以下内容。

BGP-4 能够运行在一组环境中，其可达目的地可能通过单个 IP 前缀表示。网络类别或子网划分的概念对 BGP-4 是外来的。为了适应这些能力，BGP-4 改变了与 AS\_PATH 属性相关联的语义和编码。新文本已经添加与 IP 前缀相关联的语义的定义。这些能力允许 BGP-4 支持建议的超网方案[RFC1518, RFC1519]。

为了简化配置，该版本引入了一个新的属性，LOCAL\_PREF，有助于路由选择过程。

INTER\_AS\_METRIC 属性已重命名为 MULTI\_EXIT\_DISC。

引入了一个新属性 ATOMIC\_AGGREGATE 来确保某些聚合不被解聚合。另一个新属性，AGGREGATOR，可以添加到聚合路由，以通告哪个 AS 和该 AS 内的哪个 BGP speaker 发生了聚合。

为了确保 HOLD 定时器是对称的，HOLD 定时器现在在每个连接的基础上进行协商。现在 HOLD 定时器支持 0 值。

## 附录 C. 与 RFC 1163 的比较

附录 A 和 B 中列出的所有更改，以及以下内容。

BGP 连接冲突的检测和恢复，一个新的字段（BGP 标识符）已添加到 OPEN 消息中。新文本（6.8 节）已添加冲突检测和恢复的过程描述。

新文档不再限制, NEXT\_HOP 路径属性经过的路由器作为 BGP speaker 的 AS 的一部分。  
新文件优化和简化关于以前可达的路线信息交换。

## 附录 D. 与 RFC 1105 的比较

附录 A, B 和 C 中列出的所有更改, 以及以下。

需要对[RFC1105]有限状态机进行小的更改, 兼容 BSD 版本 4.3 提供的 TCP 用户接口。

RFC 1105 中提出的 Up / Down / Horizontal 关系的概念已从协议中删除。

来自 RFC 1105 的消息格式的更改如下:

1. HOLD 时间字段已从 BGP 头部中删除, 添加到 OPEN 消息。
2. 版本字段已经从 BGP 头删除, 添加到 OPEN 消息。
3. “链接类型” 字段已从 “OPEN” 消息中删除。
4. OPEN CONFIRM 消息已被消除并被隐性确认替换, 由 KEEPALIVE 消息提供。
5. UPDATE 消息的格式已显著更改。将添加新字段到 UPDATE 消息中以支持多路径属性。
6. 标记领域已经扩大, 其作用扩大了支持认证。

请注意, 通常参考 RFC 1105 中规定的 BGP 作为 BGP-1; [RFC1163]中规定的 BGP 被称为 BGP-2;

RFC 1267 中规定的 BGP 称为 BGP-3; 和文档中指定的称为 BGP-4。

## 附录 E. 可能与 BGP 一起使用的 TCP 选项

如果本地系统 TCP 用户接口支持 TCP PUSH 功能, 那么每个 BGP 消息都应该被设置为 PUSH 标志来发送。强制设置 PUSH 标志及时发送 BGP 消息给接收器。

如果本地系统 TCP 用户接口对 TCP 连接支持设置 DSCP 字段[RFC2474], 则 BGP 使用的 TCP 连接应该把 DSCP 字段的 0-2 位设置为 110 (二进制)。

一个实现必须支持 TCP MD5 选项[RFC2385]。

## 附录 F. 实现建议

本节介绍一些实现建议。

### 附录 F.1 每个消息的多个网络

BGP 协议允许具有相同的路径属性的多个地址前缀在一条消息中描述。强烈推荐使用这个能力。每个消息一个地址前缀大幅增加了接收机中的开销。由于接收多条消息, 不仅增加系统开销, 而且对更新 BGP peer 和其他路由协议 (并发送关联的消息) 时扫描路由表的开销也成倍增加。

一种为一个路由表的每个路径属性组构建包含多个地址前缀的消息的方法，随着路由表被扫描，基于每个未组织的路径属性组，构建多条消息。随着每个地址前缀被处理，一个关联的路径属性集的消息被分配，如果是不存在，并为其添加新的地址前缀。如果这样消息存在，新的地址前缀被附加到它上面。如果消息缺少保存新地址前缀的空间，发送它，新的消息被分配，并且新的地址前缀被插入到新消息中。当整个路由表已被扫描，所有已分配的消息都将被发送并释放其资源。当一组共享通用路径属性地址前缀覆盖所有目的地时，达到最大压缩率，使其可以在一个 4096 字节的消息发送多个地址前缀。

当与不支持压缩的多个地址前缀到一个消息实现的 BGP 毗邻，当 peer 被获取时或者当重要的网络拓扑变化发生时，可能有必要采取措施减少洪泛数据接收的开销。这样做的一个方法是限制更新的速度。这将消除对路由表的冗余扫描，为 BGP peer 和其他路由协议提供闪存更新时间。这种方法的一个缺点是它增加了路由信息传播延迟。通过选择最小闪存更新间隔时间并不比处理多个消息所需的时间大多少，这个延迟应该被最小化了。一个更好的方法是在发送 UPDATE 之前读取所有收到的消息。

