

源端控制的 OpenFlow 数据面

王哲^{1,2}, 梁满贵^{1,2}, 及晓萌^{1,2}, Phillipus Mateus Abisai^{1,2}

(1. 北京交通大学 计算机与信息技术学院, 北京 100044; 2. 北京市现代信息科学与网络技术重点实验室, 北京 100044)

摘要: 为了实现流表的多元快速查找, OpenFlow 交换机一般采用 TCAM 存储和查找流表, 从而带来了扩展性、成本和能耗的问题。尽管可以采取流表压缩、引入 RAM 存储器等方法, 但仍无法彻底解决使用 TCAM 造成的局限性。针对这个问题提出了源端控制的 OpenFlow 数据面模型, 即 SCOF (source-controlled OpenFlow)。它以一种源路由地址—向量地址 (VA) 作为数据分组的交换标签, VA 完全定义了通信路径。SCOF 的数据转发设备是向量交换机, 不需要存储和查找流表, 只根据 VA 即可完成数据分组转发。SCOF 模型降低了交换机硬件复杂度, 简化了流表更新过程, 克服了 OpenFlow 的扩展性问题。

关键词: 软件定义网络; 流表; 三态内容寻址存储器; 向量地址; 现场可编程门阵列

中图分类号: TP393.0

文献标识码: A

Source-controlled OpenFlow data plane

WANG Zhe^{1,2}, LIANG Man-gui^{1,2}, JI Xiao-meng^{1,2}, Phillipus Mateus Abisai^{1,2}

(1. Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China;

2. Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China)

Abstract: An OpenFlow switch typically used TCAM to speed up flow table matching. Therefore it brought problems with scale, high cost and energy consumption. To resolve these issues, a novel model called the source-controlled OpenFlow(SCOF)was proposed. SCOF used a source routing address named vector address (VA) as the switching label of the packet. VA defined the communication path. The core of SCOF data plane is vector switch. It can accomplish the data forwarding process relying only on VA, instead of flow table matching. SCOF overcomes scale problem by reducing the complexity of switch hardware and simplifying the process of updating flow table.

Key words: SDN; flow table; TCAM; vector address; NetFPGA

1 引言

软件定义网络 SDN^[1]的基础思想是控制面和数据面相分离, 它将网络设备抽象为应用程序接口提供给应用层, 构建了可编程和集中控制的网络环境。OpenFlow^[2,3]是第一个符合 SDN 要求的网络接口规范。它把网络划分为以 OpenFlow 控制器 (OF-C) 为代表的控制面、以 OpenFlow 交换机 (OF-S) 为代表的数面面和 OpenFlow 协议。控制

面通过 OpenFlow 协议, 对数据面的流表 (Flow Table^[4]) 进行添加、删除和更新等操作, 从而实现

对网络中数据流的控制。OpenFlow 交换机依据本地存储的流表匹配数据分组并进行转发。为了实现快速查表, 流表通常采用 TCAM 来进行存储和查找。TCAM 是一种三态内容寻址存储器, 可以实现高速并行查找, 并且支持掩码形式的通配符。但 TCAM 成本和能耗高, 存储容量有限, 一般交换机中的 TCAM 仅能存

收稿日期: 2013-12-04; 修回日期: 2014-01-20

基金项目: 国家重点基础研究发展计划(“973”计划)基金资助项目(2011CB302203); 国家高技术研究发展计划(“863”计划)基金资助项目(2007AA01Z203); 文化部民族民间文艺发展中心数字文化研究基地基金资助项目(136023522)

Foundation Items: The National Key Basic Research Program of China (973 Program) (2011CB302203); The National High Technology Research and Development Program of China (863 Program) (2007AA01Z203); Center for Ethnic and Folk Literature and Art Development, Ministry of Culture(136023522)

储几千条流表项，对 OpenFlow 在大规模网络中的应用造成了限制。

为此，本文提出了源端控制的 OpenFlow 数据面模型，即 SCOF (source-controlled OpenFlow)。它以一种源路由地址——向量地址 (VA, vector address) 作为数据分组的交换标签，VA 独立完整的定义了通信路径，交换机在收到数据分组后只依据 VA 即可完成转发处理。SCOF 的优点如下：1) 数据交换不需要查表，使交换机不必采用 TCAM 进行流表存储和查找，从而降低了交换机复杂度和成本，根据在 NetFPGA 平台上实现的 SCOF 交换机对比统计，其硬件资源消耗约为 OpenFlow 交换机的 30%；2) SCOF 简化了 OpenFlow 的流表更新过程，新的数据流建立时不需要下载流表项到交换机，使流表更新所需的控制信令数量大大减少。

