一种支持网络功能演进的可重构数据平面

段 通,兰巨龙,胡宇翔,刘释然
(国家数字交换系统工程技术研究中心,河南郑州 450002)

摘 要: 传统网络体系在安全、服务质量保证、流量调度等方面难以适应互联网的发展,而新型网络体系如SDN(Software Defined Networking)、NDN(Named Data Networking)等,由于路由器、交换机等传统网络设备电路固化,限制了新型网络功能的试验和部署.针对此问题,本文设计了支持网络功能演进的可重构数据平面(Reconfigurable Dataplane for network Function Evolution,RDFE),通过插入用户配置单元的方式对数据包解析、匹配和处理过程进行编程,从而支持用户自定义的功能部署;其次,针对RDFE提出基于树型结构的解析映射和匹配映射算法,将用户定制功能映射到硬件结构中;最后,基于NetFPGA-10G板卡完成了RDFE的原型实现,与现有的Kangroo、EPC(Elastic Protocol Customizable)、LabelCast等方案相比具有更高的转发速率和更低的资源利用率.

关键词: 网络数据平面;功能定制;可编程;可重构;NetFPGA
中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2016)07-1721-07
电子学报URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2016.07.029

A Reconfigurable Dataplane Enabling Network Function Evolution

DUAN Tong, LAN Ju-long, HU Yu-xiang, LIU Shi-ran
(*National Digital Switching System Engeineering & Technology Research Center, Zhengzhou, Henan* 450002, *China*)

Abstract: The traditional network system is difficult to adapt to the development of the Internet in some aspects such as security, QoS (Quality of Service) guarantee, and traffic scheduling. But emerging network architectures like SDN (Software Defined Networking) and NDN (Named Data Networking) are limited in the experiment and deployment of new functions based on the inflexible circuits of traditional network devices like routers and switches. This paper is devoted to dealing with this problem. Firstly, we design a reconfigurable network dataplane to enable network function evolution. The dataplane uses configure units to program the parser, match tables and action processor, which enables the deployment of customized functions. Secondly, we propose tree-based parser-mapping and match-mapping algorithms to map user-defined network functions to hardware. Finally, we implement a NetFPGA10G-based prototype of RDFE which achieves higher forwarding rate and lower resource utilization compared with Kangroo, EPC (Elastic Protocol Customizable) and LabelCast programmble dataplanes.

Key words: network dataplane; function customization; programmable; reconfigurable; NetFPGA

1 引言

随着互联网的迅猛发展,传统的网络体系难以适应用户不断增长的需求,如XIA(eXpressive Internet Architecture)[1]、NDN(Named Data Networking)[2]、Nebula[3]、VxLAN[4]等新的体系随之涌现.这些体系定义了新的网络功能,对网络设备在解析、匹配、动作等方面的处理能力有了新的需求.然而受到网络设备厂商和协议开发环境的限制,新型功能设备的开发和部署面临诸多困难.如果网络设备能够在解析、匹配、动作等方面支持用户定制,那么将大大降低新型网络功能的试验和部署难度,从而为网络的创新和演进提供一个更加开放的平台.

为支持网络功能的快速创新和部署,网络设备首先要能够支持灵活的数据包解析,以便提取出功能所需的匹配域.为此,CAFE[5]和SwitchBlade[6]在包头解析

收稿日期:2014-12-26;修回日期:2015-04-07;责任编辑:覃怀银
基金项目:国家重点基础研究发展规划课题(973计划)项目(No. 2012CB315901, No. 2013CB329104);国家自然科学基金(No. 61372121);国家高技术研究发展计划(863计划)课题(No. 2013AA013505)

模块中设计了任意比特抽取器,其目的就是支持数据包头部任意比特域的自由组合,从而支持用户自定义的匹配域提取方式. 为实现灵活高性能的包头解析模块,Huber 等人[7]提出了 Kangaroo 结构,利用可编程的协议树可同时解析多种数据包,达到 40Gbps 的包头解析能力. 刘中金等人[8]提出弹性协议可定制的数据包查找结构 EPC(Elastic Protocol Customizable)及映射算法,其核心思想是在数据包解析模块中加入存储匹配域偏移量信息的存储单元,并利用偏移量信息将所需的匹配域提取出来. EPC 仅能支持 4 个匹配域的提取,不能满足多匹配域的处理需求.

