边界网关协议和路由表增长

无 47　　刘前 *　　2014011216

2017 年 1 月 14 日

摘要：本文对边界网关协议 (Border Gateway Protocol，BGP)、自治域的互联及 BGP 路由表规模的增长等进行了综述。本文首先将 BGP 与传统的距离向量类路由算法进行了对比，分析了两者之间的显著差别，然后总结了自治域互联过程中在路由选择时需要关注的因素，分析了各因素对路由协议和路由算法产生的影响。同时，本文还对 BGP 规模增长的机理进行了分析说明，解释了 BGP 路由表的前缀条目增长速度大于 IP 地址块分配速度的原因。最后，本文还对 BGP 当前存在的突出问题及解决和改进方法进行了总结，并据此对未来 BGP 的发展进行了展望。

关键词：边界网关协议，距离向量类路由算法，BGP 路由表

1　距离向量类路由算法与 BGP

1.1　距离向量类路由算法

1.1.1　基本介绍

距离向量类路由算法是常见的动态路由算法之一，最早被用于 ARPANET，能够利用网络当前的负载情况动态决定进来的分组应当被传送至哪一条输出线路。距离向量路由算法的设计者为 Bellman，Ford 和 Fulkerson，因而也常被称作 Bellman-Ford 或 Ford-Fulkerson 算法。

1.1.2　算法原理

距离向量类路由算法中，每个路由器维护一张表，称为路由表 (Routing Table)。每张路由表都以子网 (subnet) 中各路由器为索引，每个路由器对应表的一个表项。表项分为两部分内容：a) 为了到达该目标路由器而首次使用的输出线路；b) 到达该目标路由器的距离估计值。关于第二部分表述的 “距离”，不同的应用场景下往往有不同的度量单位，常见度量包括跳数、时间延迟和沿该路径排队的分组数目 (队列长度) 等。

假设使用时间延迟作为距离的度量标准，若每个路由器都知道它到所有邻居路由器的时间延迟，那么路由器在向每个邻居发送列表时，列表中包含了它到每个目标的延迟估计值；同样，该路由器从其他路由器接收一个列表时，也包含延迟估计值。举例来说，如果一个路由器 X 接收到来自邻居 Y 的一个列表，其中包含了 Y 估

*E-mail：liuqian14@mails.tsinghua.edu.cn，清华大学电子工程系。

计的到达路由器 Z 所需要的时间 T_{yz}；此时如果路由器 X 知道它到 Y 的时间延迟为 T_{xy}，那么 X 即可计算得到在 $T_{xy} = T_{xy} + T_{yz}$ 时间内经过路由器 Y 到达路由器 Z。每个路由器针对它的每个邻居都进行上面的计算，进行比较即可发现最佳的距离估计值，并在新的路由表中使用最佳的估计值。

距离向量类路由算法可以总结为：路由表中列出当前已知路由到每个目标节点的最佳距离；通过在邻居节点之间相互交换信息，路由器不断对路由表进行更新。

1.2 边界网关协议

1.2.1 基础概念

为方便理解边界网关协议，首先需要明确一些基本概念。

自治系统 (Autonomous System，AS)

对于独立的网络，每个网络可以使用完全不同的路由算法。当互联网内部的每个网络都独立于所有其他的网络时，可以将每个网络称作一个自治系统。举例来说，假设三个企业 X、Y 和 Z 的内部网络都在 Internet 内，并且相互独立，那么这三个网络通常被看作三个自治系统，它们内部可以使用不同的路由算法。自治系统内部的路由算法被称为内部网关[1]协议 (Interior Gateway Protocol，IGP)，自治系统之间的路由算法称为外部网关协议 (Exterior Gateway Protocol，EGP)。

内部网关协议 (IGP)

最早的 Internet 内部网关协议是一个基于距离向量路由算法 (Bellman-Ford Algorithm) 的路由信息协议 (Routing Information Protocol，RIP)，从 ARPNET 中继承而来。随着自治系统越来越大，RIP 逐步被开放的最短路径优先 (Open Short Path First，OSPF) 协议所替代。

