



数据平面的可编程时代

P语言基础与实战

提纲



P4项目背景

P4语言基础

P4实战演练

P4项目发展历

Nick McKeown、Jennifer Rexford、Amin Vahdat等教授在ACM SIGCOMM上联合发表论文：《P4: Programming Protocol-Independent Packet Processors》。以此拉开了高级数据平面编程语言P4高速发展的序幕。

2014年7月

2015年3月

P4项目社区正式发布目前广泛支持的P4语言标准《The P4 Language Specification Version 1.0.2》。

SIGCOMM在伦敦举行了年度会议，《P4: Programming Protocol-Independent Packet Processors》当选为年度最佳论文。第一次P4研讨会顺势召开，众多国际顶尖网络学者、专家参与讨论了P4在L4负载平衡，网络监控和分析，动态路由、以及故障排除方面的发展。

2015年8月

2016年6月

Nick教授等人创办的Barefoot公司C轮获得\$5700万融资

《The P4 Language Specification Version 1.1》发布

2016年7月

2016年10月

超过50个成员组织加入P4项目社区，探讨P4语言标准制定和发展方向。

网络交换芯片技术

▶ ASIC

专用集成电路，面向特定需求。体积小，性能强，成本低，功能固化，不支持数据平面编程。代表：SDN白盒交换机。

▶ CPU

通用集成电路，功能灵活，性能受CPU限制，效率低，支持数据平面编程。代表：OpenvSwitch。

▶ NP

网络处理器，功能灵活，性能较高，成本高，功耗高，支持数据平面编程。代表：华为S12700系列。

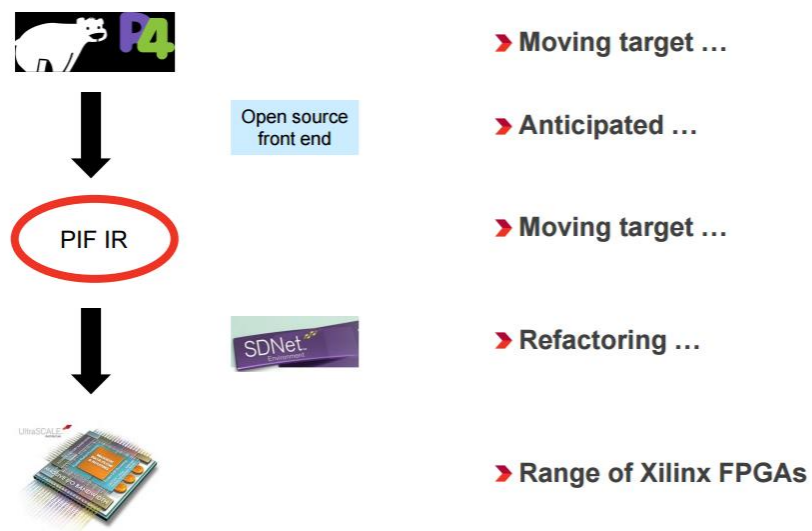
▶ FPGA

现场可编程门阵列功能灵活，数据转发性能低，支持数据平面编程。代表：NetFPGA。

▶ PISA

协议无关交换机架构，功能灵活，性能强，支持数据平面编程。代表：Tofino。

国外P4产品



Xilinx、Cornel推出支持P4的FPGA



Netronome推出的Agilio网卡及P4C—SDK

INT : P4 Code Snippet

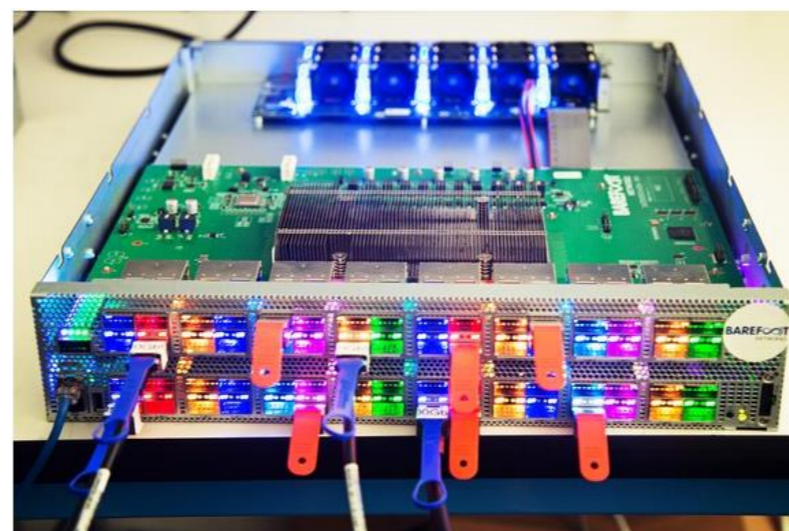
Exact-match
Table Definition

```
table int_inst {  
  reads {  
    int_header.instruction_mask : exact;  
  }  
  actions {  
    int_set_header_i0;  
    int_set_header_i1;  
    int_set_header_i2;  
    int_set_header_i3;  
    .....  
  }  
}
```

Action
Definitions

```
action int_set_header_i0() {  
}  
action int_set_header_i1() {  
  int_set_header_3();  
}  
action int_set_header_i2() {  
  int_set_header_2();  
}  
action int_set_header_i3() {  
  int_set_header_3();  
  int_set_header_2();  
}  
.....
```

Princeton、VMWare等合作推出支持P4的vSwitch



Barefoot推出6.4T可编程Tofino芯片及配套开发工具

P4项目社区成员





提纲

P4语言基础

P4是什么？

▶ P4是开源编程语言项目

P4语言是一个数据平面的编程语言，想让大家像用C语言对CPU编程一样对目标设备编程。

▶ 目标设备

一切可编程的网络设备，目前包括可编程网络芯片，网卡，FPGA，NPU，vSwitch 。

▶ 目标

P4的目标是作为一个统一的高度抽象的高级语言，聚焦于描述数据包解析处理过程，将底层的操作交由目标设备的编译器完成。

基础数据类型

数据类型	描述
bool	布尔型；0代表false，1代表true，可进行与、或、非等逻辑运算。
bit<W>	任意宽度W的无符号整型；位串是以比特位形式表示的任意长度的数（如：bit<127>，表示长度为127比特的位串），但如果需要对位串进行某些数学运算时，位串长度必须是8的整数倍（如：16、32、64bit）
int(W)	任意宽度W有符号整型；
varbit<W>	变长位串（bit-strings）不支持算术、比较、按位运算，甚至不支持类型转换。该数据类型在定义时会指定一个静态的最大宽度值。
int	无限精度整数常量型；

