

弹性协议可定制的网络数据平面结构及其映射算法

刘中金 李勇* 苏厉 金德鹏 曾烈光
(清华大学电子工程系 北京 100084)

摘要: 随着网络功能的不断扩展, 新型网络协议的不断涌现, 这些协议中的数据包具有新的格式定义, 需要网络设备能够支持相应的解析和查找。软件定义网络(Software Defined Networking, SDN)基于流表的转发设计使得网络的创新变得简单, 但是仍然难以支持任意协议的可编程解析和处理。该文联合考虑数据包的解析和查找过程, 提出一种支持协议弹性定制的数据包查找硬件结构, 通过比特粒度的解析和基于元操作的查找过程, 使得任意协议能够在硬件结构上得到处理; 此外, 该文针对所提硬件结构提出一种基于多叉树的映射算法, 将用户定制协议映射到硬件处理流水线和查找表中。通过实际的 FPGA 部署验证了所提结构能够支持多种协议的灵活定制, 在硬件中的处理速度可以达到 390 Gbps, 与已有方案相比, 其硬件资源利用率有明显降低。该结构对未来的软件定义网络的数据平面设计有重要的意义。

关键词: 软件定义网络; 可编程; 协议定制; 多叉树; 映射算法

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2014)07-1713-07

DOI: 10.3724/SP.J.1146.2013.01151

Design on the Elastic Protocol Customizable Data Plane and Its Mapping Algorithm

Liu Zhong-jin Li Yong Su Li Jin De-peng Zeng Lie-guang
(Department of Electronic Engineering, Tsinghua University, Beijing 100084, China)

Abstract: Along with the growing expansion of network functions and continuous emerging of new protocols, the novel packet types and processing requirements are established in these protocols. The Software Defined Networking (SDN) makes the network innovation simpler based on the programmable flow table. However, it is still difficult for the SDN to support the arbitrary protocol processing. This paper takes into account the parsing and lookup procedure together, and proposes a hardware structure that could support elastic customization of the arbitrary protocols. By transforming the protocol processing procedure into a multi branched tree, users can arbitrarily define their own parsing and lookup processing, which is mapped into a hardware pipeline. The prototype on FPGA platform shows its flexibility in customizing protocols. The packet processing speed can achieve 390Gbps in hardware and the hardware resource utilization is significantly reduced compared to the existing scheme. The proposed architecture is quite meaningful for the data plane design of the SDN.

Key words: Software Defined Networking (SDN); Programmable; Protocol customization; Multi-branched tree; Mapping algorithm

1 引言

随着互联网的迅猛发展, 传统的网络协议难以适应用户不断增长的需求, 因此网络需要不断创新以满足这些需求, 新的体系结构和网络协议随之涌现出来。在这些协议中, 数据包不再局限于特定的数据包格式, 如 IP, MPLS 等, 且由于数据包的定

义面向新的应用而不再适应网络设备的处理能力, 如文献[1,2]等, 使得现有的网络设备难以处理这类数据包。另一方面, 为了满足 VPN 加密或数据中心网络互连等需求, 协议的嵌套也变得十分常见, 如 TRILL, FabricPath 等^[3,4], 网络设备需要做相应改动才能够适应这些协议。如果网络设备支持灵活的、可编程定制的报文解析和处理能力, 那么可以大大降低新型协议的部署难度, 网络可以在一个更灵活、更开放的框架下进行创新和演进。

软件定义网络(Software Defined Networking, SDN)^[5]通过将网络的控制平面与数据平面相分离, 使得控制功能抽象并集中到 SDN 控制器中, 控制器

2013-07-31 收到, 2014-01-23 改回

国家 973 计划项目(2013CB329105), 国家 863 计划项目(2013AA010601, 2013AA010605) 和国家科技重大专项项目(2013ZX03002001)资助课题

*通信作者: 李勇 liyong07@tsinghua.edu.cn

可以对各个网络设备进行集中控制,从而实现网络的灵活、智能可控和全局优化,但是,对于数据平面来讲,SDN 采用了基于流表的转发功能,它只能提供对 IP 和 MPLS 固定格式的支持,难以支持任意定义的数据包格式。

