SDN网络

2024/3/15：SDN网络拓扑发现

1. 无法正常给出网络links列表信息：



Solution：

1. 优先启动Mininet拓扑（这将要求你在ryuapp启动时输入额外的参数

--ofp-tcp-listen-port 6653 本地监听端口为6653而非默认的6633）

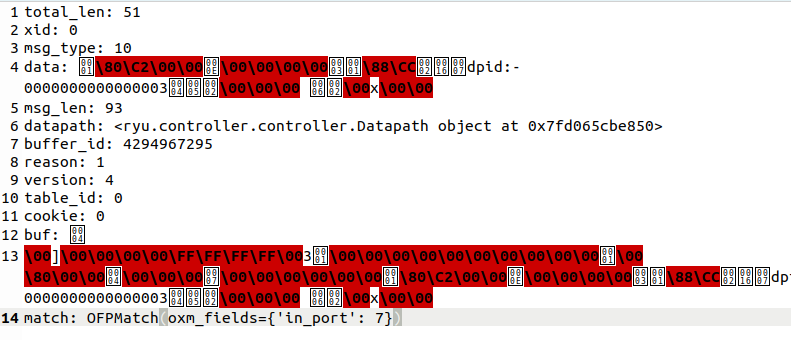
2. 输入额外links参数

Total：

2024/3/16：SDN网络拓扑发现

1. 常见数据类型的属性

**ev.msg:**



buffer\_id: 用来唯一标识 OpenFlow 协议缓存中相关数据包的 ID

data: 原始以太网帧数据

datapath: 表示发送此消息的交换机的 datapath

in\_port: 数据包进入交换机的端口号

match: 表示此数据包对应的匹配字段

reason: 数据包被发送给控制器的原因

total\_len: 表示数据包的总长度

cookie: 与本事件相关的 cookie 值，可以用于关联它与某个流表项

**Switch** (from get\_switch()):

dpid: 交换机的 Datapath ID，这是交换机的唯一标识符。

ports: 交换机上端口的列表

**Port**：

port\_no: 端口号标识，该端口在交换机内的唯一编号。

hw\_addr: 端口的硬件地址，通常是MAC地址。

name: 人类可读的端口名称，如 's1-eth1' 对于一个名为 s1 的交换机上的第一个端口。

config: 端口配置状态，表示某些端口配置的位掩码，如端口是否被管理员关闭。

state: 端口的当前状态，也是位掩码，指示端口状态，如链路状态（上或下）。

curr: 表示端口当前支持的功能，如链路速度和模式（10Mb, 100Mb, 1Gb, 全双工等）。

advertised: 端口声称支持的功能（通过 LLDP 之类的协议）。

supported: 端口硬件支持的功能。

peer: 对等端口告知支持的功能。

curr\_speed: 端口的当前速度（单位是kbps）。

max\_speed: 端口能够支持的最大速度（单位是kbps）

**Link** (from get\_link()):

src: 表示链路源端的 SwitchPort 对象，它通常包含以下字段：

dst: 表示链路目标端的 SwitchPort 对象，它包含的字段与 src 类似：

2024/3/17：SDN网络拓扑发现

资料网站：

1. [软件定义网络(SDN)与OSPF协议的深度解析-百度开发者中心 (baidu.com)](https://developer.baidu.com/article/details/2989170)

[SDN实验（二）：动态变更转发路径 - greyishsong](https://greyishsong.ink/SDN%E5%AE%9E%E9%AA%8C%EF%BC%88%E4%BA%8C%EF%BC%89%EF%BC%9A%E5%8A%A8%E6%80%81%E5%8F%98%E6%9B%B4%E8%BD%AC%E5%8F%91%E8%B7%AF%E5%BE%84/)

