

# 一种解析与执行联动的 SDN 可编程数据平面

孙鹏浩, 兰巨龙, 胡宇翔, 段 通

(国家数字交换系统工程技术研究中心, 河南郑州 450002)

**摘 要:** 当前互联网技术发展迅速, 新型网络协议的不断出现, 要求网络转发设备能够及时提供对新协议的支持. 目前, 软件定义网络要兼顾可编程协议解析和数据转发性能仍然面临诸多困难. 对此, 本文提出了基于解析和执行联动结构的可编程数据平面 (CLIFE), 通过在硬件的解析器上部署用户可定义模块, 可实时更新硬件中解析逻辑中的协议多叉树, 从而实现协议解析的用户定制性; 并且, 通过解析器和动作执行器联动的创新结构, 减少了整个处理架构的冗余性, 从而减小动作执行时延, 提高了硬件资源利用率, 与现有方案相比, 节约了 11% 的逻辑资源和 24% 的 BRAM 资源. 最后, 本文基于 NetFPGA-10G 板卡完成了本方案的原型机实现.

**关键词:** 网络数据平面; 灵活解析; 动作处理器; 可编程

**中图分类号:** TP393      **文献标识码:** A      **文章编号:** 0372-2112 (2017)05-1103-06

**电子学报 URL:** <http://www.ejournal.org.cn>      **DOI:** 10.3969/j.issn.0372-2112.2017.05.012

## A Configurable SDN Dataplane Based on Linkage Between Parsing and Executing

SUN Peng-hao, LAN Ju-long, HU Yu-xiang, DUAN Tong

(National Digital Switching System Engineering & Technology Research Center, Zhengzhou, Henan 450002, China)

**Abstract:** With the development of Internet technology, novel types of protocol emerges one by one, which requires basic functional support from dataplane equipment. By now, SDN (Software Defined Networking) still has difficulty achieving both programmable protocol parsing and high forwarding performance. This paper proposes a Configurable dataplane based on LInkage between Parsing and Executing (CLIFE). By installing a user-configurable module in parser, CLIFE can achieve real-time update of multi-way tree in protocol parsing, thus making parsing protocol a user-configurable process. Besides, with the linkage between parser and action executor, the redundancy of overall architecture is reduced, as a result of which action delay and resource utilization are improved. Based on NetFPGA-10G board, the prototype of this architecture is implemented, and compared to other schemes such as EPC and POF, the advantage of resource utilization and forwarding rate is proved.

**Key words:** network dataplane; flexible parser; action processor; programmable

## 1 引言

当前互联网技术发展迅速, 新的体系结构和网络协议不断涌现, 传统的网络体系已经难以适应新的功能要求. 其中 XIA (eXpressive Internet Architecture)<sup>[1]</sup>, VxLAN<sup>[2]</sup>, FabricPath<sup>[3]</sup> 等新设计的出现使网络中的数据包包格式不断更新, 网络转发设备面临在新型数据格式的处理能力上的挑战. 如果网络设备在数据报文的解析、查表、动作等的处理环节上能够支持用户灵活自定义, 则不断更新的网络协议给硬件测试平台和实际应用设备部署带来的压力将会大大降低.

网络设备的主要功能是数据包的识别、处理和转发, 在新型网络结构迅猛发展的背景下, 其可扩展性、可编程性和转发性能是网络设备发展的重要方向. 目前, 软件定义网络 (Software Defined Networking, SDN)<sup>[4]</sup> 实现了网络控制和转发功能的分离, 从而实现了控制逻辑的用户定制. 然而在数据平面, 以 OpenFlow<sup>[5]</sup> 为代表的实现方案仅能处理现有的 MPLS、IP 等数据包, 对于新的网络协议支持并不灵活, 且动作执行器执行时延偏高. 为实现数据包解析的灵活性, CAFE<sup>[6]</sup>、Switch-Blade<sup>[7]</sup> 等提出了支持数据包头部任意比特域组合的结构, 初步实现了用户自定义的匹配域提取方式. 刘中金

