

流驱动和表驱动相结合的流表控制算法研究

李云峰 唐畅 张益 瞿辉 王亚昕 马枢清

重庆金美通信有限责任公司

DOI:10.12238/acair.v2i4.10304

[摘要] 路由交换设备业务交换出接口按照路由表的引导完成,常用的路由表包括管理用途的路由表、转发用途的软件转发表和硬件转发表,三者在一概情况下一致。硬件转发引擎包括通用交换芯片和FPGA,在硬件转发引擎资源受限的情况下,硬件转发表项容量远少于软件转发表容量。现有的流表下发设计方式会造成硬件转发引擎的资源浪费,表项使用率不高。为实现硬件转发表的有效利用,本文提出基于流驱动和表驱动的流表控制算法,通过表项选择模块及时更新硬件转发表,使之当前活跃的路由转发表项一致,以增加表项匹配的活跃度,提升硬件转发有效性。

[关键词] 路由交换设备; 硬件转发引擎; 硬件转发; 路由转发表

中图分类号: TN393 文献标识码: A

Research on flow and table control algorithm combining flow driven and table driven

Yunfeng Li Chang Tang Yi Zhang Hui Qu Yaxin Wang Shuqing Ma

Chongqing jinmei communication Co.,Ltd.

[Abstract] The outgoing interface of the service switching device is guided by the routing table. Commonly used routing tables include the management routing table, the software forwarding table, and the hardware forwarding table. Generally, the three tables are the same. Hardware forwarding engine includes universal switching chip and FPGA. When the resources of hardware forwarding engine are limited, the capacity of hardware forwarding table is much less than that of software forwarding table. The current flow table delivery design will cause the hardware forwarding engine resources waste, the entry usage is not high. In order to realize the effective utilization of hardware forwarding table, we propose a flow table control algorithm based on flow driven and table driven, which updates the hardware forwarding table in time through the entry selection module to make the current active routing forwarding table entries consistent, so as to increase the activity of table entry matching and improve the effectiveness of hardware forwarding.

[Key words] Routing and Switching equipment; Hardware forwarding table; Hardware forwarding; Routing forwarding table

引言

随着网络规模的增大以及用户数量的增加,互联网所承载的数据量也呈指数级增长,灵活地控制路由交换设备对业务的开展和部署至关重要。其中,路由表是路由交换设备的控制核心^[1],存放着来自各种路由协议(如OSPF, BGP等)的路由,通常存放在路由引擎的DRAM内存中,而转发表是将路由表中选择出来的最佳路由附加上相关的转发信息的表。路由交换设备通过转发表来实现对网络包的转发,转发表的查找效率直接关系到路由交换设备的转发效率。初代路由设备采用CPU软件完成路由转发,转发的速度受到CPU处理能力的限制,随着网络传输速度不断提高,依靠软件转发的路由交换设备出现传输瓶颈。硬件转发引擎(通用交换芯片和FPGA)可以进行高速数据传输并具有强

大并行处理的能力,是当前解决高速数据流的主流方案^[2,3]。

路由转发表的下发方式分为业务流驱动和路由表驱动两种方式^[4]: 采用业务流驱动时,第一帧数据通过软件转发表进行转发,再实现对硬件转发表的操作,这种方式能有效利用转发表但第一帧转发速度较慢、延时大^[5]。而采用表驱动的业务直接由硬件转发引擎转发,表项容量取决于硬件转发引擎的存储能力。

因此,本文基于以硬件转发引擎为核心的数据转发平台,结合以上两种驱动方式的特点,设计了流驱动和表驱动相结合的方式生成动态的硬件转发表功能,能够提高转发效率以及有效利用硬件转发引擎的存储容量,具有重要的应用前景。