## 附录 F.2 减少路线震荡

为了避免过多的路由振荡，需要一个 BGP speaker 撤销目的地并发送一个 UPDATE，其应该将更具体或笼统的路由组合成相同的 UPDATE 消息。

## 附录 F.3 路径属性排序

组合 UPDATE 消息的实现（如上所述 第 6.1 节）可能更喜欢看到所有路径属性都是有序的。这允许他们快速识别一组来自不同 UPDATE 消息但语义上相同的属性。为了方便起见，这是一个有用的优化根据路径属性类型代码排序。这个优化是完全可选的。

## 附录 F.4 AS\_SET 排序

另一个有用的优化可以做到简化这种情况，对 AS\_SET 中的 AS 号排序。这个优化是完全可选的。

## 附录 F.5 控制版本谈判

因为 BGP-4 能够承载不能被 BGP-3 正确表示的聚合路由，一个实现支持 BGP-4 和另一个 BGP 版本基于每个 peer 应该提供只能 BGP-4 宣告的能力。

## 附录 F.6 复杂的 AS\_PATH 聚合

选择提供路径聚合并保留大量路径信息的算法，可能希望使用以下步骤：

为了聚合两条路由的 AS\_PATH 属性，我们将每个 AS 建模一个元组<type, value>，其中“type”标识 AS 所属的路径段的一种类型（例如 AS\_SEQUENCE, AS\_SET），“值”是 AS 号。也就是说如果两个 AS 相应的<type, value>元组是相同的则为相同。

两个 AS\_PATH 属性的聚合算法用作如下：



a) 识别每个 AS\_PATH 属性中的相同 ASes（如上所述），在两个 AS\_PATH 属性内以一种相同的相对顺序。也就是说，两个 AS，X 和 Y，如果同样的顺序：

- 在两个 AS\_PATH 属性中，X 都在 Y 之前
- 在两个 AS\_PATH 属性中，Y 都在 X 之前。

b) 聚合的 AS\_PATH 属性由（a）中所标识的 ASes 组成，与它们出现在要聚合的 AS\_PATH 属性中的顺序完全相同。如果连续两个（a）中识别的 ASes 在要聚合两个 AS\_PATH 属性中没有紧挨着，则两个属性中间的 ASes（连续两个连续相同的 AS 之间的 ASes）合并为一个 AS\_SET 路径段，由两个 AS\_PATH 属性中介于中间的 ASes 组成。然后放置此段在（a）中聚合属性确定的两个连续 AS 之间。如果两个连续的在（a）中被识别的 ASes，在一个属性中紧挨着，但是另一个没有紧挨着，那么后者介于中间的 ASes 组合成 AS\_SET 路径段。这个将段放置在两个连续的在（a）中标识 AS 的聚合属性之间。

c) 对于聚合 AS\_PATH 中的每对相邻元组，如果两个元组的类型相同，则将它们合并在一起且要保证不会生成长度大于 255 端。

作为上述步骤结果，如果给定的 AS 号码不止一次出现在聚合的 AS\_PATH 属性中，除了最后一个实例（最右边的出现）的 AS 号，其余的都应该从聚合的 AS\_PATH 属性中删除。

## 安全注意事项

BGP 实现必须支持在 RFC 2385 [RFC2385]中规定认证机制。在每个 peer 的基础上提供的身份验证机制。

BGP 利用 TCP 与 peer 路由器可靠地传输其流量。提供面向连接的完整性和数据原点认证在点对点的基础上，BGP 指定使用 RFC 2385 中定义的机制。这些服务旨在检测和拒绝针对路由器 TCP 连接间的主动窃听攻击。没有使用这些机制提高有效的安全服务，攻击者可能会中断这些 TCP 连接和/或伪装成合法的 peer 路由器。因为在 RFC 中定义机制不提供 peer 身份验证，这些连接可能会受到某些形式的重放攻击，其不会在 TCP 层被检测到。这种攻击可能会导致发送（来自 TCP）的“破坏”或“欺骗”的 BGP 消息。

RFC 2385 中定义的机制增加了通常的 TCP 校验和，为 16 字节消息认证码（MAC）与 TCP 校验和计算相同的数据。该 MAC 基于一个方式散列函数（MD5）和密钥应用。密钥在 peer 路由器之间是共享的，用于生成 MAC 值，其无法由没有访问密钥的攻击者轻易计算出来。一个兼容的实现必须支持这个机制，而且必须允许网络管理员在每个 peer 的基础上激活它。

RFC 2385 没有指定一种用于计算 MAC 的密钥的管理手段（例如：生成、分发和替换）。RFC 3562[RFC3562]（信息文件）提供了一些指导，并对这一指导提供理论支持。它注意到应该使用一个独特的密钥来与每个受保护的 peer 进行交互。如果同一个密钥用于多个 peer，则提供的安全服务可能会降级，例如，由于在一个路由器上妥协的风险增加可能对其他路由器造成不利影响。