等人提出了弹性协议可定制的网络数据平面结构及其映射算法,简称 EPC (Elastic Protocol Customizable)<sup>[8]</sup>,通过在解析模块中利用 RAM 实现对匹配域偏移量的缓存,从而利用偏移量将所需的匹配域信息提取,实现匹配域的用户定义,但是其匹配域难以满足多匹配域的要求,且无动作执行器的优化.为实现数据包的动作执行的灵活性,华为借助 SDN 网络与 PC 机结构的比拟,提出了 POF<sup>[9]</sup>模型.通过定义一套通用的动作集合,实现了协议无关的动作执行器,但是执行一个动作动辄需要几十个时钟,延迟过高. PLUG<sup>[10]</sup>, LabelCast<sup>[11]</sup>分别从查表匹配的硬件架构和流表的抽象方面提出了新的设计方案,但都没有实用的硬件实现.

基于以上分析,本文提出了具有高转发性能且协议无关的解析与执行联合结构,并且提出了相应的协议可定制的映射算法,主要具有以下两方面的贡献:(1)实现了协议识别和匹配域提取的自由定制.(2)实现了高转发性能和协议无关的低延时动作执行器.本文通过在 NetFPGA-10G<sup>[12]</sup>平台上完成对所提实现方案的仿真和验证,结果显示与已有方案相比,本方案在保证低处理时延的条件下具有更高的资源利用率和较高的转发速率.

## 2 设计目标

本文的设计目标是设计一种支持协议定制的高速数据平面处理结构及映射算法,主要有如下三个子目标:(1)实现协议解析的灵活定制性.(2)支持用户自定义的执行动作.(3)实现高转发性能和较低的硬件资源占用.

(1)协议解析的灵活定制性.解析的过程包括提取包头的类型域和匹配域.传统的协议处理方式中,各层协议的类型和组合方式都是固定的,如果协议格式发生变化或者有新的协议添加,则传统的解析模式无法进行解析.

(2)支持用户自定义的执行动作.目前的数据平面网络设备中,动作的执行主要可以分为三类,分别是添加、删除和修改字段.这样,对于一个执行动作的定义,

就可以使用偏移(offset)、类型(type)、长度(length)和内容(content)四个字段来描述,以四元组  $\langle \text{offset}, \text{type}, \text{length}, \text{content} \rangle$  表示.其中,偏移指动作执行位置在整个数据包中的偏移量,类型描述动作执行方式是添加、删除还是修改,长度是指以比特为单位的动作内容的长度,而内容是按比特给出实际的动作内容.通过使用  $\langle \text{offset}, \text{type}, \text{length}, \text{content} \rangle$  的四元组集合来描述一个执行动作,实现了对动作执行的高度抽象,达到了动作执行协议无关的目标.

(3)高转发性能和较低的硬件资源占用.以文献[13]的实现方案为代表的许多数据平面架构,其解析器已经完成了对数据包包头类型域和匹配域的提取工作,在执行动作时,动作执行器采用的逻辑仍然是按照协议树从根节点到子节点进行搜寻,在每一级节点进行逐次对比.这种动作执行方式造成了数据格式解析的重复,既增加了逻辑布线资源的占用,又增加了数据包处理时延.本文提出了解析器与动作执行器联合的结构,使动作执行模块可以直接从解析器中获得数据包头的字段信息,从而在数据包动作执行环节直接定位到执行位置,进行准确的动作执行.

## 3 CLIPE 的整体结构

基于以上的目标,本文提出了如图1所示的 CLIPE 硬件逻辑结构,主要有解析器、匹配器和动作执行器三部分逻辑模块.其中,解析器在解析数据包包头信息的同时,对树节点的节点信息进行特定的哈希运算,将节点所含包头偏移信息储存在 RAM 空间上,在流表匹配环节直接将匹配到的流表与 RAM 中的包头信息对比,提取动作执行的偏移信息,转送动作执行模块.动作执行模块得到  $\langle \text{offset}, \text{type}, \text{length}, \text{content} \rangle$  四元组的动作信息后,直接在数据包特定位置开始动作执行,进而将数据包输出.多级流表在执行过程中,只需将本结构的动作处理模块进行级联,由于单级动作执行模块的资源消耗量小、动作执行快,所以级联所带来的硬件资源压力大大减小.