本文着眼于数据平面的可编程设计,使其能够灵活支持各种用户自定义的数据包格式。数据包在数据平面中的处理的过程主要包括解析、查找、转发和交换 4 个方面,其中解析和查找的编程能力决定了数据平面的灵活性。解析是指网络节点按照协议规范将数据包中指定的数据域提取出来,用于查找转发操作;查找过程是通过解析后字段的查找处理来决定数据包的输出端口。随着数据包格式呈现出多样性,需要研究高速、灵活的协议的解析结构;同时由于查找类型的不断变化,从传统的哈希匹配,最长匹配等变为数值比较、字符串匹配等操作,需要研究能够灵活定制的查找过程以满足这一需求。

为了提高数据包解析过程的处理速度和灵活性,文献[6]提出了 Kangaroo 结构,通过预读的机制同时解析多种数据包,利用 CAM (Content Addressable Memory), ALU (Arithmetic-Logic Unit)和哈希表进行查找的结构可以提供 40 Gbps 的报文解析能力,但是由于 CAM 资源占用较大,限制了它的扩展性。文献[7]提出了一种报文解析的编译器 PP(Packet Parsing),它可以将不同的协议映射到具有固定结构的硬件流水线上,由于流水线的划分以协议字段为边界,使得流水级数较多,导致算法的资源占用率较高。文献[8]提出了一种基于二叉树的报文解析结构,可以在较低的资源占用情况下达到 456 Gbps 的处理能力,但是二叉树本身只能提供两个条件转移的能力,对于多种协议同时解析的情况难以提供支持。

OpenFlow^[9,10]通过流表的方式提供了可编程定制查找过程的能力,但是其采用的相对固定的匹配域定义以及单一的匹配操作难以满足新协议的需求;Switchblade^[11]集成了一系列通用的包处理功能,用户可以从这些功能中选择相应的功能并进行组合,从而支持最长前缀匹配和精确匹配等功能,但由于这些功能模块编程粒度大,难以支持任意定义

的协议类型;文献[12-14]对路由器的可编程能力做了深入研究。

以上的研究将解析过程和查找过程分离开来,而灵活定制数据包的处理过程需要动态的协议解析过程和灵活的查找定义能力进行组合。本文针对这一问题做出了两个主要贡献:(1)提出一种支持协议弹性定制的比特粒度解析和查找的硬件结构;(2)提出一种映射算法,可以将任意的协议格式和查找过程映射到所提的硬件结构上。本文通过 FPGA 平台对所提结构进行实现和仿真验证,结果显示与已有方案相比我们可以达到 390 Gbps 的转发速率,同时降低了 50%的资源利用率。

2 设计目标

本文的目标是提出一种任意协议可定制的数据平面结构及映射算法,共有 3 个设计子目标:(1)支持任意比特的提取和查找定义;(2)支持多种协议格式的同时解析和处理;(3)高转发性能和低资源利用率。

(1)任意比特提取和查找 从之前的编程结构来看,主要有两种思路:(a)基于模块组合的方式,(b)基于协议的层级处理的方式。对于模块组合的方式来讲,模块的功能是固定的,如 TTL 更新,Checksum 计算,任何功能的变化都需要对原有模块进行修改;对于协议层级处理的方式而言,协议的定制需要考虑不同层次之间的相关性,比如 IP 协议后通常解析 TCP 协议,但是由于新的协议定义格式发生变化,多层嵌套的数据包头打破了原有协议的相关性,使得这种方式难以适应不断变化的包头处理需求。本文从数据包的处理过程出发,考虑协议的解析和查找过程本质上是将数据包的每个比特进行操作定义,通过提取特定的比特组合,并且指定该比特组合的查找操作便可以定义整个数据包查找过程,基于此,本文提出如图 1 所示的弹性协议定制的数据包解析查找模型。

首先,包的格式是可以灵活定制的,用户可以通过定义比特偏移量来定义数据包中的操作域,如图中 Field1~5;其次,这些匹配域的查找方式也可以通过从一些基本的操作集合中选择,在图 2 中给出了哈希匹配、数值比较或者用户自定义操作等;再次,在包头的操作上,用户可对包头的指定

