

# 360的网络运维自动化之路



360 互联网安全中心

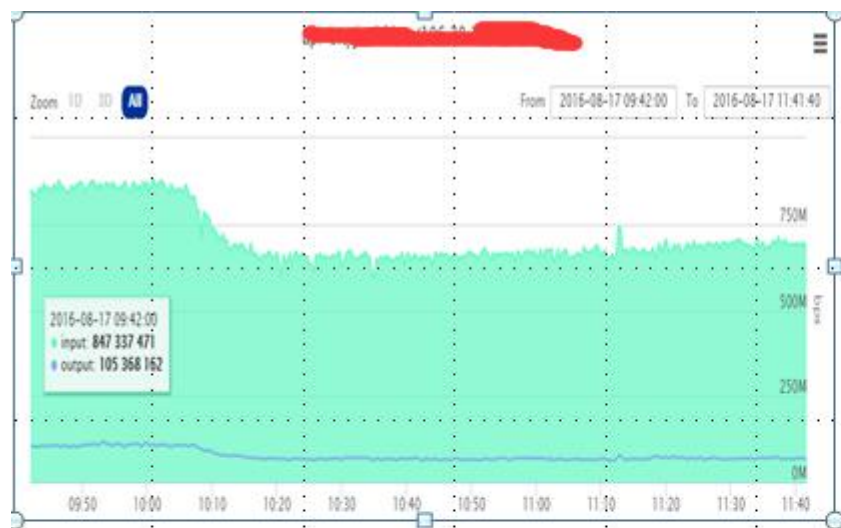
# 大纲

- 1、软件在奇虎网络中发挥的作用
- 2、奇虎在硬件设备上的VXLAN实践
- 3、“SDN”