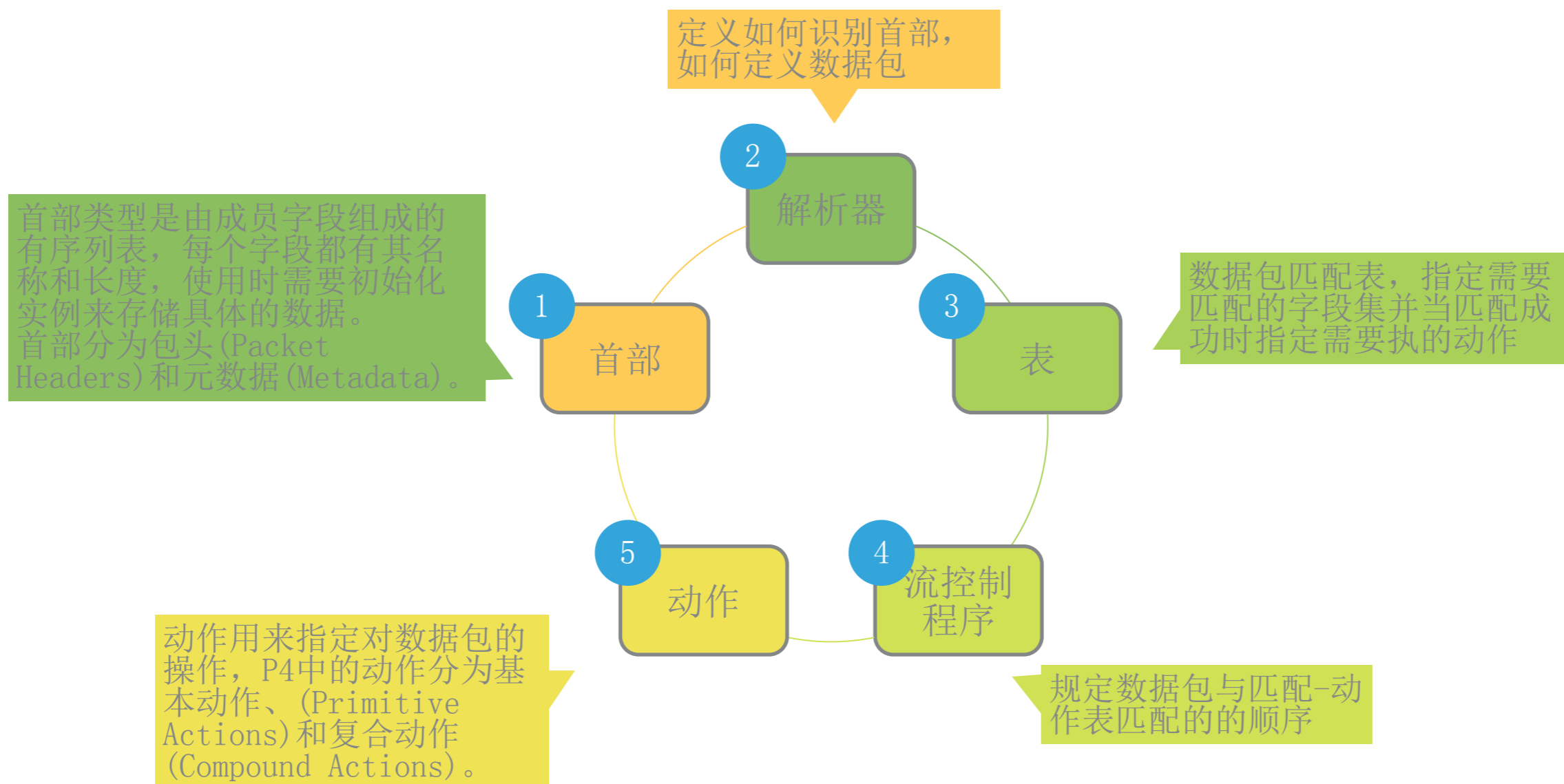
```

parser start {
  extract(ethernet);
  return select(latest.etherType) {
    0x8100, 0x9100 : parse_vlan_tag;
    0x0800         : parse_ipv4;
    0x86DD         : parse_ipv6;
    default       : ingress;
  }
}
    
```

常用方法及标识符

名称	描述
extract()	将包头(packet header)实例作为传参，从当前偏移量开始，将数据包中的数据拷贝到包头实例中并将偏移量指向该头部的末尾。Extract方法中使用next标识符指向头部堆栈，以表示下一个可解析位置。
select()	将字段列表作为传参，并以逗号分割；按顺序将字段值与程序中的设定值比较，找出匹配条目。
latest	包头实例指针，指向解析器中最近提取的包头实例，必须在解析器中必须在extract()方法调用后使用。
current()	字段指针，指向当前还未被解析的字的最高比特位，即即将被解析的第一个比特位、当前解析器解析到的比特位。可以传参，传参一代表距离当前解析比特位的正偏移量，传参二代表字段长度。

