CERNET第二十六届学术年会

SDN网络中SAVI绑定表安全研究

华中科技大学　李冬

2019.11.13

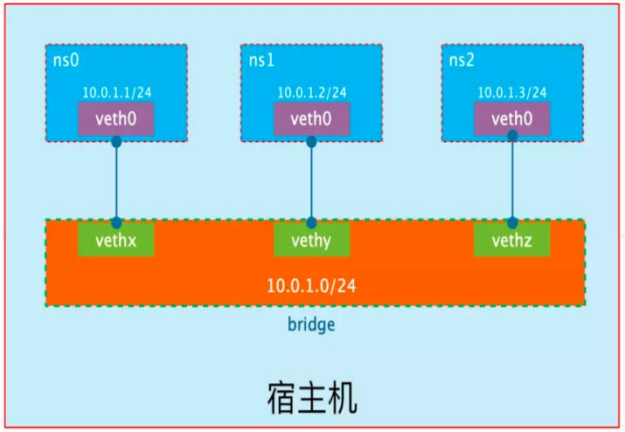
内容提纲

1 源地址验证（IPv6）新问题

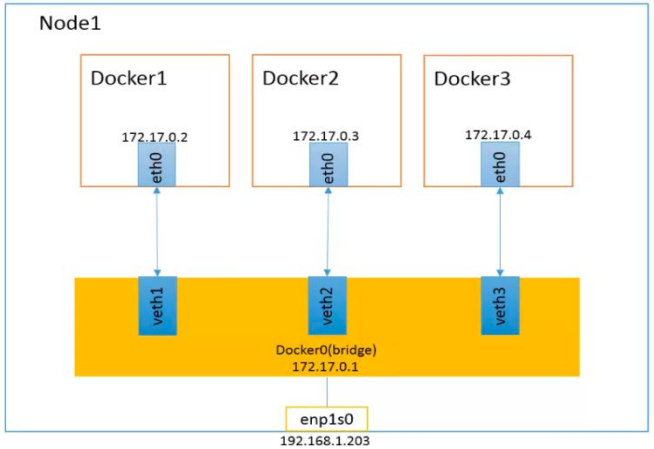
2 绑定表的安全问题分析

3 SDN网络中绑定表安全机制

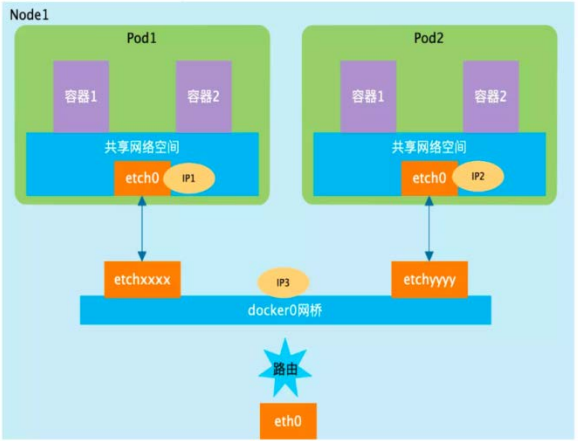
云网技术的发展



Linux虚拟网络

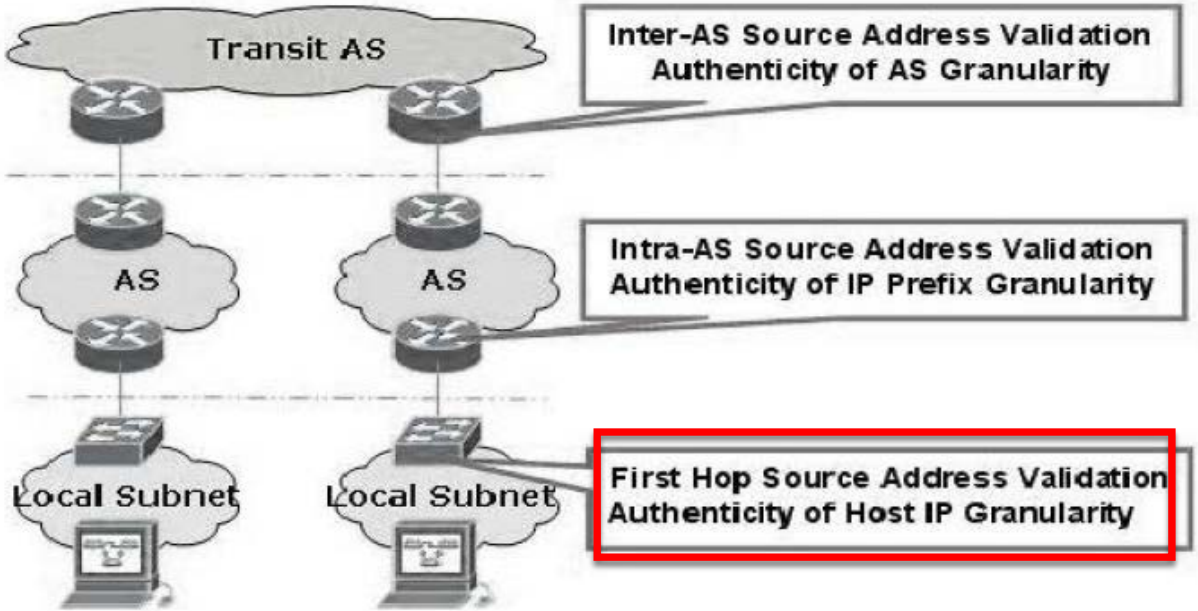


Docker虚拟网络



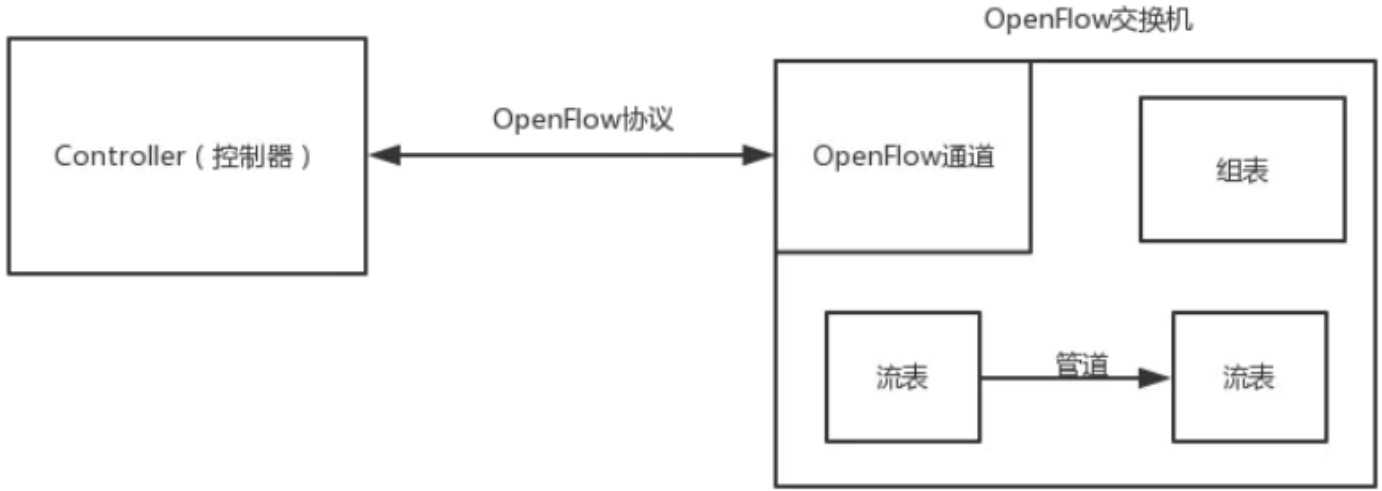
kubernetes虚拟网络

源地址验证结构



Source Address Validation Architecture

SDN技术的发展



1. SDN技术能够较好的适应云网技术发展；
2. SDN技术能够较好的适应真实源地址验证技术需求。

内容提纲

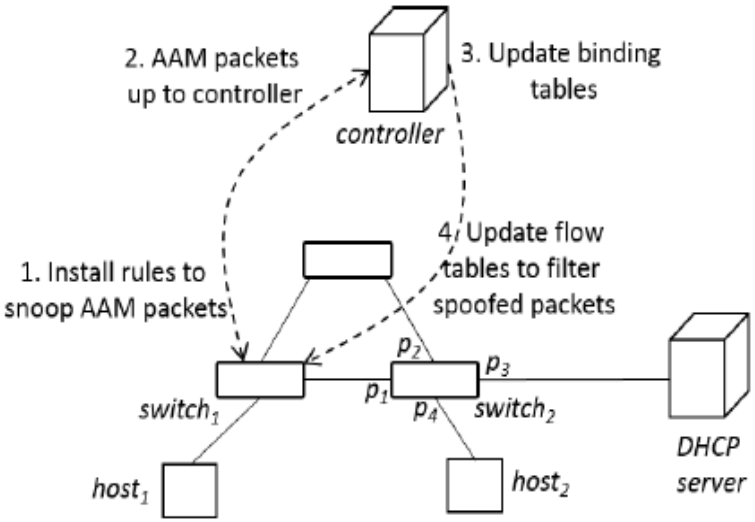
1 源地址验证（IPv6）新问题

2 绑定表的安全问题分析

3 SDN网络中绑定表安全机制

研究背景

- Bingyang Liu, Jun Bi, Yu Zhou. Source Address Validation in Software Defined Networks, SIGCOMM,2016