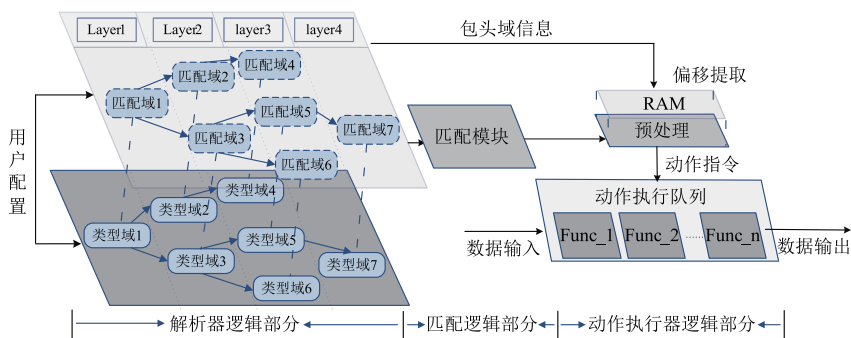


图1 CLIPE总体逻辑架构

### 3.1 解析器

在本方案中,解析器获取用户从上层给出类型域和匹配域信息,完成解析树的配置,从而实现数据包解析的可定制性.

用户配置信息通过用户总线将上层配置信息载入用户配置模块实现协议定制.其中,用户配置模块以 TCAM(Ternary Content Addressable Memory)和 RAM(Random Access Memory)为核心,用户自定义协议的类型域内容信息储存在 TCAM 中,字段长度信息储存在 RAM 中.首字段的偏移信息单独储存在一个寄存器(OFFSET)中,用作首段数据的解析起始地址.数据包进入解析器后,经过类型域提取模块的提取,将类型字段送往 TCAM 查表辨别类型后,根据不同内容查询不同 RAM 地址进而获得该协议的匹配域长度信息和下一字段起始偏移.从 RAM1 中读取的匹配信息送往匹配域提取模块,同时将相同的信息送往哈希处理模块.匹配域提取模块根据 RAM1 的偏移信息将数据包的匹配域提取,送往组合模块最终将所有解析得到的匹配域组成整体的包头域.哈希处理模块从用户配置模块的解析过程中得到每种协议匹配域的名称和偏移量,储存在 RAM2(即图 1 偏移提取环节中的 RAM)中,以备后续动作执行模块的提取使用.见图 2.

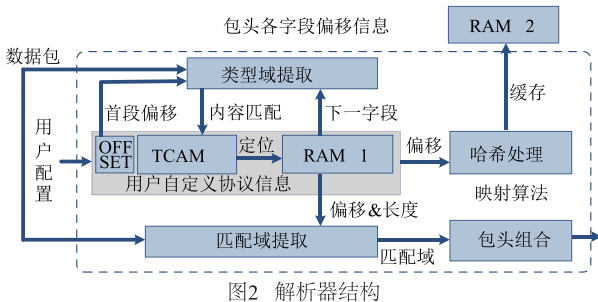


图2 解析器结构

### 3.2 动作执行器

本文提出的动作执行结构实现了协议无关性,并且借助与解析器的联合,大大减小动作执行模块的逻辑复杂度.动作执行器通过与解析器的联合,直接从 RAM 中提取包头偏移信息.数据包在匹配模块完成流表匹配和动作内容提取,结合解析器给出的包头偏移信息组合成动作指令四元组,以  $\langle \text{offset}, \text{type}, \text{length}, \text{content} \rangle$  的数据格式传输到动作执行器.动作执行器在四元组拆分模块分离  $\langle \text{offset}, \text{type}, \text{length}, \text{content} \rangle$  四元组中各字段,完成相应功能子模块的指令分配.其中,偏移计算模块的核心功能是计数和比较,通过以字节为单位对数据包进行计数,与 offset 进行对比以快速定位动作执行位置.type, length, content 信息缓存在类型选择模块中,类型选择模块等待计数器给出动作执

行的信号后,根据 type 选择将 length 和 content 字段有选择性地加载到添加、删除或者修改字段模块中,进而在对应的动作执行类型模块中完成动作执行.动作执行器结构如图 3 所示.