P4语言组件



P4项目源码目录结构

p4c-bm

- 后端编译器，可将高级语言或高级语言中间表示转为JSON格式或PD格式的配置文件

ptf

- Python测试框架，基于unittest框架实现，该框架中的大部分代码从floodlight项目中的OFTTest框架移植而来

p4-hlir

- 前端编译器，将高级抽象语言转化成高级语言中间表示

P4ofagent

- OpenFlow协议插件，目前实现的功能有限

p4c-behavior

- 第一代bm的编译器，现已基本废弃

ntf

- 网络测试框架，集成了mininet和docker，包含较多bmv2应用测试脚本

behavioural-model

- 模拟P4数据平面的用户态软件交换机bmv2，识别JSON格式配置文件

switch

- Switch示例，基本完成交换机的绝大部分功能

p4factory

- 快速开始，内含6个可快速启动的项目
basic_routing、copy_to_cpu、l2_switch、sai_p4、simple_router、switch

P4-build

- 需要手动生成的基础设施库，为执行P4程序编译、安装PD库

scapy-vxlan

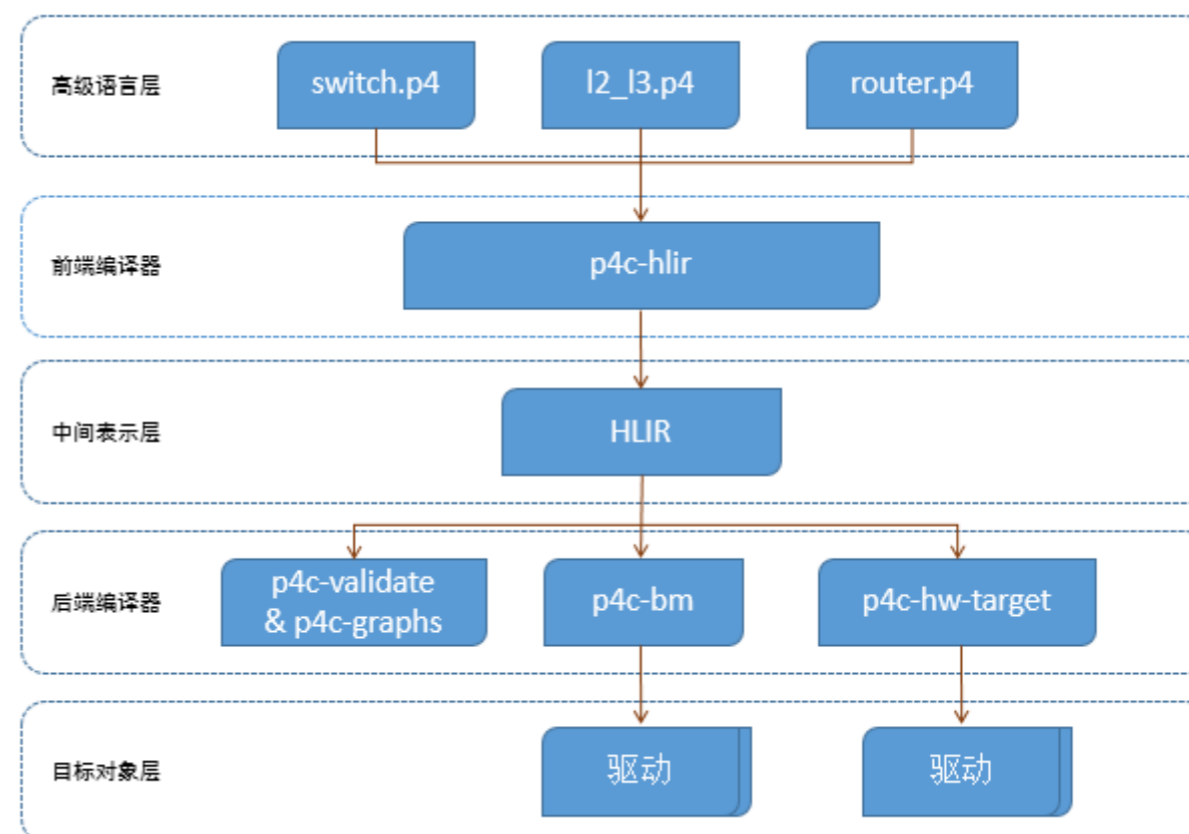
- 扩展VXLAN和ERSPAN-like协议包头处理

tutorials

- 示例教程，内含8个基础示例教程：copy_to_cpu、meter、TLV_parsing、register、counter、action_profile、resubmit、simple_nat

P4项目架构

- 高级语言层：高度抽象的P4语言编写的程序
- 前端编译器：对高级语言进行与目标无关的语义分析并生成中间表示
- 中间表示层：高级语言中间表示，可转换成多种其他语言
- 后端编译器：将中间表示转换为目标平台机器码
- 目标对象层：受控制硬件/软件设备



P4语言特性

▶ 协议无关

网络设备不与任何特定的网络协议绑定，用户可以使用P4语言描述任何网络数据平面协议和数据包处理行为。这一特性通过自定义包解析器、匹配-动作表的匹配流程和流控制程序实现。

▶ 目标独立

用户不需要关心底层硬件的细节就可实现对数据包的处理方式的编程描述。这一特性通过P4前后端编译器实现，前端编译器将P4高级语言程序转换成中间表示IR，后端编译器将IR编译成设备配置，自动配置目标设备