以上研究工作仅针对包头解析部分进行研究,而事实上网络功能的实现还需要设备具备包头解析后的后续处理能力(如匹配、查找、操作等). 软件定义网络(Software-Defined Networking,SDN[9])通过将网络设备的控制逻辑转移到控制器内,从而支持用户功能定制. 但是在数据平面,其网络设备仅能处理现有的 MPLS 和 TCP/IP 数据包,对于新型网络体系的数据包处理并不支持. PLUG[10]为实现匹配查找资源对不同协议的适配,提出协议自适应的硬件架构及编译器,它将硬件架构分为处理能力相同的 Tile,利用 Tile 之间的灵活组合来实现对不同匹配查找需求的适配,其结构复杂度较高,且需要全新的编译系统与之匹配,且目前尚未实现. Pat 等人[11]设计了可编程的多级流表架构,各级流表包含匹配、查找、动作等资源,且流表之间可动态组合. 其架构偏理想化,需要巨大的存储资源和处理时延,因此并未得到业界的广泛响应. LabelCast[12]针对现有 SDN 的不足,通过定义 Label 表和 Cast 表对除 SDN 之外的新型网络体系的转发行为进行抽象和普适. 其数据平面在服务器中实现,转发性能较低.

基于以上分析,本文提出支持功能演进的可重构数据平面结构以及用户功能定制的映射算法,其主要有两方面的贡献:(1)实现了任意协议的识别和匹配域提取;(2)实现了处理能力和处理开销对任意功能的适配. 其次,基于 NetFPGA-10G[13]平台完成了所提结构的原型实现,结果显示与已有方案相比 RDFE 具有更高的转发速率和更低的资源利用率.

2 设计目标

本文的目标是设计一种新型的网络数据平面,以支持网络功能的演进. 对此,本文将其分为三个子目标:(1)支持用户定制的协议解析;(2)实现灵活可编程的数据包处理;(3)内部资源可动态组合.

(1)用户可定制的协议解析. 支持用户定制的协议解析是实现新功能的关键. 考虑网络数据包包头内包含类型域和匹配域,其中类型域用于标识数据包的协议类型,用 T 表示;匹配域用于匹配查找并对数据包进行相应的处理,用 F 表示. 多数据包的协议解析过程可用多叉树表示,例如图 1 左侧图所示,每个类型域或匹配域都是一个树节点. 两个数据包的第一层协议包格式相同,均为$\{F_1,T_1\}$;第二层协议包格式不同,通过 T_1 的值来识别,对应 2 棵以 T_1 为根节点的子树. 左边子树代表数据包的第二层协议对应匹配域 F_2 和 F_3;右边子树代表数据包的第二层协议对应匹配域 F_{II} 和类型域 T_{II},类型域 T_{II} 的值指示第三层协议对应匹配域 F_{III}. 如果能够利用解析树进行协议相应匹配域的精确提取,即可实现对任意协议数据包处理的支持.

