西安交通大学

开题报告

题　目　支持可编程数据平面的网络模拟器设计与开发

电信 学院 计算机科学与技术 系（专业）计算机 45 班

学生姓名　况鹏

学　　号　2140505105

指导教师　毕军、赵鹏

设计所在单位　清华大学网络科学与网络空间研究院

2018 年 3 月

目录

1 课题任务

本课题针对目前网络模拟器开发难度大，开发代码无法迁移到真实网络设备、不支持可编程网络技术等问题，提出将可编程数据平面技术集成到网络模拟器中，设计了支持可编程数据平面的网络模拟器，达到了简化开发难度、便于代码移植、支持可编程数据平面的目的，主要包括以下课题任务：

（1）研究可编程数据平面转发模型以及领域特定语言编译架构，熟悉 bmv2 工作原理以及 P4 编程方法。

（2）研究传统网络模拟器系统结构、运行机制和编程方法，总结现有网络模拟器存在的问题。

（3）研究基于软件实现的可编程数据平面行为模型系统架构、运行机制与实现方法，探索将可编程数据平面技术集成到网络模拟器中方法，总结设计挑战，主要包括基于网络模拟器构建真实 P4 设备模型、存在 P4 运行工具不便于应用在网络模拟器中、大规模 P4 设备的流表下发问题等。

（4）设计将可编程数据平面技术与现有模拟器集成的方法，设计 P4 流水线中缓冲器与队列的调度安排策略， 设计大规模网络拓扑（fat-tree）的构建方法。

（5）基于（4）中设计，实现支持可编程数据平面的网络模拟器原型。

（6)使用网络模拟器原型，构建大规模网络拓扑(fat-tree)结构、部署 Silkroad 网络结构环境、构建 NS3 与 NS4 对比实验，构建 NS4 与 mininet 对比实验，验证网络模拟器原型设计。

2 对课题的理解

2.1 课题背景及目的

网络模拟广泛应用于网络研究、教育、工业的各个方面，网络模拟器不仅可以抽象出真实世界网络拓扑模型，而且可以模拟出实际网络运行效果。网络模拟器通常应用于两个方面，一个是验证正在开发的网络协议，一个是在大规模生产之前测试网络设备设计正确性。在传统网络模拟器（ns3）中，网络功能模型开发紧密耦合于模拟器内部特征，导致开发代码不能直接迁移到真实网络设备上，因此产生了能支持可编程数据平面的网络模拟器需求。领域特定语言 P4，用于定义可编程数据平面行为，具有协议无关性、目标独立性、可重构性等三个特点。如果能够将 P4 嵌入到网络模拟器（ns3）中，使用 P4 定义网络设备行为，使用 ns3 定义网络拓扑结构，那么即可解决传统网络模拟器存在的问题，同时也为 P4 提供了一个有用的研究开发环境，无论是对网络模拟的研究还是对 P4 功能的探索都有着重要意义。

本课题希望通过对传统网络模拟器（ns3）以及 P4 软件交换机（bmv2）的系统结构、内部机制以及编程方法的研究，寻求一种将 P4 软件交换机（bmv2）的核心功能嵌入到网络模拟器(ns3)中的方法，设计并实现支持可编程数据平面技术的网络模拟器的原型，该模型能够充分地利用 P4 的相关特性，以此解决传统网络模拟器存在的问题。

本课题旨在培养学生独立思考、科研创新、解决问题的能力，激发学生对科研工作兴趣，提高学生编程实践能力。

2.2 研究现状

传统的网络模拟器有 ns、OPNet、REAL、PFPSim 等，ns 是一系列的开源的基

于离散事件的网络模拟器，主要用于网络研究、教育等方面，ns 目前包括 ns1、ns2、ns3，ns3 由 C++、Python 编写，内部已集成一些常见的网络功能，可方便地进行网络拓扑定义以及网络场景的模拟与验证。OPNet 是面向对象的通用目的网络模拟器，它使用动态进程分配技术构建虚拟电路传输模型，以便进行离散事件的模拟，它基于 Proto-C 语言，可实现几乎所有网络功能及协议。REAL 是一个用于研究流动态行为以及拥塞控制的网络模拟器，它使用 NetLanguage 描述网络拓扑、协议、数据和控制参数，目前已提供 30 多个模块来进行流控制协议的模拟。PFPSim 是一个使用 C++以及 P4 编写的主机编译型的网络模拟器，它使用 C++进行网络结构中模块定义，使用 P4 进行包处理行为定义，用于网络功能的模拟及调试，可供硬件厂商模拟验证及优化设计。