▶ 现场配置

分允许用户随时改变包解析和处理的程序，并在编译后配置交换机，真正实现现场可重配能力。

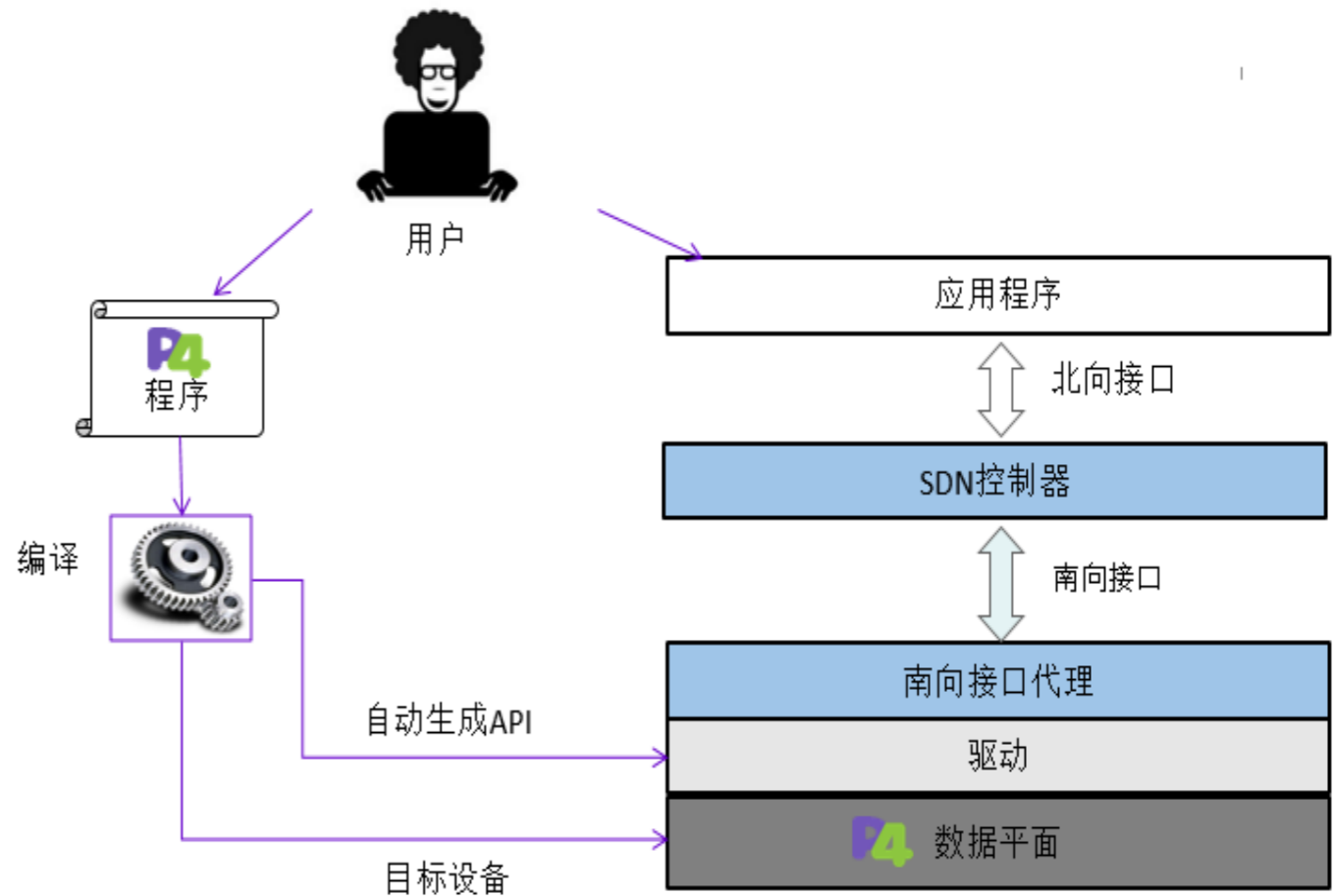
P4与OPENFLOW差别

定义差别

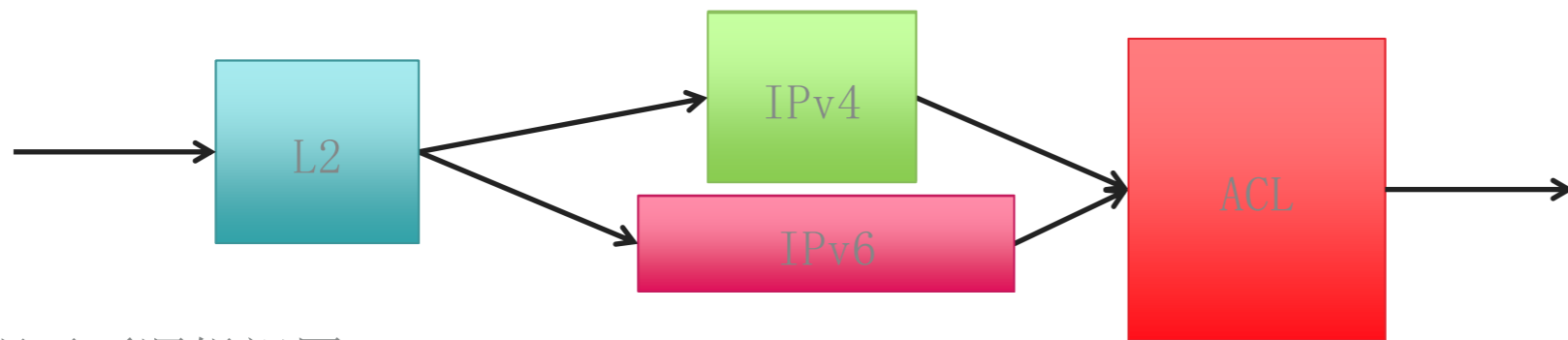
- OpenFlow是南向协议
- OpenFlow用以控制器与网络设备通信，描述网络中的通信过程
- P4是高级编程语言
- P4用以描述数据包解析处理过程

内容差别

- P4项目中有个openflow.p4程序，该程序对PISA芯片进行编程以支持OpenFlow。对于P4语言而言，OpenFlow是一个程序。OpenFlow和P4通过这种方式同时在网中络工作。