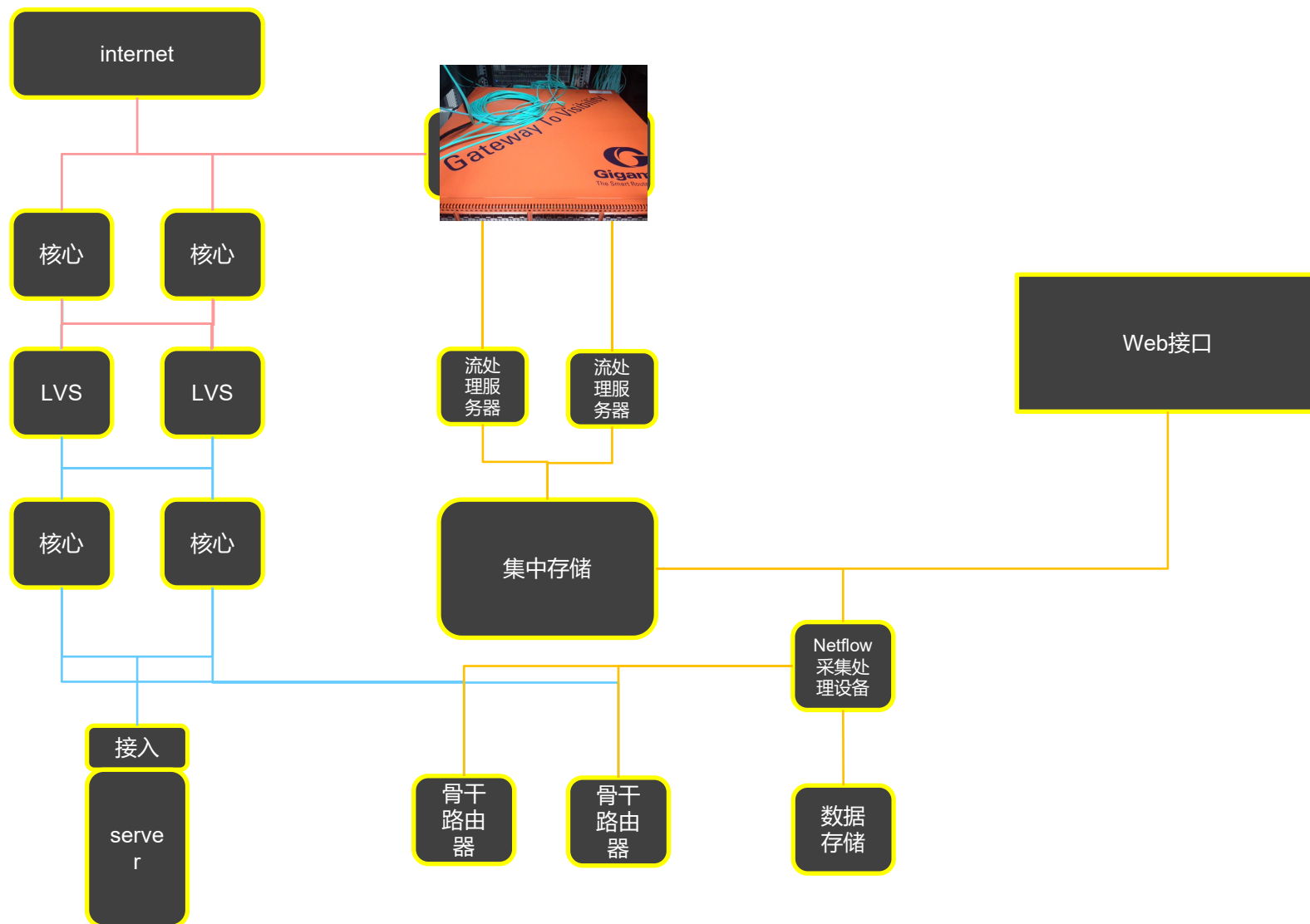
# 出口流量监控



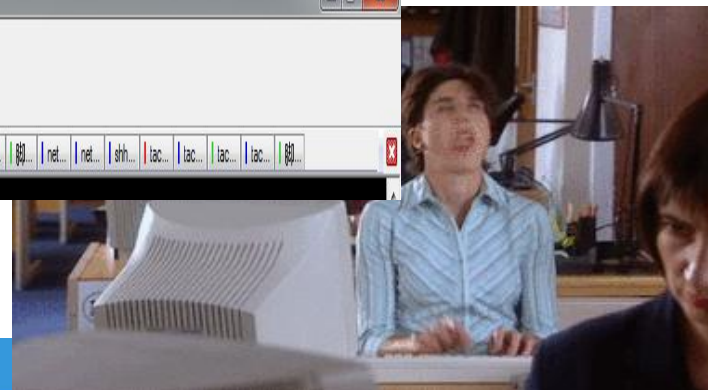
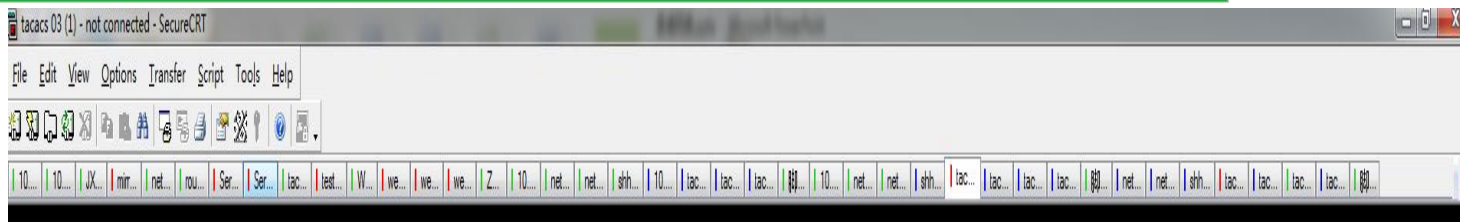
360  
互联网安全中心