2 OpenFlow 存在问题简述

OpenFlow 交换机中存储有一张或多张流表，每张流表由若干流表项 (flow entry^[4]) 组成。流表项组成结构如图 1 所示。在每条流表项中，匹配域 (match fields) 用来定义流的特征，它包含数据分组的入端口 (ingress port)、L2/L3/L4、VLAN ID 等多项，此外还可以扩展包含流表之间传送的元数据 (metadata)；计数器 (counters) 用来对数据流量进行统计，操作字段 (instructions) 则定义了对数据分组的转发操作。

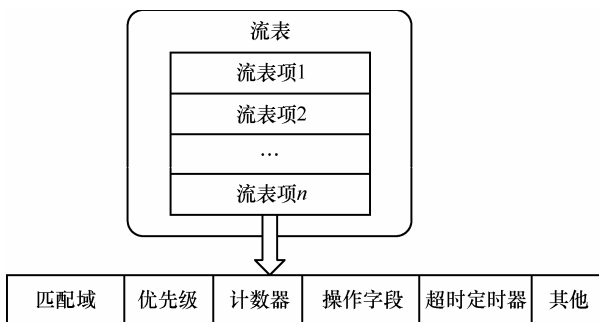


图 1 流表项的组成结构

当交换机收到一个数据分组，它根据入端口和分组头等信息，查找每个流表项的匹配域。若匹配成功，更新 counters，按照 instructions 字段执行转发处理（发送到某个端口、丢弃、更新分组头 TTL 等）。当有多项匹配成功时，根据优先级字段选择一项。若匹配失败，OF-S 将此分组发送给 OF-C，OF-C 启动路由过程，通过 OpenFlow 协议在交换机

中增加一条新的流表项，用来指示交换机如何处理此分组（转发或丢弃）。

OpenFlow 交换机为了实现流表的多元并行查找，通常采用 TCAM 作为流表存储器。TCAM 可以在一个查找周期内完成整个表的匹配，具有查找速度快、指令操作简单的优点，因此广泛应用于路由和 ACL 中。然而它也存在以下缺陷：1) TCAM 存储容量有限，较大的 TCAM 芯片容量仅为 36 MB，一般交换机采用的 TCAM 容量更小；2) TCAM 要求将分组分类规则转换为三元格式，这带来了范围扩张问题 (range expansion problem^[5])，经过格式转换后 TCAM 表项数量大大增加， w bit 表示的范围转换为前缀规则，在最坏情况下需要 2^w-2 条表项，加重了对 TCAM 存储容量的压力；3) TCAM 价格高昂，耗能高，每 MB 耗能约是 SRAM 的 30 倍^[6]，随之带来了散热的问题。这些问题使采用 TCAM 的 OpenFlow 交换机流表存储容量有限，限制了其可扩展性，无法适用于大规模的网络环境。

研究者们为此提出了一些应对方法，如流表压缩算法^[6,7]，结合 TCAM、SRAM、和 NPU 等设计流表存储器^[8,9,10]，调整流的辨识粒度^[11]等。例如文献 [6] 中 Bit Weaving 算法平均压缩比达到 23.6%。但随着网络规模的增大和应用的复杂，以及网络安全控制机制的实施，流的分类规则变得更加复杂，以上方法并不能彻底解决流表存储容量的限制问题。

可以看到，OpenFlow 交换机依据流表查找转发的特性是限制其大规模应用的一个重要因素。在 OpenFlow 中，网络根据交换机中的流表和分组头共同标识一条数据传输路径，流表在数据转发过程中必不可少。因此，本文提出以向量地址 VA 作为一种新的交换标签，只通过 VA 即可完整的标识一条数据分组传输路径，使数据转发不再依赖于流表。