- https://github.com/netarchlab-savi/savi-floodlight



我们的研究：

1. 端口(TRUST,UNTRUST)： 动态SAVI

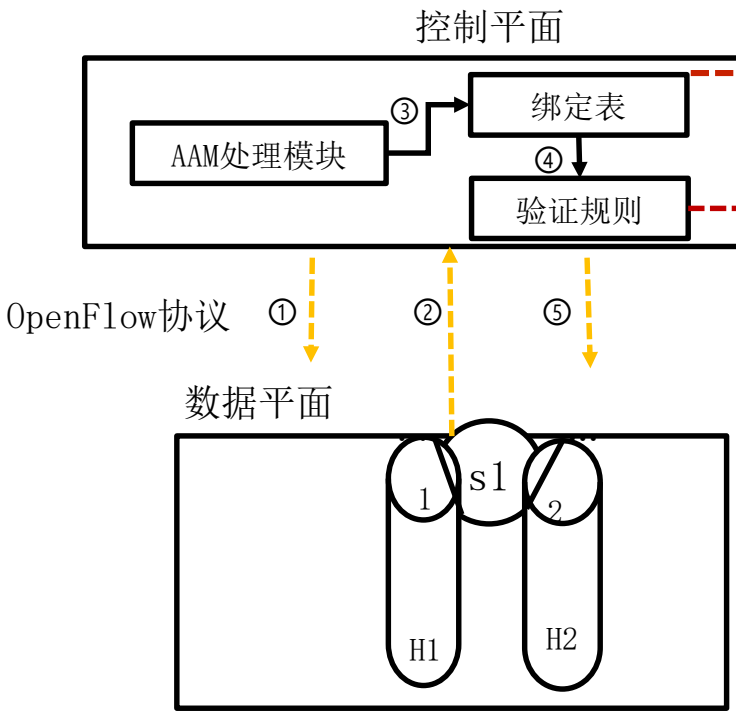
2. SAVI绑定表的安全性

真实性： 消息验证

完整性： 更新机制

可靠性： 验证效率

研究背景



绑定表

SwitchPort	MAC	IP
[s1,1]	MAC1	IP1
[s1,2]	MAC2	IP2

源地址验证流表

匹配域			动作
Port	MAC	IP	
1	MAC1	IP1	goto table 2
2	MAC2	IP2	goto table 2
*			drop

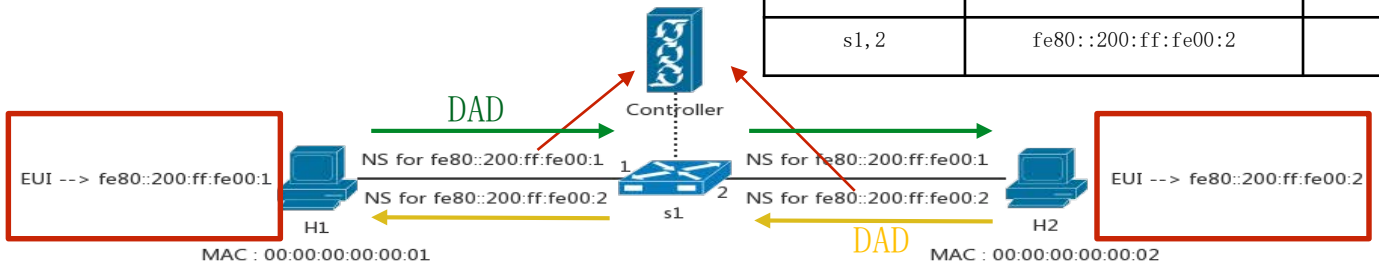
①控制器向交换机下发监听流表，监听AAM报文
②主机H1和H2上线请求地址配置的信息被交换机以Packet_In的形式上传给控制器
③控制器中AAM处理模块根据对应Packet_In生成绑定表
④控制器根据绑定表中绑定表生成验证流表
⑤将验证流表下发到对应交换机

验证过程：主机H1发出的合法报文匹配验证流表中第一条流表，交换机执行匹配动作**转发**。H1发出的伪造报文不匹配前两条流表，最后匹配table_miss，交换机执行匹配动作**drop**。**转发真实报文，丢弃伪造报文即达到源地址验证的目的。**

