计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2017,26(10):275−279 [doi: 10.15888/j.cnki.csa.006058]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 VPP 的虚拟路由器数据平面加速方法①

张宇巍, 曾　一, 杨燕宁

(重庆大学 计算机科学与工程学院, 重庆 400000)

摘　要: 本文提出了一种使用 VPP 对虚拟路由器数据平面加速的方法, 此方法将数据平面与控制平面分开, 然后将数据平面的转发工作转移到 Linux 用户空间下的 VPP 上面, 并通过监听控制平面信息, 实现数据平面路由表的更新. 通过此种方法, 可消除虚拟路由器在报文转发速率上的瓶颈, 使得使用标准 x86 平台服务器替代专用网络设备成为可能, 从一定程度上促进了网络功能虚拟化的发展, 使得网络资源更加具有弹性, 易于管理.

关键词: 虚拟路由器; 网络功能虚拟化; 计算机网络; VPP

引用格式: 张宇巍,曾一,杨燕宁.基于 VPP 的虚拟路由器数据平面加速方法.计算机系统应用,2017,26(10):275–279. http://www.c-s-a.org.cn/1003-3254/6058.html

Method to Accelerate Virtual Router Data Plan Based on VPP

ZHANG Yu-Wei, ZENG Yi, YANG Yan-Ning

(Computer Science and Teconology, Chongqing University, Chongqing 40000, China)

Abstract: This paper proposes a method to accelerate the virtual router's data plan by VPP. This method separates the virtual router into control plan and data plan, then moves the data plane's transform work to the VPP running in Linux userspace and alters the VPP's route table by listening to the control plan info. In this way, it could eliminate the bottleneck of the traditional virtual route's rate of transform packets which make it possible to replace the network device with standard x86 platform server. It can promote the network function virtualization in some way and make the network resource more flexible and easier to be managed.

Key words: virtual router; network function virtualization; computer network; VPP

1　引言

在过去的十几年里, 互联网已经成为整个社会的基本支柱之一, 伴随着数据中心与日俱增的数据流量, 传统互联网架构已经难以满足各大互联网公司数据中心的基本需求, 所以业界对互联网架构进行了更深入的研究, 但互联网在应对安全性、移动性和服务质量等方面的挑战时, 采用修补解决的方式阻碍了创新型网络架构的部署和评估, 无法有效解决互联网本身固有的问题[1]. 目前的趋势指出, 一个灵活的, 创新的网络架构才是符合现实需求的[2]. 对此, 各种新型的网络结构及对现有网络的改进方法被提了出来, 其中软件定义网络 (Software define network) 及网络功能虚拟化 (Network function virtualization) 无疑是最热门的话题之一. SDN 将传统的网络结构划分为数据平面和控制平面, 其中, 数据平面是负责网络内容的转发即网络中的主要流量, 报文内容可能是视频流, 即时通信信息等, 控制平面转发的内容则是路由信息等网络控制报文, 例如 BGP, RIP 等路由协议报文, 目的是为设备维护和更新路由表. SDN 网络中的路由器只负责数据流量的转发, 将控制平面交给中央控制器负责, 中央控制器通过下发流表的方式对路由器进行控制. 传统的网络设备不仅价格昂贵, 而且封闭性较强, 用户只能以各大网络设备商允许的方式对网络设备进行配置和使用, 这无疑不利于互联网的发展, NFV 技术颠覆了传统电信

① 收稿时间: 2017-02-11; 采用时间: 2017-03-20

网络封闭专用的思想, 同时引入了资源弹性管理的思想, 将各种网络功能通过软件定义的方式, 结合虚拟化的思想, 运行于通用服务器之上. 相较于专用网络设备, 通用服务器有着价格相对低廉, 开放性较强及可灵活定制的优点.

路由功能是整个互联网的基础骨架结构, 所以, 随着 NFV 的发展, 路由功能的虚拟化越来越被重视起来. 各种虚拟路由器被发布出来, 例如 Quagga, RouterOS 等. 同时, 伴随着虚拟化技术的发展, 使得多台虚拟路由器可以同时运行在一台主机之上, 每一台虚拟路由器逻辑上都是独立的, 可以满足不同的应用需求[3]. 将不同路由器运行于同一个物理平台上可以使每个虚拟路由器运行于不同的网络环境中, 有效的节省了硬件资源[4-6]. 而有关路由器虚拟化的设计与研发, 路由器厂商和科研机构已经着手对支持异构网络和服务且运行在同一共享底层平台的虚拟路由器进行深入的研究[7-9]. 虚拟路由器相较于传统路由器最大的问题在于网络接口收发包速度的差距和路由选择速度的差距. 本文将针对减小网络接口收发包速度的差距这一方面, 提出一种改进方法. Quagga 作为一款虚拟路由器, 支持 RIP, OSPF, BGP 等大部分主流路由协议, 故本文选择 Quagga 为研究对象并进行改进.