1.2.2 边界网关协议

因为内部网关协议和外部网关协议的目标是不相同的，所以在自治系统之间，需要一个与自治系统内部完全不同的协议。在一个自治系统内部，经常使用的路由协议是 OSPF 协议，而在自治系统之间则常使用边界网关路由协议 (Border Gateway Protocol，BGP)。

从一台 BGP 路由器的角度来看，网络可以看成是由不同的自治系统和自治系统之间的连接线路共同构成的。若一条线路两端分别对应不同自治系统的 BGP 路由器，则认为两个自治系统是相连的。BGP 路由器之间通过线路建立 TCP 连接进行通信，提供了可靠的通信，并且隐藏了中间网络的所有细节。

与普通的路由器相比，BGP 路由器需要考虑很多额外的因素，包括政治、经济和安全性等。举个例子，一个企业的自治系统希望能够向所有的 Internet 站点发送分组，同时希望能够从所有的 Internet 站点接收到分组。但是，当起始端和目标端所在的自治系统不同时，路由器可能由于诸多原因不希望承担两者之间的中转任务。基于以上特定的需求，BGP 协议提供了多种路由策略，能够强制用于跨自治系统的传送。这些策

[1]“网关” 是 “路由器” 的老式称呼。

略往往以手工的方式被配置到每台 BGP 路由器中，其本身不属于 BGP 协议的一部分。

1.3 BGP 与传统距离向量类路由算法的比较

BGP 本质上是一种以距离向量路由算法为基础的距离向量协议。但是，BGP 与传统距离向量类路由算法之间仍有很多明显的不同，主要总结为以下三点。

1.3.1 路由器传递的信息不同

在传统的距离向量类路由算法中，路由器在路由表中列出了已知的到每个目标的最佳距离及使用线路，但是并没有给出到每个目标的完整的路径，只提供了路由器到每个目标的 “距离” 信息。但是，在 BGP 中，每台 BGP 路由器不仅维护它到每个目标的距离，还记录下所使用的确切路径。也就是说，每台 BGP 路由器不仅定期告诉邻居关于每个可能目标的估计开销，还将它使用的准确路径一并告诉邻居。

1.3.2 路由器选择路径的方式不同

距离向量类算法的目标比较简单，尽可能有效地将分组从源端传送到目标端。因此，路由器在更新路由表选择路径时，必然根据距离最短的原则选择最佳路径。但是 BGP 路由器在选择路径是会复杂许多。

对于 BGP，当邻居提供的所有路径信息到达路由器后，该路由器会对这些路径进行检查。每一台 BGP 路由器包含一个功能模块，该模块能够检查所有到达指定目标的路径，并通过使用评价函数对路径进行评价，评价后为每条路径返回一个数值，代表对应的 “距离” 开销。BGP 的路由策略相当于对路径的选择增加了限制条件，如果某路径违反了 BGP 路由器中预先配置的策略限制，则该路径的分值会被赋值为无穷大。对所有路径使用评价函数之后，路由器将会采用 “距离开销” 原则选择最短的路径。

1.3.3 无穷计算问题的处理

传统的距离向量类路由算法存在着一个严重的缺陷，即无穷计算问题。虽然距离向量路由算法保证总能得到正确的答案，但是它收敛到正确答案的速度可能会非常缓慢。但是 BGP 能够很高效地解决无穷计算问题。

无穷计算问题通常是指，距离向量路由算法对于好消息反应非常快，但是对坏消息反应非常迟缓。

1) 对于好消息

假设路由器 A 到目标 X 的时间延迟很大，如果下一次交换信息时，A 的一个邻居 B 突然报告说它到 X 有一个非常短的延迟 (称为好消息)，那么路由器 A 只需将发送给 X 的流量切换到通向 B 的线路，好消息就能够起作用了。事实上，好消息的扩散速度是每交换一次向远处前进一跳；因而如果一个子网中最长的路径是 N 跳，那么经过 N 次交换之后，每个路由器都将获取到好消息。