(2)灵活可编程的数据包处理. 数据包处理过程主要包括匹配、查找、动作三个方面,其中匹配查找是实现灵活可编程数据包处理的关键. 对此,可用匹配树表示

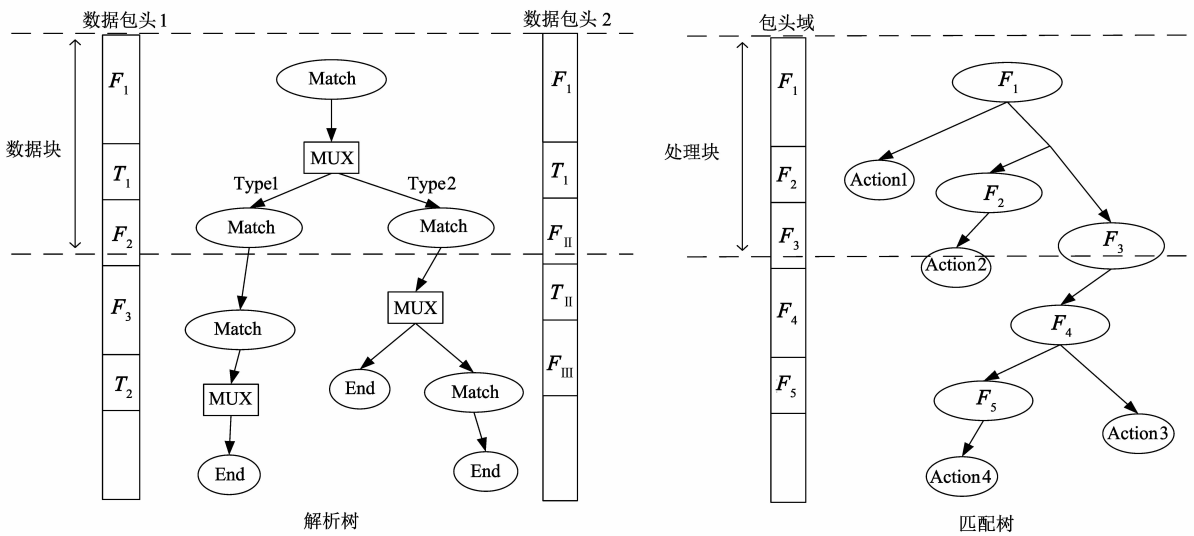


图1 解析树和匹配树示例

数据包的处理过程,每一个匹配域都是一个树节点,每一种功能对应一个匹配树中的子树,树的匹配域节点表示该功能所需要的匹配域,树的叶子节点则指向相应的操作类型. 例如图 1 右侧图所示的 4 种功能,其中功能 1 对应左边子树,其操作域为匹配域 F_1,动作类型为 action1;功能 4 的操作域则由 $F_1/F_3/F_4/F_5$ 这四种匹配域组成,动作类型为 action4. 如果能够将匹配树映射到数据包处理单元,即可实现灵活可编程的数据包处理.

(3)内部资源可动态组合. 任何功能的数据包处理过程都可抽象成“匹配 + 查找 + 动作”的过程,但不同功能所定义的匹配域数量和长度以及动作类型都不相同. 在不额外占用资源的情况下,如果能利用匹配查找资源的组合来实现对不同功能所需资源的适配以达到可重构的效果,那么将大大减少额外的处理开销.

3　RDFE 整体架构

基于以上目标,本文提出如图 2 左图所示的支持网络功能演进的可重构数据平面(Reconfigurable Dataplane for network Function Evolution,RDFE),它主要由包头解析器(parser)和元处理单元(cell)组成. 其中,包头解析器用于识别数据包的协议类型,同时根据数据包的协议类型得到相应所需的匹配域并将其组合成包头域,向后级元处理单元输出. 元处理单元是最基本的数据包处理单元,用来实现“匹配 + 查找 + 动作”的操作. 元处理单元之间通过元数据(metadata)进行信息传递,实现元处理单元之间的组合.

通过对包头解析器的配置和元处理单元的组合可达到内部逻辑重构的目的,从而实现用户功能定制. 以图 2 右侧两个子图为例,假设一个元处理单元能够处理 24bit 的匹配域,则对于 MPLS 功能,其核心 LSR 所需的匹配域长度为 20bit,动作是简单的数据转发,利用一个元处理单元即可实现,见左图;而右图则是针对私有网络访问企业网的一个用例,先利用 NAT 功能将数据包的私有 IP 转化为公用 IP,再用 ACL 功能实现访问接入控制,该处理过程需要使用四个元处理单元的组合实现:前用两个元处理单元做 IP 地址的匹配及修改,实现 NAT 功能;后两个元处理单元做 IP 地址和 TCP 端口号的匹配及数据包转发/丢弃操作,实现 ACL 功能.