2 虚拟路由器存在的问题及 VPP 数据平面加速工具技术分析

2.1 虚拟路由器

虚拟路由器将传统路由器路由的功能通过软件的方式实现, 减少了对专用封闭设备的依赖, 提高了设备的可定制性. Linux 系统本身通过路由表的形式提供了路由的功能, 但是 Linux 系统只能通过手动加入或修改静态路由的方式更新路由表. 故此, 以 Quagga 为代表的虚拟路由器通过维护 Linux 路由表, 并用软件实现各种动态路由算法的方式, 实现了虚拟路由器动态路由的功能.

相较于传统路由器, 虚拟路由器存在以下问题: 1) 传统路由器通过专有硬件加速的方式实现高速的路由选择, 虚拟路由器只能以较慢的速度通过软件方式进行路由选择. 2) 虚拟路由器通过中断的方式进行数据报文的收发, 每次需要陷入 Linux 内核态并进行 memory copy, 导致极大的 I/O 延迟. 本文将针对第二点, 对虚拟路由器 Quagga 进行优化, 使用户态空间收发包的形式对数据平面进行加速.

2.2 数据平面加速工具

随着高速网卡技术的发展, 10G, 20G 乃至 100G 的网卡纷纷进入市场并被各大数据中心使用, 针对高速网卡, 传统的内核态中断方式处理数据包的方式难以满足高速网卡的要求, 针对这个需求, 多种用户态处理数据包的工具被开发出来, 其中比较有代表性的就是 Intel 公司推出的 DPDK(Data plan development kit). DPDK 通过将中断改为轮训的处理方式, 将需要内核态处理的部分转移到用户态空间, 减少了 I/O 中断时上下文保存和恢复以及 memory copy 的时间, 大大提升了数据报文处理速率. 同时 DPDK 还运用 Linux 提供的 hugepage 技术, Hugepage 使用有别于传统页表的 2MB 大小的页表, 使用此种技术, 将会极大的增加了 cache 的命中率, 从而对报文处理的速率进行提升. DPDK 主要基于链路层进行快速收发包的工作, Cisco 利用 DPDK 高速收发包的能力, 开发出一款数据平面的报文处理工具 VPP(Vector packet process), 实现了 IP 层路由的功能, 从而使得运行 VPP 的服务器具有高速收发数据报文并根据路由表进行转发的能力. VPP 采用 Graph Node 的形式进行报文处理, 每一个 Node 作为一个逻辑单元而存在, 可完成相对独立的任务, 例如 VLAN 标签的处理, MPLS 标签处理等. VPP 将所接收到的报文按组的形式组织起来形成向量, 每个向量有固定数量的报文组成, 以向量为单位通过每个 Node, 通过这种方式, 使得每个节点同时处理一定数量的报文, 可以较明显的提高 Cache 和内存的命中率, 减少了页表换出换入的操作, 从而提高了整体性能. VPP 通过 Graph Node 的形式, 提供给了开发者较强的可定制性, 开发者可以向 VPP 加入自己制定的 Node, 并将其和其它 Node 进行串联, 从而对特定环境下的网络报文进行处理. VPP 平台可以用于构建任何类型的数据包处理应用, 比如负载均衡、防火墙、IDS、主机栈. 也可以是一个组合, 比如给负载均衡添加一个 vSwitch. 本文选择 VPP 作为数据平面的加速工具来对虚拟路由器进行优化.

3 虚拟路由器架构设计

3.1 传统虚拟路由器架构

Quagga 作为传统虚拟路由器的代表, 整体架构如图 1 所示. VTYSH 作为与用户的交互模块, 提供了以

命令行的形式配置路由器的途径. VTYSH 仿照大多数商业路由器的通用模式 (例如 Cisco 和 Huawei 的商用路由器) 进行配置. OSPF, BGP, RIP 作为动态路由协议由 Quagga 进行实现, 并可以通过配置文件的形式决定启用哪一个动态路由协议. Zebra 作为 Quagga 的管理模块, 负责 Quagga 的配置管理的生成和更新路由表的操作, Zebra 通过 Netlink 的方式与内核进行通信以达到修改内核路由表的目的.