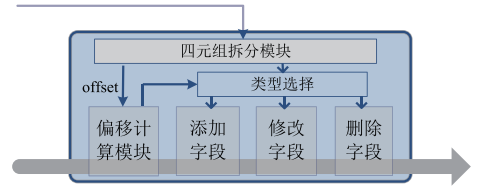


图3 动作执行器结构

## 4 解析映射算法

解析器的协议可定制性要求需要将用户的协议配置信息以一定的映射方式通过南向接口储存在转发设备中.其映射过程主要是将用户对类型树和匹配树的定义信息以一定的格式储存在 TCAM 和 RAM 中.图 1 的结构图中给出了四层协议树的逻辑示意(Layer1 ~ Layer4).相关算法的变量定义如表 1 所示.

表 1 算法主要符号及含义

符号	含义
$T_m$	匹配树
$T_t$	类型树
$f$	匹配域
$D(i)$	节点 $i$ 的子节点
$M$	匹配域集合
$\text{node}(i)$	匹配树节点
$W_D$	数据传输位宽
$\text{num}(M)$	集合 $M$ 中有效元素个数
$\text{layer}(i)$	节点 $i$ 所在的网络协议层
$l_t$	类型域的位宽

**定义 1** 如果匹配域  $f_m, f_{m+1}, \dots, f_n$  是可聚合(总宽度小于  $W_D$ )的相邻域,则称存在  $C(f_m \text{ to } f_n)$ ,相应的聚合后集合记做  $C$ .

**定义 2** 为了准确快速根据协议编号定位协议位置,使用基于生成多项式取余运算的方式,直接采用各比特并行逻辑运算的方式来实现数学取余运算的结果,计算过程函数记做 Quick\_hash().

算法 1 给出了解析树的映射算法.算法的逻辑过程按照从树的根节点向子节点进行的顺序逐层执行.算法分别对实际硬件中的数据拆分和聚合两种情况进行了优化.其中,考虑到实际硬件中传输位宽固定,类型字段和匹配字段有可能会被拆分为两个分字段,分别作为不同的偏移量储存;同时,考虑到相对于储存位宽,树的节点偏移量数据宽度可能较小,则在算法中通

过聚合的方式将不同的匹配域存放到相同 RAM 地址以节省 RAM 资源。

#### 算法 1 解析树映射算法

```

1.  push  $f[0]$  to  $M$ ;    //初始化匹配域集合
2.  counter = 0;
3.  while ( num( $M$ ) ! =  $N$ ) do //逐层映射树节点
4.      for all  $f(i) \in M$ ;
5.          for  $j$ : = counter to  $N$ 
6.              if node( $j$ ) = Child ( node( $i$ )) && node( $j$ )  $\in T_i$ 
7.                  push  $f(j)$  to  $M$ ;
8.                  temp =  $j$ ;
9.          counter = temp;
10. while (  $i$  ! =  $N$ ) do //分配物理地址
11.     if  $C(f(i)$  to  $f(n)$ )
12.         push  $f(i)$  to  $f(n)$  to  $C$ ;
13.         assign ram_addr to address of  $f'(i)$ ;
14.         ram_addr = ram_addr + 1;
15.     else
16.         assign ram_addr to address of  $f(i)$ ;
17.         ram_addr = ram_addr + 1;
18.      $i = n$ ;
19. for  $i$ : = 1 to  $N$ 
20.     if node( $i$ ) ! = NULL
21.         assign node( $i$ ) to TCAM;

```

算法 2 给出了解析器将匹配域偏移量解析完毕后映射到联合动作执行器 RAM 中的算法. 本算法主要以哈希为主, 将特定协议特定字段按照层级-协议标识-偏移量三个内容映射到特定的 RAM 地址.