研究背景

本地链路地址绑定

SwitchPort	IP	MAC
s1,1	fe80::200:ff:fe00:1	00:00:00:00:00:01
s1,2	fe80::200:ff:fe00:2	00:00:00:00:00:02



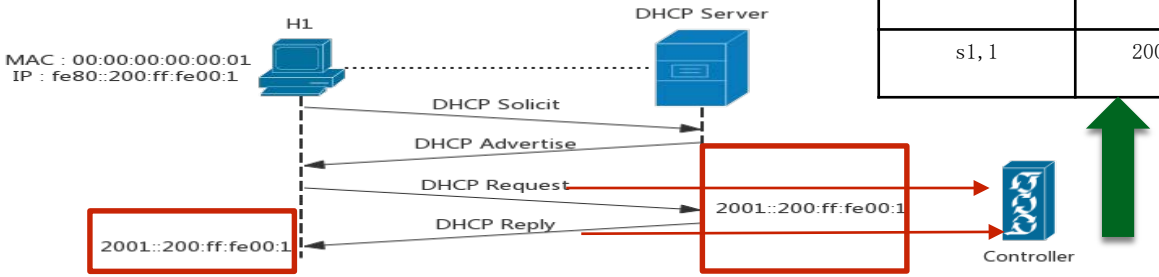
SLAAC配置的IPv6地址绑定

新增一条绑定表项

SwitchPort	IP	MAC
s1,1	2001::200:ff:fe00:1	00:00:00:00:00:01

问题分析

DHCPv6配置IPv6地址绑定



新增一条绑定表项

SwitchPort	IP	MAC
s1, 1	2001::200:ff:fe00:1	00:00:00:00:00:01

主要问题

I. 缺乏验证机制

- ✓ 控制器没有对AAM报文进行验证，导致绑定表可能被污染、破坏，同时处理不恰当主机配置地址错误。

II. 缺乏更新机制

- ✓ 没有记录以及更新绑定IP地址的有效期或主机状态，导致绑定表记录无法更新。

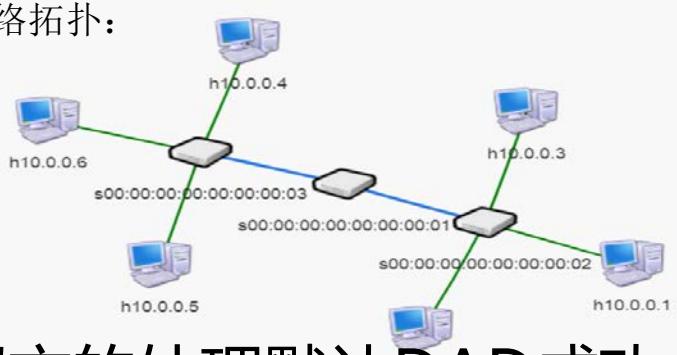
III. 缺乏对报文优化处理机制

- ✓ 没有仔细过滤AAM报文，导致控制器容易受到DoS攻击。

问题分析

- 缺乏AAM报文验证，容易遭受伪造消息攻击

网络拓扑：



发起攻击：

伪造的随机IP

```
root@ubuntu:~/sdn/savi# python ndSproof.py 1
ip:282b:5684:310c:8870:d3ce:de78:8d4d:adec
.
Sent 1 packets.
root@ubuntu:~/sdn/savi# ifconfig
h1-eth0   Link encap:Ethernet  HWaddr 00:00:00:00:00:01
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::200:ff:fe00:1/64 Scope:Link
          inet6 addr: 2001:db1::200:ff:fe00:1/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:271 errors:0 dropped:22 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:24752 (24.7 KB)  TX bytes:1416 (1.4 KB)
```

- NS报文的处理默认DAD成功，主机地址配置错误

绑定表：

SwitchPort	MAC	IP
s1, 1	00:00:00:00:00:01	fe80::200:ff:fe00:1
s1, 1	00:00:00:00:00:01	2001:db1::200:ff:fe00:2
s1, 2	00:00:00:00:00:03	
...	...	
s2, 3	00:00:00:00:00:0	0:ff:fe00:6
s2, 3	00:00:00:00:00:0	200:ff:fe00:
s1 ,1	00:00:00:00:00:0	282b:5684:310c:8870:d3ce:de78:8d4d:adec

绑定的主机H1的两个IP地址

检测到相同IP不绑定并丢弃

H2配置了和H1相同的IP地址，而绑定表不含绑定导致主机H2不能用这个IP地址通信

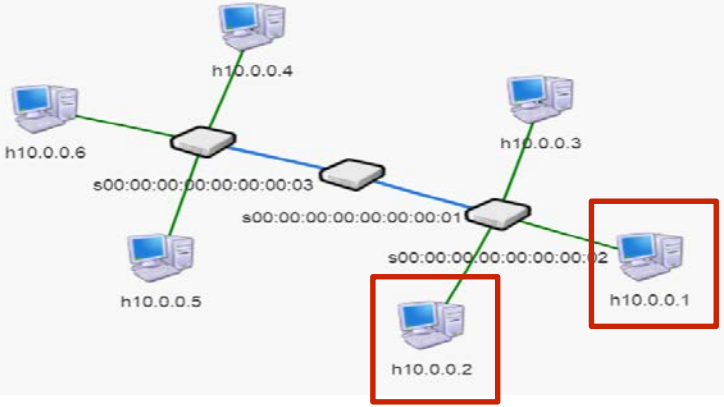
绑定表维护的是主机的IP地址

DAD成功，配置这个IP地址

伪造消息	结果
NS	端口绑定伪造的IP
	恶意删除绑定项
	伪造ip有效期

问题分析

绑定表更新问题




```
root@ubuntu:~/sdn/savi# ifconfig
h1-eth0   Link encap:Ethernet  HWaddr 00:00:00:00:00:01
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: 2001:db1::200:ff:fe00:1/64 Scope:Global
          inet6 addr: fe80::200:ff:fe00:1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1221 errors:0 dropped:12 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:106557 (106.5 KB)  TX bytes:1420 (1.4 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@ubuntu:~/sdn/savi# ifconfig
h1-eth0   Link encap:Ethernet  HWaddr 00:00:00:00:00:01
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::200:ff:fe00:1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1223 errors:0 dropped:14 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:106715 (106.7 KB)  TX bytes:1420 (1.4 KB)
```

SwitchPort	MAC	IP
s1, 1	00:00:00:00:00:01	fe80::200:ff:fe00:1
s1, 1	00:00:00:00:00:01	2001:db1::200:ff:fe00:2
s1, 2	00:00:00:00:00:03	fe80::200:ff:fe00:3
s1, 2	00:00:00:00:00:03	2001:db1::200:ff:fe00:3
...	...	...
s2, 3	00:00:00:00:00:06	2001:db1::200:ff:fe00:6