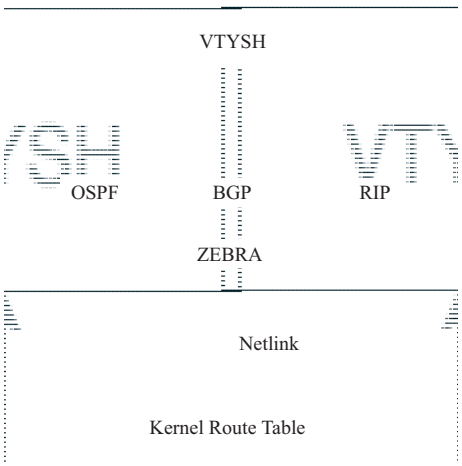

图 1 Quagga 架构图

将 Quagga 数据平面转移到 VPP 上有两个问题需要解决

当虚拟路由器启动后, Quagga 将启动用户所配置的路由协议, 通过与其他路由器的通信建立初始路由表, 并在运行中持续更新路由表.

Netlink 是 Linux 提供的用户空间与内核进行通信的方式之一, Netlink 是基于 socket 的通信机制, 由于 socket 本身的双共性、突发性、不阻塞特点, 因此能够很好的满足内核与用户空间小量数据的及时交互, 因此在 Linux 2.6 内核中广泛使用.

3.2 基于 VPP 加速数据平面的虚拟路由器架构设计

Quagga 的数据平面运行于内核空间下, 用户空间向内核空间转换过程中的环境保存和 memory copy 过程限制了报文转发过程的速度, 成为了 Quagga 转发速率的瓶颈, 所以, 将数据平面转移到用户空间下成为加速 Quagga 转发速率的最有效方法.

经过分析, 有两种方式可以作为转移 Quagga 数据平面到 VPP 的选择: 1) 将 Quagga 中操作 Linux 路由表的部分用 VPP 提供的操作自身路由表的 API 替换. 2) 不修改 Quagga 代码, 通过增加 VPP 插件的方式, 同步 VPP 与 Linux 路由表.

通过比较, 方法 1) 相较于方法 2), 由于直接操作 VPP 路由表, 会有一定的性能优势, 但是, 需要较大程度的修改 Quagga 源代码, 对程序有较大的侵入性, 往往不利于开源软件的发布. 所以我们选择侵入性较小的通过增加 VPP 插件来同步路由表的方式.

(1) 如何通过 Quagga 同步 VPP 自身所维护的路由表.

(2) VPP 绑定的网卡需要运行特定的驱动而不是 Linux 提供的驱动, 所以无法被 Quagga 识别.

下面部分将针对上述两个问题进行解决.

针对第一个问题, VPP 为了减少陷入内核态所需要的开销完全运行于用户空间下, 所以需要自身维护一张路由表, 该路由表与 Linux 内核所维护的路由表结构相似. 如前文所述, Zebra 通过 Netlink 方式与 Linux 内核进行通信, 同时 Netlink 支持多播组的形式, 所以选择将 Zebra 改为通过 Netlink 多播的方式与 Linux 通信, 然后将 VPP 加入到该多播组中. 以此方式, 可以使 VPP 获取到 Quagga 对内核路由表的每一次更新, 然后 VPP 可以根据每次收到的 Netlink 消息更新自身的路由表, 以达到同步 VPP 路由表和 Linux 路由表的目的.

针对第二个问题, VPP 基于 DPDK 运行, 所以需要将所使用的网卡绑定 DPDK 所提供的驱动 igb_uio, 该驱动运行于用户空间下. 但是由于该网卡未绑定 Linux 所提供的驱动, 所以无法被 Linux 系统所识别, 导致不能被 Quagga 配置和管理. 当接收到的报文转发所需要的路由条目已存在与 VPP 路由表中, 将不会存在这个问题, VPP 将根据路由条目对该报文直接进行转发. 当 VPP 路由表中不存在与接收到的报文目的 IP 地址相符的路由条目, 则需要 VPP 将该报文传递给 Quagga 通过动态路由协议获取所需要的路由信息. 为解决这个问题, 本文选择使用 Linux TAP 虚拟网卡技术来解决这个问题. Tap 驱动程序的数据接收和发送并不直接和真实网卡打交道, 他在 Linux 内核中添加了一个 TUN/TAP 虚拟网络设备的驱动程序和一个与之相关连的字符设备/dev/net/tun, 字符设备 tun 作为用户空间和内核空间交换数据的接口. 当内核将数据包发送到虚拟网络设备时, 数据包被保存在设备相关的一个队列中, 直到用户空间程序通过打开的字符设备 tun

的描述符读取时, 它才会被拷贝到用户空间的缓冲区中, 其效果就相当于, 数据包直接发送到了用户空间. 运用此种技术, 在内核中创建一个虚拟网卡作为 VPP 绑定网卡的一个映射, 所有针对虚拟网卡的配置和操作将映射到 VPP 绑定的物理网卡上面. 以此种方式, 可以使 Quagga 对物理网卡进行控制.