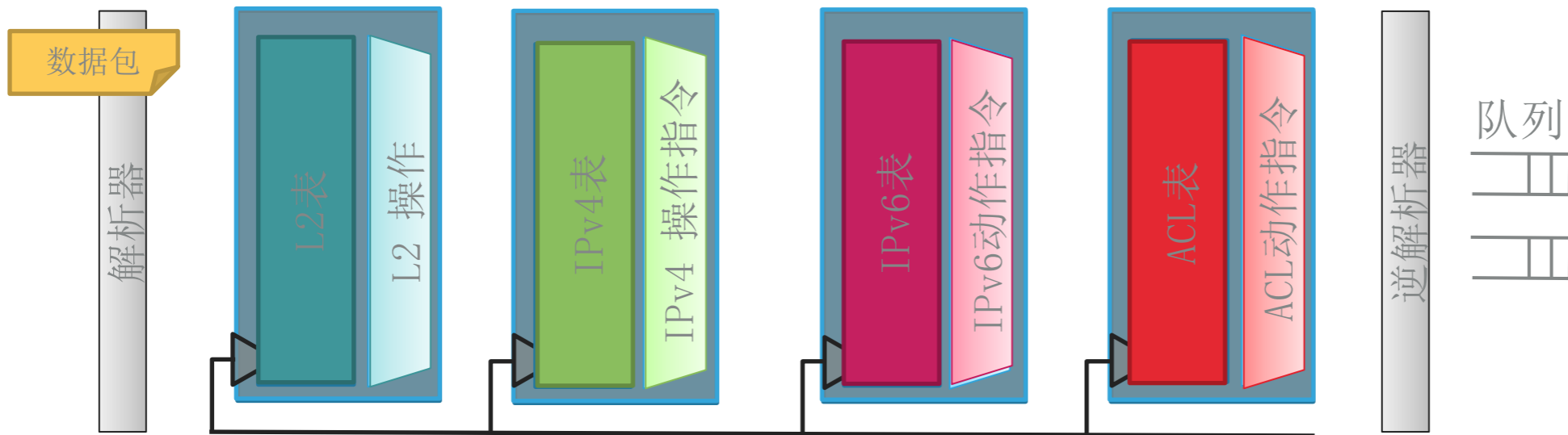


P4程序工作流程



数据平面逻辑视图

交换机流水线





提纲

P4实战演练

快速开始 (P4FACTORY)

▶ 系统

推荐系统: Ubuntu14.04+

▶ 安装

```
$. /install_deps.sh
```

```
$. /autogen.sh
```

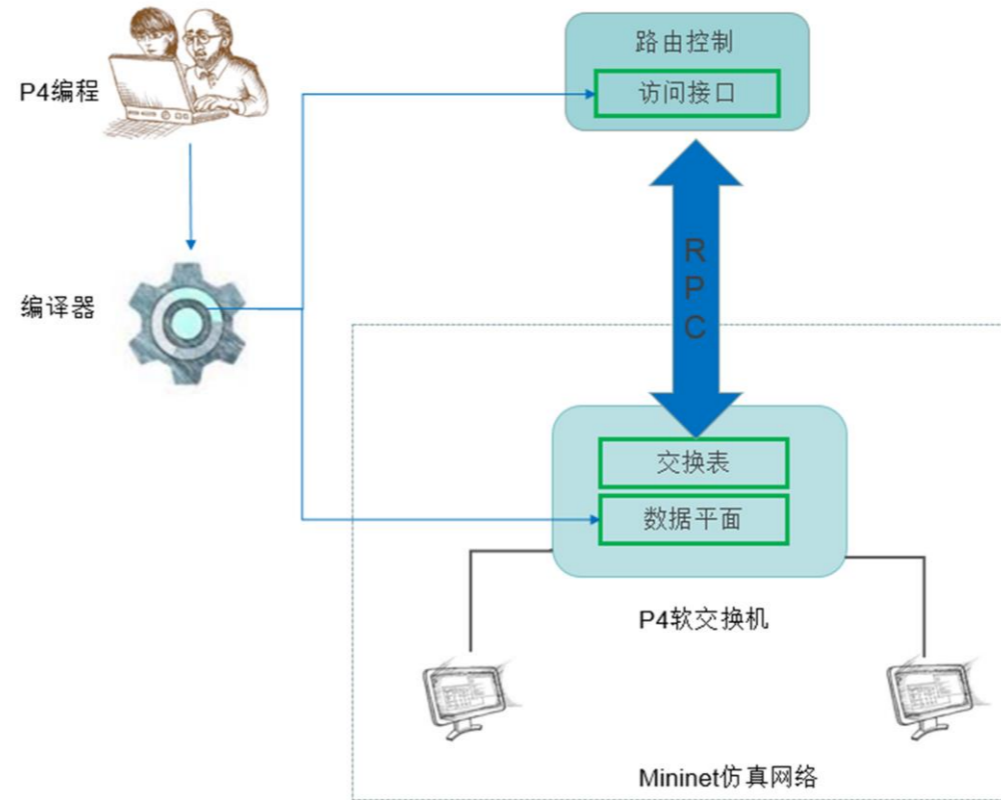
```
$. /configure
```

▶ 运行

```
$cd targets/basic_routing
```

```
$make bm
```

```
sudo ./behavioural-modal
```



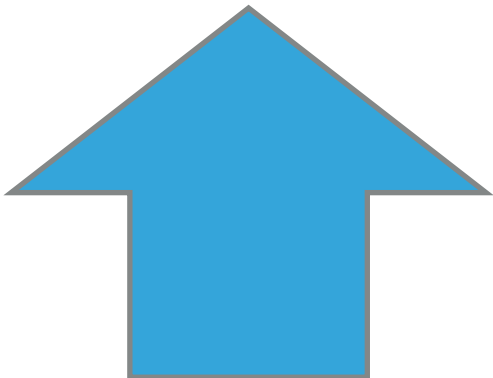
```
p4@ubuntu:~/p4factory/targets/simple_router$ sudo ./behavioral-model
[sudo] password for p4:
No PD RPC server address specified, using 127.0.0.1:9090
No listener specified, switch will run in standalone mode

P4 Program: simple_router


Starting RPC server on port 9090
In client_init
In ipv4_lpm get_entry
```

```
(Cmd) show_entry ipv4_lpm 1
key:
  ipv4_dstAddr: 0x00000000 (0 0 0 0),
  prefix_length:
    0
action:
  set_nhop
action data:
  nhop_ipv4: 0x00000000 (0 0 0 0), port: 0x00000000 (0 0 0 0),
```