仍然存在

仍然存在于绑定表中

更新问题可能带来的危害：

1. 新主机冒用端口旧主机身份
2. 消耗绑定表空间
3. 没有删除的无效IP不能被绑定

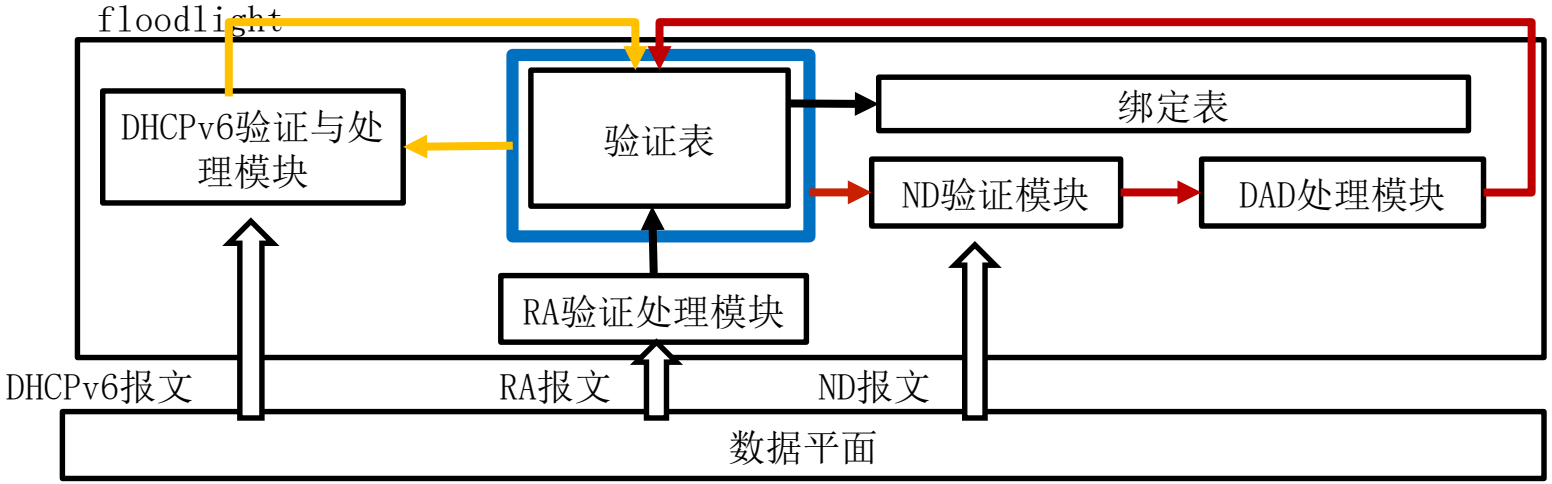
内容提纲

1 源地址验证（IPv6）问题

2 绑定表的安全问题分析

3 SDN网络中绑定表安全机制

绑定表验证机制总体架构



1.建立消息验证表，记录主机的真实信息。
2.监听处理路由通告报文，分类主机并记录前缀信息。
3.ND报文的验证：利用消息验证表并根据RA报文处理结果验证ND报文。
4.ND报文的处理：通过DAD处理模块处理ND报文，处理的结果用于更新验证表和构建绑定表项。
5.DHCPv6报文的验证和处理：利用消息验证表并根据RA报文处理结果验证DHCPv6报文，并根据具体DHCPv6消息具体处理。处理结果更新验证表，部分消息处理结果更新绑定表。

绑定表验证机制

I. 建立消息验证表

SwitchPort	MAC	IPs
port	mac	IPList

size<=5

IP	Valid Lifetime	Status	Type
IP1	Vt1	0	0
IP2	Vt2	1	1
...			

0：本地链路地址
1：全球单播地址
2：临时地址

0:绑定完成
1:尚未绑定，待检测

- ✓ 每一个表项代表一个已验证的主机
- ✓ 每一项记录一个主机所处交换机端口、mac地址以及IP地址信息列表
- ✓ IP地址信息列表里包含新旧公网地址和新旧临时地址以及它们的有效时间、绑定状态和类型

II. 监听和处理路由通告消息

- 向边缘交换机部署RA的监听流表，监听路由通告消息

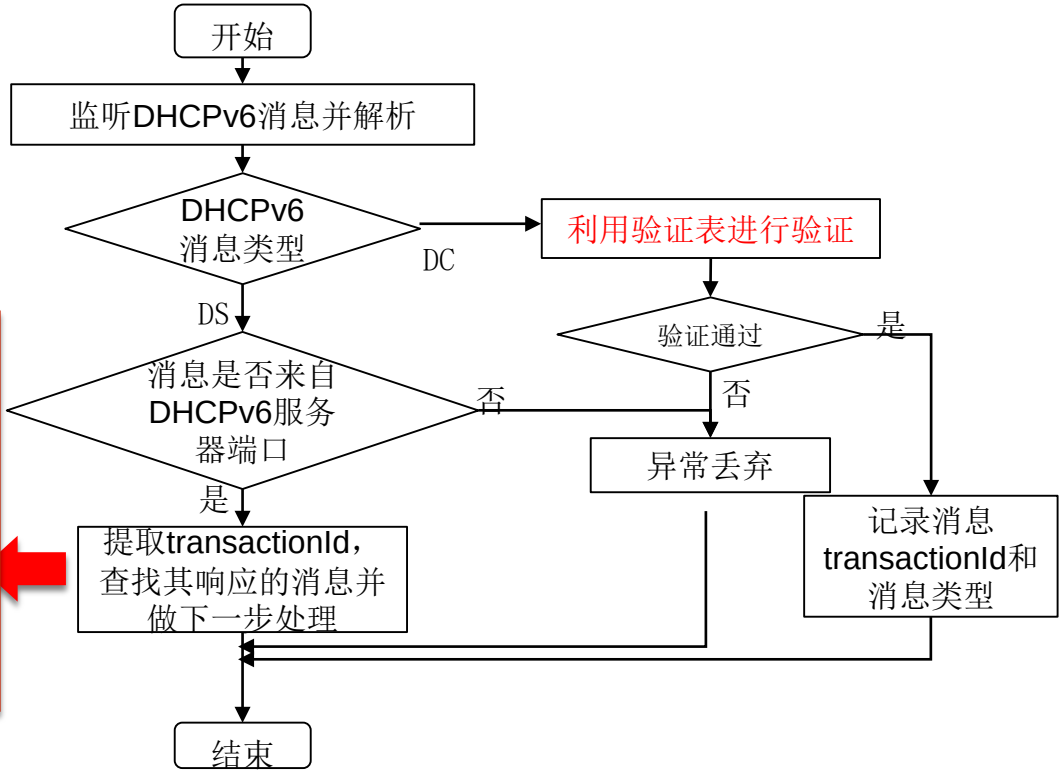
Ethernet Type	Next Header	ICMPv6 Type	Instructions
IPv6	ICMPv6	134	goto controller