3 VA 和 SCOF 模型

3.1 一种源路由地址 VA

传统分组交换网有 2 种类型的网络地址编码方式：1) 以网络节点为地址编码对象，如 IP 网，为每个节点分配一个地址；2) 以节点间的虚链路为地址编码对象，如 MPLS、ATM。这 2 类地址编码均不能独立标识一条网络路径，例如在 IP 网中，网络路径由分组头 IP 地址和路由器中的路由表共同表示，在 MPLS 中，网络路径由分组头的交换标签和路由器中的标签转发表 LFIB 共同表示。OpenFlow 中数

据流的转发路径由数据分组和交换机中的流表共同表示，因此其数据面转发行为同时依赖于数据分组特征字段和流表。

本文采用另外一种新的网络地址编码，称为向量地址 VA^[12,13]，它是一种源路由地址。这种地址编码方法不对节点和虚链路编码，而是对交换机的端口进行编码，交换机为其拥有的网络端口依次分配一个本地序号 (PI, port index)。将通信路径上交换机的相应本地序号依次组合而成的序列，称为向量地址。下面结合图 2 说明 VA 的含义。

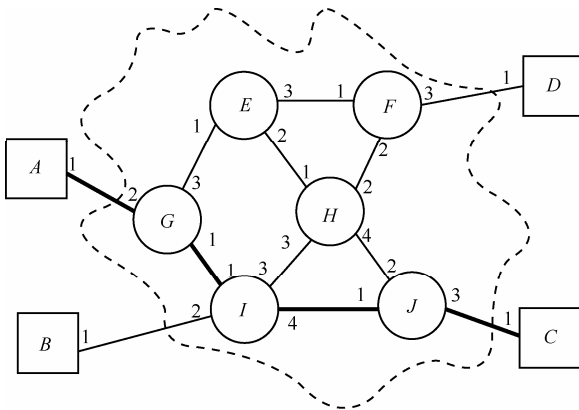


图 2 向量地址 VA

如图 2 所示，包括终端系统 ABCD，交换机 EFHIJG。以从端系统 A 到端系统 C 通信为例，其通信路径设为 A→G→I→J→C，各个节点相应的输出端口的本地序号为 A: 1、G: 1、I: 4 和 J: 3，将这些序号组合起来，就是从 A 到 C 的向量地址 VA={1,1,4,3}，这个数字序列完整的指示了从节点 A 到 C 的一条路径，所以它可以作为从节点 A 到节点 C 的路径标识。

源终端发送数据分组时，将 VA 作为交换标签存放于分组头中，如图 3 所示。当交换机收到数据分组，读取分组头的 VA，取出第一个端口序号 PI₁，将 PI₁ 从 VA 中切掉，将剩余部分 VA'={PI₂~PI_n} 重新封装入数据分组头，沿 PI₁ 所指定的出端口将数据分组转发出去。随着数据分组在传输路径上不断转发，VA 也不断消耗和变小，直至到达目的节点，VA 消耗完毕。这就是使用向量地址的数据转发过程，称为向量交换。可以看到，在向量交换中，交换机完全不需要查转发表，只根据分组头中的 VA 即可完成转发。即网络只需要根据 VA，就可以确定一条唯一的数据传输路径，因此向量地址是一种源路由地址。使用向量地址作为交换标签的数据分组称为向量分组。将 VA 应

用于 OpenFlow 的数据面，即可定义 SCOF 模型。

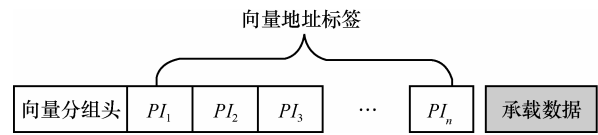


图 3 数据分组头的向量地址字段

3.2 SCOF 模型

3.2.1 SCOF 网络构成

如图 4 所示，SCOF 网络由终端机、OpenFlow 交换机 (OF-S)、向量交换机 (VS, vector switch)、OpenFlow 控制器 (OF-C) 和 SCOF 控制器 (SCOF-C) 组成。