#### 算法 2 动作偏移提取

```

1.  for  $i = 1$  to  $N$ 
2.      if node( $i$ )  $\in T_m$ 
3.          address( $i$ ) = hash ( node_id( $i$ ));
4.          assign address( $i$ ) to node( $i$ ) in RAM;
5.  Function hash (node_id)
6.      switch layer (node_id) //借助网络协议层解析缓存
7.          1: address[4:3] = 0;
8.          2: address[4:3] = 1;
9.          3: address[4:3] = 2;
10.         4: address[4:3] = 3;
11.         address[2:0] = Quick_hash ( address[15:0] );
12.         return address;

```

## 5 性能分析

本文以 NetFPGA-10G 为实验平台开发出了 CLIPE 的原型, 实现了解析与动作执行联动的整体数据包处理流程. 其中, NetFPGA-10G 板卡核心芯片为 Xilinx Vir-

tex5 XCV5TX240T-2. 结合理论分析, 本文在此给出 CLIPE 的性能评估.

### 5.1 算法储存开销

在本方案实现的解析器和动作执行器中, 储存开销主要有储存类型域偏移量的 RAM、储存类型域内容字段的 TCAM、储存匹配域偏移量的 RAM 和储存动作模块包头偏移量的 RAM 四部分. 这四部分的储存开销总量分析如下:

(1) 在 RAM 中, 储存一个偏移量需要记录其起始位置和结束位置, 即一组 < 起始偏移, 结束偏移 > 偏移量对. 假设待解析的数据包头总长度为  $L$ , 硬件中设置的数据传输位宽为  $W$ , 则解析一个数据包最多需  $\lceil L/W \rceil$  次传输, 记录  $W$  位数据需要  $\log_2(W)$  位地址. 因此储存偏移量对的 RAM 开销最多为  $2 \times \log_2(W) \times \lceil L/W \rceil$ .

(2) 对于匹配域偏移量, 每个匹配域对应一个匹配树节点. 经过合并后, 算法中的树节点数量为  $N(T_m)$ , 则储存所有匹配域偏移量对所需 RAM 空间为  $2 \times \log_2(W) \times N(T_m)$ .

(3) 类型域的匹配需要 TCAM 储存, 设类型域解析树的节点总个数  $N(T)$ , TCAM 位宽为  $B$ , 则 TCAM 储存所需开销为  $N(T) \times B$ .

(4) 动作模块储存的动作偏移量与解析的协议数量相关. 其中, 储存单个包头偏移量只需记录起始位置, 因此为  $\log_2(W)$ . 匹配树中最终提取的匹配域数量最多为  $\text{num}(M)$ , 因此总的储存开销为  $\log_2 \times \text{num}(M)$ . 综上所述, 解析器和动作执行器所需的总 RAM 储存空间上限为  $2 \times \log_2(W) \times \lceil L/W \rceil + 2 \times \log_2(W) \times N(T_m) + \log_2 \times \text{num}(M)$ , 所需 TCAM 空间为  $N(T) \times B$ .

本算法与 EPC 所使用的解析结构算法进行了储存开销对比. 以目前数据平面比较有代表性的 7 种协议为例, 算法比较了 802.3、MPLS、802.1Q、IPv4、IPv6、TCP 和 UDP 的解析存储. 这些协议对应总位宽 78Byte, 其中, 类型域共占用 5Byte, 解析树节点数量为 41. 在数据总线宽度为 64bit 的情况下, 二者储存空间如表 2 所示:

表 2 储存空间对比

实现方案	压缩前/后 RAM 需求(Byte)	压缩前/后 TCAM 需求(Byte)
EPC	7.596/6.228	205/115
CLIPE	6.872/5.726	102/102

以上结果表明, 采用本方案后 RAM 需求减少了 17%, 而 TCAM 需求不变. 本方案的整体 RAM 需求低于 EPC, 为 EPC 的 92%, 而 TCAM 资源需求是 EPC 的 88%.

### 5.2 硬件资源开销

(1) 包头解析器: 通过对包头解析器进行综合和布

局布线,得到本方案 CLIPE 与 EPC 在包头解析器上的资源利用对比,结果如表 3 所示:

表 3 硬件开销对比

方案	slice	BRAM
EPC	4013	29
CLIPE	3572	22

由表 3 可知,本方案的解析器共占用了实验平台 slice 资源的 9.5%,是 EPC 使用量的 89%;BRAM 资源占用比 EPC 节约了 24%.