2) 对于坏消息

坏消息之所以传播速度速度缓慢，是因为没

有路由器知道比它所有邻居的最小值大超过 1 的值是多少。逐渐地，所有路径长度都会趋向无穷大，无穷大用一个数值来代替，坏消息扩散所需的交换次数依赖于代表无穷大的数值。

对于距离向量路由算法，无穷计算问题是一个重大缺陷。面对这一问题，距离向量算法往往用最长的路径加 1 代表无穷大。若使用时间作为“距离” 的度量标准，则需要一个较大的值代表无穷大，没有明确定义的上限值，以便避免将一条延迟较长的路径当成断路处理。

事实上，无穷计数问题的核心在于，当路由器 X 告诉 Y 它有一条路径的时候，Y 无法知道自己是否在这条路径之上。对于 BGP，这一问题得到了很好的解决，因为路由器在接收邻居的信息时，能够立即获取确切的路径，从而立即判断出路径是否经过该路由器本身，如果经过本身，则该路径是没有意义的，BGP 能够做出正确的决定，避免无穷计算问题。

1.4 小结

本部分对传统的距离向量类路由算法和 BGP 进行了基本的介绍，并对两者的差别作了详细的说明。可以看出，BGP 相对于距离向量路由算法，在实际应用中会更加完备和适用。传统距离向量算法在理论上可行，但实际应用会存在一定缺陷，BGP 不仅尽力克服了一些缺陷，还从政治、经济、安全角度增加一些路由策略，使得 BGP 路由的功能更加多样化。

2 自治域互联的路由选择

自治域互联过程在进行路由选择时，往往需要考虑方方面面的因素，除去最基本的路径长度，还包括简单性、稳定性、灵活性以及某些特定的路由策略。下面将分别对以上因素进行简要分析，研究它们对路由协议和路由算法的影响。

2.1 路径长度

路径长度是路由选择时最常用也是最基本的标准。路径长度越短，各种消耗越小，路由器越应当选择该路径。此处路径长度同样是一个广义的概念，在不同的情境下路径长度有不同的衡量标准，一般包括跳数、物理距离、传输延迟等。

不论是基于路径向量路由算法的内部网关路由协议，还是以 BGP 为代表的自治系统之间的外部网关路由协议，路由选择都是以路径长度为基本原则，再附加一些其他的限制。因而，对于路由协议或路由算法来说，路径长度是进行路由选择的首要标准。

2.2 简单性

在满足各种功能的前提下，路由选择的算法或者协议应当尽可能地简单。因为在路由选择时，不可避免地要向网络中增加一些额外开销，为了避免过多增加网络的开销，路由选择也应当选择低耗的路径，路由算法应当简单易实现。

2.3 稳定性

当网络中某一路径或者路由器出现问题时，路由选择应当尽量保证不会因网络结构的改变而影响正常到达目标，即路由选择所增加的延迟不应过多。前文指出距离向量路由算法存在无穷计算问题，如果不特殊处理，路由算法的稳定性会较差。因此，在设计路由协议或者路由算法时，需要特别关注稳定性因素，从而在路由选择时性能更加稳定健壮。

2.4 路由策略

在自治域互联的路由选择中，往往需要根据不同需求增加额外的某些路由策略，这些策略相当于在原有路由算法或协议的基础之上增加一些限制条件。典型的路由策略限制可能会涉及政治、安全和经济等。为了在路由选择时满足这些限制条件，路由器往往需要进行额外的设置。比如，在边界网关协议中，以上策略都是以手工的方式添加配置到每台 BGP 路由器中，本身不是路由协议的一部分。因而路由策略也成为路由选择时需要关注的重要因素。