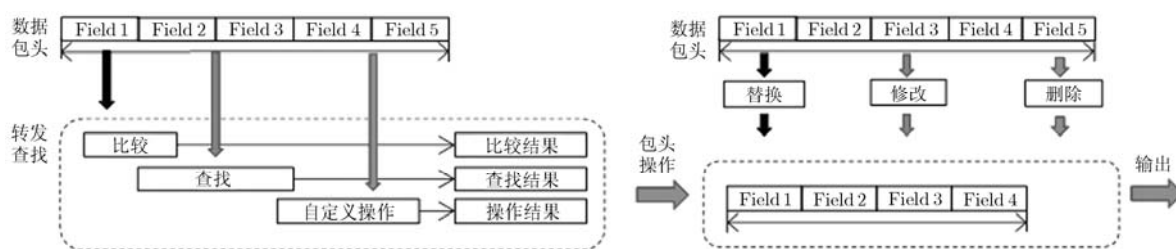


图 1 弹性协议定制处理模型

域进行操作选择及组合，比如数据域的替换、修改、删除或增加等。利用这一模型用户可以定制任何协议的解析、查找和修改过程。

(2)多协议同时解析和处理 通常在路由器中不会仅仅处理一种网络协议，网络设备往往需要支持多种数据包处理格式。但是多个协议的同时解析面临着冲突和选择的问题，设备需要能够准确地识别数据包所从属的协议类型以及进行相应的查找操作。为了解决这一问题，本文采用多叉树来表示多个协议的处理过程。如图 2 所示，有两种协议需要在设备中同时处理，左边为数据包头 1，属于协议类型 1，右边为数据包头 2，属于协议类型 2。在数据包头中 T 代表数据包头的类型域，用于判断数据包的协议类型； F 代表匹配域，表示除类型域外的操作域。两类协议具有不同的包头结构：二者具有相同的第 1 层包头信息 $\{F1, T1\}$ ，在协议树中，每个匹配域或类型域都用一个树节点表示；通过对 $T1$ 域的包类型判断会有不同的进一步包头解析的结果，对应 3 棵以 $T1$ 为根节点的子树， $\{F2, T2\}$ 和 $\{FII, TII\}$ ，如此进行迭代可以得到一棵完整的协议树。通过这种方式，多种协议的解析和处理过程都可以在一棵协议树中表示。

(3)高转发性能及较低的资源利用率 在传统路由器中，数据包进入路由器后首先进行数据包解析，按照各个数据包中的类型域的偏移量将所有的操作域提取出来，这些解析后的结果将会在不同的处理逻辑中进行处理。在这一解析过程中，所有的操作域被提取出来，但是在实际的数据包查找过程中，这些操作域又被重新聚合，本来花费代价从数据包中提取出来的数据域又被重新组合进行查找处理，反而浪费了大量的处理资源。表 1 分析了一些现有的协议的处理需求，从该表可以看出，实际的包处理过程涉及到的包处理操作种类是十分有限的(3, 4 种)，主要区别在于匹配域的数目和处理的位宽。在长度有限的情况下，多个短位宽的逻辑操作与单个长位宽的相比会需要更多的逻辑资源和更多

表 1 各种协议涉及到的操作类型

协议类型	匹配域数目	最长匹配位宽(bit)	操作类型
MPLS	4	20	大于/等于
Q-IN-Q	8	16	等于
802.1Q	4	16	等于
802.3	2	48	等于
IPv4	13	32	大于/等于/LPM
IPv6	8	128	大于/等于/LPM
NDN	3	>100	等于/LPM

的控制信号。因此，如果能够将不同的处理域进行组合处理可以节省逻辑资源，并且提高数据包的处理速度。

3 弹性协议定制硬件结构

为了实现任意协议的解析和多个域的任意组合，本文提出如图 3 所示的弹性协议定制结构(Elastic Protocol Customization, EPC)来实现数据包的可编程查找。

数据包通过总线发送到查找结构中，这一结构分为解析和查找两部分，解析功能由多个解析处理单元构成的流水线完成。在流水线的每级处理单元中，宽度为总线位宽的数据包字段被处理，这些处理是指根据用户定义的参数进行类型判断、域提取等；匹配域组合模块的操作是将各级流水线输出的匹配域进行移位和拼接成一个较长的查找和处理字段。数据包查找转发模块将拼接成的字段按照类型区分进行统一的查找处理，比如哈希查找、比较操作或 LPM 查找等。

在上面的结构中，解析模块最为关键，其详细的设计结构如图 4 所示。主要包括 3 个部分，域提取模块、类型查找模块以及匹配域输出模块。其中域提取部分由 1 个双端口的随机存取器(RAM), 1 个多路复用器(MUX), 1 个类型域提取模块和 1 个匹配域预提取模块组成。RAM 中存储有两类偏移

