LabelCast：一种普适的SDN转发平面抽象

吕高锋 孙志刚 李 韬 毛健彪 杨 安
（国防科技大学计算机学院 长沙 410073）

摘 要 在互联网体系结构演进过程中试验和部署新型网络协议比较困难，基于Openflow的软件定义网络SDN提供了一种简单易行的方法. OpenFlow基于流表实现了多级流水转发处理，然而Openflow不支持对网络中计算和存储等资源的描述，因此很难支持以内容为中心的新型网络. 为扩展SDN能力，提出了一种普适的转发平面抽象LabelCast，以将多态网络地址映射到定长标签. 转发平面基于Label表来查找和调度弱语义的转发操作相关的指令，Cast表对网络协议语义或状态相关的服务进行组织，支持动态扩展新型服务以满足新型网络体系结构复杂的处理功能. LabelCast能够支持基于流的端到端的交换和以数据或服务为中心的非端到端的新型网络体系结构，为SDN提供了一种普适的转发平面抽象.

关键词 软件定义网络；转发平面抽象；标签；服务
中图法分类号 TP393 **DOI号**：10.3724/SP.J.1016.2012.02037

LabelCast：A General Abstraction for the Forwarding Plane of SDN

LV Gao-Feng SUN Zhi-Gang LI Tao MAO Jian-Biao YANG An
（*School of Computer*, *National University of Defense Technology*, *Changsha* 410073）

Abstract In the evolution of Internet, it is very hard to test and deploy new network protocols in production networks for researchers, SDN based on Openflow tries to provide a feasible way. Openflow implements flows forwarding based on multiple tables by multiple pipelines, while Openflow does not describe the ability of computing and storing within networks, which could not support content-centric networks. To enhance the scalability of SDN, a general abstraction of the forwarding layer, LabelCast, is proposed, which maps network addresses to fixed-length labels and characterizes the ability of forwarding operations and processing functions with Label and Cast tables. Forwarding layer lookups based on fixed-length labels and schedules services, including light-semantics instructions of general forwarding operations, and network protocol semantics or status-related processing functions arranged by the Cast table, which could be extended to support the complexity processing of new networks. LabelCast supports end-to-end forwarding based on flows and non-end-to-end forwarding in content or information centric network architectures, which supplies a general abstraction layer for SDN.

Keywords software defined network; abstraction of forwarding plane; label; service

收稿日期：2012-06-30；最终修改稿收到日期：2012-08-12. 本课题得到国家“八六三”高技术研究发展计划项目基金（2011AA01A101、2009AA01A334）、国家“九七三”重点基础研究发展规划项目基金（2012CB315906，2009CB320503）资助.**吕高锋**，男，1980年生，博士，助理研究员，主要研究方向为新型互联网体系结构、高性能路由与交换技术. E-mail：lvever@nudt.edu.cn. **孙志刚**，男，1973年生，博士，研究员，主要研究领域为网络体系结构、高性能网络交换. **李 韬**，男，1983年生，博士，助理研究员，主要研究方向为计算机网络、网络处理器. **毛健彪**，男，1988年生，硕士研究生，主要研究方向为新型互联网体系结构. **杨 安**，男，1988年生，硕士研究生，主要研究方向为计算机网络.

1 引 言

在互联网演化过程中,研究人员提出了众多新型协议和技术,然而很难在生产网络中试验和部署,主要是由于网络设备和协议开发环境由设备制造商控制.为了加速网络协议创新和开发,人们提出了软件定义网络 SDN(Software Defined Network).SDN 基于 Openflow[1]转发层抽象统一了网络转发层配置接口,通过引入网络操作系统向网络应用控制器提供了简单统一的编程开发和配置接口以及全局网络视图等.SDN 支持网络虚拟化等技术,便于开发人员在已有网络隔离的资源中开发和试验新型网络协议.

Openflow 作为 SDN 的转发层抽象,采用三元组规则〈Matching,Instructions,Counters〉构成的流表描述网络转发功能,Openflow 控制器通过下载规则到 Openflow 交换机来控制转发行为.Openflow 交换机提取关键字过程中需要根据已知协议类型来解析报文,无法识别新型网络协议,而且基于变长的三元组规则的查表转发的硬件实现复杂.Openflow 交换机多级流水结构功能简单,将新型网络协议的处理映射到已有的多级流水处理结构又比较复杂,存在多级流水处理不支持新型网络协议所需要的数据存储和名字转发等特殊处理、流表项与新型网络协议表项不匹配等问题.

另一方面,下一代互联网新型协议试验和部署对 SDN 网络转发层抽象提出了更高的要求.目前,下一代互联网体系结构演进是网络领域研究的热点,如 NDN[2]、XIA[3]、Nebula[4]等 FIA 研究项目,其中,XIA 提出了有向图地址、NDN 提出了支持 Everything over Name 的名字网络,Nebula 提出了网络数据中心云概念.新型互联网体系结构设计目标是保持沙漏型体系结构和可扩展的端-端模式,开发新的控制功能,如内容路由功能.新型网络协议开发和试验需要改变数据平面转发行为,而已有网络硬件无法满足新型网络协议需要的计算和存储等特殊处理.

下一代互联网细腰演进不再是打补丁,而是在网络层解决制约互联网能力扩展的命名和寻址等基本问题[5-6],新型网络协议设计实现、试验部署等面临多方面的挑战,需要一种普适的转发平面抽象.而普适的转发平面抽象需要具备如下特点:

(1)适应网络地址的演化;

(2)支持转发平面功能扩展;

(3)为控制平面提供统一的管理配置接口.

我们提出了一种新型网络转发平面抽象 LabelCast,用 Label 表和 Cast 表分别表示网络节点转发操作和处理服务的能力.转发平面基于 Label 表来查找和调度弱语义的转发操作相关的指令,Cast 表对网络协议语义或状态相关的特殊处理的服务进行组织,支持动态扩展新型服务以支持新型网络体系结构特殊处理的需求.逻辑集中式 LabelCast 控制器将多态网络地址映射到定长标签,以支持 SDN 中新型网络协议开发和试验.