1 问题描述与分析

硬件转发引擎芯片可以进行高速数据传输并具有强大并行处理的能力,但存储资源非常有限^[6]。在有限面积情况下设计资源更多的芯片需要投入更多的成本和技术,因此客观上存在经济上的限制。此外,基于交换芯片对高速查询和处理的强烈需求,其芯片上的存取速度最快的存储器容量最小且无法无限扩大。

由于硬件转发表项容量远少于软件转发表容量^[7],传统的流表下发设计方式会导致硬件转发表中存在长时间内未使用的表项,称为非活跃表,这些非活跃表的存在将导致部分无效资源被占用,使得其中硬件转发表项实际使用率低,造成硬件转发引擎硬件存储资源浪费。

基于以上有限且复杂的存储资源条件下,如果硬件转发表项长期固定不变,当有新路由需要存储时就不可避免地会出现因内存不足导致无法完全存放转发表或因阶段不足而无法满足不同依赖关系的情况,这些问题会导致转发表查询失败,从而导致路由交换设备无法实现其功能。为实现硬件转发表的有效利用,提高转发效率,如何及时更新硬件转发表,使之与动态路由协议保持一致,成为硬件IP包正确选路与转发的关键。

2 总体实现设计方法

总体设计实现采用业务流驱动和表驱动相结合的方式,生成可用的动态硬件转发表。当数据包由硬件来进行转发时,通过内容可寻址存储器进行快速查找相应的路由信息实现高速转发。为了使硬件中维护的转发表得到路由协议的动态更新,与内核路由表保持同步,使得修改内核路由表的同时动态更新硬件转发表,这一内容依靠表项选择模块和硬件转发引擎驱动模块增删机制来实现硬件转发表的及时更新。其所在整个系统的层次结构及总体处理流程如图1所示。

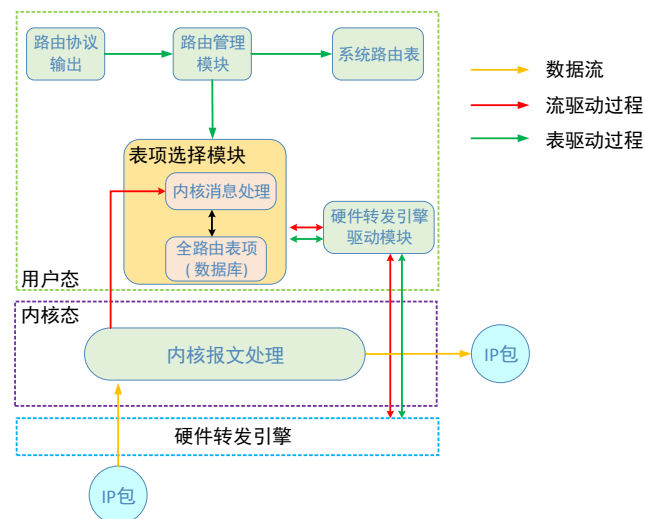


图1 方案总体设计

首先,由路由管理模块接收路由协议输出的路由后生成最优路由,将该路由添加进系统路由表,同时将最优路由发给表项选择模块,由表项选择模块存储该最优路由。当网络设备收到业务报文,首先在硬件转发表中查表,如果在硬件存储有其匹配的

路由转发表,则直接根据目的节点地址查表获得下一跳需要发往的节点地址。如果在硬件转发表查不到匹配项则发往CPU软件处理,CPU软件截取内核处理的相关信息,将查表相关的消息及所有命中路由由表通过netlink发送给用户态程序的表项选择模块。表项选择模块处理内核发来的消息,同时根据路由的情况生成硬件转发表项及硬件转发表项配置消息发往硬件转发引擎驱动模块,最后由硬件转发引擎驱动模块根据收到的配置消息完成硬件转发表的更新,最终完成动态硬件转发表项的生成。