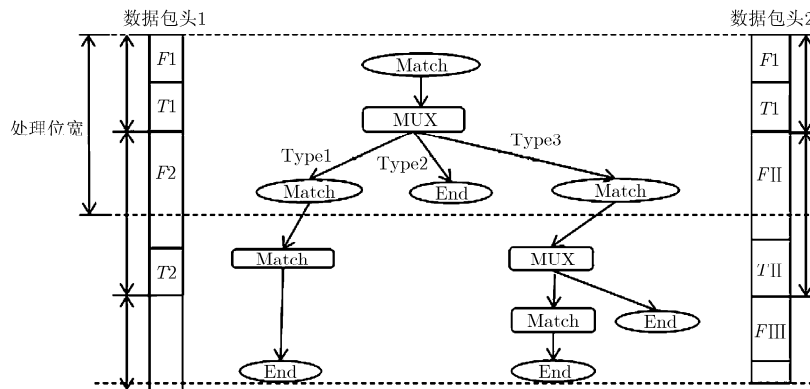


图 2 多个协议处理过程的多叉树表示

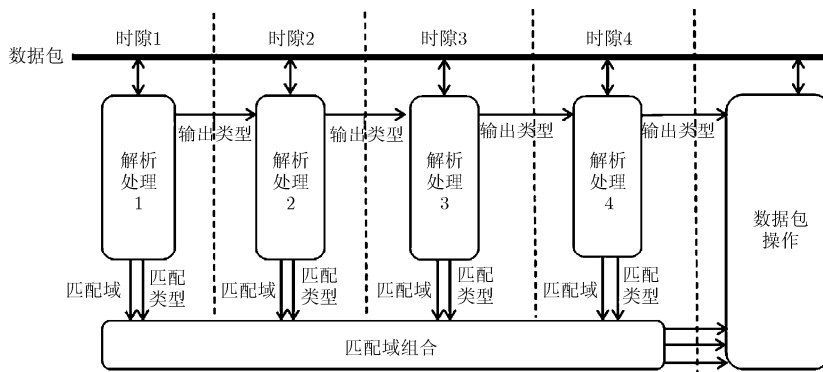


图 3 协议可定制的数据包查找结构

量：类型域的偏移量和匹配域的偏移量，MUX 根据前一级的包类型解析结果从 RAM 中分别提取类型域和匹配域的偏移量信息，类型域提取模块根据这些偏移量信息分别进行类型域提取，组成一个 TCAM 查找表项；预提取模块同时提取多个可能的匹配域，并存储在 RAM 的不同地址中。提取出的类型域在 TCAM 中进行最长前缀匹配查找，得到当前数据包对应的协议类型，从而从 TCAM 对应的 RAM 中得到预提取的 RAM 入口地址以及提取长度。最终通过这些操作后将得到的数据包类型发送到下一级流水线单元中，提取出的匹配域则按照匹配域操作类型分类发送到匹配域的组合模块中。

4 弹性协议定制映射算法

用户在定制协议时不会直接面向硬件进行编程，因此需要提供映射机制将用户定制的协议存储到硬件结构中去，本文于多叉树结构将协议存储到硬件结构中去。映射过程有 3 个主要的考虑因素，一是按照总线位宽进行协议树的分割，当总线位宽较短时，协议处理模块无法在单个时钟周期内处理所有的包头信息，因此需要将数据包按照总线宽度进行分割，分别在不同的流水线单元中进行处理；二是考虑是节点的压缩，同样的操作可以进行合并处理以节省查找资源；三是存储器的配置，包括

RAM 和 TCAM 的配置。基于以上考虑本文提出两个算法，第 1 个是将协议树划分为多个总线位宽的子树，第 2 个是将每个划分后的子树进行压缩映射到硬件结构中。协议树中的符号及具体涵义如表 2 所示。

表 2 协议树中的一些符号以及具体涵义

符号	具体涵义
T	协议树
$T(v)$	以节点 v 为根节点子树
$d(T)$	协议树的深度(比特为单位)
$d\{T\}$	一组协议树集合的深度(取所有树深度的最大值)
$d(v/T)$	在树 T 中节点 v 的比特深度
$P(v)$	节点 v 的父亲节点
$\{P(v)\}$	节点 v 的所有前级节点集合
$C(v)$	节点 v 的所有子节点集合