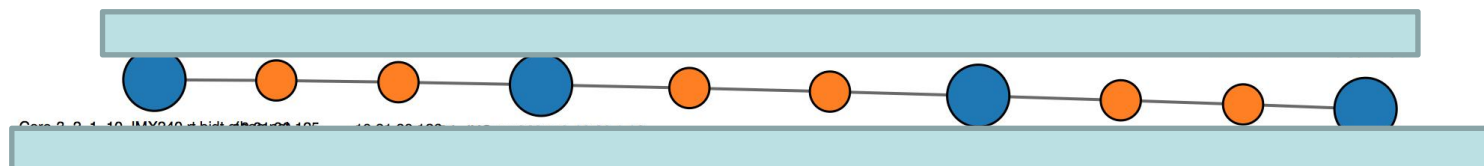
# 出口流量监控



# 主机间故障排查



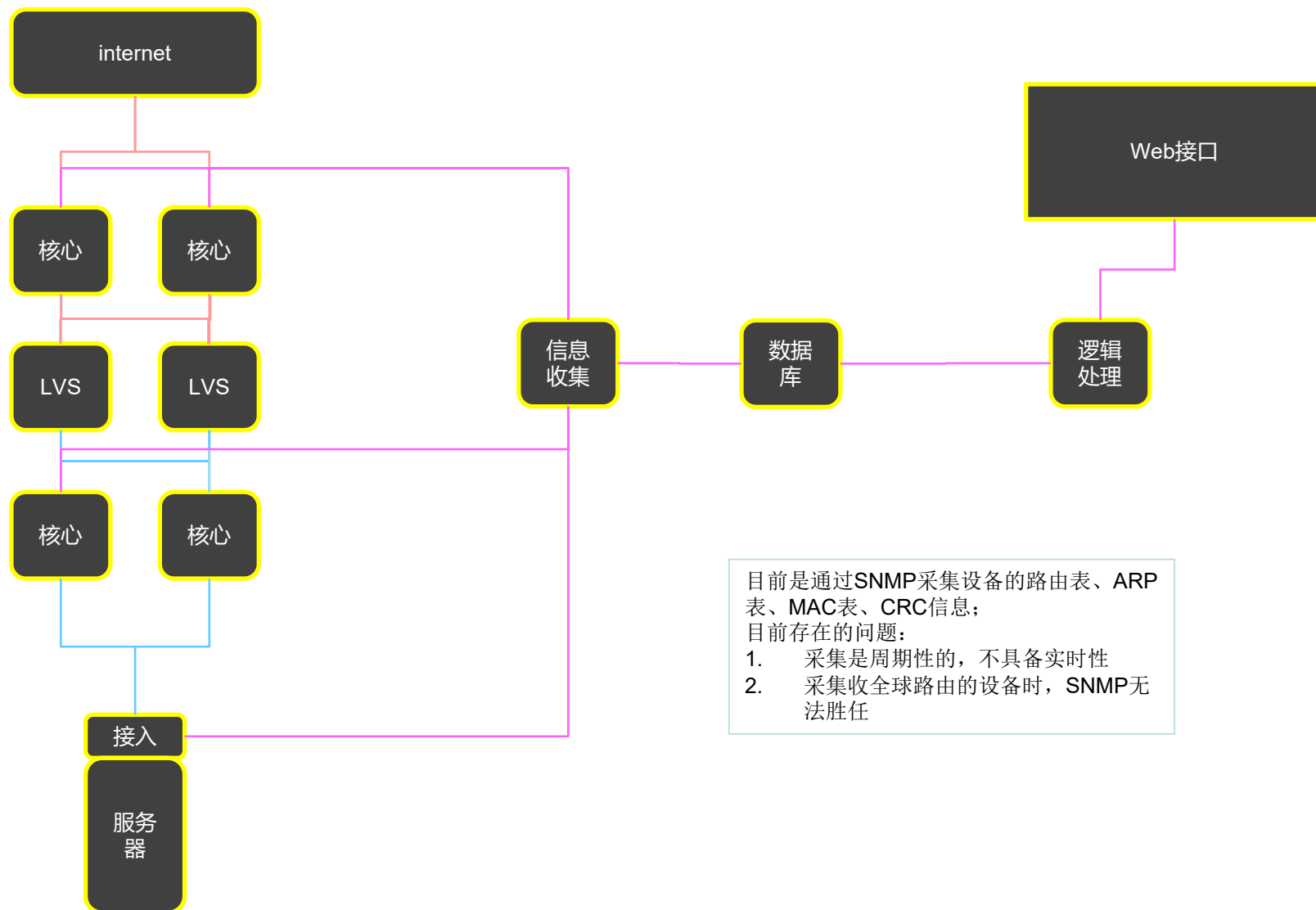
new-bjdt-to-gz2\_2



new- 14



# 主机间故障排查

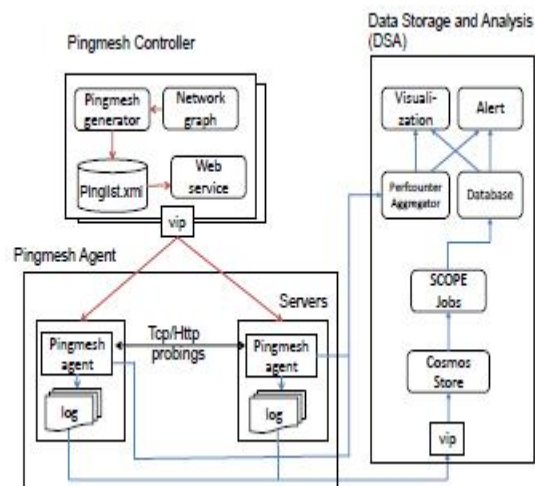


局限性：

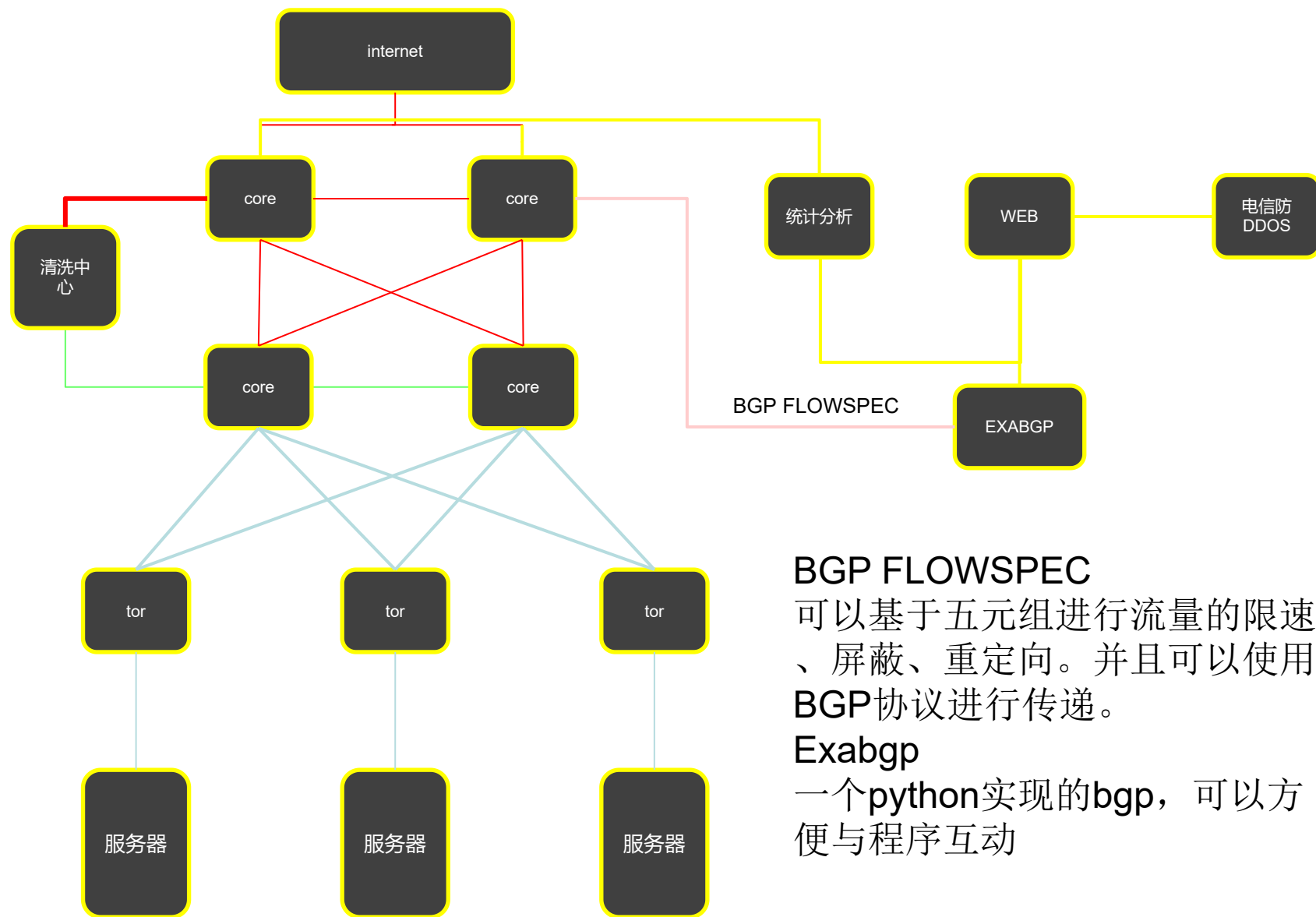
1. 覆盖范围不够广，有时候不能说明问题
2. 不能发现交换机丢包问题。
  - a. parity error造成的特定源IP和目标IP之间不通
  - b. 特定业务流不通
  - c. Tail-drop

解决方法：

参照微软的pingmesh方法



# 应对DDOS



## BGP FLOWSPEC

可以基于五元组进行流量的限速、屏蔽、重定向。并且可以使用BGP协议进行传递。

## Exabgp

一个python实现的bgp，可以方便与程序互动



# 我们学到了什么



1. 网络可以借助ES,HADOOP, STORM等大数据分析应用，挖掘更多有用的东西
2. 与设备交互真的不容易

python管理设备的方法		
设备管理方法	管理方法	数据格式
telnetlib	telnet	字符串
pexpect	telnet/ssh	字符串
Paramiko	ssh	字符串
ansible	ssh	字符串
SNMP	snmp	snmp格式
netconf	ssh、ssl等	xml、yang结构化数据