- 在控制器中增加RA报文的解析模块
  - ✓ 新增RA消息的类型，继承ICMPv6类型
  - ✓ 重写序列化和反序列化方法，解析出各字段的值
- 过滤不合法RA报文
  - ✓ 判断RA消息源端口，丢弃边缘交换机端口的RA消息
- 判断主机地址配置方式
  - ✓ 交换机s1的p4端口收到RA
  - ✓ 解析RA消息的managed Address Configuration字段

绑定表验证机制

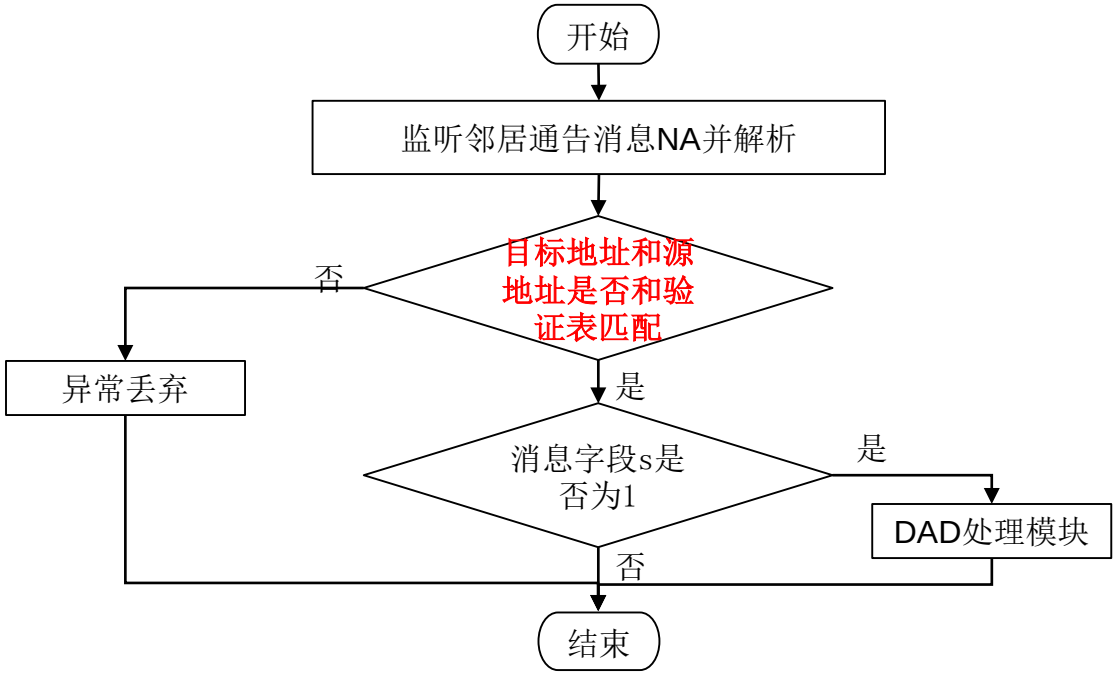
III. DHCPv6消息的监听和处理

响应的消息类型	下一步的处理
Request	新增验证表地址信息，并将状态置为1
Renew/Rebind	提取验证表和绑定表对应项，并更新其有效时间
Decline/Release	删除验证表和绑定表对应IPv6地址
Confirm	修改验证表和绑定表的绑定端口

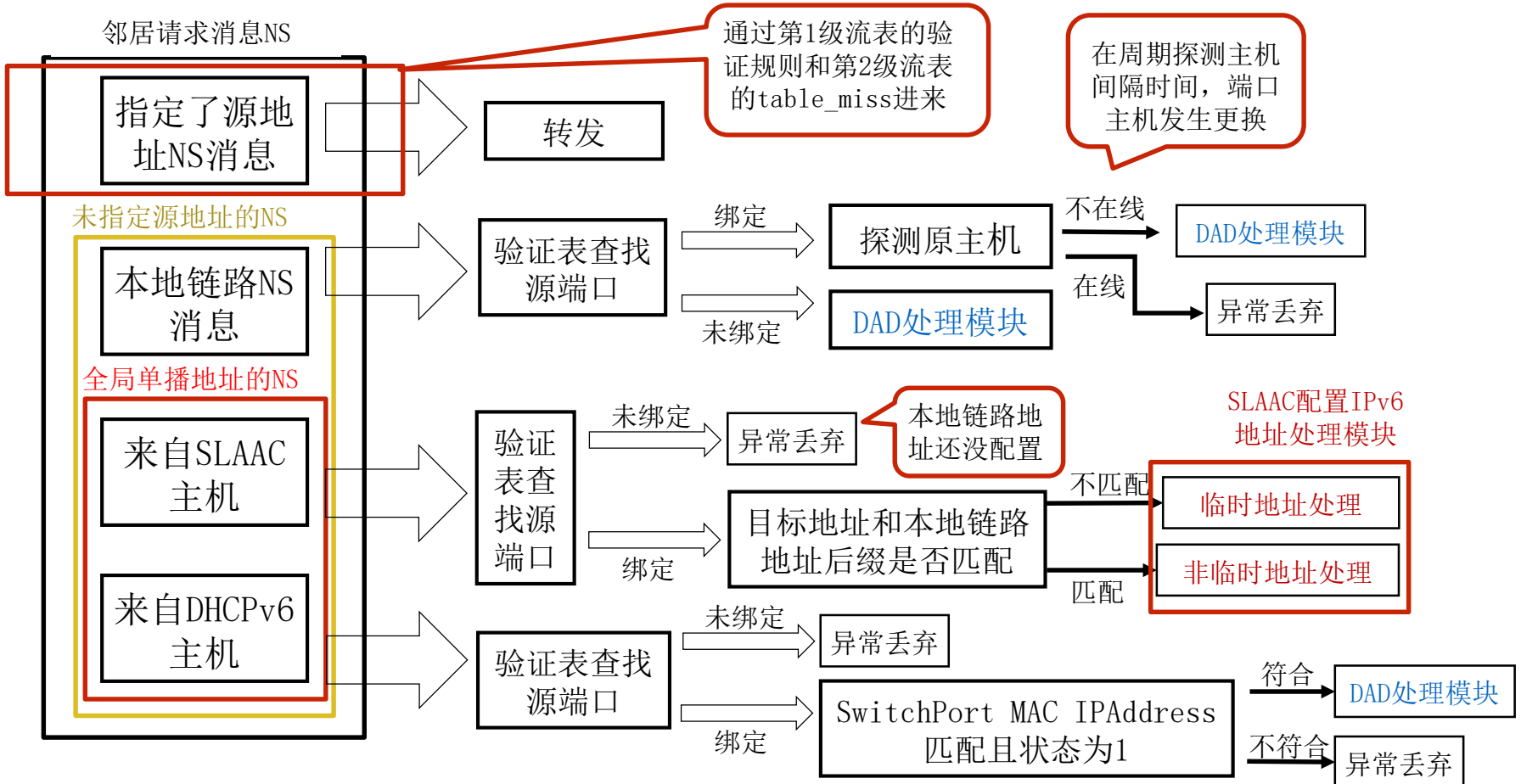


III.ND监听和处理(NA,NS)