P4与ONOS



ONOS1.6版新增了“BMv2 Device Context Service”北向接口，应用程序调用此接口指定bmv2运行时的JSON配置。Bmv2接口将在下个版本中作为ONOS的核心接口。



ONOS南向通过thrift与bmv2通信，ONOS1.6版中修改了P4项目中的simple_switch模块，添加了对连接SDN控制器的支持。

ONOS1.6支持的P4特性

- 设备发现：连接/断开事件
- JSON配置转换
- Packet-in和packet-out
- 匹配-动作表填充（通过流规则，流目标或意图）
- 端口统计数据收集
- 流统计数据收集

快速开始

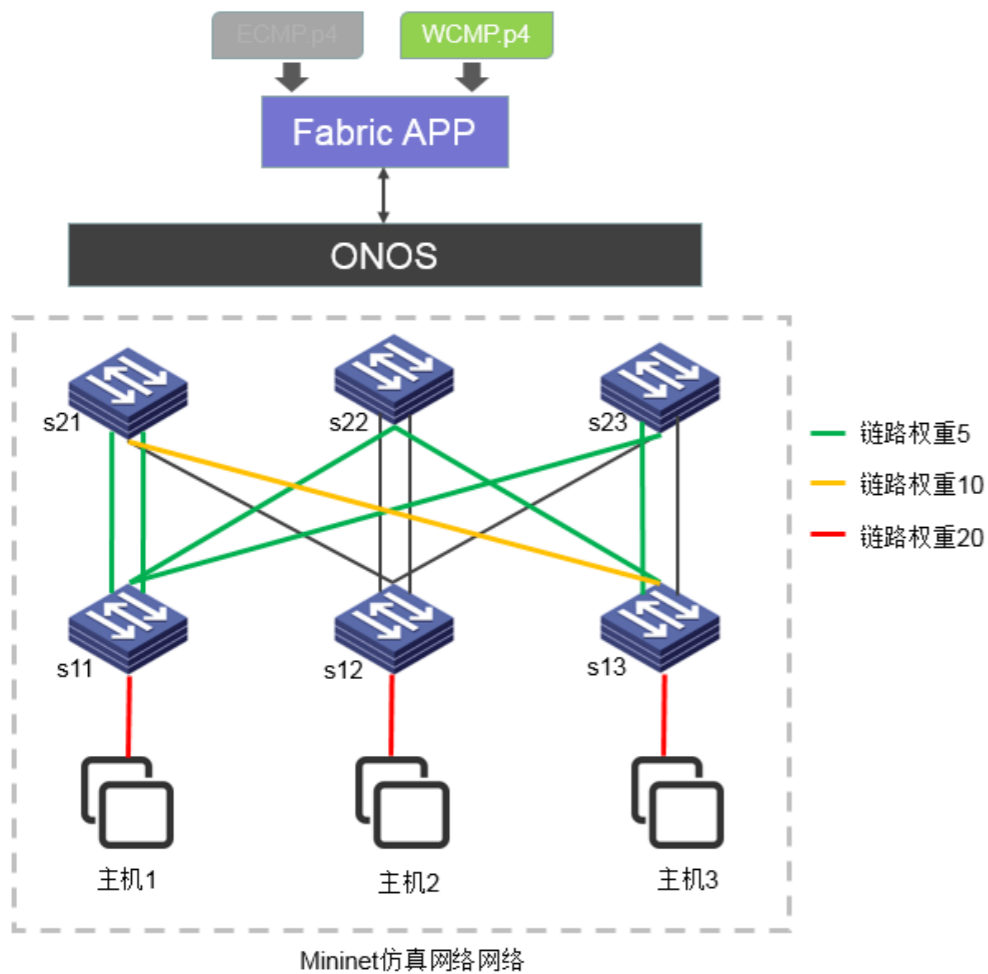
- bmv2及p4c-bmv2编译脚本
- Mininet拓扑脚本bmv2.py
- P4样例程序
- Mininet网络快速debug命令