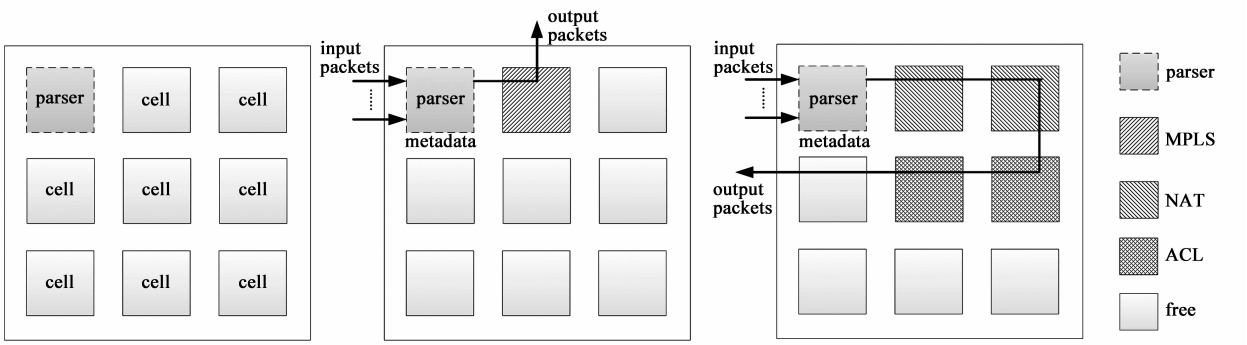


图2　整体架构

3.1　包头解析器

在以上结构中,包头解析器是实现对新型数据包格式支持的关键模块. 它根据用户的配置识别数据包的类型域,同时根据类型域提取相应匹配域并将其组合得到包头域(header)向后级元处理单元输出.

包头解析器结构如图 3 所示,它包含 4 个部分:类型域提取模块、匹配查找模块、匹配域提取模块和匹配域组合模块. 其中类型域提取模块用于识别数据包头并提取类型域. 数据包通过总线传输,每次传输数据总线位宽大小的数据块. 首先,类型域提取模块将状态设置为第一个类型域所在的数据块编号;当数据块到达,将数据块编号与当前状态对比,若相等则根据从 RAM1 中读取的偏移量将数据块中的类型域提取出来,并将类型域和当前状态一起送往匹配查找模块. 当接收到从匹配查找模块输出的下一状态,类型域提取模块将当前状态更新至下一状态. 匹配查找模块包含一个 TCAM 单元和一个 RAM2 存储单元. 其中 TCAM 中存放状态信息和用户定制的类型域信息,RAM2 中存放类型域所对应的匹配域的偏移量信息. 匹配查找模块利用 TCAM 匹配类型域和状态,根据匹配结果在 RAM2 中读取得到下一状态和对应匹配域的偏移量,并分别向类型域提取模块和匹配域提取模块输出. 匹配域提取模

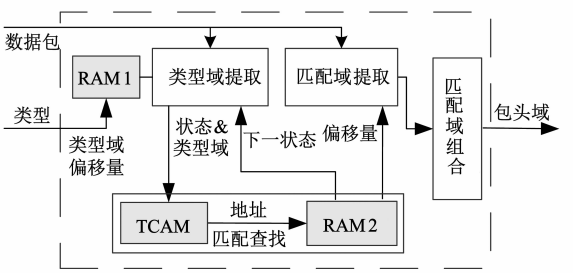
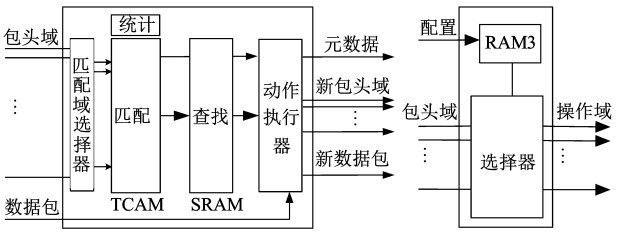


图3　包头解析器整体结构

块根据匹配域的偏移量将所需匹配域提取出来. 最后,匹配域组合模块将提取得到的匹配域组合成包头域并送往后级元处理单元处理.

3.2 元处理单元

元处理单元是最基本的数据包处理单元,它可抽象为“匹配 + 查找 + 动作”的处理过程,如图 4 所示. 元处理单元由匹配域选择器、流表匹配单元、动作处理器组成.



(*a*) 整体结构 (*b*) 匹配域选择器

图4 元处理单元结构

其中匹配域选择器将包头域中的用户关心的匹配域提取出来组成操作域,如图 4(*b*)所示. 当包头域到达,选择器会从 RAM 读取用户配置的匹配域选择信息,并根据这些匹配域选择信息将包头域中的所需字段提取出来组成操作域. 流表匹配单元使用“TCAM + SRAM”实现“匹配 + 查找”,其中 TCAM 存放用户下发的处理域,SRAM 存放动作字段. 动作执行器接收到动作字段后根据动作字段进行数据包的处理以及元数据的修改. 当所需处理域宽度超过一个元处理单元匹配能力时,可通过两个元处理单元相连,使同一个匹配域在两个元处理单元中组合.