2.5 小结

自治域互联过程中，路由选择需要关注的因素多种多样。以上简要分析了一些主要的因素，这些因素共同决定了在实际网络中路由协议的设计和路由算法的具体实施。

3 BGP 路由表规模增长

3.1 基础概念

3.1.1 BGP 路由表结构

BGP 是全球通用的自治系统之间的路由协议。BGP 使用前缀来区别同一目标网络中的 IP 地址块，以 IPv4 为例，前缀通常使用 32 位的 IP 地址和一个掩码位。举例来说，10.0.0.0/8 表示范围在 10.0.0.0 至 10.255.255.255 之间的 IP 地址块。

BGP 路由表包含了如何到达各路由的相关信息，主要包括前缀条目、自治系统的路径信息等，从而保证能够抵达相应的前缀。

3.1.2 IP 地址分配

IP 地址的分配是通过以前缀为基础的 IP 地址块来实现的。起初，IP 地址空间被分为 3 个不同的地址类型，在 32 位 IP 地址 (IPv4) 中，每种类型的地址在网络前缀 (network-prefix) 和客户编号之间都有固定的边界。后来 CIDR(Classless Inter-Domain Routing) 实现了网络前缀和客户编号之间灵活的界限，从而使得 IP 地址分配更具灵活性，同时使得地址空间的利用更加充分。

3.2 BGP 路由表规模增长的机理

[] 基于 RouteViews 收集的 BGP 路由表的相关数据，分析发现 BGP 路由表增长是两个方面因素共同作用的结果：新前缀条目的增加和旧前缀条目的消失。[] 更详细地总结了造成 BGP 路由表增长的重要因素，主要包括多宿

主 (multi-homing)、整合路由表失败 (Failure to Aggregate to Routing Table Size) 和负载均衡 (Load Balancing) 等等，并且分析了各因素在 BGP 路由表中的组成及特点。

后文简要分析以上对路由表规模影响较大的因素，然后将这些因素的变化与路由表规模的增长相比较，分析得出 BGP 路由表规模增长的内部机理。

3.2.1 多宿主

多宿主是指自治系统连接多个提供者 (providers)，这是考虑到容错性而实行的机制。多宿主为什么会导致 BGP 路由表规模的增长呢? 多宿主往往会导致路由表中出现空洞 (hole)，空洞是指一些地址块已经包含在另一地址块中但是却在路由表中被单独声明。多宿主机制下，自治系统的前缀会在连接的多个提供者中被声明，从而意味着 BGP 路由表规模的增长。

3.2.2 整合失败

BGP 要求将前缀按照自治系统的不同在路由表中进行分类，形成前缀簇 (prefix cluster)，并且分别声明。前缀簇是相同自治系统的前缀的最大集合，实现前缀簇分类的过程称为整合或者并入 (aggregation)。前缀一般采用迭代的方法进行整合，一直进行到无法再整合。整合完成之后，前缀的总数目等于原来的数目减去无法整合的前缀数。

3.2.3 负载均衡

此处负载均衡是指相同自治系统内的前缀由于声明不同导致无法整合。这种情形下，原本的前缀包括整合后的前缀、整合失败的前缀和负载均衡导致无法整合的前缀。[] 中指出，负载均衡大约会引入 20%25% 的新前缀，导致路由表规模的增长。

3.2.4 小结及拓展

[] 根据数据分析了以上各因素和路由表规模增长的对比，结果发现，多宿主和负载均衡是导致路由表规模增长的最为明显的原因。

[] 同样通过分析数据，认为新 IP 地址的分配是 BGP 路由表增长的主要原因。新 IP 地址的分配与新前缀出现和旧前缀的消失都有密不可分的关系；并且，新前缀的数目明显多于消失的旧前缀的数目，这也造成了 BGP 路由表的增长。