▶ ECMP/WCMP程序

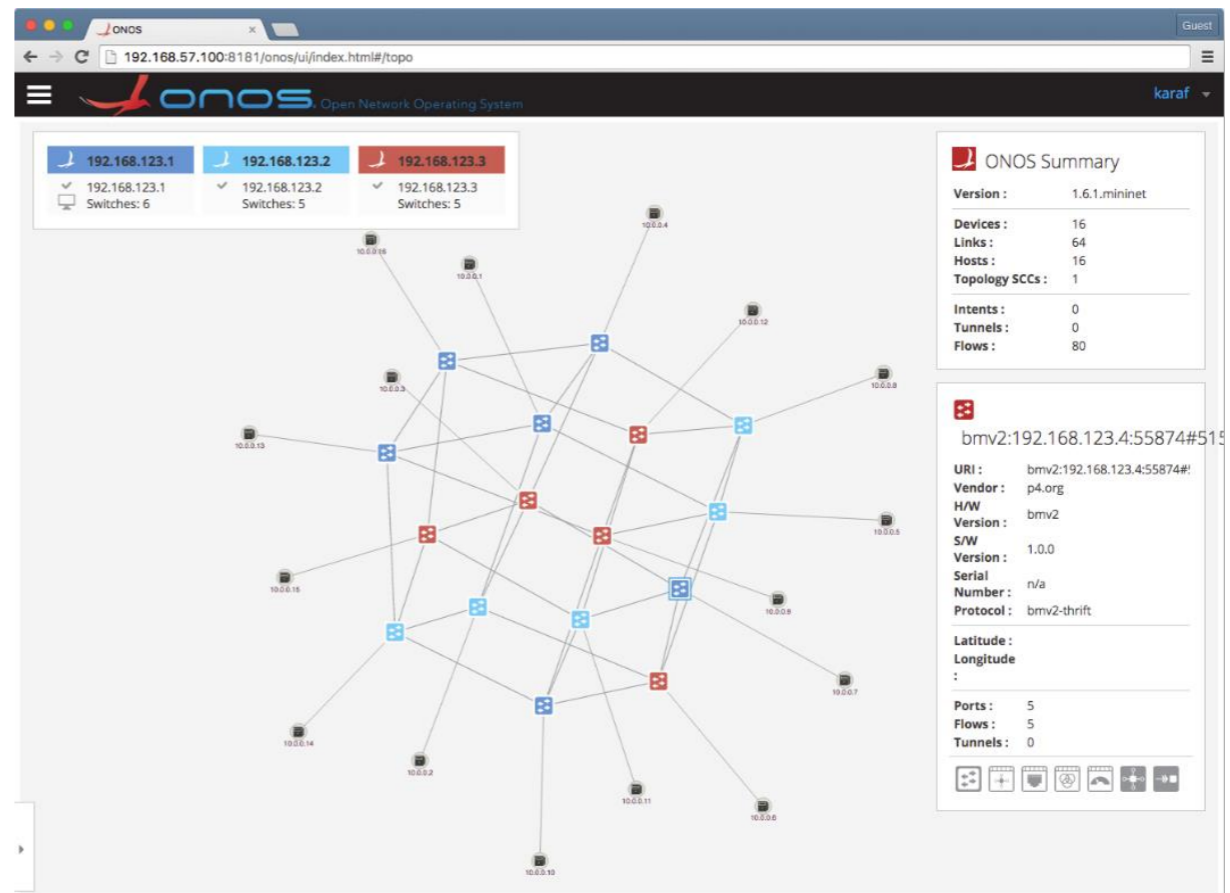
ONOS1.6中附带了两个P4示例程序：ECMP.p4和WCMP.p4。

用户只需编译安装ONOS并激活*BMv2 Drivers*应用，然后使用onos提供的自定义拓扑脚本创建mininet仿真网络即可。

WCMP示例架构图



网络拓扑



基于P4网络监测

传统网络监测

- 客户端/服务器模式
- Pull 模式

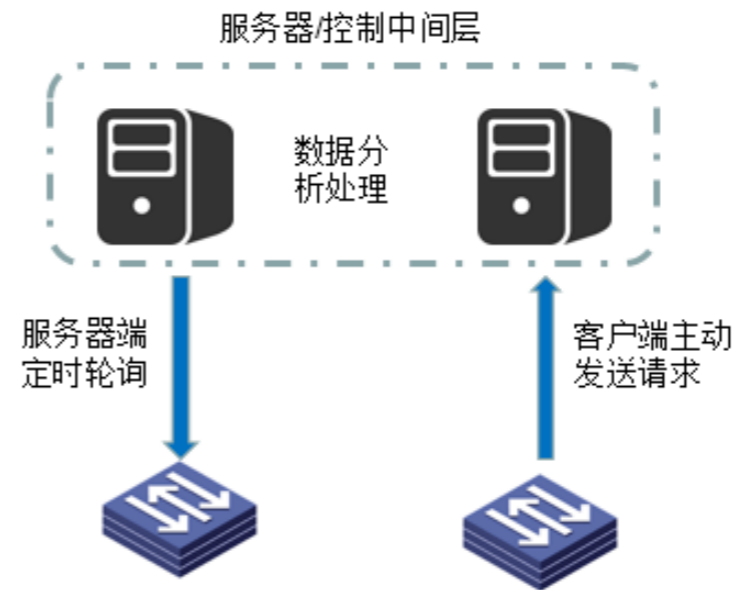
缺点

实时性

- 涉及CPU和控制层面
- 无法监控快速变化的网络状态

端到端网络状态

- 无法将流的实际路径和元素状态关联

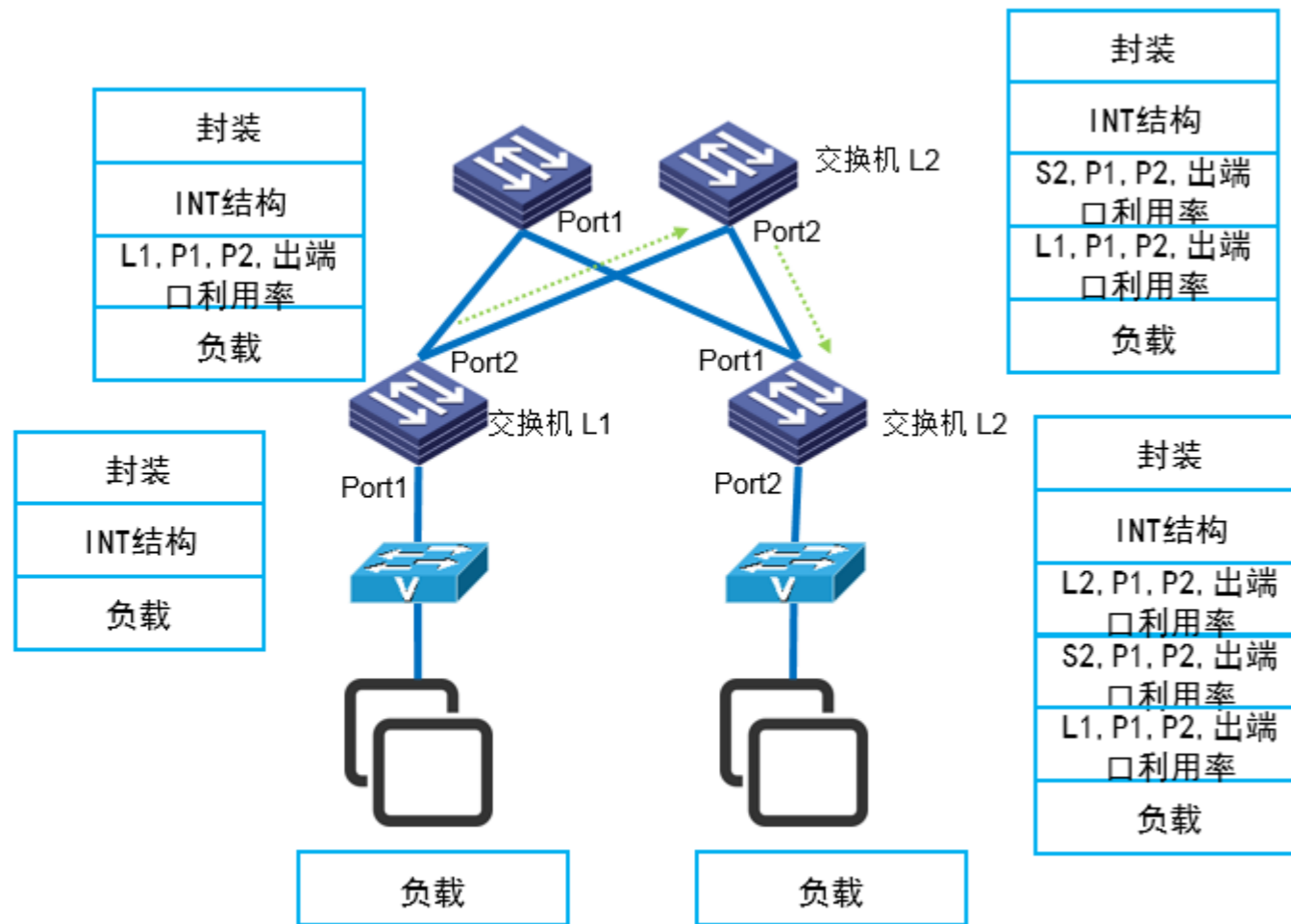


需要监测的网络状态数据

- 交换机ID
- 入端口ID
- 出端口ID
- 每跳延迟
- 出端口队列占用
- 出端口队列拥塞状态
-

P4网络监测场景实现

► 应用场景



In-band模式网络状态监测机制

- 目的交换机将收集的数据发送到本地CPU本地设备上处理
- 目的交换机将数据导出到远程服务器上的网管系统或大数据分析工具
- 目的交换机将数据反馈到源交换机，以作为源交换机处理其他数据参考

网络状态元素

- 交换机ID+入端口ID+出端口ID——确定交换机端到端之间的不同路径。而传统的基于IP的路由追踪监测方式只能监控3层路径，无法区分Port-channel里的多条链路。
- 链路利用率——交换机端到端的不同路径的链路利用率既可以用于基础的监控，也可以用于选择新流量的路径选择，而不是盲目地将流量导向任意等价路径上。
- 延迟——交换机端到端的不同路径的链路延迟既可以用于基础的监控，也可以用于时间敏感类业务的智能路由。

谢谢观看！



江苏省未来网络创新研究院-杨帅@SDNLAB
yangshuai@sdnlab.com

我是分割线

P4集成MININET与DOCKER

▶ 启动mininet

```
$sudo python ../../mininet/l2_demo.py --num-  
hosts 4 --behavioral-exe $PWD/behavioral-model
```

```
p4@ubuntu:~/p4factory/targets/simple_router$ sudo python ../../mininet/l2w_demo.py --behavioral-exe $PWD/behavioral-model  
*** Creating network  
*** Adding hosts:  
h1 h2  
*** Adding switches:  
s1  
*** Adding links:  
(h1, s1) (h2, s1)  
*** Configuring hosts  
h1 h2  
*** Starting controller  
*** Starting 1 switches  
s1 Starting P4 switch s1  
/home/p4/p4factory/targets/simple_router/behavioral-model --name s1 --dpid 0000000000000001 -i s1-eth1 -i s1-eth2 --listen  
127.0.0.1:11111 --pd-server 127.0.0.1:22222  
switch has been started  
  
*****  
h1  
default interface: eth0 10.0.0.10      00:04:00:00:00:00  