3 软件模块设计

3.1 内核模块修改

内核模块主要有两种修改方式,一种是通过挂载K0的方式,在内核态将完成查表的业务报文勾出,由用户态程序根据目的地址匹配路由由表生成路由表项。通过K0挂载的方式修改内核模块实现起来比较容易且通用性强,但由于内核集成netfilter组件,影响软件处理效率,且用户态有再查表操作。另外一种方式是直接修改内核代码,在fib_lookup函数内部,将RT_TABLE_MAIN三层查表成功的表项返回给用户态程序,用户态程序直接获取路由表项。这种方式相对K0挂载方式通用性较差,需要根据项目Linux版本修改内核,但由于某些项目无法获取源码而无法直接修改内核代码,用户态有再查表操作(大小网段)。

3.2 表项选择模块设计

表项选择模块由内核消息处理模块、路由处理模块及转发路由表项模块组成,其负责接收路由管理模块发送的路由,通过处理内核模块发送的消息,生成硬件转发表项。具体实现过程如下:

步骤1:路由处理模块接收路由管理模块输出的路由,并将路由进行本地存储;判断路由选择模块的路由存储条目数量或者活动硬件表项数量是否达到硬件转发引擎支撑总条数的80%,若路由条目未达到硬件转发引擎支撑总条数的80%,则直接进入步骤5,否则进入步骤2;

步骤2:由内核消息处理模块接收内核消息,提取消息内容,若消息为路由消息,则进入步骤3;否则解析报文,提取目的IP,进入步骤4;

步骤3:路由处理模块根据路由消息查询路由存储模块的路由,找出该路由相关的所有大小网段的路由,进入步骤5。

步骤4:路由处理模块根据从内核消息提取的目的IP,查询路由存储模块,找出该目的IP相关的所有大小网段路由,将该大小网段路由生成硬件路由表项,进入步骤5。

步骤5:将存储的硬件路由表项发往硬件转发引擎驱动模块;

步骤6:由硬件转发引擎驱动模块接收下表模块反馈的活动硬件表项数量,详细实现见章节3.3;

3.3 硬件转发引擎驱动模块设计

硬件转发引擎驱动模块由表项维护模块、表项管理模块和表项下发模块组成,其负责下发从表项选择模块收到的路由表项,主要完成硬件转发表的定时维护,包括硬件转发表的动态更

新, 转发表项的一致性检查, 转发表项的呈现等。其具体处理流程如图2所示。

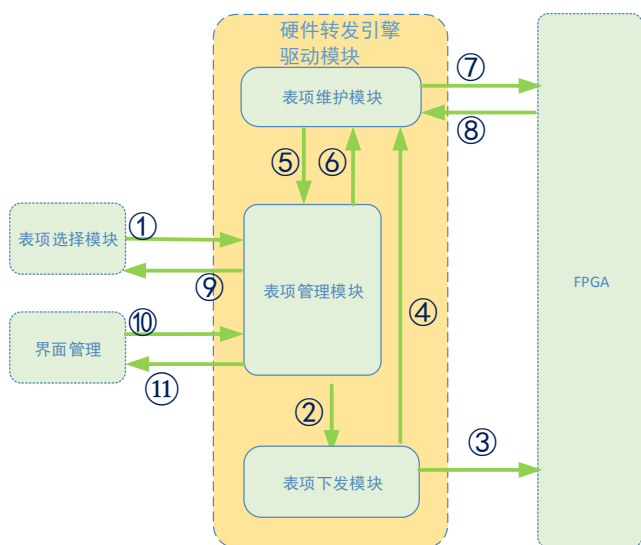


图2 硬件转发引擎驱动模块处理图

步骤1: 由表项管理模块接收表项选择模块的消息, 根据获取的表项资源生成硬件转发表项;

步骤2: 表项管理模块将更新的表项发往表项下发模块;

步骤3: 表项下发模块完成硬件写表操作;

步骤4: 表项下发模块向表项维护模块发送下表操作通告;