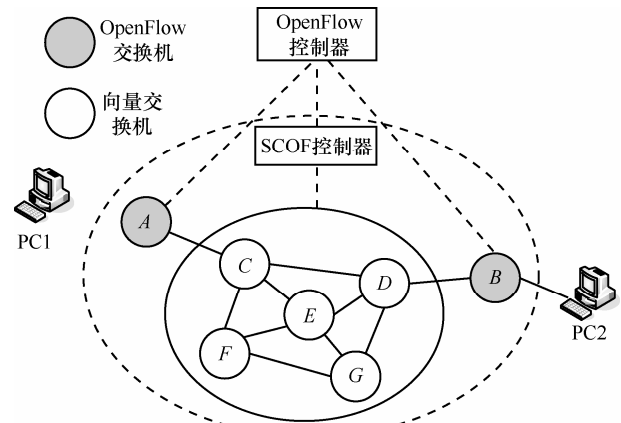


图 4 SCOF 网络组成结构

OF-S 是边沿网关 (图 4 中的 A 和 B)，它连接了终端机和 VS，负责路由请求、向量分组和其他数据分组 (如 IP、MPLS) 的转换等；

VS 是支持向量交换的交换机。它位于 SCOF 网络核心，负责高速数据转发，采用向量交换方式，在分组转发过程中，不需要查表，因此相比于 OF-S，VS 不需要维护流表；

OF-C 负责响应 OF-S 的路由请求，在 SCOF-C 的协助下完成路由的计算，并维护 OF-S 中的流表项，它和 OF-S 之间通过 OpenFlow 协议通信；

SCOF-C 负责由 VS 组成的核心网的拓扑搜集、路由和路径表示等，它和 OF-C 之间通过 SCOF 通道通信，协助 OF-C 完成 VS 间的路径计算，提供 VA。

以图 4 为例，详细说明 SCOF 网络的组成结构。在图 4 所示的网络中，PC1 和 PC2 是 2 台 IP 终端，A 和 B 是 2 台 OpenFlow 交换机，C 至 G 是 5 台向量交换机，OF-C 是 OpenFlow 控制器，SCOF-C 是 SCOF 控制器。

1) OpenFlow 交换机

OF-S 作为边沿网关,负责连接用户和核心网的 VS。它的任务有: 1) 查找流表, 实现向量分组和其他类型数据分组之间的转换; 2) 收到新的数据流时, 启动路由过程, 向 OF-C 发出路由请求。通过 OF-S, SCOF 网络可以提供对 IP、MPLS 等各种类型终端机的支持。

OF-S 中的流表项与 OpenFlow 标准 (图 1) 相比, 在 instructions 字段中增加了向量分组的封装和解封操作。当它收到源终端发来的原始数据分组, 首先根据匹配域在流表中查找。若命中, 则根据相应表项的 instructions 在原始数据分组头处添加 Head+VA (Head 是向量分组的分组头标记, 如图 3 所示), 从而将数据分组封装为向量分组, 并转发至相应输出端口; 若查表未命中, 则 OF-S 向 OF-C 发出一条路由请求消息, 控制器经过路由得到一条新的流表项 (其中包含了路径信息, 以 VA 形式标识), 配置到 OF-S 的流表中, 指示其如何处理此分组。

另一方面, 当 OF-S 收到经过 VS 从核心网发来的向量分组, 此时分组头中的 VA 地址已经在转发过程中全部消耗, 去除分组头的 Head 字段, 将向量分组解封为原始数据分组, 最终交付到目的终端机。

2) 向量交换机

SCOF 网络边沿使用了 OpenFlow 交换机, 当进入到核心网后, 随着流量的集中, 在高带宽大流量的情况下, 采用向量交换机。原始数据分组经过边沿网关的封装, 转换为向量分组, 进入核心网。VS 之间进行向量交换, 当 VS 收到向量分组, 不再查找流表, 而是根据向量交换方法进行转发。

在 OpenFlow 中, 当一条新的数据流出现, 途径的 OF-S 都需要增加流表项。在核心网中, 随着数据流的汇聚, OF-S 有限的流表存储容量往往不能满足需求, 这就形成了瓶颈。而在 SCOF 中, 当一条新的数据流出现, 只需要在边沿网关 OF-S 上配置一条流表项记录, 而核心网中的 VS 不需要增加。即网络数据流的复杂度不会增加交换机表项存储和查找代价, 使 VS 克服了 OF-S 流表存储容量的限制, 因此具有在大规模网络中部署应用的能力, 如数据中心、骨干网等流量很大的网络。同时由于 VS 在转发中不需要查表, 从而避免了使用 TCAM 存储器带来的成本、能耗和散热等问题, 简化了交换机结构, 降低了其复杂度。