用于 MAC 计算的密钥应定期更改，最小化密钥妥协或密码分析成功带来攻击造成的影响。RFC 3562 建议一个加密期（一个密钥是使用的间隔），最多 90 天。频繁的密钥更换是可行的，来减少重播攻击（如上所述）。但是，如果没有 peer 之间协调发生变化方式的标准机制，不能假设 BGP-4 实现将符合此 RFC 及支持频繁的密钥变化。

显然，每个密钥都应该选择一个攻击者难以猜测的。RFC 1750 讲述的随机数生成技术提供了生成密钥数值的指导。RFC 2385 要求实现支持“由一串 80 字节或更少的可打印 ASCII 码组成。”密钥。

RFC 3562 建议在本上下文中使用的密钥是 12 到 24 个字节的随机（伪随机）位。这与建议的模拟的 MAC 算法相当一致，通常密钥采用的范围为 16 到 20 个字节。为了在这个范围的低端提供足够提供足够的随机位，RFC 3562 还遵守到，典型的 ASCII 文本字符串必须接近上面 RFC 2385 指出的密钥长度的上限。

BGP 漏洞分析在[RFC4272]中讨论。

IANA 注意事项

所有的 BGP 消息都包含一个 8 位消息类型，为此 IANA 创建并正在维护名为“BGP 消息类型”的注册表。本文档定义了以下消息类型：

名称	值	定义
OPEN	1	见第 4.2 节
UPDATE	2	见第 4.3 节
NOTIFICATION	3	见第 4.5 节
KEEPALIVE	4	见第 4.4 节

未来的分配是使用在[RFC2434]中定义的“标准行动”处理，或早期 IANA 在[RFC4020]中定义分配处理。分配由名称和值组成。

BGP UPDATE 消息可以携带一个或多个路径属性，其中每个属性包含一个 8 位属性类型代码。IANA 已经维护了这样一个名为“BGP 路径属性”的注册表。本文档定义了以下路径属性类型代码：

名称	值	定义
ORIGIN	1	见第 5.1.1 节
AS_PATH	2	见第 5.1.2 节
NEXT_HOP	3	见第 5.1.3 节
MULTI_EXIT_DISC	4	见第 5.1.4 节
LOCAL_PREF	5	见第 5.1.5 节
ATOMIC_AGGREGATE	6	见第 5.1.6 节
AGGREGATOR	7	见第 5.1.7 节

未来的分配是使用在[RFC2434]中定义的“标准行动”处理，或早期 IANA 在[RFC4020]中定义分配处理。分配由名称和值组成。



BGP NOTIFICATION 消息携带 8 位错误代码，IANA 已经创建并正在维护名为“BGP 错误代码”的注册表。本文档定义了以下错误代码：

名称	值	定义
消息头错误	1	第 6.1 节
OPEN 消息错误	2	第 6.2 节
UPDATE 消息错误	3	第 6.3 节
HOLD 定时器已过期	4	第 6.5 节
FSM 错误	5	第 6.6 节
Cease	6	第 6.7 节

未来的分配是使用在[RFC2434]中定义的“标准行动”处理，或早期 IANA 在[RFC4020]中定义分配处理。分配由名称和值组成。

BGP NOTIFICATION 消息携带 8 位错误子码，其中每个子代码必须在特定的错误代码上下文中定义，因此必须在该上下文中是唯一的。

IANA 创建并正在维护一组注册表，“错误子代码”，每个 BGP 错误代码都有一个单独的注册表。未来的分配是使用在[RFC2434]中定义的“标准行动”处理，或早期 IANA 在[RFC4020]中定义分配处理。分配由名称和值组成。

本文档定义了以下消息头错误子代码：

名称	值	定义
连接未同步	1	见第 6.1 节
错误消息长度	2	见第 6.1 节
错误消息类型	3	见第 6.1 节

本文档定义了以下 OPEN 消息错误子代码：

名称	值	定义
不支持的版本号	1	见第 6.2 节
坏的 Peer AS	2	见第 6.2 节
坏的 BGP 标识符	3	见第 6.2 节
不支持的可选参数	4	见第 6.2 节
[已弃用]	5	见附录 A
不可接受的 HOLD 时间	6	见第 6.2 节

本文档定义了以下 UPDATE 消息错误子代码：

名称	值	定义
属性列表格式错误	1	见第 6.3 节
无法识别的公认属性	2	见第 6.3 节

缺少公认属性	3	见第 6.3 节
属性标志错误	4	见第 6.3 节
属性长度错误	5	见第 6.3 节
ORIGIN 属性无效	6	见第 6.3 节
[已弃用]	7	见附录 A
NEXT_HOP 属性无效	8	见第 6.3 节
可选属性错误	9	见第 6.3 节
无效的网络字段	10	见第 6.3 节
AS_PATH 格式错误	11	见第 6.3 节