LabelCast 转发平面包括基于 OLV(Offset Length Value)的基本报文选项修改和输出控制等动作指令,以及基于计算和存储资源的面向任意字段修改和匹配的服务,分别实现弱语义的通用操作和网络协议语义或状态相关的特殊处理.转发平面基于弱语义定长标签来聚合不同端系统的请求同一内容或服务的报文,并调度动作指令和服务,具有硬件实现简单等优点.另外,服务能够灵活重组,实现关键数据路径上协议相关的报文特殊处理的服务,如新型网络体系结构需要的数据 Cache 和标签替换等,又能够以计算和存储资源为基础动态更新服务以支持未来新型网络协议特殊处理需求,具有良好的可扩展性.

LabelCast 能够支持基于流的端到端的转发和以数据或服务为中心的非端到端等多种网络体系结构,为 SDN 提供了一种普适的转发平面抽象,支持新型网络协议的开发和试验.文献[7]与 LabelCast 思想不同,提出了一种支持普适服务的传输层架构,主要针对端系统传输层协议.

本文第 2 节介绍相关研究;第 3 节提出普适转发平面抽象 LabelCast,设计了 Label 表、Cast 表、服务扩展机制以及基于标签的多协议统一承载机制;第 4 节基于 LabelCast 设计基于规则的报文转发和基于名字的 NDN 报文转发,并与其它实现方法进行比较和分析;第 5 节设计实现 LabelCast 转发原型系统 NetMagic-Pro;最后是总结和下一步工作.

2 相关研究

IETF 提出的转发与控制分离模型 ForCES(Forwarding and Control Element Separation)[8]是最早关于网络转发与控制平面的抽象.在 ForCES 中转发平面在 FIB 表中查找目的 IP 地址获得输出

端口号来转发报文，而控制平面运行路由协议，计算路由来配置转发平面 FIB 表. ForCES 模型提出了转发与控制平面的接口，支持控制器对不同网络节点转发单元的控制，简化了路由器设计与实现. OpenRouter[9]模型进一步细化了转发与控制接口，与 ForCES 功能相似.

Openflow 提供了以三元组规则〈Matching，Instructions，Counters〉表示的网络转发层抽象. 在 Openflow 发展过程中对规则进行不断扩展. 匹配域从 Openflow 规范 1.0 中以太网、IP 和 TCP 等选项构成的 12 元组扩展到 Openflow 规范 1.1 中 15 元组，现在又扩展成 Openflow 规范 1.2 中基于 TLV 表示的可变的匹配域. 交换机硬件实现复杂的匹配操作，试图支持未来新型网络协议. 然而，仅扩展匹配域选项并不一定能够预测并适应未知的新型网络协议开发和部署的需求.

Openflow 交换机功能也在不断发展，但还只是局限于对基本报文选项的处理，如修改和替换等，并不能提供新型网络应用，如数据 cache 等. Openflow 处理流程也从以前的单表处理扩展到多表流水处理，而多表流水处理结构主要是为了解决规则扩展过程中组合爆炸问题，以简单的流水处理方式实现单级复杂处理.

Openflow＋[10]针对 Openflow 流表实现复杂等问题，从转发功能、控制功能、转发与控制交互接口和流表等方面对 Openflow 进行扩展，引入 TLV 表示方法，支持将规则分解到 ACL 和 FIB 等表，能够在商用路由器中实现基于规则的报文转发. Openflow＋简化了 Openflow 实现，能够加速 Openflow 商业化部署.

Openflow 试图提出通用的网络转发层抽象，以支持网络体系结构演进. 然而，不断扩展匹配域的方法只使得匹配域变长，硬件实现更加复杂. 仅仅在转发层的扩展并不能解决未来新型网络协议开发和部署中遇到的问题. 另外，Openflow 报文解析以已知协议类型为基础，Openflow 交换机不支持新型网络协议报文的解析，需要集中式 Openflow 控制器来处理.

Juniper 从服务角度对网络转发平面进行了扩展，提出了服务平面[11]的概念，通过在路由器中集成计算和存储模块，提供计算和存储等能力，为第三方服务提供了运行平台，同时提供了开发工具 SDK. 在 Juniper 路由器中支持第三方服务驻留路由器服务平面. Juniper 服务平面能够支持多种服务，并具有良好的可扩展性.

Juniper 服务平面提供了与网络节点紧耦合的计算和存储资源，提供的服务主要是网络管理相关方面的服务，不涉及关键数据路径上深度报文处理，如缓存或组播等. 另外，Juniper 服务平面应用的开发者仅限于经过 Juniper 认证的合作者，缺乏开放性.

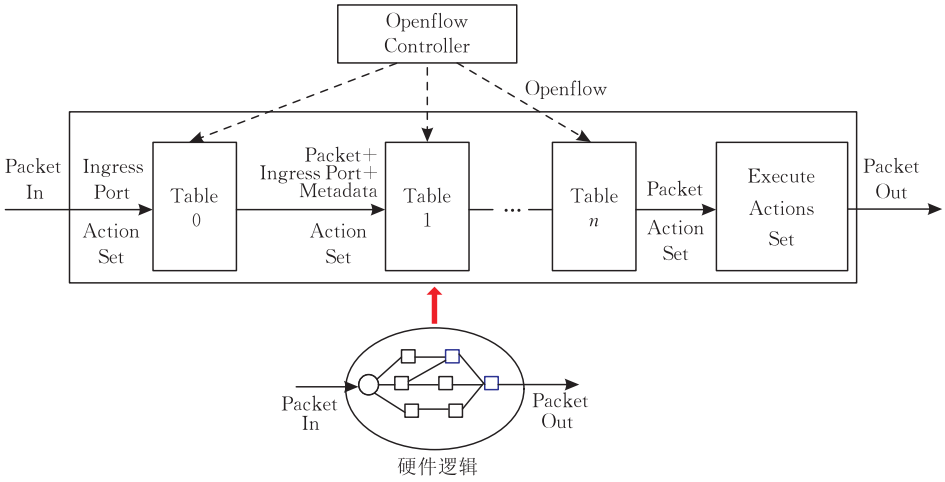
下一代互联网体系结构，如 XIA、NDN 和 Nebula 等，提出了不同的转发平面抽象. NDN 面向名字转发，将交换机硬件抽象为 Content Store 表(CS)、Pending Interest 表(PIT)和转发表. NDN 在 CS 表中提供名字对应的数据的存储状态，在 PIT 表中记录 Interest 的请求，用于消除冗余的请求，根据 FIB 表转发 Interest 报文. NDN 的转发层抽象不仅要支持转发，还要支持存储和冗余请求的消除等高级处理功能，这些功能是 Openflow 转发平面无法提供的. XIA 采用有向图标识源和目的通信地址. 有向图中的每个节点为一个实体标识，支持的实体标识类型包括地址、服务标识、内容标识等. 在有向图中通过增加“fallback”机制实现分组在不同的寻址机制中切换. 有向图表示地址的机制可以有效解决路由器对端到端通信和非端到端通信的统一处理问题. AIP[12]提出了管理域 ID 加端系统 ID 的层次地址结构，首次采用完全自验证的编址方式，把安全作为网络层第一考虑的要素，支持接入路由器对用户身份的验证，提供管理域边界路由器对跨域的分组的身份验证，可以有效防止身份欺骗和 DOS 攻击等.