3) OpenFlow 控制器和 SCOF 控制器

在 SCOF 控制面 (如图 5 所示) 中, OpenFlow

控制器负责控制 OpenFlow 交换机, 并在 SCOF 控制器的协助下, 搜集网络拓扑和进行路由。

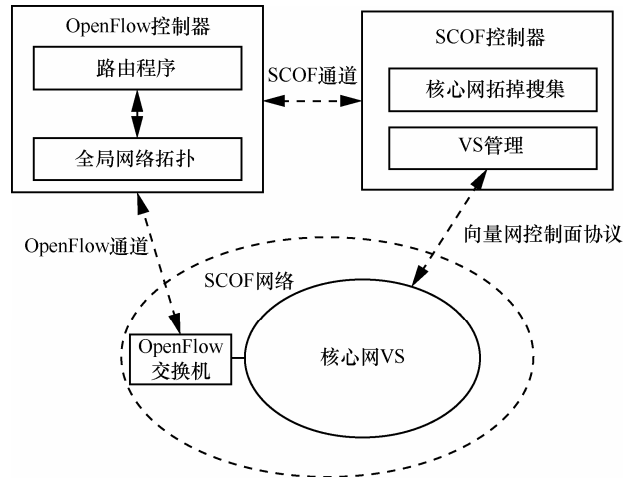


图 5 SCOF 控制面

OF-C 通过 SCOF-C 通道和 SCOF-C 通信, 维护全局网络拓扑, 在路由程序的支持下, 响应 OF-S 的路由请求, 生成新的流表项并通过 OpenFlow 通道配置到 OF-S 中。流表项中包含了对原始数据分组的向量分组封装和解封操作。

SCOF-C 通过向量网控制面协议^[14,15], 管理核心网中的 VS 并进行拓扑搜集, 将拓扑搜集结果上报到 OF-C 中。VS 不需要与 OF-C 直接通信, 只需配合 SCOF 控制器完成极少部分信令处理即可, 相比 OF-S 信令处理能力得到了简化。

3.2.2 SCOF 网络通信过程

以图 4 所示的 SCOF 网络中, 2 个 IP 终端 PC1 和 PC2 通信为例, 说明 SCOF 网络的数据分组传输过程 (假设其通信路径为 $A \rightarrow C \rightarrow D \rightarrow B$):

- 1) PC1 发出到 PC 2 的 IP 数据分组 Pkt_{ip} ;
- 2) A 收到 Pkt_{ip} 后, 查找流表, 若查找命中, 则直接转 5); 若查找未命中, 向 OF-C 发出路由请求;
- 3) OF-C 运行路由程序, 在 SCOF-C 协助下得到一条路径, 回应 A 的路由请求, 并分别向 A 和 B 下发新的流表项;

4) A 收到 OF-C 发来的路由回应, 将新的流表项添加到本地流表, 并开始 Pkt_{ip} 的转发处理;

5) A 根据查找命中的流表项, 在 Pkt_{ip} 中添加向量分组头 Head 和向量地址字段 VA, 将 IP 分组转换为向量分组 Pkt_{va} , 即 $Pkt_{va} = Head + VA + Pkt_{ip}$ 。设 VA 给出的 A 到 B 的通信路径经过 C 和 D, 则转发给向量交换机 C;

6) C 收到向量分组 Pkt_{va} ，以向量交换的方式转发到 D ，依次类推，注意随着每一跳的转发， VA 由于消耗会逐渐变短，直至到达 B ；

7) B 收到向量分组 Pkt_{va} 。此时分组头中的 VA 在转发过程中消耗干净，查找流表，将向量分组头 Head 去除，恢复出原始 IP 分组 Pkt_{ip} ，转发给 $PC2$ ；

8) $PC2$ 收到 Pkt_{ip} ，数据转发全部完成。

同样， B 到 A 的数据传输过程相同，只是方向相反，这样，对于 A 和 B 就建立起相互的 IP 通信关系。

3.2.3 SCOF 的路由过程

根据文献[4]中协议，OpenFlow 路由过程如下：

1) 当 OF-S 收到一个新的数据流（未能匹配到流表项），其启动路由过程，向 OF-C 发出一条 PACKET-IN 消息，请求控制器进行路由；2) OF-C 收到 PACKET-IN 消息后，计算得到一条路径，通过 modify state message 向此路径上的所有 OF-S 中添加流表项，从而形成这个新的数据流的路由；3) OF-C 向 OF-S 发出 PACKET-OUT 消息，OF-S 收到后，根据新的流表项进行数据分组转发。注意对于此数据流的途经路径上的所有 OF-S，OF-C 都需要发送一条 modify state message 增加新的流表项。