目前能够编译运行 P4 的工具有 bmv2、P4 Runtime、SDE compiler 等，bmv2 是一个基于 C++编写的能够支持 P4 的软件交换机，主要用于 P4 验证以及网络模拟。P4 Runtime 是一个基于 C++编写的动态运行库，为 P4 提供了一个控制平面的框架和工具。SDE compiler 是 Barefoot 推出的能够将 P4 程序编译进 Tofino 芯片的编译器。

2.3 要解决的问题

支持可编程数据平面的网络模拟器开发需要将 bmv2 的核心功能嵌入到 ns3 中，主要需要解决以下问题：

1) bmv2、ns3 的系统结构、运行机制、编程方法的深入研究与理解。
2) 基于 ns3 模拟器构建真实 P4 设备的行为模型。
3) P4 流水线中缓冲器与队列机制的设计。
4) 将 P4 运行时交互操作（P4Runtime）转换成离散事件型操作。
5) 大规模网络拓扑中流表项的自动下发。

3 阅读文献综述

1) P4: programming protocol-independent packet processors

这篇文章提出可编程的协议无关的包处理高层次语言 P4，用于充当控制器

与交换机之间的通用接口，从可重配置性、协议无关性、目标独立性等目标出发进行 P4 语言的设计，描述了抽象转发模型、P4 语言规范定义、P4 编译器、P4 简单用例等，解决了 OpenFlow 表达性、灵活性不足（无法自定义数据包头部字段、无法自定义协议解析操作等）的问题。

2) **NS4: A P4-driven Network Simulator**

这篇文章提出 P4 驱动的网络模拟器 NS4，从数据平面和控制平面两个部分详细描述了 NS4 整体架构设计，致力于解决传统网络模拟器网络功能开发难度大、开发代码不能迁移到真实设备、不支持可编程数据平面等问题。

3) **PFPSim: A Programmable Forwarding Plane Simulator**

这篇文章提出了一个基于可编程转发平面结构的、用于数据包处理应用程序分析和验证的网络模拟器 PFPSim。PFPSim 使用转发结构描述语言（FAD）定义转发设备体系结构，使用 C++或 P4 定义应用程序代码（数据包处理逻辑），通过平台产生器实体，自动将 FAD 结构定义转换成等价 SystemC 代码，通过 P4 软件交换机编译器，将 P4 描述翻译成 C 代码，进而编译成静态库和头文件，以导入转发设备的 SystemC 模型中。

这篇文章关注的是转发设备体系结构的建模，并使用 PFPSim 模拟了 NPU 以及 RMT 结构模型，但没有对所支持的 P4 语言特征作出说明，没有对更大范围的网络拓扑结构作出阐述，也没有提及模拟器中控制器的具体设计。

4) **Network Simulations with the ns-3 Simulator**

这篇文章介绍了基于离散事件网络模拟器 ns3，从软件核心、软件集成、虚拟化支持、试验台集成、属性系统、追踪架构等方面描述了 ns3 与 ns2 的区别，重点介绍了 ns3 支持的新型特征。

5) **Network Simulation and its Limitations**

这篇文章围绕 ns3 及其核心功能介绍了网络模拟过程及其限制因素，首先介绍了网络模拟使用场景，然后详细说明了 ns3 的设计、结构和工作流，最后总结了网络模拟存在的限制性。

6) **Silkroad: Making stateful layer-4 load balancing fast and cheap using switching asics**

这篇文章提出了一个 L4 层的由 400 行 P4 程序定义的可实现在 ASIC 交换机中的负载平衡器 Silkroad，以代替数以百计的软件交换机，来解决大型数据中心中负载平衡代价过大的问题。Silkroad 充分利用 ASIC 交换机高吞吐、

低延迟的特点，使用 SRAM 存储五元组哈希值而不是五元组信息来减少内存消耗，通过位于芯片的布隆过滤器记录未决定的网络连接，可在保持自连接一致性的同时，以线性速率处理千万级别的网络连接。