4 功能定制映射算法

用户在功能定制时不会直接对硬件进行编程,而是通过映射机制将用户定制的数据包协议格式及所需处理过程到硬件存储结构中去. 这一过程涉及到两个操作:解析树的映射和匹配树的映射. 本文定义如下变量:

表 1 主要的符号定义及具体含义

符号	含义
W_D	数据块宽度
W_T	元处理单元处理域的宽度
T_P	解析树
T_M	匹配树
$R(T)$	T 的所有子树集合$\{T_1,T_2,\cdots T_N\}$
$F(T)$	T 中所有匹配域节点
$M(T)$	T 中所有类型域节点
root(T)	T 的根节点
$C(v)$	节点 v 的子节点

定义 1 如果匹配域 f 存在于连续两个数据块内,则称 f 可拆分,记为 Cut(f).

定义 2 如果匹配域 $f_1,f_2,\cdots,f_m$ 相邻且在同一个数据块内,则称 $f_1,f_2,\cdots,f_m$ 可合并,记为 Combine$(f_1,f_2,\cdots,f_m)$.

算法 1 给出了解析树的映射过程,映射过程考虑两方面的因素:(1)考虑数据总线位宽的限制,当数据总线位宽较短时,一个时钟周期内无法将整个数据包头取出,那么就需要将数据包头根据数据总线位宽进行分割,将被分割的匹配域进行拆分,生成两个偏移量对;(2)考虑匹配域宽度,如果将不同匹配域的偏移量映射到 RAM 中的不同行中,那么将会造成存储资源浪费,当匹配域宽度较小时,可以利用相邻匹配域节点的合并以减少存储开销.

算法 1 解析树映射算法

```
0. for k = 1:N T = T_k ∈ R(T_P) /* 分别映射子树 */
1.   Function(node * node_c)/* node_c 是当前节点 */
2.     if node_c ∈ F(T)
3.       while C(node_c) ∈ F(T)
4.         put node_c to F_c:{f_1,f_2,…f_n};
5.         node_c = C(node_c);
6.       end while
7.       if ∃f_m,Cut(f_m)
8.         cut f_m to f_m1,f_m2;
9.       else if ∃Combine(f_i to f_j)/* j≥i≥1 */
10.        combine f_i to f_j into f_i';
11.        assign f_i' offset to RAM2;
12.    end if
13.  end if
14.  if C(node_c) ∈ M(T)
15.    node_c = C(node_c);
16.    assign node_c offset to RAM1;
17.    assign node_c value to TCAM;
18.    if C(node_c)! = NULL
19.      Function(C(node_c));
20.  else return
21. end Function
22. end for
```

算法 2 给出了匹配树的映射过程. 映射过程考虑两方面的因素:一是考虑处理域掩码的设定,这是由于匹配树的不同子树所需处理域和最终的处理动作都不同,这就需要设定合适的掩码以支持多功能并存;二是考虑元处理单元中处理域宽度的限制,根据处理域宽度进行匹配树的映射及元处理单元之间的组合,以减少 TCAM 存储开销.

算法 2 匹配树映射算法

```
0. 匹配树共有 M 个不同的匹配域,记为{f_1,f_2,…f_M}
1. i = 1;k = 1;
```

2. while $i \leqslant M$　/* 分别映射匹配域至元处理单元 */
3.　if $\exists j, \text{Length}(f_i \text{ to } f_j) \leqslant W_T$
　　& $\text{Length}(f_i \text{ to } f_{j+1}) > W_T$
4.　　cut f_{j+1} to f_j' and f_{j+1}';
5.　　assign f_i to f_j' offset to $\text{Cell}_k\_\text{RAM3}$;
6.　　$k = k + 1; i = j + 1;$
7.　end if
10. end while
11. for $k = 1; N$　$T = T_k \in R(T_M)$ /* 分别映射子树 */
12.　node_c = root(T);
13.　while node_c ! = NULL
14.　if C(node_c) = = NULL
15.　　assign node_c value to SRAM the k^{th} row;
16.　else
17.　　assign node_c value/mask to TCAM the k^{th} row;
18.　end while
19. end for

5　性能分析

本文基于 NetFGPA-10G[13] 板卡完成了 RDFE 的原型实现,包括收发单元、处理单元以及配置单元,其中收发单元包含 4 个带宽为 10Gbps 的物理端口和 1 个虚拟端口,物理端口与外部网络相连,虚拟端口通过 DMA 与主机虚拟网卡相连;处理单元是 RDFE 的逻辑承载单元,包含包头解析器和 4 级元处理单元;配置单元接收上层用户的配置信息并将表项下发到处理单元中去.