而在 SCOF 模型中，数据流的传输路由由向量分组头中的携带的 VA 决定，核心网 VS 收到向量分组，解析 VA 并进行转发。 VA 由边沿网关 OF-S 通过路由得到，其过程如下：1) OF-S 收到终端发来的新的数据分组，启动路由过程，向 OF-C 发出一条 PACKET-IN 消息，请求控制器进行路由；2) OF-C 收到 PACKET-IN 消息后，在 SCOF-C 协助下计算得到一条路径，通过 modify state message 向路径上的 OF-S 添加流表项；3) OF-C 向 OF-S 发出

PACKET-OUT 消息，OF-S 收到后，根据新的映射表项进行数据分组转发处理。

在 SCOF 路由过程中，OF-C 只需要在边沿的 OF-S 中配置流表，对网络中的 VS 不需要进行配置，这样就去除了 VS 的流表下载，相比 OpenFlow 的路由更新过程得到了简化。

4 SCOF 和 OpenFlow 的比较

SCOF 模型采用 VA 作为交换标签对 OpenFlow 进行了改进，核心网数据面设备采用向量交换机，进行向量交换。下面从交换机复杂度和流表更新 2 个方面对 SCOF 和 OpenFlow 进行了比较，说明 SCOF 的优势。

4.1 基于 NetFPGA 的交换机 VS 和 OF-S 的比较

NetFPGA^[16]是一个为研究者提供吉比特网络实验的硬件实验平台。它采用了 Xilinx Virtex-2 Pro FPGA 作为主处理芯片，板载资源包括 SRAM 4.5 MB、DRAM 64 MB、4 个千兆以太网接口和一个 PCI 插口。其中 Virtex-2 Pro 主控芯片包含 23 616 个 Slice、47 232 个 DFF 和 LUT、232 个 BRAM。

文献[10]提出了一种在 NetFPGA 上实现的 OpenFlow 交换机的参考设计。它采用了 2 种流表存储器，基于 SRAM 的精确查找表和基于 TCAM 的通配符查找表。前者采用片外 SRAM 实现，最多可以存储 32 768 条表项；TCAM 采用片内 Xilinx SRL16e 实现，可以存储 32 条 32 bit 的表项。另外，NetFPGA 小组还给出了 IPv4 路由器和千兆以太网交换机 2 种参考设计。

在 NetFPGA 平台上设计实现了向量交换机，它能够完成 4 个吉比特网口的交换功能，其结构如图 6 所示。

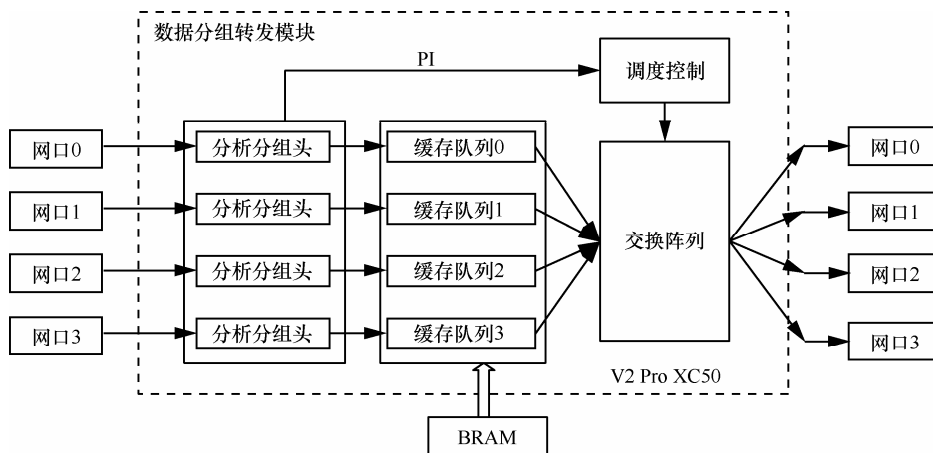


图 6 基于 NetFPGA 的向量交换机

网卡物理端口收到数据分组后,转发模块首先分析分组头,读取分组头中 VA 的当前分量 PI,调度控制模块依据 PI 进行交换阵列的调度。VA 处理完毕后,数据分组分别进入缓存队列 0~3 中,等待交换阵列的调度,发送到相应的输出端口。缓冲队列均采用片上的 BRAM 实现,未使用片外的 DRAM 和 SRAM,从而避免了片外存取的速度瓶颈和资源开销。