7) **HULA: Scalable Load Balancing Using Programmable Data Planes**

这篇文章提出了支持可编程数据平面的负载平衡器 HULA，以解决多源网络拓扑中网络带宽利用率不高的问题。它是一个数据平面层次的、使用 P4 定义的、运行在可编程交换机上的负载平衡器。它采用链路利用率反映网络拥塞状况，采用了探针请求、flowlet 交换等核心技术来进行负载平衡，核心思想是使每一个 HULA 交换机只记录到达目的地最好下一跳的链路利用率信息，充分地利用了分布式网络路由和拥塞意识负载平衡特性，以实现可扩展性、主动性、自适应性、可编程性目标。

8) **DC.p4: Programming the Forwarding Plane of a Data-Center Switch**

这篇文章提出了一个使用 P4 来表达数据中心交换机转发平面行为的案例学习。它首先回顾了 P4 语言定义、抽象转发模型，概述了 P4 需要增加的语言特性以支持数据中心交换机特定功能，然后描述了编译执行 P4 程序的开发环境，最后基于描述数据中心交换机行为的程序 DC.p4 开发经验，探索了 P4 未来的发展方向。

9) **Compiling packet programs to reconfigurable switches**

这篇文章探索的是面向可重配置交换机结构的编译器设计，以解决将数据包程序映射到目标交换机的问题。它首先描述了可重配置交换机芯片、数据包处理语言，然后从有向无循环图 DAG、内存类型、分配开销等方面定义交换机硬件抽象模型，接着从交换机中每阶段内存资源、延迟这两个角度详细分析了 RMT、FlexPipe 结构的硬件限制，最后从流水线阶段数、流水线延迟、电力消耗量等角度出发，使用整数线性规划 ILP 以及贪心算法来优化编译器设计。

10) **P4FPGA : A Rapid Prototyping Framework for P4**

这篇文章提出一个开源的，将 P4 映射成 FPGA 的，用于开发、评估数据平面应用程序的工具 P4FPGA。它实质上包含一个编译器和一个运行库，编译器用于将 P4 程序编译成 Bluespec FPGA 代码，允许用户包含用任何语言编写的任意硬件模块，运行库提供了一个设备无关的硬件抽象，允许 FPGA 支持能合成到 Xilinx 或 Altera FPGAs 的设计。

4 实施方案

4.1 前期工作

在前期的工作中，通过对相关研究以及工具的调研，我们做出了以下三方面的工作：

1) 阅读了介绍网络模拟器的相关文献，总结了现有网络模拟器存在的问题，重点研究了 ns3 的系统结构、运行机制与编程方法。
2) 阅读了介绍可编程数据平面技术的相关文献，总结了可编程数据平面技术的特征、应用场景，重点学习了领域特定语言 P4，总结了 P4 特性及优势，学习了 P4 编程方法。
3) 调研并学习了能够编译运行 P4 的软件工具，重点研究了 bmv2、P4 Runtime 的使用方法以及内部运行机制。

4.2 课题基本要求

根据本课题的具体任务，我们总结出来的课题基本要求包括以下四个方面：

1) 文献调研要求，总结现有网络模拟器存在的问题以及 P4 应用范围、优势特征。

2) 系统设计方面的要求，包括可编程数据平面技术与网络模拟器的集成方案的完整性设计、P4 流水线中缓冲器与队列的调度优化设计、大规模网络拓扑（fat-tree）设计等。

3) 系统实现方面的要求，能够完整地实现基于可编程数据平面的网络模拟器原型，可支持各个 P4 程序，可方便研究者学习与使用，可方便开发者灵活开发拓展相应功能。

4) 实验验证方面的要求，需在虚拟机、服务器等不同实验环境下、从系统资源使用情况、模拟运行时间、网络吞吐量、网络带宽等多种实验指标、采用 Silkroad 部署、大规模网络拓扑（fat-tree）构建等评估方法来测试网络模拟器原型的实现效果。

4.3 研究构想与思路

基于前期工作，我们计划重构 bmv2 里面 simple switch 的核心代码，将其整合进 ns3 中，开发出一个支持可编程数据平面技术的网络模拟器，具体的思路构想如下：

1) 参考 bmv2 里面 simple switch 的核心代码，实现 P4 Model。
2) 基于实现的 P4 Model，参考 ns3 的编程样例，构建真实 P4 设备的模型。
3) 根据 P4 流水线的行为模型，设计并实现缓冲器以及队列的调度机制，将该机制编程实现并嵌入进 P4 Model 中。
4) 基于 P4 设备模型，编写 P4 Example 测试程序，检验所编写的 P4 设备模型。
5) 基于 P4 设备模型，构建大规模网络拓扑，进行网络行为以及性能的测试。