步骤5: 表项维护模块收到下表操作通告, 延时120s向硬件转发引擎查询路由表写表情况, 判断获取的已写入硬件转发表项与本地存储的表是否一致, 如果不一致, 则重新下表; 否则定时120s向硬件转发引擎查询路由表项使用统计消息;

步骤6: 表项维护模块收到表项管理模块路由表项消息;

步骤7: 表项维护模块统计表项命中记录, 15分钟内命中记录未变化, 认为该表为非活跃表, 根据规则生成删除表项发往表项管理模块, 完成相关操作;

步骤8: 表项管理模块将删除操作反馈给表项维护模块, 同时完成步骤2;

步骤9: 表项管理模块将活动表项的数量告知表项选择模块, 用于判断下表方式;

步骤10: 界面管理发送硬件表项查看请求;

步骤11: 界面收到硬件转发表项, 在界面显示。

3. 4 硬件交换逻辑适配一致性设计

交换芯片的硬件转发容量(8K)符合路由交换设备的最大指标要求, 因此针对交换芯片转发表的工作, 重点在于表的维护工作。目前相关的转发表有3张: 上层发送的硬件转发配置表(简称上层配置表), 转发配置记录表(简称本地配置表), 配置确认生效芯片自带的ipuc转发表(简称IPUC表)。上层配置表是上层软件下发的要配置的硬件转发表项; 本地配置表是基于上层配置

表配置后形成的表, 里面包含了配置成功或者失败信息; IPUC表是芯片提供的转发表, 呈现真实的IPUC配置成功表项。上层配置表和本地配置表表项一致, 本地配置表的表项大于或者等于IPUC配置表。

上层配置表将表项消息发往本地配置表, 本地配置表存储配置信息, 并将配置消息发往芯片。本地配置表接收IPUC配置表的配置反馈, 并记录配置成功和失败的表项, 将配置失败的表项发往失败缓存队列, 缓存队列在间隔一定时间后, 向本地配置表发起配置操作, 本地配置表重新配置表项, 表项配置成功后, 本地配置表将记录的该配置失败表项置位, 同时向失败表项缓存队列删除该失败表项。以此实现交换芯片的IPUC表和本地配置表的一致性。

4 结论

本文设计一种基于硬件转发引擎硬件转发的动态表项设计方案, 通过设计的表项选择模块以及硬件转发引擎驱动模块及时更新硬件转发表, 使硬件转发表与动态路由协议保持一致以增加表项匹配的活跃度, 增加表项的实际使用率, 能够有效提升硬件转发效率。

【参考文献】

[1]安淑芝. 计算机网络技术与应用[M]. 北京: 清华大学出版社, 2009: 102-107.

[2]谢立军, 胡宇翔, 汪斌强, 等. 网络可编程技术研究[J]. 电信科学, 2016, 32(2): 1-12.

[3]张威. 基于FPGA的高速以太网接口设计和实现[D]. 电子科技大学, 2016.

[4]范明慧, 彭澎, 董国英. 基于表驱动的路由协议设计与FPGA实现[J]. 制导与引信, 2024, 45(2): 40-49.

[5]杜旭, 杨明, 周宗照. 路由器软硬件转发表同步的设计与Linux实现[J]. 计算机应用研究, 2003, 10(1): 222-224.

[6]彭鼎祥. 大容量高带宽路由查找算法设计与FPGA实现[J]. 现代电子技术, 2023, 46(15): 20-24.

[7]刘渊. 一种支持动态名字查找的NDN网络路由转发表设计[J]. 小型微型计算机系统, 2017, 38(6): 1192-1197.

作者简介:

李云峰(1988--), 男, 本科, 工程师, 研究方向: 软件集成、路由组网交换。

唐畅(1999--), 女, 硕士, 助理工程师, 研究方向: IP路由、IP组播。

瞿辉(1979--), 男, 管理学学士, 高级工程师, 研究方向: IP路由、IP组播。

王亚昕(1981--), 男, 硕士, 高级工程师, 研究方向: SDN技术、网络内生安全。