对在 NetFPGA 上实现的 VS 进行了硬件资源消耗的统计,统计内容包括 Slice、LUT、DFF 和 BRAM 等,并和 OF-S、IPv4 router、以太网交换机等参考设计进行了对比,结果如表 1 所示。

表 1 几种交换机的 NetFPGA 实现的硬件资源消耗统计

资源	向量交换机	OpenFlow 交换机	IPv4 路由器	以太网交换机
LUT	2 778	10 721	14 622	4 050
DFF	2 443	7 578	3 908	1 136
BRAM	34	18	20	100
Slice	1 924	6 924	—	—
片外存储	—	SRAM, DRAM	SRAM, DRAM	—

注:表中 VS 是全部模块消耗的资源统计,而 OF-S、IPv4 Router 和 Ethernet Switch 均只统计了 Output Port Lookup 模块消耗的资源。

从表 1 中可以看到,相比于 OF-S,VS 的优势如下:1)片内资源消耗少,VS 所消耗的片内 LUT、DFF 等资源约为 OF-S 的 30%;2)VS 不需要采用片外存储 SRAM,而 OF-S 需要使用 SRAM 作为流表存储器,并且其流表更新需要通过 PCI 总线进行,增加一条新的流表项至少需要 11 μ s,这限制了流表更新的速度;另外,OF-S 需要使用片外的 DRAM 缓冲数据分组,而 VS 的数据分组缓冲区大小仅为 40 KB,采用片内 BRAM 实现,避免了对片外 DRAM 的占用,存取速度也更快。

4.2 SCOF 和 OpenFlow 的路由比较

为了比较 SCOF 和 OpenFlow 的路由过程,分别统计在相同的网络拓扑和数据流路径条件下,2 种网络在选路时所消耗的信令条数(只计算了在已知拓扑的条件下,配置转发路径的信令条数,不包括拓扑搜集和路由计算,假设这部分二者的开销相同)。设数据流都经过 n 个网络交换节点。

在 OpenFlow 网络中,第一台收到数据分组的 OF-S 路由所需要的信令条数为 3 (PACKET-IN, modify state message 和 PACKET-OUT),其他 $n-1$ 台 OF-S 所需要的信令条数为 1 (modify state mes-

sage),共消耗信令条数为 $n+2$ 。即在 OpenFlow 网络中,路由和流表更新的信令条数和路径跳数 n 为线性关系。

在 SCOF 网络中,边沿 2 台 OF-S 所需要的信令条数为 4 (1 条 PACKET-IN, 2 条 modify state message 和 1 条 PACKET-OUT),核心网中的 $n-2$ 台 VS 不需要流表更新信令,共消耗信令条数为 4。即在 SCOF 网络中,路由和流表更新的信令条数和网络跳数无关。与 OpenFlow 相比,可以得到 SCOF 和 OpenFlow 在路由和流表更新中所消耗的信令数量之比为 4:($n+2$), n 为数据流经过的网络交换节点数量。文献[17]的研究表明,中国 Internet 网络跳数平均为 13。若取 n 为 13,则在新的数据流建立时,SCOF 所需的信令数量约是 OpenFlow 的 26%,即 SCOF 的流表更新过程相比 OpenFlow 得到大大简化。

5 结束语

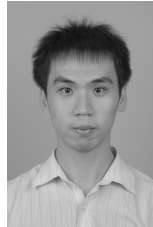
本文在 OpenFlow 研究成果的基础上,针对 OpenFlow 交换机流表存储容量的局限性,以一种源路由地址 VA 作为交换标签,对 OpenFlow 数据面做出了改进,提出了 SCOF 模型及其相应的路由方法。SCOF 网络的核心设备是支持向量交换的 VS,不需要查找流表即可完成数据流的转发处理,从而降低了交换机复杂度,根据对 NetFPGA 平台上实现的 VS 的统计,其硬件资源消耗约为 OpenFlow 交换机 OF-S 的 30%;另外,在流表更新过程中,SCOF 所需的控制信令数量约为 OpenFlow 的 26%。