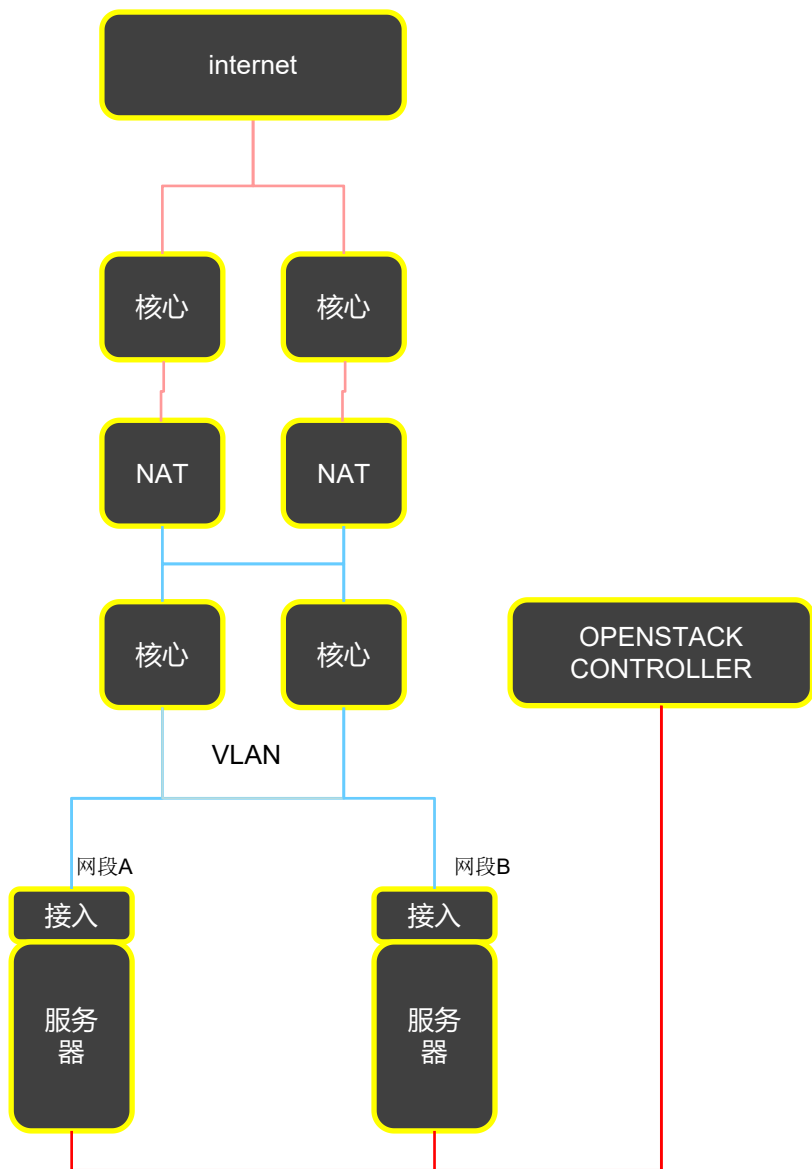
## OpenConfig

OpenConfig is an informal working group of network operators sharing the goal of moving our networks toward a more dynamic, programmable infrastructure by adopting software-defined networking principles such as declarative configuration and model-driven management and operations.

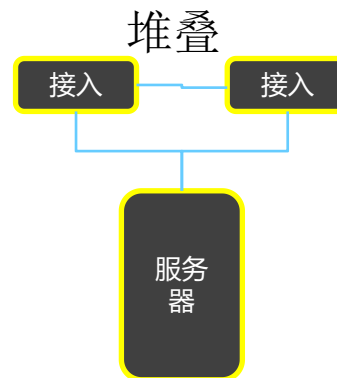
## Streaming Telemetry

Streaming telemetry is a new paradigm for network monitoring in which data is streamed from devices continuously with efficient, incremental updates. Operators can subscribe to the specific data items they need, using OpenConfig data models as the common interface.