3.3 问题讨论

关于 BGP 路由表规模的增长，有一个问题需要单独加以讨论。基于前文的说明，IP 地址块的分配需要更多的新前缀。[] 中通过调查 IP 地址块分配增长速度和前缀条目增长速度之间的关系，对 IP 地址空间的增长进行了研究。数据表明，在 2002-2005 年间，前缀条目的数目增长超过 100%，但是 IP 地址块的分配只增加了大约 25%，表明 BGP 路由表中前缀条目的增长速度大于 IP 地址块的分配速度。前缀条目的数目

与 IP 地址分配块之间的关系就是怎样的？下面对这一问题给出两点解释。

3.3.1 新前缀不一定对应 IP 地址块分配的增加

路由表中新前缀的出现，不一定会增加 IP 地址块的分配。因为新出现的前缀所对应的 IP 地址空间可能已经被现有的前缀所覆盖，因而没有必要再给这些新前缀分配地址空间。这就导致前缀条目数增加较快时，IP 地址分配可能并没有相应得快速增加。[Impa] 通过 RouteViews 提供的路由表的数据，发现超过 67.5% 新出现的前缀条目都对应于旧的分配的地址，不需要相应地新的地址分配。

3.3.2 旧前缀消失导致的前缀条目减少影响较小

[Imp] 还得到结论，当 IP 地址块分配增加时，消失的前缀所对应的地址空间仍然有 78.5% 的部分被剩余未消失的前缀所覆盖，略多于 20% 的会导致前缀条目的缺失，这对前缀条目的增加不会造成很大的影响。

总之，在以上两方面因素的影响之下，也就解释了 BGP 路由表中的前缀条目的增长速度大于 IP 地址块的分配速度这一现象出现的原因。

4 BGP 存在的问题与改进方法

BGP 作为自治系统之间的外部网关协议，虽然具有很多功能和优点，但仍然不可避免存在一定的缺陷。本文主要介绍 BGP 的两大缺陷：域间路由问题和路由策略冲突。

4.1 域间路由问题

4.1.1 基本介绍

随着互联网规模的日益扩大，自治系统之间的路由选择需要考虑的因素更加复杂。考虑到不同网络的性能差异很大，并且结合政治、经济等因素，自治系统之间的路由选择必须要制定相应的路由策略，选出较优的路径。BGP 正是基于这一目的所设计的。然而，BGP 在设计上存在着较大的安全缺陷，并已经导致了很多安全事故，对自治域互联的路由安全造成了威胁。[] 中总结了当前自治域间路由问题的主要原因包括：路由泄露、BGP 错误配置、前缀劫持及自治系统路径篡改等。

4.1.2 改进方法

以路由泄露为代表的域间路由问题是由 BGP 路由系统存在的系统漏洞所导致的，轻则导致网络不可达，重则导致整个互联网的瘫痪，对经济等造成巨大损失。

由于不少大型公司因此蒙受过损失，因而针对这一问题也投入了不少力度研发了不少安全机制。常见的包括前缀过滤机制、路径过滤机制路由注册表过滤机制及 RPL 保护机制等。Geoff Huston 还于 2012 年提出了两种较为复杂的安全机制，在解决域间路由问题中卓有成效。

4.2 路由策略冲突

4.2.1 基本介绍

BGP 路由策略能够根据各自治系统的实际情况由使用者自主配置，使各个自治系统根据各自需要进行灵活的路由选择，但是这可能会引发 BGP 路由策略冲突问题。

BGP 路由策略冲突通常是由于各个自治域配置了互相冲突的路由策略，引发路由振荡，致使路由始终发散。这一问题将直接导致转发延迟甚至丢失，引起网络拥塞，极大降低网络的性能和服务质量。[] 中针对路由策略冲突问题，给出了三个改进方法。

4.2.2 改进方法

静态分析法、指导规则法及动态适应法是当前解决 BGP 路由策略冲突的三个主要方法。

静态分析法 静态分析法是指对各个自治系统的路由策略进行集中地分析，直到发现其中路由策略的冲突并加以解决。静态方法在理论上有效，但是要求在分析之前收集到所有自治系统配置的所有路由策略，这不仅实施起来非常困难，而且还会暴露各自治系统的路由策略，导致 BGP 整体的安全性降低，因而静态分析法的应用相对较少。