本节对 RDFE 进行性能评估,并与现有可编程数据平面做对比评价.

5.1　存储开销分析

在用户定制的映射算法中,其存储空间主要分为四个部分:存储类型域偏移量的 RAM,存储类型域字段的 TCAM,存储匹配域偏移量的 RAM,存储匹配域字段的 TCAM. 令解析的数据包长度为 L,数据总线位宽为 W. (1)设协议种类为 K,则用于提取类型域的偏移量总长度最大为 $2\log2(W) \times \lceil L/W \rceil$. (2)每个偏移量对的位宽为 $2\log2(W)$,合并后解析树的节点个数为 $N(T_P)$,那么整体的偏移量 RAM 存储需求为 $2\log2(W) \times N(T_P)$. (3)类型域的总个数为 M,那么所需类型域存储量为 $\sum_{t=1}^{M} l_t$,其中 l_t 为第 t 个类型域长度. (4)合并后匹配树的节点个数为 $N(T_M)$,那么所需匹配域存储量为 $\sum_{k=1}^{N(T_M)} l_k$,其中 l_k 为第 k 个匹配域长度. 因此整体的 RAM 存储需求为 $2\log2(W) \times (N(T_P) + \lceil L/W \rceil)$,整体的 TCAM 存储需求为 $\sum_{t=1}^{M} l_t + \sum_{k=1}^{N(T_M)} l_k$.

本文与文献[8]中所提解析结构 EPC 及其算法做对比分析,假设需要设备提取 7 种协议包(802.3、MPLS、802.1Q、IPv4、IPv6、TCP 和 UDP)的所有匹配域,这些协议对应的总位宽为 78byte,类型域宽度为 5byte,解析树节点数目为 41. 表 2 对比了总线位宽为 64bit 时两种结构及对应算法所需的存储空间.

表 2　存储资源需求

系统	压缩前/后的 RAM 需求 byte	压缩前/后的 TCAM 需求 byte
EPC	7.596/6.228	205/115
RDFE	0.598/0.487	78/78

结果表明,通过解析树和匹配树的压缩,可以使整体 RAM 存储需求减少 18%,TCAM 存储需求不变. 且与 EPC 的存储需求相比,RDFE 的 RAM 存储需求要减少 12.7 倍,TCAM 存储需求减少 32.1%.

5.2　资源与性能分析

本节首先分析包头解析器的资源占用和转发性能;然后,分析元处理单元处理域宽度对资源利用率产生的影响,并选定 64bit 为元处理单元的处理域宽度;最后分析元处理单元级数对资源开销的影响.

(1)包头解析器性能分析

将数据总线位宽设为 1024bit,在不添加元处理单元的情况下对包头解析器进行布局布线,布局布线时钟为 178.6MHz,也即理论上最高可满足 182.8Gbps 的转发速率. 图 5 对比了 RDFE 的包头解析器与现有可编程包头解析器 Kangaroo[7] 和 EPC[8] 的性能.

从实验结果可以看出,相比于 Kangaroo,RDFE 是在 Slice 资源开销提高 24%、BRAM 资源降低 58% 的同时具有 4 倍的转发速率;相比于 EPC,虽然 RDFE 的转发速率要低 10%,但同时资源开销也降低了 24%.

(2)元处理单元处理域宽度对资源的影响

假设单个功能所需的匹配域数目和长度一定,那么当元处理单元的处理域宽度较小时,需要多个元处理单元组合实现,其资源开销较大;但由于表项粒度较小,因此表项利用率较高. 相反,当处理域宽度较大时,资源开销较小,但表项利用率较低. 考虑到单个网络功能所需的匹配域数目十分有限[8],为不失一般性,现从最常用的匹配域 PORT、VLAN ID、MAC/IP 源/目的地址、TCP 源/目的端口中随机选出 3 个匹配域作为处理

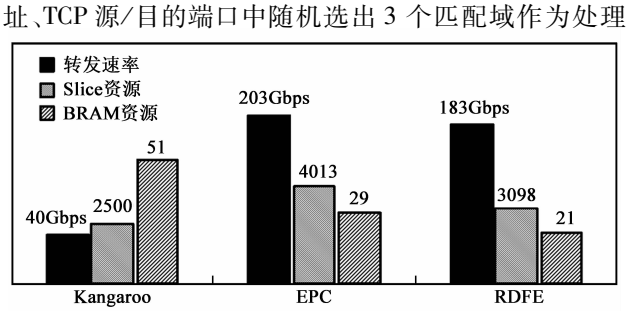


图5　包头解析器资源与性能对比

域,统计不同处理域长度下元处理单元的平均资源开销和平均表项利用率,见表 3.