新型网络体系结构不仅需要转发平面实现基本的转发功能，还需要转发平面有一定的报文深度处理能力. 而且新型网络协议对网络演进目标有不同的见解，对于网络实体命名和寻址、协议承载层等有不同的设计，因此新型网络转发平面应具有良好的可扩展性.

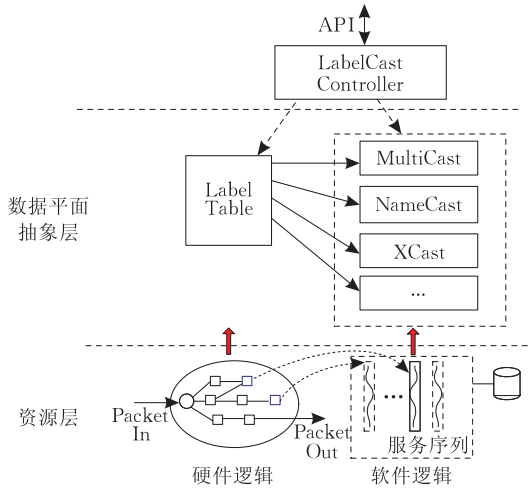
3　普适的转发平面抽象 LabelCast

随着微处理器设计和制造工艺水平的发展，通用多核处理器的性能越来越高，能够高速实现越来越复杂的网络处理. 另一方面，下一代互联网演化提出了以内容或服务为中心的新型网络体系结构，需要转发平面性能更高，功能更加灵活. 新型互联网协议服务与 Openflow 多级流水处理结构的映射很复杂，如图 1(a)所示，存在多级流水处理不支持新型网络协议，以及流表与新型网络协议规则表项不匹配等问题. 面向 SDN 中新型网络协议开发，为减小硬件查找匹配实现的复杂度，扩展网络转发平面功

能,我们提出了普适的转发平面抽象 LabelCast.以定长标签为基础,调度网络状态无关的对报文转发操作的动作指令和状态相关的对报文深度处理的服务,统一承载多种网络协议,支持基于流的端到端的转发和以内容或服务为中心的非端到端的新型网络转发.如图 1(b)所示,LabelCast 能够有效利用网络节点中的计算和存储资源,分别实现新型网络协议语义或状态相关的特殊处理和弱语义的通用操作,而且服务原语能够动态扩展以满足新型网络体系结构特殊处理需求.



(a) Openflow模型



(b) LabelCast模型

图 1 Openflow 模型与 LabelCast 模型比较

3.1 转发平面抽象

随着网络处理器性能的不断提高和存储资源的增加,网络节点除了基本的转发操作(Forwarding Operations),还具有强大的计算和存储能力[13],能够为网络服务例程运行提供平台,从而实现丰富的关键数据路径深度报文处理功能(Processing Functions).

定义 1. 指令.转发平面中对报文基本选项的修改,报文输入输出控制等转发操作(Forwarding Operations).

定义 2. 服务.基于网络中集成的计算和存储资源运行的数据路径上的报文的任意字段查找和修改等特殊处理(Processing Functions),与网络协议语义或状态相关.服务是对计算和存储资源的抽象,为用户自定义服务开发提供系统库和应用库级支持,表示为 $atomService$,即 $atomService=app(compResouce,storeResource,pkt)$.

LabelCast 用 Label 表和 Cast 表对报文转发等操作和基于网络节点集成的计算和存储资源的深度处理等网络服务进行描述和管理,如图 2 所示.

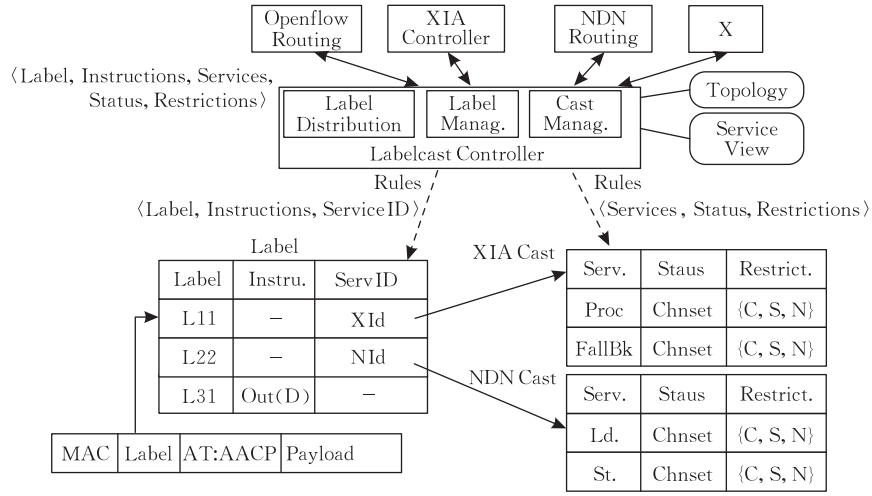


图 2　LabelCast 控制与处理模型

定义 3.　Label 表. 表示转发平面处理报文的动作指令或者服务索引，表项包括标签、选项域、动作指令域、服务索引 ID 域，记为〈Labels，Instructions，ServiceID〉.

定义 4.　Cast 表. 表示转发平面深度处理报文的服务，表项包括服务域、状态域和资源约束域，记为〈Services，Status，Restrictions〉.