根据以上两个方式设计出的路由器架构如图 2. 图中 VPP Plugin 作为增加的节点, 用于处理接收到的报文的路由工作. Physical NICs 和 Virtual Device 为一一对应的关系, 将由 VPP 管理的物理网卡映射到内核中进行操作.

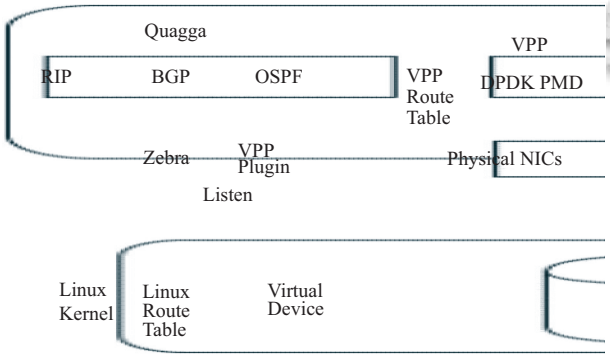

图 2 基于 VPP 优化的 Quagga 架构图

VPP Plugin 作为一个新增加的 VPP 节点完成以下工作: VPP 将所有接收到的报文进行预处理, 将物理层和数据链路程头部和尾部去掉, 然后将报文传递到新增加的节点, 新增节点工作流程如下:

1) 根据报文目的 ip 地址在 VPP 路由表中进行查找, 若存在则 2), 否则 3).

2) VPP 根据路由表进行路由转发.

3) 将报文上传给创建的虚拟设备, 并由该设备传递给 Quagga 进行路由.

4) Quagga 根据运行的动态路由协议与邻居进行交互并获取所允许的路由信息, 然后通过 Netlink 协议更新到 Linux 路由表中.

5) 新增节点通过监控 Netlink 多播协议, 将更新到 Linux 路由表中的内容同步到 VPP 路由表中, 然后 1).

工作流程图如图 3 所示至此, 虚拟路由器的数据平面已经转移到了用户空间下的 VPP 上, 而控制平面则保留在 Quagga 上面.

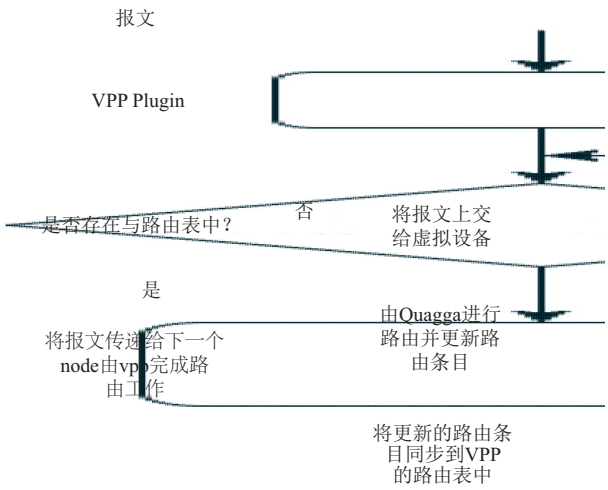

图 3 VPP 插件工作流程图

4 实验与分析

为了模拟真实情况下的虚拟路由器性能数据, 本文设计如下的拓扑结构进行实验: 两台标准 x86 服务器作为实验环境, 每台服务器上有一个两口的 intel10G 网卡, 将两台服务器各自的一个网口通过光纤与另一个服务器网口相连, 组成一个线性拓扑结构. 两台机器的另一个接口分别于 Traffic Generator 相连. 本实验选择 IXIA 设备作为 Traffic Generator 来进行试验, IXIA 是专业的网络测试设备, 可根据配置产生高速动态变化的网络报文. 本实验选择 BGP 作为动态路由协议进行测试. BGP 协议是基于 TCP 协议的动态路由协议, 作为自制系统间的动态路由协议而广泛应用于大型网络中. 为了进行精确测试, 本实验选择对单 core 下的性能进行分析.