(2)动作执行器:由于动作执行器优化设计,使硬件资源开销明显减小.在多级流表实现中,单个动作执行器的资源优势经过联后就尤为突出.图 4 和图 5 分别给出了 slice 和 BRAM 的利用与数据位宽和流表级数之间的关系.

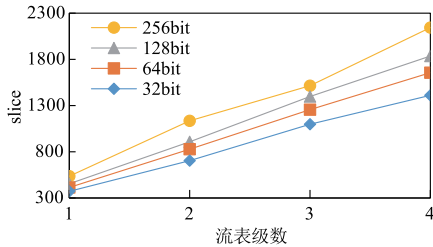


图4 slice资源使用

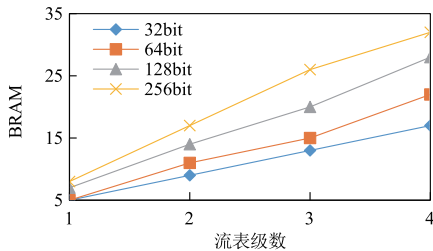


图5 BRAM资源使用

由此可以看出,本方案的动作执行模块使用的 slice 和 BRAM 数量随着流表级数增加基本呈线性增长. slice 和 BRAM 的使用量还受数据位宽影响,数据位宽越大, slice 消耗量越多.由此判断,实现相同功能的情况下,数据位宽越小,动作执行器的资源消耗量越小.

而整体方案的数据位宽选择中,位宽过小会造成解析模块的类型域和匹配域过于分散,增加了解析树节点的压缩难度,提高了储存偏移的 RAM 使用量.表 4 给出了 5.1 实现背景下解析器中用于储存类型域和匹配域偏移量的 RAM 资源使用量与数据位宽的关系.由表可以看出,数据位宽从 64bit 切换为 32bit 时, RAM 使用量增加比较明显.

由此可知,在数据位宽的设置上,要综合考虑整体结构的性能,兼顾解析器和动作执行器的需求.本实验中,最终方案选择 64bit 数据位宽比较合适.

表 4 RAM 使用与数据位宽

数据位宽	RAM 使用量 (Byte)
32	7.232
64	5.726
128	5.322
256	5.134

### 5.3 转发性能

转发速率和转发时延是数据转发的重要性能指标.本文使用 ISE13.7 开发环境对解析与执行联动结构进行了线路综合和布局布线测试,在数据位宽设置为 1024bit 时,系统能达到的最高布线时钟为 187.8 MHz,表明理论上本方案最多能实现 192.3 Gbps 的转发速率.而数据处理延时方面,解析与执行联动结构带来动作执行器的处理简化,从而显著减小了动作执行器的处理时延,使本方案中一个动作执行只需要两个时钟周期.由于 EPC 论文中着重论述了其解析器结构在 NetFPGA-10G 平台的实现,而动作执行器没有做出优化处理,因此对比内容中 EPC 的动作执行时延数据做缺省处理.而对于参考文献[13]中的 openflow 实现方案,其动作执行器完成一个动作执行时延为 7~9 个时钟周期.以常见的修改 IP 首部的 TTL 字段的执行动作为例,现将不同方案性能对比如下:

由图 6 可以看出,本方案在节约了硬件资源消耗的情况下,转发速率接近 EPC,而高于参考文献[9]中提到的 POF 实现方案的转发速率.同时,在动作执行时延上,CLIPE 执行延迟为 2 个时钟,要明显低于 POF 的 31 个时钟.

综上所述,CLIPE 在实现了协议可编程的基础上,兼顾了动作执行的高效性,保证了转发率和处理时延.