Label 表项包括标签 ID、指令域、服务索引. 匹配域是报文标签，指令域 Instructions 表示转发报文的动作集合. Label 聚合不同端系统请求数据或服务的报文，并标识转发平面服务. 输入报文匹配表项后，根据表项中的指令进行处理，如修改标签状态属性、端口状态属性、替换标签和输出等通用的弱语义操作. 服务索引指向实现特殊处理的服务. LabelCast 控制器用 Label 表控制 LabelCast 转发平面处理行为.

Cast 表项包括服务域、状态域和资源约束域. 服务域表示深度处理报文的服务. 服务基于网络单元集成的计算和存储资源实现新型网络协议语义或状态相关的特殊处理功能，如任意字段的修改和查表、存取数据等. 状态域包含 Metadata、资源使用状态和统计计数器. LabelCast 控制器用 Cast 表对转发平面服务进行调度.

Openflow 转发采用多级流水方式，主要实现报文选项修改等操作. 将新型网络协议处理映射到固定的多级流水处理很复杂. LabelCast 中动作指令实现弱语义的对报文的通用操作，服务实现网络协议语义或状态相关的深度处理，能够根据新网络协议需求动态增加 Cast 表项，扩展支持新型协议特殊处理的服务原语，具有良好的可扩展性.

3.2　Label 表

在 LabelCast 中，标签与报文选项是关联的，如将目的地址映射到定长标签 . 在 Label 表中用标签对处理报文的指令进行标识和调度.

3.2.1　标签映射

LabelCast 控制器将新型网络体系结构中节点地址或报文选项映射到标签，并与处理报文的指令一起来配置 Label 表，如图 3 所示. 标签可以作为报文网络层协议头，并与新型网络体系结构中节点地址对应. 标签由两部分构成：值和选项. 标签值是为网络应用分配的定长标识，与网络应用一一对应，选项是与网络应用弱语义标识关联的状态信息，是对标识的补充信息，因此网络应用的标签对应于标签空间的一部分. 以 NDN 为例对标签映射进行说明，对于结构化命名，标签域是结构化名字前缀的 Hash 值，如 parc/videos/WidgetA. mpg/v2/s1，用前缀 parc/videos/WidgetA. mpg 的 Hash 值作为标签；对于〈P：L〉形式的非结构化命名，标签用真实实体标识 RWI 公钥的 Hash 结果 P 表示.

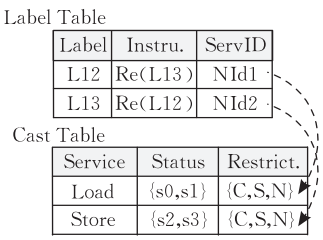


图 3　Label 和 Cast 表

标签分配方式能够为请求相同目的节点中数据或服务的报文分配相同的标签，实现不同端系统间请求报文的聚合和转发，能够支撑以数据或服务为中心的非端到端的新型网络体系结构. 另外，Label-

Cast 将可变的网络地址映射到了定长标签，可以在报文已有选项或扩展选项中携带.

3.2.2 指令设计

转发平面报文处理通常采用多级流水处理方式，包括简单的报文基本选项修改(如 TTL 减 1)和查表输出，以及标签替换等操作.我们将多级流水处理分解为粗粒度处理流程，首先是报文弱语义的基本选项修改、规则匹配等操作，然后是语义关联的深度报文处理，最后是弱语义输出控制.在 LabeCast 中，将报文基本选项修改和输出控制等转发操作作为指令，将中间阶段深度报文处理作为网络服务.指令由若干基本处理报文动作构成，如输出和丢弃动作等，指令实现了无状态报文转发等操作.

在 LabelCast 处理模型中，首先是查找 Label 表，对报文进行快速转发.根据报文携带的标签查找 Label 表，执行对应的处理报文指令，实现转发操作等简单处理.基于流表的 Openflow 转发采用了多级流水方式，利用 15 元组查找流表项选择规则，确定报文处理动作，其中包括精确匹配和带掩码匹配.与 Openflow 转发相比，LabelCast 快速转发采用精确匹配的方式，根据定长标签查找 Label 表，查表实现简单.

3.3 Cast 表

在 LabelCast 中，Cast 表对报文深度处理等网络服务进行组织，如图 3 所示，包括服务、服务状态和资源约束等.服务能够扩展，用户可以利用服务原语开发自定义服务，实现新型网络报文处理功能.

3.3.1 服务原语

LabelCast 定义了基本的服务原语 *atomService*，作为用户开发自定义服务的应用库，用户也可以定义新的服务原语.

定义 5. 缓冲区原语.表示申请用户共享缓冲区或专用缓冲区的内存操作，记为 *bufferAlloc*，*atomService*=*bufferAlloc*(*restrictions*)对服务申请存储资源的操作进行抽象.

定义 6. 线程原语.表示创建线程的操作，记为 *createThd*，*atomService* = *createThd* (*restrictions*)对服务申请计算资源的操作进行抽象.

定义 7. 注册原语.表示将用户自定义的功能加入到创建的线程中的操作，记为 *registerFun*，*atomService*=*registerFun* 对服务功能的动态加载.

在报文处理过程中涉及多种服务.如 IP over MPLS[14] 处理过程中，首先是报文特定字段的修改，然后是标签的出栈或入栈操作，最后是输出控制等操作.多服务原语顺序执行构成服务序列，实现更复杂的处理，服务序列中服务原语之间通过 Metadata 传递中间处理结果，Metadata 在 Cast 表项的状态域中记录.

3.3.2 服务扩展

LabelCast 中网络服务以网络节点中计算和存储资源为运行基础，以转发平面服务例程方式实现网络协议相关的关键数据路径上深度报文处理.转发平面服务程序能够动态升级，扩展新的服务，支持新型网络协议相关的报文处理，避免转发平面硬件不断升级.

服务是基于扩展的 LabelCast 资源容器开发和运行的.扩展的 LabelCast 资源容器提供服务运行所需的计算、存储和网络资源，利用缓冲区原语和线程原语等对网络中存储和计算等资源进行管理和配置，为服务运行提供了独立的资源空间.扩展的 LabelCast 资源容器还提供了基本的系统功能，如消息服务和传输服务，能够确保网络应用的控制消息能够传递给 Cast 服务，同时将 Cast 服务收到的未能处理的报文转发给网络应用.扩展的 LabelCast 资源容器实现了 LabelCast 控制平面与数据平面的控制与数据交互，为用户自定义的网络应用程序提供了运行平台.