- 监听和处理NA消息，验证主机地址的正确性



绑定表验证机制

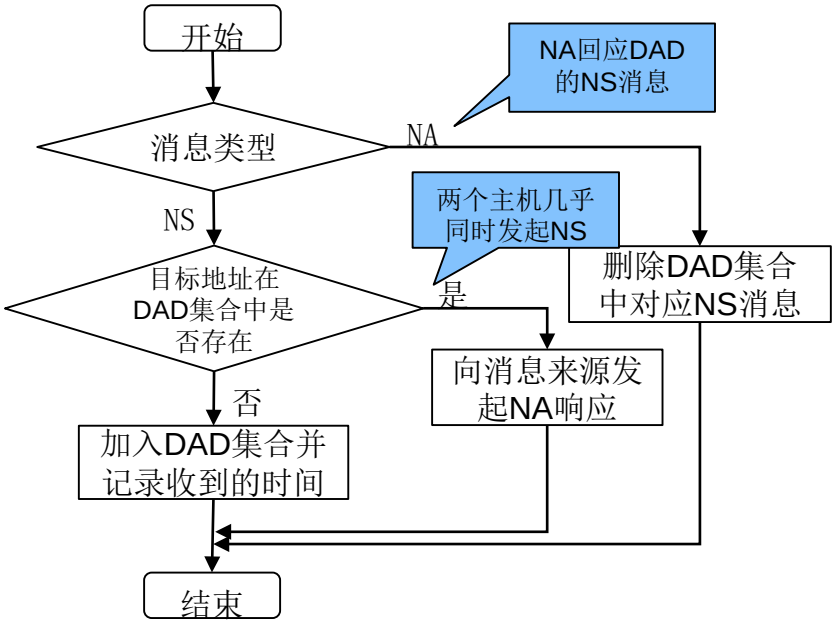

邻居请求消息NS
指定了源地址NS消息
未指定源地址的NS
本地链路NS消息
全局单播地址的NS
来自SLAAC主机
来自DHCPv6主机
转发
通过第1级流表的验证规则和第2级流表的table_miss进来
在周期探测主机间隔时间，端口主机发生更换
验证表查找源端口
绑定
探测原主机
不在线
DAD处理模块
在线
异常丢弃
未绑定
DAD处理模块
验证表查找源端口
未绑定
异常丢弃
本地链路地址还没配置
SLAAC配置IPv6地址处理模块
绑定
目标地址和本地链路地址后缀是否匹配
不匹配
临时地址处理
匹配
非临时地址处理
验证表查找源端口
未绑定
异常丢弃
绑定
SwitchPort MAC IPAddress匹配且状态为1
符合
DAD处理模块
不符合
异常丢弃