算法 1 实现了协议树的分解，它将处于边界上的树节点进行分解，使得协议树被分割多个对应总线位宽的子树，分割后的子树被随后映射到硬件的单个解析单元中。本文所涉及的协议树分割算法的基本描述如表 3 的算法 1 所示。

定义 1 如果某一操作 $Oper(\bullet)$ 的处理结果与操作数的长度无关，则称该操作为可合并操作。

对于任意的两个匹配域 $a1$ 和 $a2$ ，如果二者操作结果相同，即 $Oper(a1)=Oper(a2)$ ，那么可合并操作对于按照 bit 拼接后的组合 $\langle a1, a2 \rangle$ 也会有相同的处理结果，即 $Oper(a1)=Oper(a2)=Oper(\langle a1, a2 \rangle)$ 。比如哈希匹配，数值比较都是可合并的操作；而最长前缀匹配则是不可合并的操作。

算法 2 描述了一棵树映射到硬件结构的映射过程，如表 4 所示，在每个切分出的子树中，将类型

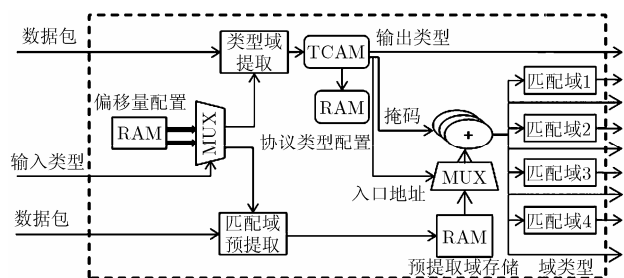


图 4 解析模块内部结构

表 3 协议树分割算法

算法 1 协议树分割算法 $\text{Cut}(T)$ 主程序

- (1) $T_r = T$
 - (2) While $d\{T_r\} > W_{\text{bus}}$ Do
 - (3) for T' in $\{T_r\}$
 - (4) if $d(T') > W_{\text{bus}}$ then
 - (5) find $v : d(P(v)/T') < W_{\text{bus}}, d(v/T') > W_{\text{bus}}$;
 - (6) cut v into v_u and v_d :
 - (7) $v_u : d(v_u/T') = W_{\text{bus}}$;
 - (8) $v_d : d(v_d/T') > W_{\text{bus}}$;
 - (9) $\text{Cut}(T'/T(v_d))$;
 - (10) put $T(v_d)$ into $\{T_i\}$
 - (11) else
 - (12) $\text{Map}(T')$
 - (13) end if
 - (14) end for
 - (15) $\{T_r\} = \{T_i\}$
 - (16) end while
-

域单独提取出来，放置到树的根节点处，以根节点的选择建立相应数目的子树。建立单个子树时，如果相邻的几个匹配域是可合并的，那么将这些匹配域进行合并，为了节省空间，在不同子树中共用的匹配域存储在一个节点上。

以图 5 为例，有两种数据包需要在硬件中解析，二者共用相同的类型域 $T1$ ，第 1 步将 $T1$ 提取出来作为根节点，以它的 3 种跳转条件建立 3 棵子树，如图 5(b)所示，由于 $F1$ 和 $F2$ 两个域是相邻的，而且操作都是精确匹配，所以两者可以进行组合，因此左侧子树可以用两个节点存储原来的 3 个节点的内容，又由于 3 棵子树的第 1 个节点内容是相同的，所以 3 棵子树的第 1 个节点可以合并为一个节点，硬件存储时，包括类型节点偏移量一共存储 3 个节点的偏移量；在预提取区域存储 2 个值，对应左侧和右侧两棵子树；在 TCAM 相连的 SRAM 中存储 3 对偏移量，分别对应 3 个匹配节点的长度。