在 LabelCast 处理模型中，在快速转发之后，转发平面根据 Cast 表进行转发决策，调度对报文深度处理的服务.服务实现了复杂的报文处理功能，如报文任意字段修改、标签替换、基于规则的转发控制和报文存储等语义或状态相关的深度报文处理，还实现了报文的单播、组播或基于内容的转发等.服务根据协议状态和处理策略可以动态修改转发平面 Label 表规则，调整后续报文的处理动作.另外，服务可以解析报文，产生新报文.服务增强了网络转发平面处理报文的能力，能够满足新型网络协议特殊的处理报文的功能需求.

3.4 基于 LabelCast 的多协议承载

在 LabelCast 控制器中可以将多种网络体系结构中节点地址映射到标签，如图 4 所示.当未命中的报文被送到 LabelCast 控制器后，控制器根据报文中地址，分配定长标签，并调用相应的网络应用控制器(程序)解析报文和计算报文处理规则，即动作指令或服务，最后将报文标签与规则中对报文处理动作等下载到转发平面 Label 表，同时构造 Cast 表来组织服务序列，实现网络协议相关的深度处理.另外，控制器中网络应用控制器(程序)还会计算报文转发

路径，LabelCast 控制器根据转发路径计算出标签，将标签替换等输出控制写入 Label 表. 网络应用控制器在解析报文时获得了标签选项的偏移量，采用 OLV(Offset Length Value)表示规则的匹配域，避免了转发平面查找匹配时对报文的解析，且增强匹配操作的灵活性.

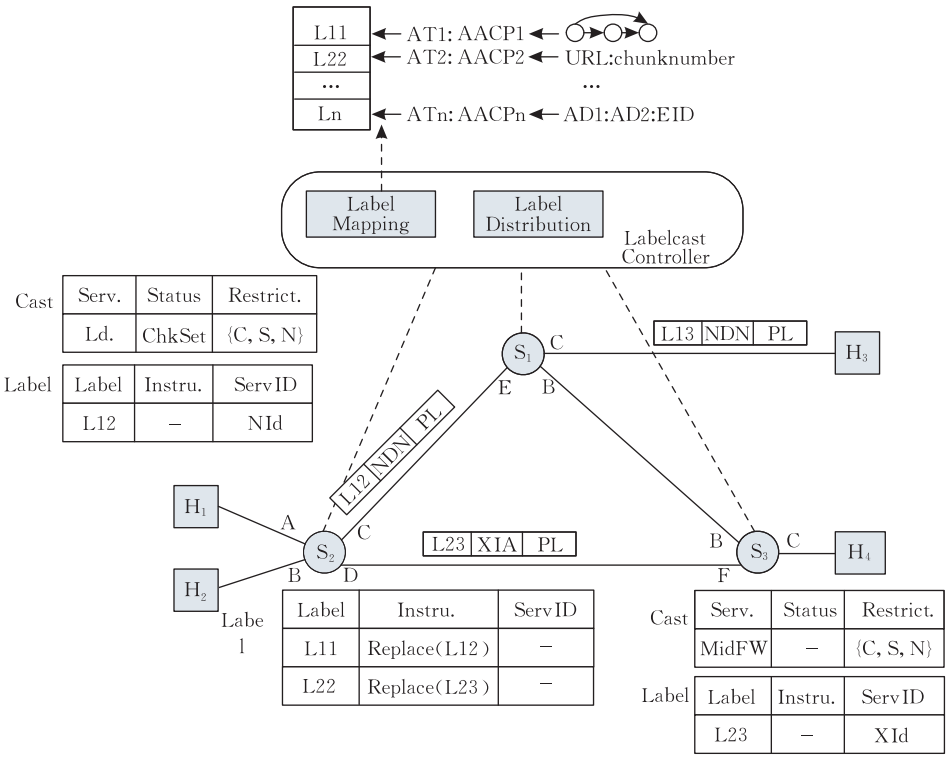


图 4　多协议统一承载

在转发平面标签与处理报文的服务关联，实现对报文协议相关的处理，在控制平面标签与处理报文的网络应用控制器关联，实现对报文转发的控制. 标签只有在报文转发和控制处理中才与特定网络体系结构的处理流程和控制方式映射，是一种弱语义的承载层，能够支持基于流的端到端的转发和以数据或服务为中心的非端到端的转发等，支持多协议统一承载，能够简化硬件实现，增强网络节点的可扩展性.

4　LabelCast 应用

LabelCast 支持基于规则的报文转发和基于名字的 NDN 报文转发等，能够支持多种网络类型，简化报文转发处理.

4.1　基于规则的报文转发

4.1.1　规则与标签映射

基于规则的报文转发控制器作为 LabelCast 控制器中一种网络控制应用来运行，LabelCast 控制器接收到 packet-in 事件后根据报文类型以及注册的网络应用，选择对应的网络应用控制程序. 规则转发控制器根据报文头选项以及管理策略设计报文转发规则，并将规则通告给 LabelCast 控制器. LabelCast 控制器为基于规则的报文转发应用分配标签 Label，并采用 Hash 方式计算规则匹配域的 ID，将 ID 作为与规则转发应用关联的标签的选项.

LabelCast 控制器将规则转发控制器产生的规则中匹配域用标签替换，规则中动作域保持不变，来配置转发平面 Label 表，如图 5(a)所示. LabelCast 控制器将与规则对应的标签通告给端系统，端系统在后续发送的报文中携带该标签. 由于规则(匹配域除输入端口号外)在转发路径上多跳节点中是一致的，处理报文的动作也是相同的，LabelCast 控制器对转发路径中所有节点下载相同的与标签关联的转发规则.

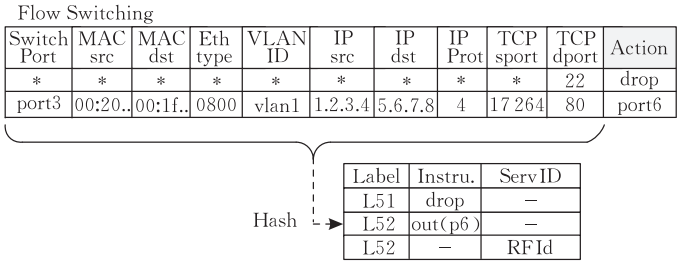
基于规则的报文转发处理过程不仅涉及输入控制和输出控制等简单操作，还可能包括 MPLS 标签替换、出栈和入栈，VLAN ID 处理和流量整形等复杂处理，在 LabelCast 中设计规则转发服务实现复杂的报文处理，并以该服务和服务状态等来配置 Cast 表.