绑定表验证机制

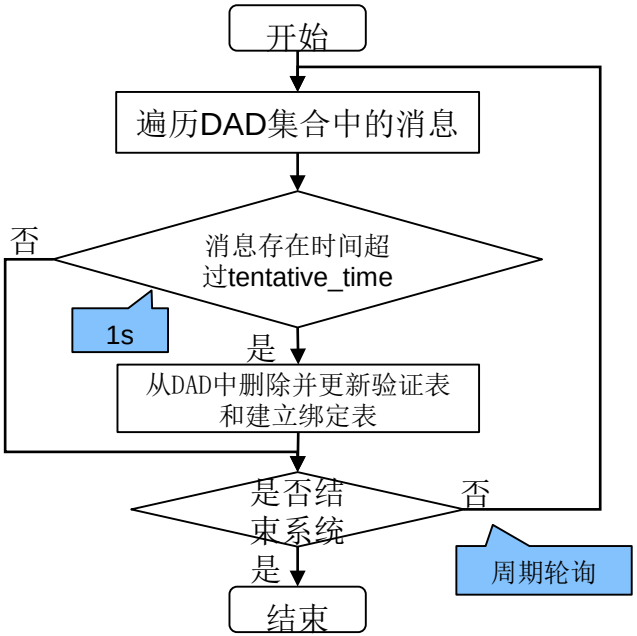
IV. DAD报文的处理

- 对真实DAD报文进行处理，构建绑定表

消息处理线程

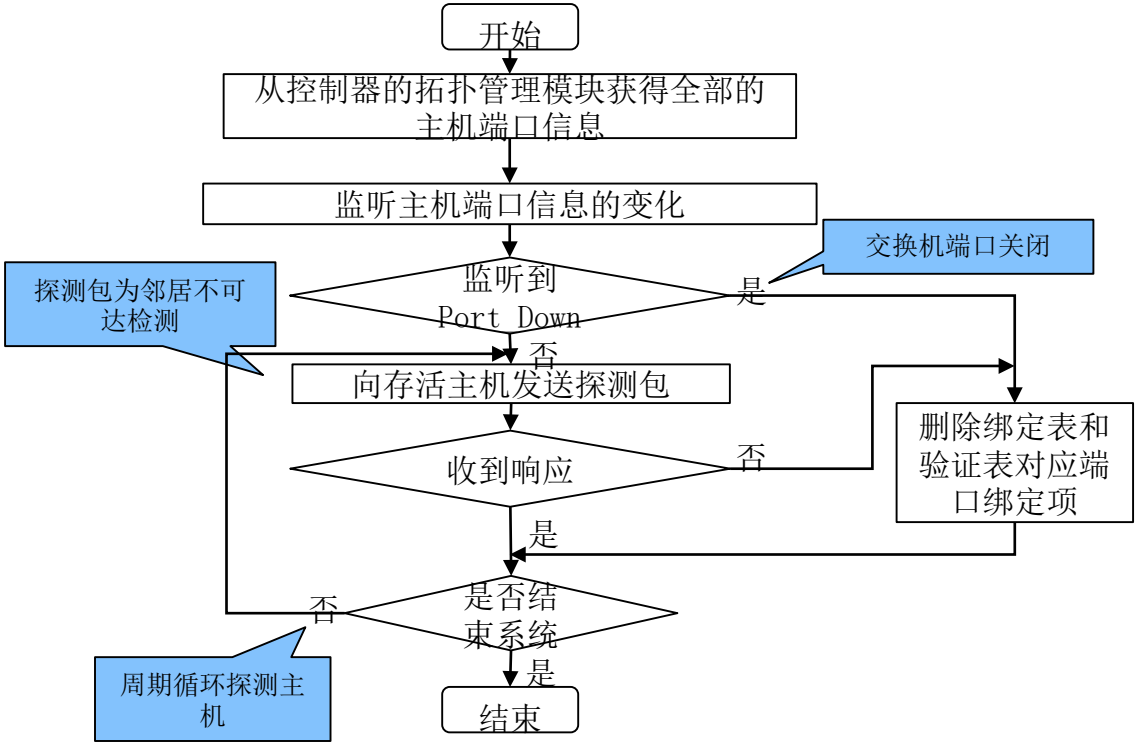


周期轮询线程



绑定表更新机制

I. 监听端口变化，并发送探测包以探测主机在线状态



绑定表更新机制

II. 更新主机IP地址有效期

- 本地链路地址有效时间为永久，无需更新
- SLAAC地址配置的主机地址有效时间的更新
  - ✓ 全局单播地址有效时间 → DAD检测时开始，有效时间为RA报文携带前缀信息Valid Lifetime的值；在地址失效前收到RA相同前缀时更新有效期。
  - ✓ 临时地址有效时间→ 不超过RA前缀信息字段Valid Lifetime值；在收到RA相同前缀时也会更新。
- DHCPv6地址配置的主机有效时间设置和更新
  - ✓ 全局单播地址有效时间 → DHCPv6 reply报文

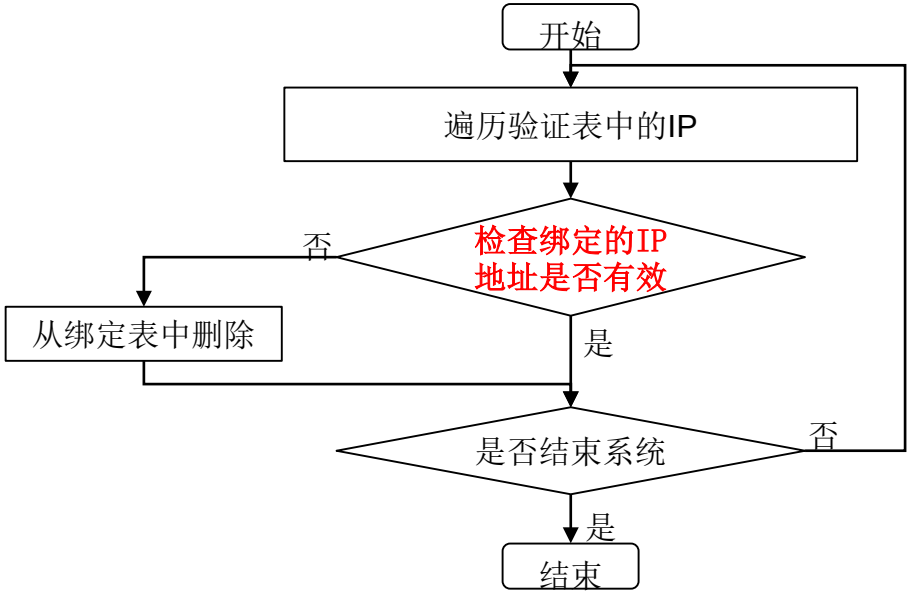
reply对应的有效时间处理

reply响应的消息类型	对应的处理
Request	设置初始有效时间
Renew	更新有效时间
Rebind	更新有效时间
Release	地址失效

绑定表更新机制研究

III.周期检查绑定IP的有效时间

- 创建一个周期轮询的线程，周期检查IP地址的有效期，删除已经失效的IP对应绑定表项，保证绑定表的及时更新



安全优化机制

I. 减小监听数据包范围

- 划分交换机类型，只向边缘交换机下发监听规则 → 非边缘交换机上的数据包可以直接转发，不需要经过控制器
- 构建匹配域更精确的流表监听规则

DHCPv6消息的流表匹配域

Ethernet Type	Next Header	Src Port	Dest Port	Action
IPv6	UDP	546	547	goto Controller
IPv6	UDP	547	546	goto Controller

NS/NA消息的流表匹配域

Ethernet Type	Next Header	ICMPv6 Type	Dst IP	Action
IPv6	ICMPv6	136	ff02::1	goto Controller

Ethernet Type	Next Header	ICMPv6 Type	Src IP	Dst IP	Action
IPv6	ICMPv6	135	::	ff02::/16	goto Controller