1. [Fat-tree：A Scalable, Commodity Data Center Network Architecture 解读-CSDN博客](https://blog.csdn.net/baidu_20163013/article/details/110004560)
2. <https://www.cnblogs.com/qq952693358/p/7354070.html> K条最短路径算法(KSP, k-shortest pathes)：Yen's Algorithm
3. <https://www.sdnlab.com/15140.html> 基于网络流量的SDN最短路径转发应用
4. <https://www.cnblogs.com/jmilkfan-fanguiju/p/11825042.html> 数据中心网络架构的问题与演进 — CLOS 网络与 Fat-Tree、Spine-Leaf 架构
5. [Twilghts/DQN: 利用DQN算法对网络中的数据包进行动态路由，从而获得最佳的服务质量（仿真实验） (github.com)](https://github.com/Twilghts/DQN/tree/master)
6. <https://www.osgeo.cn/networkx/tutorial.html> Networkx中文文档
7. <https://www.cnblogs.com/aleza/p/17757704.html> 04-networkX-查找k短路
8. 链路发现信息：
9. 常态化：LLDP 数据包（用于发现交换机、探查网络拓扑的数据包）

参考：

[ethernet(dst='01:80:c2:00:00:0e',ethertype=35020,src='00:00:00:00:01:04'), lldp(tlvs=[ChassisID(chassis\_id='dpid:0000000000000001',len=22,subtype=7,tlv\_info='\x07dpid:0000000000000001',typelen=534), PortID(len=5,port\_id='\x00\x00\x00\x04',subtype=2,tlv\_info='\x02\x00\x00\x00\x04',typelen=1029), TTL(len=2,tlv\_info='\x00x',ttl=120,typelen=1538), End(len=0,tlv\_info='',typelen=0)])]

（2）请求前：ARP数据包

参考：

[ethernet(dst='ff:ff:ff:ff:ff:ff',ethertype=2054,src='00:00:00:00:00:01'), arp(dst\_ip='192.168.2.1',dst\_mac='00:00:00:00:00:00',hlen=6,hwtype=1,opcode=1,plen=4,proto=2048,src\_ip='192.168.1.1',src\_mac='00:00:00:00:00:01')]

源 MAC 地址为00:00:00:00:00:01，源 IP 地址为192.168.1.1）想要找到具有 IP 地址 192.168.2.1 的设备的 MAC 地址

（3）PING：ICMP数据包

参考：

[ethernet(dst='00:00:00:00:00:12',ethertype=2048,src='00:00:00:00:00:01'), ipv4(csum=33730,dst='192.168.2.1',flags=2,header\_length=5,identification=12948,offset=0,option=None,proto=1,src='192.168.1.1',tos=0,total\_length=84,ttl=64,version=4), icmp(code=0,csum=44180,data=echo(data='|\xfc\xf6e\x00\x00\x00\x00=#\t\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#$%&'()+,-./01234567',id=54033,seq=1),type=8)]

以太网帧（Ethernet frame）:

dst='00:00:00:00:00:12': 目的 MAC 地址，即数据包应发送到的物理设备地址。

src='00:00:00:00:00:01': 源 MAC 地址，即发出数据包的设备的物理地址。

ethertype=2048: 表明有效载荷（payload）是一个 IPv4 数据报，以太网类型为 0x0800 对应的十进制为 2048。

IPv4 数据报（IPv4 Datagram）:

dst='192.168.2.1': 目的 IP 地址。

src='192.168.1.1': 源 IP 地址。

ttl=64: 数据报的生存时间（Time To Live），指在丢弃数据报之前允许通过的网络跳数。

proto=1: 表示承载的是 ICMP 协议。

total\_length=84: IPv4 数据报的总长度（包括头部和数据）。

flags=2: 在这里表示不分片（DF, Don't Fragment）。

ICMP 段（ICMP Segment）:

type=8: ICMP 类型 8，代表 Echo Request（回声请求，用于 ping 命令）。

code=0: 对于 ICMP Echo Request，代码（code）字段为 0。

csum=44180: 校验和（Checksum），用来校验 ICMP 消息的完整性。

id=54033: 标识符，用于区分不同的 ICMP 会话。

seq=1: 序列号，用于标识 ICMP Echo Request 的顺序。

data: 包含实际传输的数据，通常是一些填充字节，不同操作系统和 ping 工具的具体数据内容可能会有所不同。

2024/3/21 : ：SDN复杂网络拓扑

Spine\_leaf架构：无法正常路由

双联路拓扑：（正常）

S4流表信息

root@ubuntu:/home/onism/Desktop# ovs-ofctl dump-flows s4

cookie=0x0, duration=27.772s, table=0, n\_packets=9552, n\_bytes=487152, priority=65535,dl\_dst=01:80:c2:00:00:0e,dl\_type=0x88cc actions=CONTROLLER:65535