4.1.2　与 Openflow 的比较

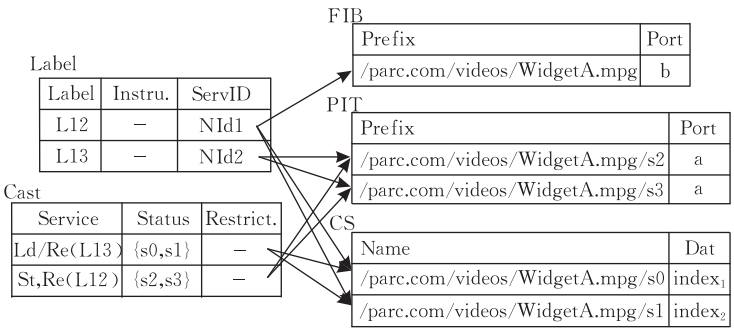
Openflow 规范 1.2 定义了以十五元组作为匹配域的转发规则，并采用 TLV〈Type，Length，Value〉表示，支持灵活扩展. 通常，在转发功能实现中，将

十五元组匹配域组织成树形结构.树形结构减小了规则存储的空间,具有很好的扩展性,但是增加了查表的次数,另外,需要在树的每一级进行带掩码的模糊匹配或精确匹配,因此 Openflow 转发实现较为复杂.

对于 LabelCast 中基于规则的报文转发,由 LabelCast 控制器中规则转发控制器实现转发控制,LabelCast 控制器将报文转发规则映射到与定长标签关联的 LabelCast 规则.报文携带的标签表明了报文按照规则转发处理,标签选项指出了处理该报文的规则,根据定长标签查找 Label 表获得的动作指令和服务表明处理报文的方式,因此 LabelCast 基于定长标签的转发更加简单.



(a) 转发规则与Label表的映射



(b) FIB、PIT和CS表与Label和Cast表的映射

图 5 LabelCast 规则配置

4.2 基于名字的报文转发

4.2.1 名字聚合与标签分配

在 NDN 网络中 Interest 报文根据 FIB 表转发,Data 报文根据 PIT 表转发[15].PIT 表是根据 Interest 报文以及其输入端口号产生,并采用最长前缀匹配查表方式,这是因为内容提供者仅仅发布内容的名字,不包括内容的数据块,而且提供者保证了内容中不同数据块在同一节点中存储.对于结构化的名字可以采用前缀方式聚合,将名字的前缀映射到固定长度的标签.采用基于前缀分配标签的方式使得端系统后续访问该内容中不同数据块的 Interest 报文能够分配同一个标签,并按照相同的路径和处理方式等转发策略在网络中转发,实现访问同一内容的 Interest 报文名字的聚合.如给访问不同视频片段 /parc.com/videos/WidgetA.mpg/s0 和/parc.com/videos/WidgetA.mpg/s1 的 Interest 报文分配相同的标签 L12,后续报文只需通过标签匹配进行转发.

4.2.2 Label 表和 Cast 表配置

LabelCast 控制器用分配的标签和 NDN 服务索引配置 Label 表,如图 5(b)所示.LabelCast 控制器用 Store 服务构造 Cast 表转发规则,实现 PIT 表功能以转发 Data 报文,还可以配置 Data 报文的处理策略,如缓存概率等.另外,用 Load 服务构造 Cast 表转发规则,实现 CS 表功能以转发 Interest 报文,也可以确定 Interest 报文是否需要访问网络节点中缓存的数据.若 Interest 报文不访问缓存的数据,则实现端-端的通信模式.

Interest 报文和 Data 报文的转发路径是由 LabelCast 控制器计算.为了简化计算,为 Egress Data 报文和 Ingress Interest 报文分配相同的标签,即 Data 报文按照 Interest 报文的路径返回.对于能够聚合的名字,当 Interest 报文到达时,更新该标签和端口的请求状态,即向端口的状态集合中增加新的数据块号(Chunk number).当 Data 报文到达时,修改标签和选择输出端口号,并从端口的状态集合中删除到达的 Data 报文对应的数据块号.用 Bloomfilter[16] 记录端口的状态集合,Bloomfilter 作为 NDN 报文转发规则中状态域 Status.

5　LabelCast 原型系统 NetMagic-Pro

5.1　原型系统设计

5.1.1　原型系统结构

基于通用多核处理器 FT1000 和网络加速引擎 NPE 设计实现了支持 LabelCast 转发的原型系统 NetMagic-Pro，如图 6 所示. 在 NPE 中设计了 Label 表，在转发平面调度报文转发操作，以及报文输出控制等指令. 同时利用通过 PCI-E 总线连接的 FT1000 提供的计算和存储资源，设计了 Cast 表，实现协议、状态相关的深度处理等服务.

LabelCast 以 Label 表和 Cast 表向控制平面提供了统一的管理和配置接口，以及计算、存储和网络资源状态等信息. 基于统一的网络转发平面接口，LabelCast 控制器向在其上运行的网络应用控制器(程序)提供了统一的视图，以及转发平面编程接口. LabelCast 控制器接收到查表未命中的报文后，经过报文解析将报文提交给新型网络体系结构对应的网络应用控制器(程序). 网络应用控制器(程序)生成报文处理规则，LabelCast 控制器为报文分配标签，并将规则中匹配域替换为标签，最后下载规则到转发平面.

标签对访问相同内容的报文聚合，LabelCast 控制器上网络应用控制器(程序)只需要根据网络协议类型对首报文进行处理和分配标签，后续报文在转发平面由标签指示的网络协议相关的服务进行处理. 原型系统 NetMagic-Pro 硬件实现简单，且支持全局服务的动态调度和扩展.