*****  
*****  
h2  
default interface: eth0 10.0.1.10      00:04:00:00:00:01  
*****  
Ready !  
*** Starting CLI:  
mininet>
```

▶ 制作docker镜像

```
//修改MakeFile文件
```

```
DOCKER_IMAGE := bm-l2-switch
```

```
include ${MAKEFILES_DIR}/docker.mk
```

```
//创建镜像文件
```

```
make docker-image
```

```
p4@ubuntu:~/p4factory/targets/l2_switch$ make docker-image  
Building docker image for target bm-l2-switch  
Sending build context to Docker daemon 1.134 GB  
Step 1 : FROM ubuntu:14.04  
14.04: Pulling from library/ubuntu  
04c996abc244: Pull complete  
d394d3da86fe: Pull complete  
bac77aae22d4: Pull complete  
b48b86b78e97: Pull complete  
09b3dd842bf5: Pull complete  
Digest: sha256:bd00486535fd3ab00463b0572d94a62715cb790e482d5419c9179cd22c74520b  
Status: Downloaded newer image for ubuntu:14.04  
--> f2d8ce9fa988  
Step 2 : MAINTAINER Antonin Bas <antonin@barefootnetworks.com>  
--> Running in aa6436880caf  
--> 7b9a34507a7a  
Removing intermediate container aa6436880caf  
Step 3 : RUN apt-get update  
--> Running in 6a29d60a4755
```


P4项目发展历

Nick McKeown、Jennifer Rexford、Amin Vahdat等教授在ACM SIGCOMM上联合发表论文：《P4: Programming Protocol-Independent Packet Processors》。以此拉开了高级数据平面编程语言P4高速发展的序幕。

2014年7月

2015年3月

P4项目社区正式发布目前广泛支持的P4语言标准《The P4 Language Specification Version 1.0.2》。

SIGCOMM在伦敦举行了年度会议，《P4: Programming Protocol-Independent Packet Processors》当选为年度最佳论文。第一次P4研讨会顺势召开，众多国际顶尖网络学者、专家参与讨论了P4在L4负载平衡，网络监控和分析，动态路由、以及故障排除方面的发展。

2015年8月

2016年6月

Nick教授等人创办的Barefoot公司C轮获得\$5700万融资

《The P4 Language Specification Version 1.1》发布

2016年7月

2016年10月

超过50个成员组织加入P4项目社区，探讨P4语言标准制定和发展方向。