cookie=0x0, duration=4.692s, table=0, n\_packets=19, n\_bytes=1078, priority=1,in\_port="s4-eth14",dl\_src=00:00:00:00:00:12,dl\_dst=00:00:00:00:00:01 actions=output:"s4-eth13"

cookie=0x0, duration=4.518s, table=0, n\_packets=4, n\_bytes=392, priority=1,in\_port="s4-eth13",dl\_src=00:00:00:00:00:01,dl\_dst=00:00:00:00:00:12 actions=output:"s4-eth14"

cookie=0x0, duration=28.904s, table=0, n\_packets=4354, n\_bytes=477518, priority=0 actions=CONTROLLER:65535

叶脊拓扑：（无法通路）

S3流表信息

root@ubuntu:/home/onism/Desktop# ovs-ofctl dump-flows s3

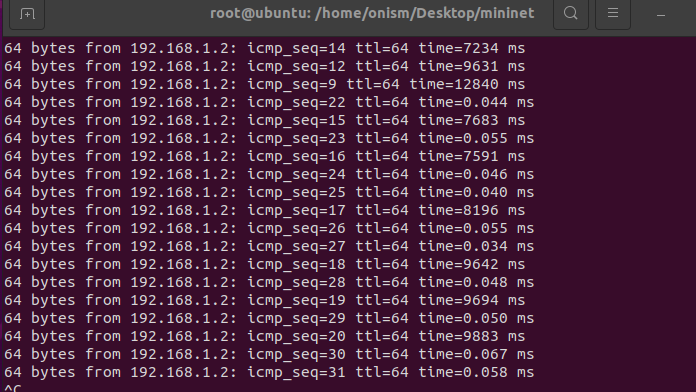
cookie=0x0, duration=75.033s, table=0, n\_packets=15839, n\_bytes=807789, priority=65535,dl\_dst=01:80:c2:00:00:0e,dl\_type=0x88cc actions=CONTROLLER:65535

cookie=0x0, duration=76.769s, table=0, n\_packets=28377, n\_bytes=3467464, priority=0 actions=CONTROLLER:65535

网络状态整理：

1. 网络拓扑（正常）
2. 交换机ID（正常）
3. OSP计算（正常）
4. 流表交互（失效）
5. restAPI调用（失效）

考虑到环路风暴问题，大量的lldp报文影响到控制器对交换机的交互，对报文处理加入忽略lldp协议



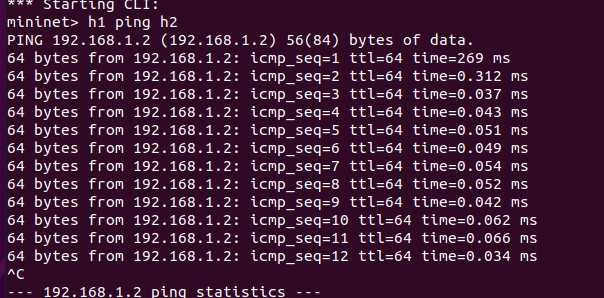
勉强可以ping通

存在大量的MDNS报文

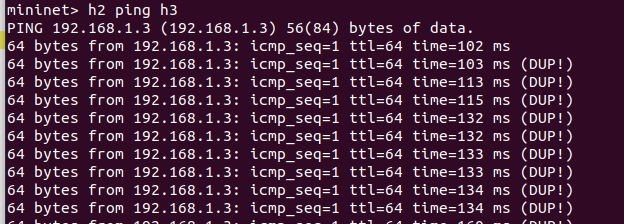
图形用户界面, 文本, 应用程序

描述已自动生成

进一步限制报文



发现最初报文传输稳定  
重启控制器再进行ping命令



h2与h3之间传输大量重复ICMP信息（DUP），确定为环路问题