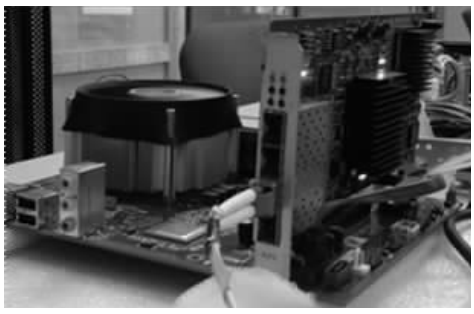


图 6　LabelCast 转发原型系统 NetMagic-Pro

5.1.2　服务开发模型

在数据平面，根据报文标签查找对应的服务 Cast 表，调度服务来处理报文，如 IPv4 与 IPv6 共存与过渡，基于名字的报文转发，基于规则的转发等. 以可扩展的资源容器为平台，用户能够扩展自定义的报文转发处理服务，在服务开发中可以调用 LabelCast 原型系统提供的应用组件库和系统库，来加速应用开发. 在原型系统 NetMagic-Pro 设计中，实现了扩展的 Linux 容器 XLC(eXtension Linux Container)对计算和存储等资源进行虚拟化和分配，简化用户自定义服务的开发.

在控制平面，用户基于 XLC 接口开发 LabelCast 网络控制器(程序). XLC 接口与网络处理器论坛 NPF 定义的 NPAS 以及 IETF 定义的 ForCES 等转发与控制层通信[17]具有相似的功能，向控制程序提供管理接口 API，监控数据转发平面统计计数器，还提供服务 API，维护数据平面转发表项，实现转发平面与控制平面之间的消息和状态传递.

5.1.3　开放的服务注册接口

XLC 还提供了统一开放的用户自定义服务注册接口，即服务原语 *registerFun*，支持用户开发的自定义网络服务的动态注册. 对于用户实现的自定义网络服务程序，需要通过该统一开放的服务注册接口向转发平面注册，就可以加载到 LabelCast 转发平面中运行. 该接口与操作系统提供的钩子函数具有相同的功能.

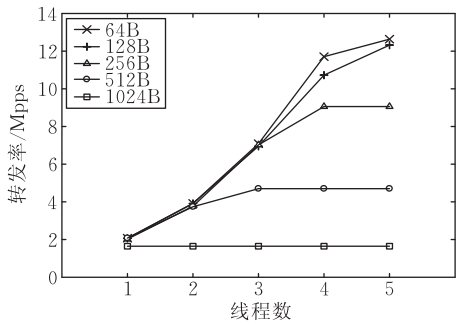
服务注册接口 *registerFun* 包含用户自定义服务的名称 *name*、用户自定义服务程序的句柄 *user_serv_proc*、用户自定义服务运行的中间状态 *Metadata* 以及 LabelCast 规则中资源约束 *Restrictions*. 资源约束主要包括 *thread_Priority*、*buffer_amount*、*virtual_queue*，分别对服务使用的计算、存储和带宽等资源进行限制. 用户可以通过该接口中消息传递函数 *npe_read* 和 *npe_write* 读取或者修改当前服务的状态，还能够增加、删除用户自定义服务中规则.

5.2　IO 性能测试与分析

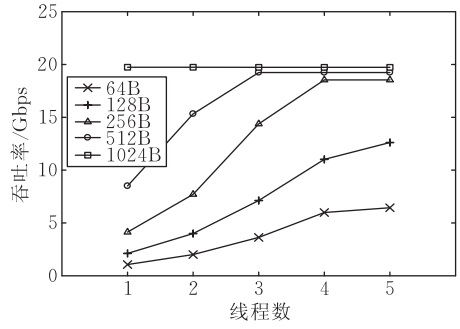
在 LabelCast 转发中转发平面提供了对报文转发的基本操作和深度处理(服务). 基本转发操作由转发平面网络加速引擎 NPE 实现，提供弱语义的通用的报文处理动作指令，包括查表、基本报文选项修改和输出控制等. 深度处理(服务)由在与 NPE 紧耦合的通用多核处理器 FT1000 之中运行的服务实现，提供与数据路径上网络协议语义或状态关联的深度报文处理的服务，如报文特定字段替换、报文缓存和基于矩阵的路由计算等功能. 因此，在原型系统 NetMagic-Pro 中 NPE 与 FT1000 间总线传输效率是系统性能提高的关键.

在原型系统 NetMagic-Pro 设计实现中，针对 NPE 与 FT1000 间传输机制进行优化，设计实现了

高效的 PacketDirect 机制，以提高系统吞吐率. 利用测试仪 2 个 10 Gbps 端口发包，对原型系统 NetMagic-Pro 在不同线程数及报文长度情况下 IO 性能进行测试，结果如图 7 所示. 随着转发线程的数目增加，处理能力的增强导致不同长度报文转发率都有一定提升；1024 Bytes 报文在 1 线程、512 Bytes 在 3 线程、256 Bytes 在 4 线程时达到最高转发率的原因在于此时报文流量已基本达到测试仪端口最大带宽，即 20 Gbps；64 Bytes 报文转发率相对较高，原因在于在内部报文缓冲区大小一定情况下，可缓存小报文数目相对较多. 对于系统吞吐率，同一线程数目下，报文长度越大，吞吐率越高，原因在于报文转发率相同情况下，系统流量和报文长度成正比；而当报文长度达到 1024 Bytes 时，单线程即可达到测试仪流量上限；其他长度报文随着线程数目（处理能力）增加，系统吞吐率也会增加. 测试结果表明，在 PacketDirect 机制支撑下，原型系统 NetMagic-Pro IO 系统满足端口吞吐率等性能要求.



(a) 转发率



(b) 吞吐率

图 7 NetMagic-Pro IO 性能测试

6 总结和下一步工作

LabelCast 是一种普适的转发平面抽象，用 Label 表和 Cast 表分别表示网络节点转发和服务的能力. 转发平面基于定长 Label 查找和调度服务，包括基于 OLV 的报文基本选项的匹配和修改等弱语义的通用动作指令，基于集成的计算和存储资源实现数据路径上任意字段查找和修改等网络协议语义或状态相关的特殊处理的服务，并且能够动态扩展新服务以支持新型网络体系结构特殊处理的需求. LabelCast 基于标签实现了多协议统一承载，以标签来聚合和转发不同端系统的请求，能够支持基于流的端到端的转发和以数据或服务为中心的非端到端的新型网络转发，为 SDN 提供了一种普适的转发平面抽象.