4.4 需要的工具与资料

基于前期调研与相关工作准备，本课题需要的工具为：

1) 支持 P4 软件交换机 bmv2，能够进行 P4 程序编译及运行。
2) 网络模拟器 ns3，能够进行网络拓扑定义以及网络场景的模拟。
3) 网络仿真器 mininet，能够进行网络拓扑定义、网络行为构建等网络仿真操作。
4) P4 运行交互环境 P4 Runtime，为 P4 提供了控制平面的框架和工具，可进行流表的交互式下发。
5) Linux 操作系统 ubuntu，提供编程调试环境。

需要的资料为：

1) P4 语言规范定义 The P4 Language Specification，详细介绍了 P4 语法规则以及特征特性。
2) Ns3 API 介绍，详细介绍了 ns3 对外提供的编程接口。
   https://www.nsnam.org/documentation/
3) Bmv2 开源项目，介绍了 bmv2 的具体使用，提供了开放的源代码。
   https://github.com/p4lang/behavioral-model

4.5 计划安排

时间	阶段	工作内容
第一、二周	调研阶段	阅读相关文献，学习相关工具，撰写开题报告
第三、四周	设计阶段	深入研究 ns3 以及 bmv2 运行机制，阅读大量源代码，设计 P4 Model 以及 P4 设备模型，设计缓冲器、队列机制等
第五~八周	编程调试阶段	编写并调试 P4 Model、P4 设备模型、P4 Example 测试程序、SilkRoad 网络功能实现、fat-tree 大规模网络拓扑构建、流表自动下实现、Mininet 下网络拓扑构建等
第九周	实验阶段	完成 Silkroad 网络场景测试、NS3 与 NS4 对比、NS4 与 Mininet 对比实验
第十~十六周	论文写作阶段	撰写毕业设计论文、参加毕设答辩

5 参考文献

1) Bosshart P, Daly D, Gibb G, et al. P4: Programming protocol-independent packet processors[J]. ACM SIGCOMM Computer Communication Review, 2014, 44(3): 87-95.
2) Fan C, Bi J, Zhou Y, et al. NS4: A P4-driven Network Simulator[C]// the SIGCOMM Posters and Demos. 2017:105-107.
3) Abdi S, Aftab U, Bailey G, et al. Pfpsim: a programmable forwarding plane simulator[C]//Proceedings of the 2016 Symposium on Architectures for Networking and Communications Systems. ACM, 2016: 55-60.
4) Henderson T R, Lacage M, Riley G F, et al. Network simulations with the ns-3 simulator[J]. SIGCOMM demonstration, 2008, 14(14): 527.
5) Rampfl S. Network simulation and its limitations[C]//Proceeding zum Seminar Future

Internet (FI), Innovative Internet Technologien und Mobilkommunikation (IITM) und Autonomous Communication Networks (ACN). 2013, 57.

6) Rui Miao, Hongyi Zeng, Changhoon Kim, Jeongkeun Lee, and MinlanYu. Silkroad: Making stateful layer-4 load balancing fast and cheap using switching asics. In Proceedings of the 2017 ACM SIGCOMM Conference, SIGCOMM '17, pages 525–538, Los Angeles, CA, USA, 2017. ACM.
7) Katta N, Hira M, Kim C, et al. Hula: Scalable load balancing using programmable data planes[C]//Proceedings of the Symposium on SDN Research. ACM, 2016: 10.
8) Sivaraman A, Kim C, Krishnamoorthy R, et al. Dc. p4: Programming the forwarding plane of a data-center switch[C]//Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research. ACM, 2015: 2.
9) Jose L, Yan L, Varghese G, et al. Compiling packet programs to reconfigurable switches[C]// Usenix Conference on Networked Systems Design and Implementation. USENIX Association, 2015:103-115.
10) Han Wang, Robert Soul´e, Huynh Tu Dang, Ki Suh Lee, Vishal Shrivastav, Nate Foster, and Hakim Weatherspoon. 2017. P4FPGA: A Rapid Prototyping Framework for P4. In Proceedings of ACM Symposium on SDN Research conference, Santa Clara, California USA, April 2017 (SOSR 2017), 14 pages.DOI: http://dx.doi.org/10.1145/3050220.3050234