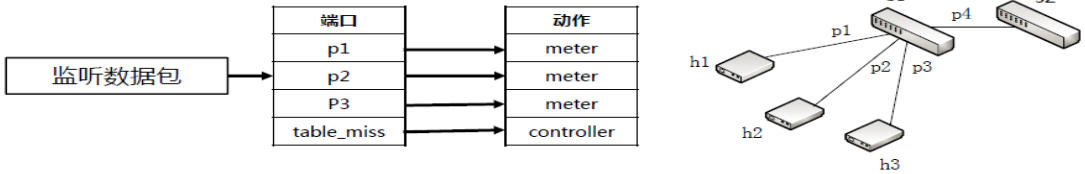
减少不相关ICMPv6消息打包成Packet_In给控制器带来负载

在交换机层面筛选出属于AAM的NS报文

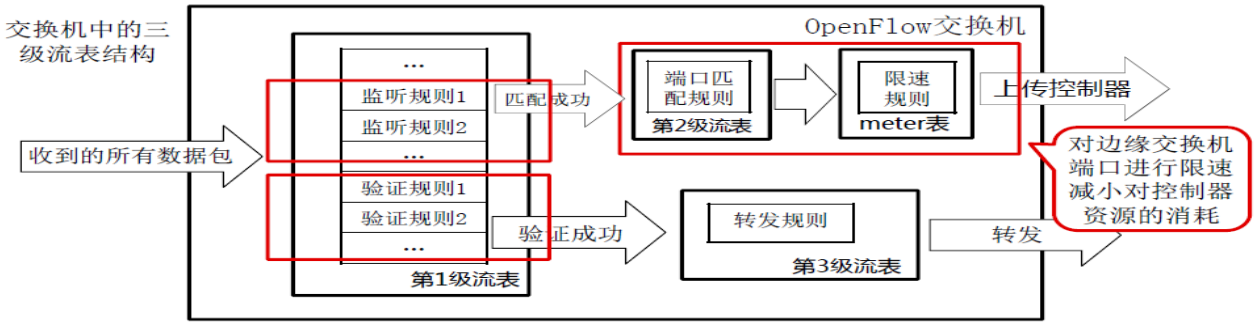
安全优化机制

II. 对监听报文进行限速

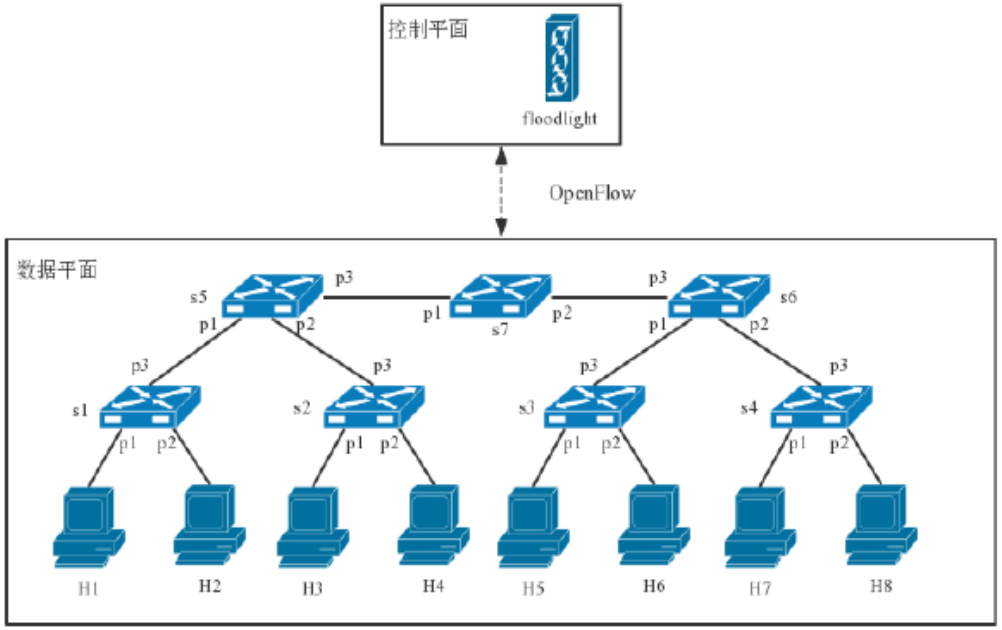
- 设置多级流表结构，根据端口信息划分监听报文对每个边缘交换机端口进行限速



- 监听流表的优先级设置较验证规则高，避免监听的数据包通过验证规则流表和转发流表不经过控制器转发出去



实验结果


控制平面
floodlight
OpenFlow
数据平面
s5
s6
s7
s1
s2
s3
s4
p1
p2
p3
H1
H2
H3
H4
H5
H6
H7
H8

实验拓扑

实验结果1

报文编号	伪造报文类型	描述
1	Request/Reply	伪造 MAC 地址
2	Request/Reply	伪造 IA 选项 IP 地址
3	Decline	伪造 IA 选项 IP 地址为其他主机 IP
4	NS	伪造 MAC 地址
5	NS	目标地址为本地链路地址的
6	NS	目标地址为全球单播地址且前缀不匹配
7	NS	目标地址为前缀匹配的全球单播地址且未被使用
8	NS	目标地址为其他主机 IP 地址

伪造报文类型

报文编号	SLAAC		DHCPv6	
	使用临时地址	不使用临时地址	使用临时地址	不使用临时地址
1	能/无	能/无	能/无	能/无
2	能/无	能/无	能/无	能/无
3	能/无	能/无	能/无	能/无
4	能/无	能/无	能/无	能/无
5	能/无	能/无	能/无	能/无
6	能/无	能/无	能/无	能/无
7	能/无	不能/有	能/无	能/无
8	能/无	不能/无	能/无	能/无

攻击防御检测结果
（能否防御/对绑定表有无影响）

实验结果2

节点：H1,H2；地址配置方式：SLACC
路由通告消息前缀信息:
ValidLifetime:120秒，
PreferredLifetime:80秒

```
=======绑定表========
SwitchPort: [s1, 1]; MAC:00:00:00:00:00:01; IP:2001:db1::200:ff:fe00:1
SwitchPort: [s1, 1]; MAC:00:00:00:00:00:01; IP:2001:db1::fd98:cd40:d6fa:4106
SwitchPort: [s1, 1]; MAC:00:00:00:00:00:01; IP:fe80::200:ff:fe00:1
SwitchPort: [s1, 2]; MAC:00:00:00:00:00:02; IP:2001:db1::200:ff:fe00:2
SwitchPort: [s1, 2]; MAC:00:00:00:00:00:02; IP:2001:db1::88d7:e53f:3a62:52c3
SwitchPort: [s1, 2]; MAC:00:00:00:00:00:02; IP:fe80::200:ff:fe00:2
```

更新前绑定表

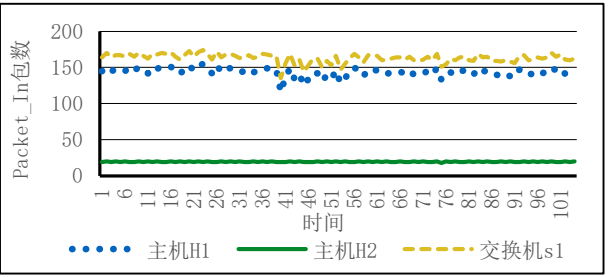
```
=====更新日志======
发现主机离线，SwitchPort：s1, 2
对应主机MAC：00:00:00:00:00:02
```

更新日志

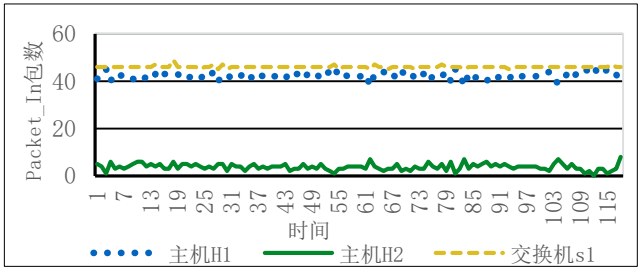
```
=======绑定表========
SwitchPort: [s1, 1]; MAC:00:00:00:00:00:01; IP:fe80::200:ff:fe00:1
```

更新后绑定表

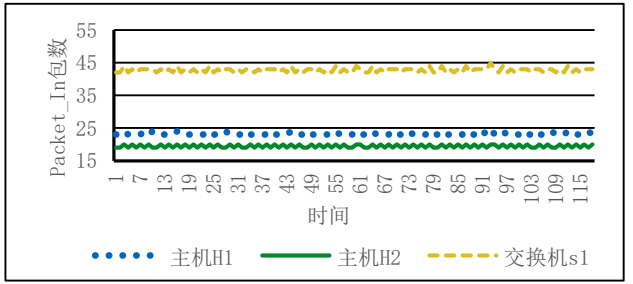
实验结果3



无限速时packet_in报文数量



交换机限速时packet_in报文数量



交换机端口限速时packet_in报文数量

项目开源地址

https://github.com/LouieYi/safeBinding

项目组成员：于俊清，鲁喻，陈清，周启钊

Thank you!