5 性能分析

在这一部分本文通过理论计算分析了算法的存储空间占用情况，并基于 FPGA 实现了所提结构的资源利用率以及处理速率。

5.1 算法存储空间分析

在算法 1 中，存储空间主要分为 3 部分：存储偏移量的 RAM，存储类型的 TCAM 以及存储预提

表 4 子树映射算法

算法 2 子树映射算法 $\text{Map}(T)$ 主程序

- (1) if $\exists\{t_1, t_2, \dots, t_n\} \subset T, (t \text{ is a type node})$
 - (2) for $i = 1$ to n
 - (3) assign one offset RAM entry;
 - (4) assign one TCAM entry;
 - (5) if $\{P(t_i)\} \not\subset \{t_1, t_2, \dots\}$
 - (6) build $T(t_i)$, assign an entry number for $T(t_i)$
 - (7) else
 - (8) $(T(t_e) \rightarrow \text{child}) \leftarrow t_i$, where $t_e \in \{P(t_i)\}$
 - (9) end for
 - (10) end if
 - (11) if $\exists\{f_1, f_2, \dots, f_n\} \subset T, (f \text{ is a match field})$
 - (12) Create a combined filed set: $F_c = \{f_1\}$;
 - (13) Mark the element number in $F_c : C_n \leftarrow 1$;
 - (14) for $i = 1$ to n
 - (15) for $k = i - 1$ to 1
 - (16) if $\text{oper}(f_i) \neq \text{oper}(f_k)$
 - (17) $F_c = \{F_c, f_i\}$;
 - (18) $C_n \leftarrow C_n + 1$;
 - (19) else
 - (20) combine f_i with f_k into f_k
 - (21) break;
 - (22) end if
 - (23) end for
 - (24) end for
 - (25) for $j = 1$ to C_n
 - (26) assign offsets to offset RAM
 - (27) assign one entry and corresponding mask to TCAM associated SRAM
 - (28) end for
 - (29) end if
-

取域的 RAM。(1)令需要解析的数据包头长度为 L ，总线位宽为 W ，那么流水线的级数为 $S = \lceil L/W \rceil$ ，每个偏移量对的位宽为 $2\log_2(W)$ 。树节点的数目为 $N(T)$ ，那么整体的偏移量存储需求为 $2 \times \log_2(W) \times N(T)$ 。(2)单个 TCAM 条目的最大空间需求为 $\sum_{t=1}^M l_t$ ，其中 M 为总的类型域个数。由于要存储 MASK 的数目为 $N(T)$ ，因此 TCAM 条目为 $N(T)$ ，因此可以得到 TCAM 的存储空间需求上界为 $N(T) \times \sum_{t=1}^M l_t$ 。(3)预提取的存储空间，

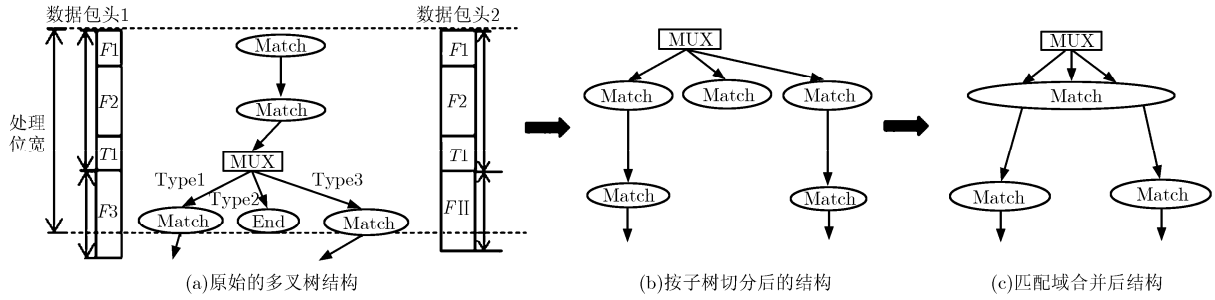


图 5 算法 2 映射过程

单条条目最大宽度为 W ，协议种类为 K ，因此最大的协议需求为 KW ，连接 TCAM 的 RAM 中存储的 Mask 数目为 $N(T) \times W$ 。因此整体的 RAM 存储需求为 $2 \times \log_2(W) \times N(T) + N(T) \times W + KW \times \lfloor L/W \rfloor$ ，TCAM 存储需求为 $N(T) \times \sum_{t=1}^M l_t$ 。

本文以在同一硬件中解析 7 种协议为例，分别为以太网，MPLS，802.1Q，IPv4，IPv6，TCP 和 UDP，这些协议对应的总的位宽为 $L=78$ Byte，节点数目为 $N(T)=41$ ，类型域的总宽度为 5 Byte。表 5 中列出了不同总线位宽情况下的资源需求情况。

表 5 在不同位宽情况下的存储资源需求

位宽(bit)	压缩前/后的 RAM 需求(kb)	压缩前/后的 TCAM 需求 Byte
64	7.596/6.228	205/115
256	16.528/11.632	205/115
512	28.898/19.358	205/115
1024	49.972/31.180	205/115
2048	99.206/61.946	205/115