实验步骤如下, 启动两台服务器的 Quagga 服务, 分别通过 shell 脚本自动注入 100 条左右的路由信息, 通过配置, 使两台服务器作为 BGP 协议中 Neighbor 关系相连并交换路由信息. 不同长度的报文长度将对性能产生较大的影响, 所以选择 64 bit 至 1513 bit 的不同长度的报文进行测试. 从 IXIA 端口向一台服务器的端口发送目的地址动态变化的报文, 在另一端的 IXIA 端口观察手包速率, 得出以下实验结果, 如表 1.

针对以上结果进行分析, 可以得出, 在 64 bit 至 256 bit 情况下, 相较于传统虚拟路由器, 转发性能有了极大的 提升. 在 256 bit 以上的情况, 单 core 的转发能

力基本可以达到线性速度. 本实验只针对单 core 性能进行测试, 通过结果可以推断出, 当增加 core 的数量时, 虚拟路由器将可以在任何报文长度条件下, 达到线性速度. 由此可以得出, 通过将虚拟路由器的数据平面转移到用户空间下, 可以较大的提升路由器性能. 此外, 通过对内核态下虚拟路由器的性能进行分析, 可以得到, 传统的数据平面转发方式已经难以满足高速网卡的需求.

表 1 实验结果

Program	Sent Framesize	Received Framesize	Cores	Input (Mp/sec)	Input (Mb/sec)	Perf (Mp/sec)	Perf (Mb/sec)
Quagga+Kernel Dataplane	64	64	1	14.882	952.383	0.745	47.887
Quagga+Kernel Dataplane	128	128	1	8.446	1081.081	0.747	90.108
Quagga+Kernel Dataplane	256	256	1	4.529	1159.422	0.703	181.641
Quagga+Kernel Dataplane	512	512	1	2.349	1203.007	0.703	348.346
Quagga+Kernel Dataplane	1024	1024	1	1.197	1226.503	0.728	759.506
Quagga+Kernel Dataplane	1512	1512	1	0.816	1233.681	0.713	1021.186
Quagga+VPP	64	64	1	14.882	952.385	5.852	374.871
Quagga+VPP	128	128	1	8.446	1081.081	5.803	743.091
Quagga+VPP	256	256	1	4.528	1159.403	4.582	1159.403
Quagga+VPP	512	512	1	2.349	1203.007	2.349	1202.993
Quagga+VPP	1024	1024	1	1.197	1226.053	1.197	1226.891
Quagga+VPP	1512	1512	1	0.816	1233.681	0.815	1233.699

5 总结与展望

本文提出了一种针对虚拟路由器数据平面加速的方法, 通过此种方法将虚拟路由器的数据平面转移到具有高速数据处理能力的用户空间数据高速转发工具上, 使得通用标准设备替代高速网络设备成为了可能. 以目前的虚拟路由器替代商业专用路由器还存在一些问题, 本文只是在收发和将报文交给 CPU 处理的环节进行了加速, 但对于整个路由器的核心功能路由选择却没有进行优化. 物理路由器通常选择使用专有硬件辅助的形式加速路由选择的过程, 而虚拟路由器单纯使用软件进行路由效率上有比较大的差距, 可以对这一方面进行深入研究, 利用多核 CPU 的优势改进路由选择算法.

参考文献

1 Gao XM, Zhang XZ, Lu ZX, *et al*. 面向虚拟路由器的转发平面改进机制. Proc. of Chinese Control Conference, 2013, (23): 0–1.

2 Mckeown N, Anderson T, Balakrishnan H, *et al*. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69–74. [doi: 10.1145/1355734]

3 Keller E, Green E. Virtualizing the data plane through source code merging. Proc. of the ACM Workshop on Programmable Routers for Extensible Services of Tomorrow. Seattle, WA, USA. 2008. 9–14.

4 Gupta M, Singh S. Greening of the internet. Proc. the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. Karlsruhe, Germany. 2003. 19–26.

5 Lyons AM, Neilson DT, Salamon TR. Energy efficient strategies for high design telecom application. Princeton: Princeton University, 2008.

6 Fu J, Rexford J. Efficient IP-address lookup with a shared forwarding table for multiple virtual routers. Proc. of the 2008 ACM CoNEXT Conference. Madrid, Spain. 2008.

7 Cisco. Multi-topology routing. http://www.cisco.com/c/en/us/td/docs/ios/12_2sr/12_2srb/feature/guide/srmtrdoc.html. [2006-08-18]

8 Turner JS, Crowley P, Dehart J, *et al*. Supercharging planetlab: A high performance, multi-application, overlay network platform. Proc. the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. Kyoto, Japan, 2007.

9 Feamster N, Gao LX, Rexford J. How to lease the internet in your spare time. ACM SIGCOMM Computer Communication Review, 2007, 37(1): 61–64. [doi: 10.1145/1198255]