下一步基于通用多核处理器和网络加速引擎继续完善 LabelCast 转发的原型系统 NetMagic-Pro，以实现 NDN 转发. 同时，研究 LabelCast 利用缓冲区原语和线程原语等对网络中集成的计算和存储等资源进行抽象和隔离，为服务扩展提供开放的开发接口，进一步增强 LabelCast 可扩展性.

参 考 文 献

[1] McKeown N, Anderson T, Balakrishnan H et al. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74

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

[3] Ashok Anand Fahad Dogar Dongsu Han et al. XIA: An architecture for an evolvable and trustworthy Internet//Proceedings of the Hotnets 2011. Cambridge, USA, 2011: 7-12

[4] Ali Ghodsi, Teemu Koponen, Barath Raghavan, Scott Shenker, Ankit Singla, James Wilcox. Information-centric networking: Seeing the forest for the trees//Proceedings of the Hotnets 2011. Cambridge, USA, 2011: 1-6

[5] Lin Chuang, Lei Lei. Research on next generation Internet architecture. Chinese Journal of Computers, 2007, 30(5): 694-711(in Chinese)
(林闯，雷蕾. 下一代互联网体系结构研究. 计算机学报，2007, 30(3): 694-711)

[6] Ali Ghodsi, Teemu Koponen, Barath Raghavan, Scott Shenker, Ankit Singla, James Wilcox. Intelligent design enables architectural evolution//Proceedings of the Hotnets 2011. Cambridge, USA, 2011: 13-18

[7] Yang Dong, Li Shi-Yong, Wang Bo, Zhang Hong-Ke. New transport layer architecture for pervasive service. Chinese Journal of Computers, 2009, 32(3): 359-370(in Chinese)
(杨冬，李世勇，王博，张宏科. 支持普适服务的新一代网络传输层架构. 计算机学报，2009, 32(3): 359-370)

[8] Halpern J, Hadi Salim J. Forwarding and control element separation (ForCES). IETF, RFC5812, 2010

[9] Wang Bao-Sheng, Xia Yi, Chen Xiao-Mei, Zhao Feng. Research and implementation of ForCES-based IPv6 router.

Journal of National University of Defense Technology, 2006, 28(3): 44-48(in Chinese)
(王宝生，夏毅，陈晓梅，赵峰. 基于 ForCES 体系结构的 IPv6 路由器的研究与实现. 国防科技大学学报，2006，28(3)：44-48)

[10] OpenRouter: OpenFlow extension and implementation based on a commercial router//Proceedings of the ICNP2011. Vancouver, Canada, 2011: 141-142

[11] James Kelly, Wladimir Araujo, Kallol Banerjee. Rapid service creation using the JUNOS SDK. ACM SIGCOMM Computer Communication Review, 2010, 40(1): 56-60

[12] David G Andersen, Hari Balakrishnan, Nick Feamster. Accountable Internet Protocol. ACM SIGCOMM Computer Communication Review, 2008, 38(4): 339-350

[13] Lu Guo-Han, Miao Rui, Xiong Yong-Qiang, Guo Chuan-Xiong. Using CPU as a traffic Co-processing unit in commodity switches//Proceedings of the HotSDN 2012. Helsinki, Finland, 2012: 31-36

[14] Rosen E, Viswana A, Callon R. Multiprotocol label switching architecture. Internet Engineering Task Force(IETF). RFC 3031, 200l

[15] Ghodsi A, Koponen T, Rajahalme J, Sarolahti P, Shenker S. Naming in content-oriented architectures//Proceedings of the SIGCOMM ICN 2011. Toronto, Canada, 2011: 1-6

[16] Hao Fang, Kodialam Murali, Song Hao-Yu. Fast dynamic multiple-set membership testing using combinatorial bloom filters. IEEE/ACM Transactions on Networking, 2012, 20(1): 295-304

[17] Xu Ke, Wu Kun, Wang Qing-Qing. High performance control-plane communication model for scalable routers. Journal of Software, 2007, 18(9): 2205-2215(in Chinese)
(徐恪，吴鲲，王青青. 可扩展路由器控制平面的高性能通信模型. 软件学报，2007，18(9)：2205-2215)

LV Gao-Feng, born in 1980, Ph.D., assistant researcher. His research interests include next generation internet architecture, high performance routing and switching.

SUN Zhi-Gang, born in 1973, Ph. D., professor. His research interests include network architecture and high performance network switching.

LI Tao, born in 1983, Ph. D., assistant professor. His research interests include computer network and network processor.

MAO Jian-Biao, born in 1988, M. S. candidate. His research interests focus on new network architecture.

Yang An, born in 1988, M. S. candidate. His research interests focus on computer network.

Background

The testing and deployment of new network protocols proposed more requirements over SDN. Researching on next generation network architecture becomes the key fields of network innovations, such as Nebula, XIA and NDN. XIA proposes addresses of directed graph, NDN proposes everything over name, and Nebula proposes data center of network. The goal of next generation network is retaining the sandglass architecture and extending the end-end model, and develops new network controllers based on the present forwarding layer. The testing and deployment of new network protocols need to enlarge the capability of forwarding layer and boost up the functions of controlling layer. The present abstraction layer of network could not support the developing and testing of new network architecture.

The evolution of next generation network is not patching, while proposes the base mechanism of naming and address to solve the limitation of network scale, which is confronted with the challenge of the designing and testing of protocols in production networks. With the supporting of 863 projects (No. 2009AA01A334, No. 2011AA01A101), Our research group proposes a general abstraction layer, LabelCast. LabelCast uses Label table and Cast table denoting the capability of network services and forwarding and loads multiple protocols on fixed-length unified labels. The logical central controller maps network protocol to fixed-length labels, to support the developing and testing of new network protocols in SDN.

Packet processing in forwarding plane of LabelCast includes action instructions of modification of packet base options and output controlling based on OLV〈Offset, Length, Value〉 and services of processing of any fields of packets running on the computing and storing resources within network elements, which implements general process of light-semantics and advance processing of network protocol semantics or status-related. Forwarding plane lookups based on fixed-length light-semantics labels to aggregate multiple requests from different hosts for the same contents or services and schedule the instruction of actions and the sequence of service atoms, which decreases the complexity of the implementation. Services in the forwarding plane process packets in depth in the key data path, such as the modification of special field and data storing, and could be extended based on the computing and storing resource to support the process of the new network protocols.