从表 5 来看，通过算法 2 的合并压缩，可以使整体的节点需求以及 RAM 存储资源需求降低 18%~37%，TCAM 的存储开销降低了 44%。

5.2 资源开销分析

本文在 NetFPGA-10G^[15] 板上做了原型实现，其 FPGA 型号为 Virtex-5VTX240T。在不同的位宽和流水线级数情况下的设计会影响到整体的资源利用率。图 6 列出了 Slice 和 BRAM 资源利用

率与比特位宽，流水线级数的关系。

可以看出，Slice 的资源利用率在不同的流水线级数条件下的不会有太大变化，但是数据位宽的变化对资源利用率的影响较大。BRAM 的占用率随着级数和位宽的变化基本都呈现出线性增长的关系。由于 Virtex-5VTX240T FPGA 提供的逻辑资源十分丰富，本文所提设计的整体的资源占用率低于 7%，使得有足够的剩余空间在单块 FPGA 上实现队列控制、数据包处理等功能，从而实现完整的路由器设计。

5.3 速率比较

本文在 1024 bit 位宽的情况下，对所提结构进行了综合以及布局布线，结果显示时钟速率可以达到 198.952 MHz，理论上的最高的转发速率可以达到 203.7 Gbps。由于硬件的并行工作特性，数据包处理可以通过增加并行流水线提高整体系统的转发速率，为此本文同时设计了两路并行的处理结构做比较，以 EPC2 表示，结果显示两路并行设计的时钟速率可以达到 192.314 MHz，因此其最高的系统转发速率为 393 Gbps。

本文将所提的 EPC, EPC2 结构与文献[6]中的 kangaroo、文献[7]中的 PP 结构进行了比较，图 7 给出了在转发速率、Slice 资源以及 BRAM 资源方面的比较，从结果来看，在比 kangaroo 资源利用率增加 60%情况下，kangaroo 只能提供 40 Gbps 的转发速率，而 EPC 可以提供 203 Gbps 的转发速率。

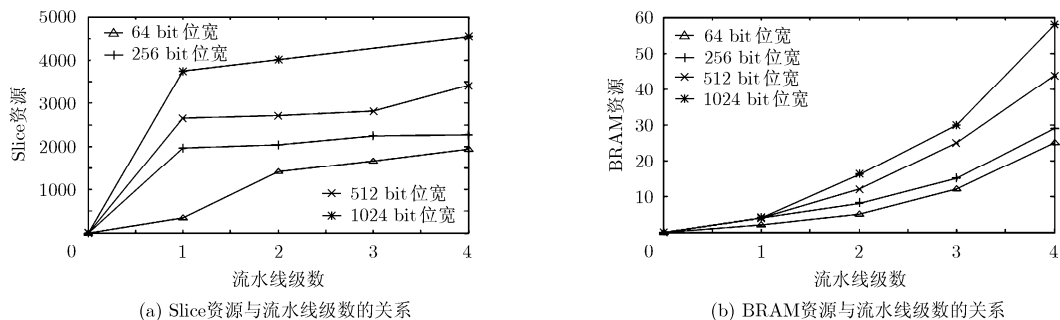


图 6 Slice 及 BRAM 资源利用情况

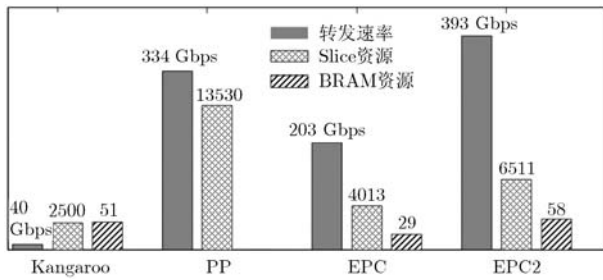


图 7 本文提出结构与 kangaroo, PP 性能和资源比较

在有两路并行通道的条件下，EPC2 的资源占用率只是 PP 的 48%，转发速率却可以达到 393 Gbps。这是由于 PP 采用的流水线结构是根据协议类型来进行分级的，在同样的条件下，5.1 节中所述的协议解析过程在 PP 中需要流水线级数为 7，而在 1024 bit 位宽下，EPC2 结构仅仅需要两路 1 级流水就可以完成。