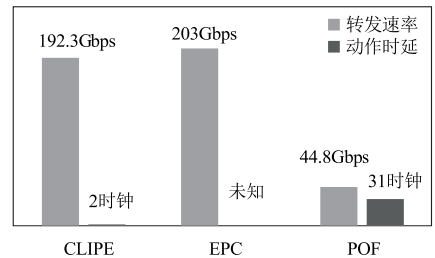


图6 转发性能对比

## 6 结束语

基于目前网络设备的更新难以满足不断出现的新型网络功能的测试和部署的问题,本文提出了支持数据包解析可编程和处理动作自定义的 CLIPE 结构,并且在 NetFPGA-10G 平台上完成了功能原型的开发.本方案实现的原型机与其他可编程数据平面方案相比,在硬件资源利用率、吞吐率和转发延迟方面体现出明

显优势. 综上所述, 本方案对于可编程数据平面的架构设计发展具有重要的意义.

### 参考文献

- [1] Aziziyeh A, Easty T, Khayat Z, et al. XIA: An architecture for an evolvable and trustworthy internet [J]. Proceedings of Acm Workshop on Hot Topics in Networks, 2011, 63 (3): 879 – 884.
- [2] RFC 7348, Virtual eXtensible Local Area Network (VX-LAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks [s].
- [3] Cisco Fabricpath [EB/OL]. <http://www.cisco.com/c/en/us/solutions/data-center-virtualization/fabricpath/index.html>. 2012-06-20.
- [4] McKeown N. Keynote talk: software defined networking [A]. Proceedings of the IEEE INFOCOM [C]. Rio de Janeiro, Brazil: IEEE, 2009. 1 – 11.
- [5] McKeown N, Anderson T, Balakrishnan H, et al. OpenFlow: enabling innovation in campus networks [J]. Acm Sigcomm Computer Communication Review, 2010, 38 (2): 69 – 74.
- [6] Lu G, Shi Y, Guo C, et al. CAFE: a configurable packet forwarding engine for data center networks [A]. Proceedings of the 2nd ACM SIGCOMM Workshop on PRESTO [C]. Barcelona, Spain: ACM, 2009. 25 – 30.
- [7] Anwer M B, Motiwala M, Tariq M B, et al. SwitchBlade: A platform for rapid deployment of network protocols on programmable hardware [J]. Acm Sigcomm Computer Communication Review, 2010, 40(4): 183 – 194.
- [8] 刘中金, 李勇, 苏厉, 金德鹏. 弹性协议可定制的网络数据平面结构及其映射算法 [J]. 电子与信息学报, 2014, 36(7): 1713 – 1719.  
Liu Z J, Li Y, Su L, et al. Design on the elastic protocol customizable data plane and its mapping algorithm [J]. Journal of Electronics & Information Technology, 2014, 36 (7): 1713 – 1719. (in Chinese)
- [9] Song H. Protocol-oblivious forwarding: unleash the power of SDN through a future-proof forwarding plane [A]. Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking ACM [C]. HongKong, China: 2013. 127 – 132.
- [10] L D Carli, Yi Pan, Kumar, et al. PLUG: flexible lookup modules for rapid deployment of new protocols in high speed routers [A]. Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication [C]. Barcelona, Spain: ACM, 2009. 207 – 218.
- [11] 吕高峰, 孙志刚, 李韬. LabelCast: 一种普适的 SDN 转发平面抽象 [J]. 计算机学报, 2012, 35(10): 2037 – 2047.  
Lü gaofeng, et al. LabelCast: A general abstraction for the forwarding plane of SDN [J]. Chinese Journal of Computers, 2012, 35(10): 2037 – 2047. (in Chinese)
- [12] NetFPGA-10G Project [EB/OL]. <https://github.com/NetFPGA/NetFPGA-public/wiki>, 2014.
- [13] Naous J, Erickson D, Covington G A, et al. Implementing an OpenFlow switch on the NetFPGA platform [A]. ACM/IEEE Symposium on Architecture for NETWORKING and Communications Systems [C]. San Jose, California, USA: 2008. 1 – 9.

### 作者简介



**孙鹏浩** 男, 1992 年生于山东即墨. 现为国家数字交换系统工程技术研究中心硕士研究生. 主要研究方向为可编程网络、数字交换技术.

E-mail: sphshine@126.com



**兰巨龙** 男, 1962 年生于河北张北. 现为国家数字交换系统工程技术研究中心总工程师、教授、博士生导师. 主要研究方向为新一代信息网络关键理论与技术.

E-mail: ndsclj@163.com