# 奇虎私有云OPENSTACK变迁-vlan模式

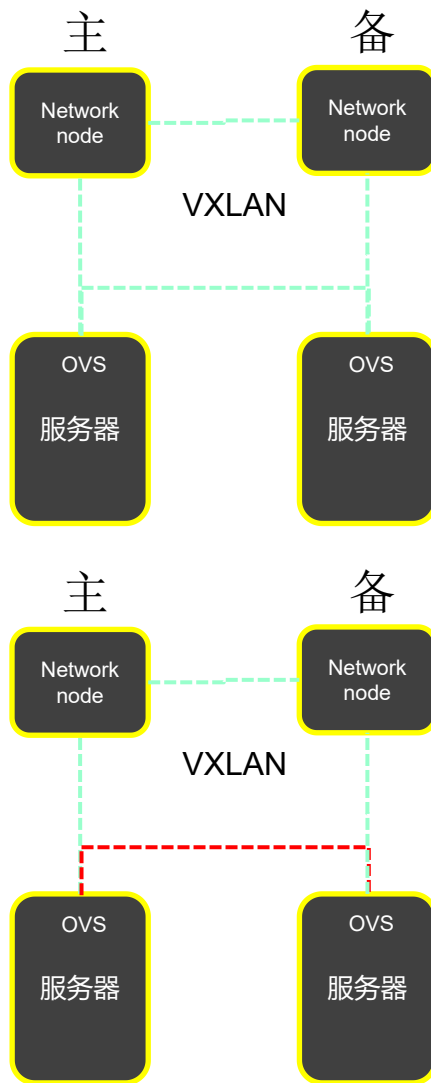
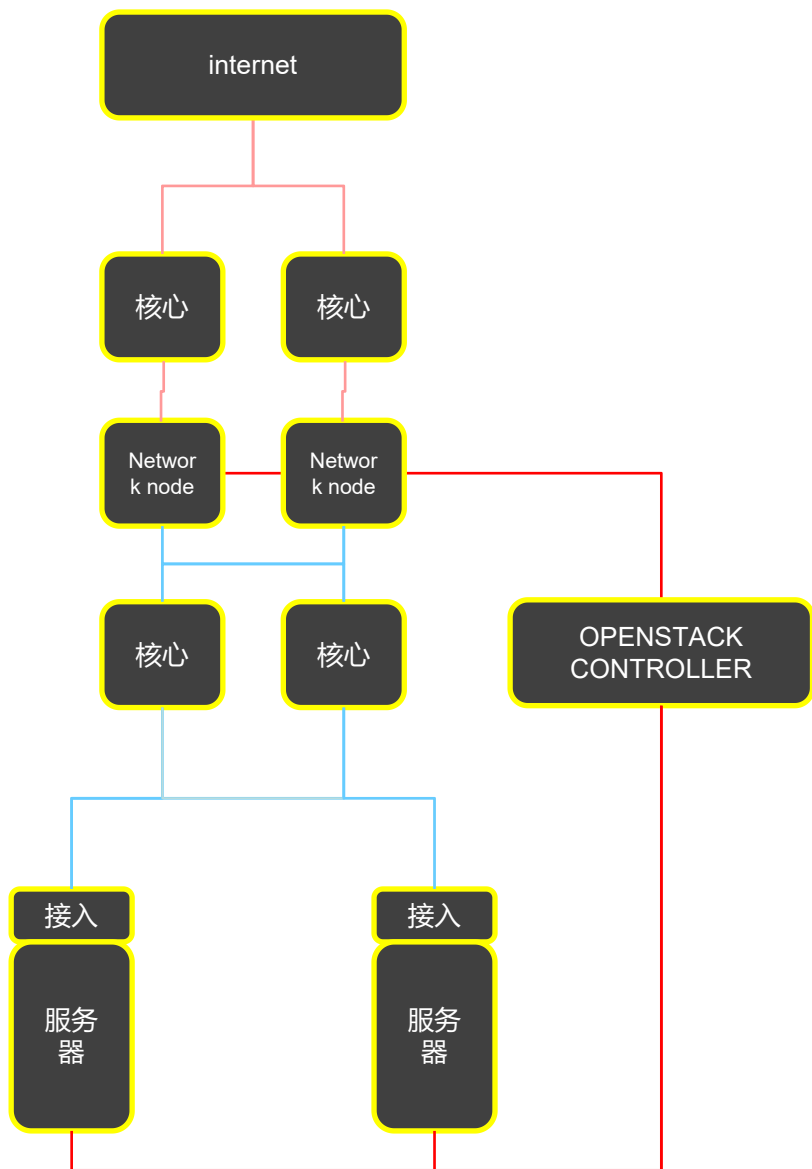