参考文献:

- [1] OPEN NETWORKING FOUNDATION. SDN Website Homepage [EB/OL]. <http://www.opennetworking.org>,2013.
- [2] NICK M,TOM A,HARI B, *et al.* OpenFlow: Enabling innovation in campus networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2):69-74.
- [3] OPEN NETWORKING FOUNDATION. Software-Defined Networking: The New Norm for Networks[EB/OL]. <http://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>, 2013.
- [4] OPEN NETWORKING FOUNDATION. OpenFlow Specification V1.4.0[EB/OL].<http://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf>. 2013.
- [5] WANG P C. Scalable packet classification for datacenter networks[J]. IEEE Journal on Selected Areas in Communications, 2014, 32(1): 124-137.

- [6] MEINERS C R, LIU A X, TORNG E. Bit weaving: A non-prefix approach to compressing packet classifiers in TCAMs[J]. IEEE/ACM Transactions on Networking (TON), 2012, 20(2): 488-500.
- [7] BREMLER B A, HENDLER D. Space-efficient TCAM-based classification using gray coding[J]. IEEE Transactions on Computers, 2012, 61(1): 18-30.
- [8] LONG F, SUN Z, ZHANG Z, *et al.* Research on TCAM-based Openflow switch platform[A]. Proceedings of IEEE International Conference on Systems and Informatics[C]. Yantai, China, 2012. 1218-1221.
- [9] KHAN A, DAVE N. Enabling hardware exploration in software-defined networking: A flexible, portable OpenFlow switch[A]. IEEE 21st Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)[C]. Seattle, Washington, 2013. 145-148.
- [10] NAOUS J, ERICKSON D, COVINGTON G A, *et al.* Implementing an OpenFlow switch on the NetFPGA platform[A]. Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems[C]. San Jose, 2008. 1-9.
- [11] KUDOU H, SHIMAMURA M, IKENAGA T, *et al.* Effects of routing granularity on communication performance in OpenFlow networks[A]. Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing[C]. Vlcctorla, Canada, 2011. 590-595.
- [12] LIANG M G, ZHANG J X, WANG S J. A new network based on vector address[A]. Proceedings of IET 2nd International Conference on Wireless, Mobile and Multimedia Networks[C]. Beijing, China, 2008.118-122.
- [13] ZHANG J, LIANG M. A hierarchical networking architecture based on new switching address[A]. Proceedings of the 2008 International Conference on Advanced Infocomm Technology[C]. Shenzhen, China, 2008. 9.
- [14] 田萌, 梁满贵. 组长探测和节点相应的向量网拓扑发现方法[EB/OL]. <http://www.paper.edu.cn/releasepaper/content/201307-414>, 2013.
- TIAN M, LIANG M G Node Answering on Leader Probing Topology Detecting Method Based on Vector Network[EB/OL]. <http://www.paper.edu.cn/releasepaper/content/201307-414>, 2013.
- [15] 黄照阳, 梁满贵. 基于向量网的网域内拓扑发现[EB/OL]. <http://www.paper.edu.cn/releasepaper/content/201204-239>, 2012.
- HUANG Z Y, LIANG M G. Intradomain Topology Discovery Based on Vector Network Technique[EB/OL]. <http://www.paper.edu.cn/releasepaper/content/201204-239>, 2012.
- [16] NetFPGA Team. netFPGA Website Homepage[EB/OL]. <http://netfpga.org/>, 2013.
- [17] ChEN X, XING L, MA Q. A distributed measurement method and analysis on Internet hop counts[A]. Proceedings of International Conference on Computer Science and Network Technology[C]. Harbin, China, 2011. 1732-1735.

作者简介:



王哲 (1987-), 男, 河北邢台人, 北京交通大学博士生, 主要研究方向为软件定义网络和 OpenFlow、网络路由等。

梁满贵 (1958-), 男, 河北张家口人, 北京交通大学教授, 主要研究方向为网络体系架构、下一代互联网、语音信号处理等。

及晓萌 (1989-), 女, 河北沧州人, 北京交通大学硕士生, 主要研究方向为 NetFPGA、网络交换等。

Fillipus Mateus Abisai (1984-), 男, 纳米比亚人, 北京交通大学博士留学生, 主要研究方向为软件定义网络和并行计算。