指导规则法 指导规则法是指在策略配置之前，所有的自治系统路由策略都受到某些指导规则的约束。在指导规则的约束之下，以保证各 BGP 路由策略不发生冲突。[] 中的 Valley-Free 指导规则从理论上能够保证各自治系统之间不存在路由策略冲突，具有很好的效果。但是，这种方法明显会对 BGP 路由策略的设置带来很大的限制，如果限制过高，很可能不会被自治系统的管理者们所接收。

动态适应法 动态适应法的思想是在 BGP 的运行过程中解决路由策略冲突问题，这样便不会像指导规则法一样失去 BGP 路由设置策略的灵活性。动态适应法往往是在路由决策阶段，使用基于优先级的算法解决路由策略冲突问题，应用相对较为广泛。

4.3 小结与展望

BGP 是当前互联网最常用的域间路由协议，对互联网安全起着至关重要的作用。但是目前 BGP 路由仍然存在着前文所述的一些关键的问题亟待解决。未来在 BGP 方面，还需要继续完善当前的算法或者提出更加完备的解决方法应对域间路由问题和路由策略冲突，改进 BGP 的安全性和稳定性，在维护互联网安全中起到关键的作用。

5 总结

本文属于综述性的文章，对边界网关协议的相关内容、自治域互联中的路由选择和 BGP 路由表内在机理等方面进行了简要的讨论，这对了解 BGP 路由协议、理解当今网络工作机理有着重要的作用。本文的最后所指出的 BGP 尚存的问题是未来 BGP 发展的重要课题，解决上述问题、改进 BGP 算法，对促进未来互联网技术发

展有着十分重要的意义!

参考文献

[1] Tanenbaum A S. Computer Networks, Fourth Edition[J]. 2003.

[2] Meng X, Xu Z, Zhang B, et al. IPv4 address allocation and the BGP routing table evolution[J]. Acm Sigcomm Computer Communication Review, 2004, 35(1): 71-80.

[3] Bu T, Gao L, Towsley D. On characterizing BGP routing table growth[J]. Computer Networks, 2002, 45(1): 45-54.

[4] Scudder J, Retana A, Walton D, et al. BGP Persistent Route Oscillation Solutions[J]. 2016.

[5] Meng X, Xu Z, Zhang L, et al. An analysis of BGP routing table evolution[J]. Techincal Report, 2003.

[6] Xu Z, Meng X, Zhang L, et al. Impact of IPv4 address allocation practice on BGP routing table growth[M]. 2003.

[7] Giotsas V, Zhou S. Valley-free violation in Internet routing —Analysis based on BGP Community data[J]. In IEEE ICC, 2012, 11(18): 1193-1197.

[8] Cittadini L, Rimondini M, Vissicchio S, et al. From Theory to Practice: Efficiently Checking BGP Configurations for Guaranteed Convergence[J]. IEEE Transactions on Network & Service Management, 2011, 8(4): 387-400.

[9] Rekhter Y, Li T. A Border Gateway Protocol 4 (BGP-4)[J]. RFC, 1994, 19(6): 193–199.

[10] Kent S, Lynn C, Seo K. Secure Border Gateway Protocol[J]. Selected Areas in Communications IEEE Journal on, 2000, 18(4): 582-592.

[11] Kuhn D R, Sriram K, Montgomery D C. Border Gateway Protocol Security[J]. Recommendations of the National Institute of Standards & Technology Special Publication, 2007, 19(6): 3-4.

[12] 于超, 王兴伟, 黄敏. 解决 BGP 路由策略冲突的振荡抑制机制 [J]. 计算机科学与探索, 2016, 10(1): 74-81.

[13] Traina P. Autonomous System Confederations for BGP[J]. 1996.