问题一：冗余性需要通过堆叠解决



问题二：虚拟机不能热漂移

# 奇虎私有云OPENSTACK变迁-原生VXLAN



1. Network Node是瓶颈点

2. OVS VXLAN东西向流量的性能瓶颈

# 如何解决OVS在VXLAN上的性能问题



迁移OVS到DPDK版本

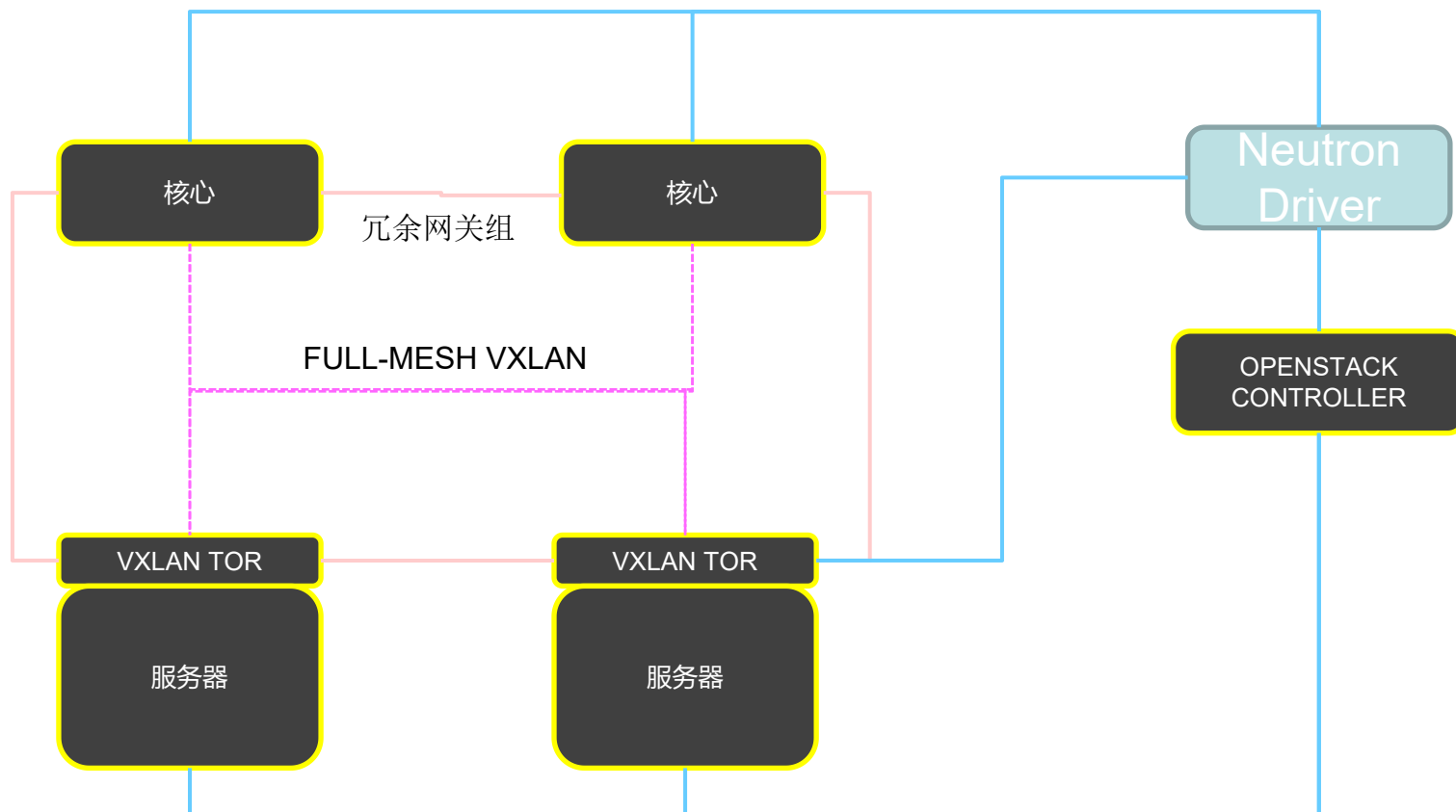


升级网卡

1. 卸载vxlan的checksum
2. 卸载vxlan封装和解封装

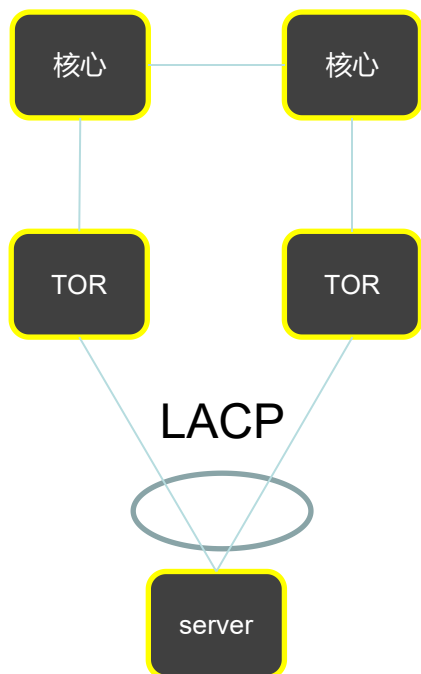