6 结束语

针对现有的网络设备难以支持协议的可编程解析和处理的问题，本文联合考虑数据包的解析和查找功能，提出了一种支持协议弹性定制的数据包查找硬件结构，并针对所提硬件结构提出基于多叉树的映射算法。通过 NetFPGA-10G 平台部署验证了所提结构能够支持多种协议的灵活定制能力以及高速转发能力，该结构对未来的软件定义网络的数据平面设计有重要的意义。

参考文献

- [1] Akella A, Andersen D, Byers J, *et al.*. Expressive Internet Architecture[OL]. <http://www.cs.cmu.edu/~xia/>, 2013, 7.
- [2] Van Jacobson, Crowley P, Zhang L, *et al.*. Named Data Networking[OL]. <http://www.named-data.net/>, 2013, 7.
- [3] Perlman R and Eastlake D. Trill[OL]. <http://www.ipjforum.org/?p=582>, 2013, 7.
- [4] Cisco. Fabricpath[OL]. <http://www.cisco.com>, 2013, 7.
- [5] McKeown N. Keynote talk: software-defined networking[C]. Proceedings of the IEEE INFOCOM, Rio de Janeiro, Brazil, 2009: 1.
- [6] Kozanitis C, Huber J, Singh S, *et al.*. Leaping multiple headers in a single bound: wire-speed parsing using the Kangaroo system[C]. Proceedings of the IEEE INFOCOM, San Diego, USA, 2010: 1-9.
- [7] Attig M and Brebner G. 400 Gb/s programmable packet parsing on a single FPGA[C]. Proceedings of the ACM/IEEE Seventh Symposium on Architectures for Networking and Communications Systems, New York, 2011: 12-23.
- [8] 董永吉, 郭云飞, 黄万伟, 等. 一种新的高速报文解析结构研究[J]. 电子与信息学报, 2013, 35(5): 1083-1089.

- [9] Dong Yong ji, Guo Yun fei, Huang Wan wei, *et al.*. A new high-speed packet parsing architecture[J]. *Journal of Electronics & Information Technology*, 2013, 35(5): 1083-1089.
- [9] McKeown N, Anderson T, Balakrishnan H, *et al.*. OpenFlow: enabling innovation in campus networks[J]. *ACM SIGCOMM Computer Communication Review*, 2008, 38(2): 69-74.
- [10] 周焯, 李勇, 王芳, 等. 基于 OpenFlow 的网络实验平台技术 [J]. 清华大学学报 (自然科学版), 2012, 52(11): 1540-1544.
- Zhou Ye, Li Yong, Wang Fang, *et al.*. OpenFlow network experiment platform[J]. *Journal of Tsinghua University (Science and Technology)*, 2012, 52(11): 1540-1544.
- [11] Anwer M B, Motiwala M, Tariq M, *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.
- [12] Liu Zhong-jin, Li Yong, Su Li, *et al.*. Grainflow: a per-bit customizable scheme for data plane innovation on programmable hardware[C]. Proceedings of the ACM CoNEXT Student Workshop, Tokyo, Japan, 2011: 1-2.
- [13] Lu Guo-han, Shi-Yun feng, Guo-Chuan xiong, *et al.*. CAFE: a configurable packet forwarding engine for data center networks[C]. Proceedings of the 2nd ACM SIGCOMM Workshop on Programmable Routers for Extensible Services of Tomorrow, Barcelona, Spain, 2009: 25-30.
- [14] Lu Guo-han, Guo Chuan-xiong, Li Yu-long, *et al.*. ServerSwitch: a programmable and high performance platform for data center networks[C]. Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation, Boston, USA, 2011: 1-14.
- [15] Lockwood J W, McKeown N, Watson G, *et al.*. NetFPGA — an open platform for gigabit-rate network switching and routing[C]. Proceedings of the IEEE International Conference on Microelectronic Systems Education, San Diego, USA, 2007: 160-161.

- 刘中金：男，1988 年生，博士生，研究方向为软件定义网络、数据中心网络、可编程虚拟化路由器等。
- 李勇：男，1985 年生，博士后，研究方向为软件定义网络、下一代 IP 网络体系结构、移动管理、移动容迟网络、网络虚拟化等。
- 苏厉：男，1976 年生，讲师，研究方向为片上网络、软件定义网络、短距离无线通信等。
- 金德鹏：男，1972 年生，教授，研究方向为软件定义网络、片上网络、短距离无线通讯等。
- 曾烈光：男，1947 年生，教授，研究方向为通信网、ASIC 设计、片上网络、下一代网络体系架构等。