表 3 不同处理域宽度对资源的影响

宽度/bit	Slice 数目	BRAM 数目	表项利用率
16	9388	60	95.43%
32	8232	48	87.65%
64	3647	25	83.35%
96	3539	23	73.75%
128	4073	26	68.97%

实验结果表明,处理域为 16bit 时,表项利用率最高,但其资源开销最大. 当处理域宽度在 64bit 时,与 32bit 相比其表项利用率仅低了 4.3%,但资源开销却减少了 56%;与 96bit 相比其资源开销仅高出 3%,表项利用率上高出了 10%. 事实上,处理域宽度的最优设置是资源开销和表项利用率的博弈,结果表明处理域宽度在 64bit 时是一个较好的折中.

(3)元处理单元级数对资源的影响

元处理单元的级数会影响整体资源开销,采用 64bit 作为每级元处理单元的处理域宽度,总线位宽为 64bit. 图 6 对比了在不同端口数下,元处理级数对资源开销的影响.

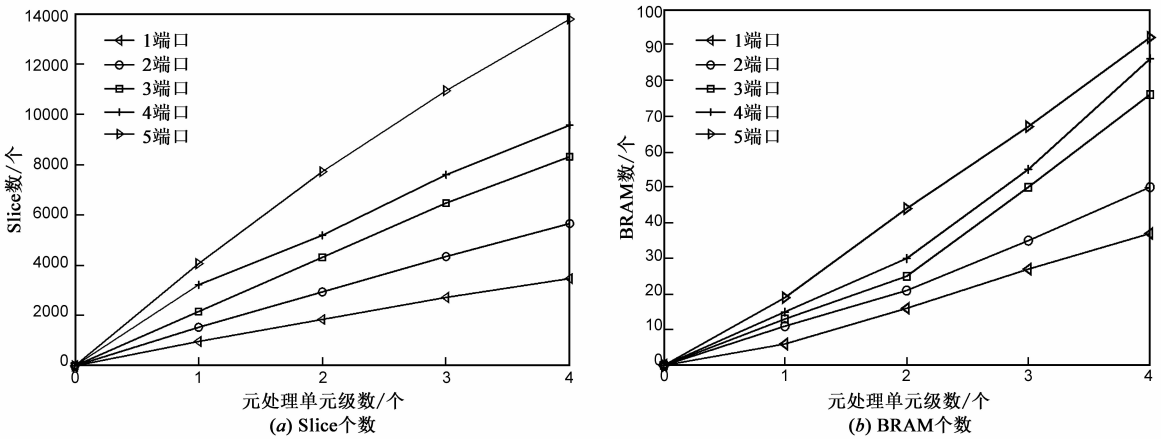


图6 元处理单元级数对资源开销的影响

结果表明,随着元处理单元级数的增加,Slice 资源和 BRAM 资源均接近线性增长. 当使用 5 个网络端口时,平均每个元处理单元需要 3439 个 Slice 和 23 个 BRAM,占总体片内资源的 9%.

5.3 整体转发性能

当进行整体实现时,考虑除 RDFE 的功能部分外还需实现收发单元和配置单元,因此为降低资源开销,数据总线位宽采用 64bit,每级元处理单元的处理域宽度定为 64bit. 当使用 1 级元处理单元(记作 RDFE1)时布局布线时钟为 172.6MHz,即最高可满足满足 54Gbps 的转发速率;当使用 4 级元处理单元(记作 RDFE4)时,布局布线时钟为 163.8MHz,也即最高可满足 51.3Gbps 的转发速率. 图 7 对比了 RDFE 与 LabelCast[12] 的整体转发速率.

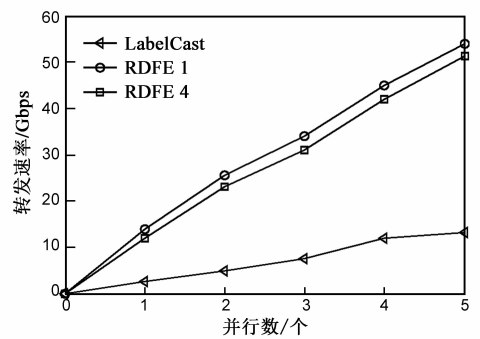


图7 整体转发性能对比

结果表明,RDFE 的转发速率随着并行数(LabelCast 是线程数)的上升接近线性增长;而随着元处理单元的增多,导致资源开销增长,RDFE 的转发速率略微下降. 与 LabelCast 相比,RDFE 的转发速率提升了近 4 倍.

6 结束语

针对当前新型网络功能的试验和部署困难的问题,本文设计并基于 NetFPGA-10G 平台实现了支持功能演进的可重构网络数据平面 RDFE,通过可编程的数据包解析和数据包处理达到内部逻辑可重构的目的,从而支持用户定制的功能部署;此外,针对 RDFE 提出协议映射和匹配映射算法,分别将用户定制的匹配域提取和数据包处理过程映射到硬件结构中. 与已有方案相比,RDFE 具有更高的转发速率和更低的资源利用率. 所提结构对未来可编程数据平面的发展具有重要意义.

参考文献

[1] Ashok Anand, Fahad Dogar, Dongs Han, et al. XIA: an ar-

chitecture for an evolvable and trustworthy Internet [A]. Proceedings of the Hotnets 2011 [C]. Cambridge, USA: ACM, 2011. 2 – 12.

[2] Van Jacobson, Diana K Smetters, James D Thornton, et al. Networking named content [J]. Communications of the ACM, 2012, 55(1): 117 – 124.

[3] Ali Ghodsi, Teemu Koponen, Barath Raghavan, et al. Information-centric networking: seeing the forest for the trees [A]. Proceedings of the Hotnets 2011 [C]. Cambridge, USA: ACM, 2011. 1 – 6.

[4] IETF. VxLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer3 Networks [S/OL]. http://tools. ietf. org/html/, 2013 – 05.

[5] 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.

[6] Anwer M B, Motiwala M, Tariq M B, et al. SwitchBlade: a platform for rapid deployment of network protocols on programmable hardware [A]. Proceedings of the ACM SIGCOMM 2010 Conference [C]. New Delhi, India: ACM, 2010. 183 – 194.

[7] Kozanitis C, Huber J, Singh S, et al. Leaping multiple headers in a single bound: wire-speed parsing using the Kangaroo system [A]. Proceedings of the IEEE INFOCOM [C]. San Deigo, USA: IEEE, 2010. 1 – 9.

[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 Zhong-jin, Li Yong, Su Li, 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] McKeown N. Keynote talk: software-defined networking [A]. Proceedings of the IEEE INFOCOM [C]. Rio de Janeiro, Brazil: IEEE, 2009. 1 – 11.

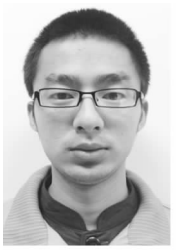
[10] L D Carli, Yi Pan, Kumar, et al. PLUG: flexible lookup modules for rapid deployment of new protocols in high-speed routers [A]. Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication [C]. Barcelona, Spain: ACM, 2009. 207 – 218.

[11] Pat Bosshart, Glen Gibb, Hun-Seok Kim, et al. Forwarding metamorphosis: fast programmable match-action processing in hardware for SDN [A]. Proceedings of the ACM SIGCOMM' 13 Conference [C]. Hong Kong, China: ACM, 2013. 99 – 110.

[12] 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.
Lv Gao-feng, Sun Zhi-gang, Li Tao. LabelCast: A general abstraction for the forwarding plane of SDN [J]. Chinese Journal of Computers, 2012, 35(10): 2037 – 2047. (in Chinese)

[13] Github, Inc. NetFPGA-10GProject [EB/OL]. https://github. com/NetFPGA/NetFPGA-public/wiki, 2014.

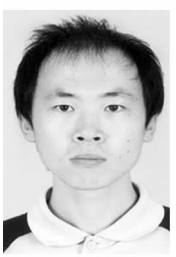
作者简介



段　通　男,1992 年生于河南驻马店. 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据平面.
E-mail: duantong21@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: ndscljl@163. com

胡宇翔　男,1982 年生于河南周口. 现为国家数字交换系统工程技术研究中心讲师. 主要研究方向为新一代信息网络关键理论与技术.



刘释然　男,1990 年生于河南洛阳. 现为国家数字交换系统工程技术研究中心硕士研究生. 主要研究方向为新一代信息网络关键理论与技术.