将VXLAN功能卸载到TOR

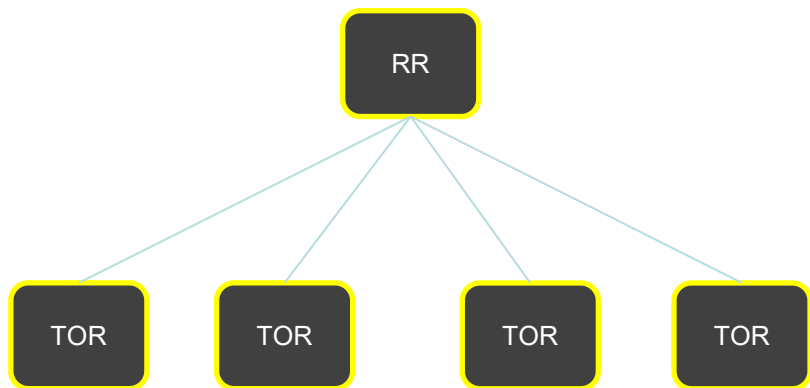


验证了VXLAN和OPENSTACK结合的可行性

# EVPN带来的好处

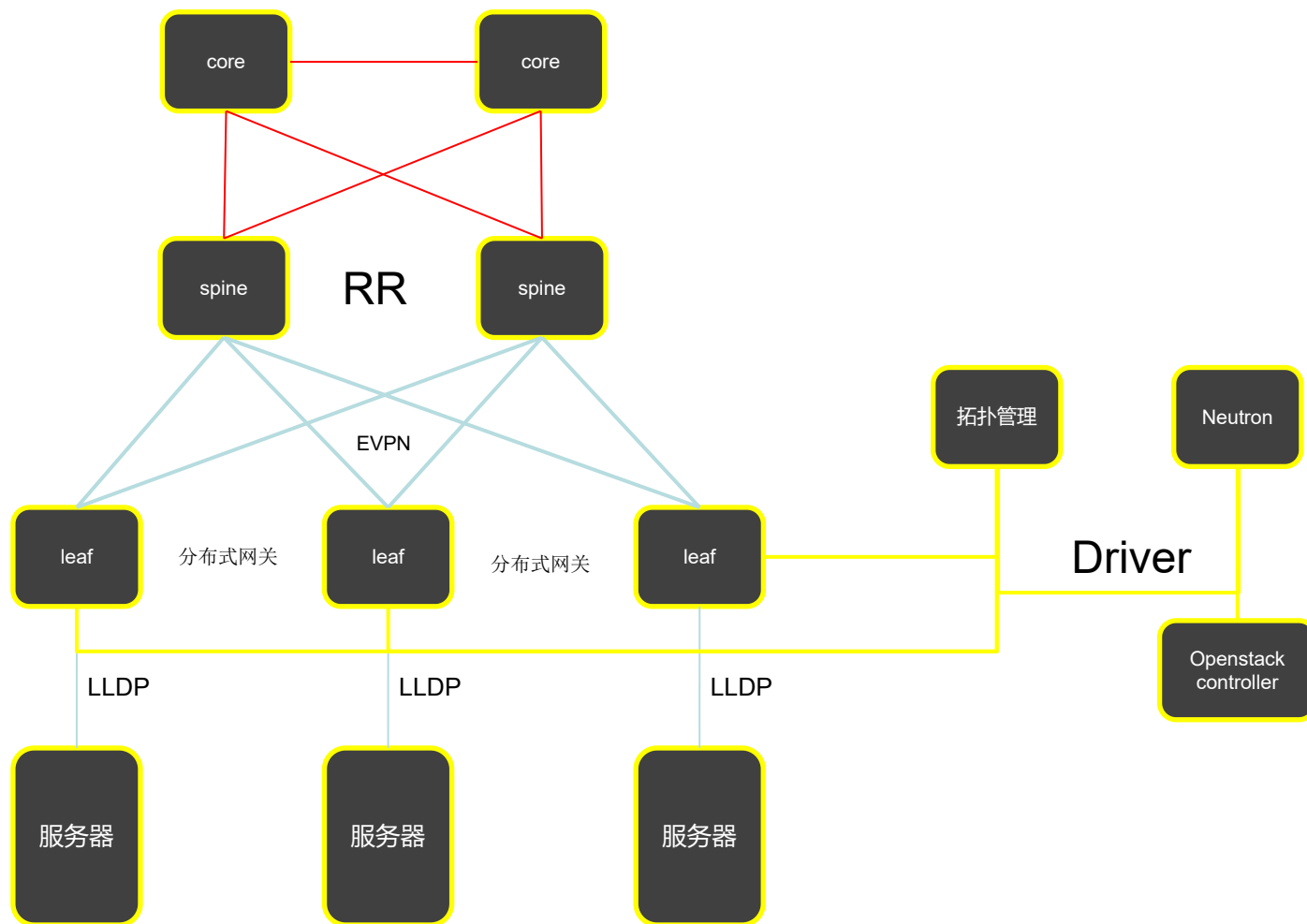


TOR层面无需再使用堆叠技术即可实现接入层的冗余。  
使用EVPN的ESI(Ethernet Segment Identifier)标识



使用BGP为控制协议，可以使用BGP的RR设计做自动发现，不用再配置full-mesh的静态配置

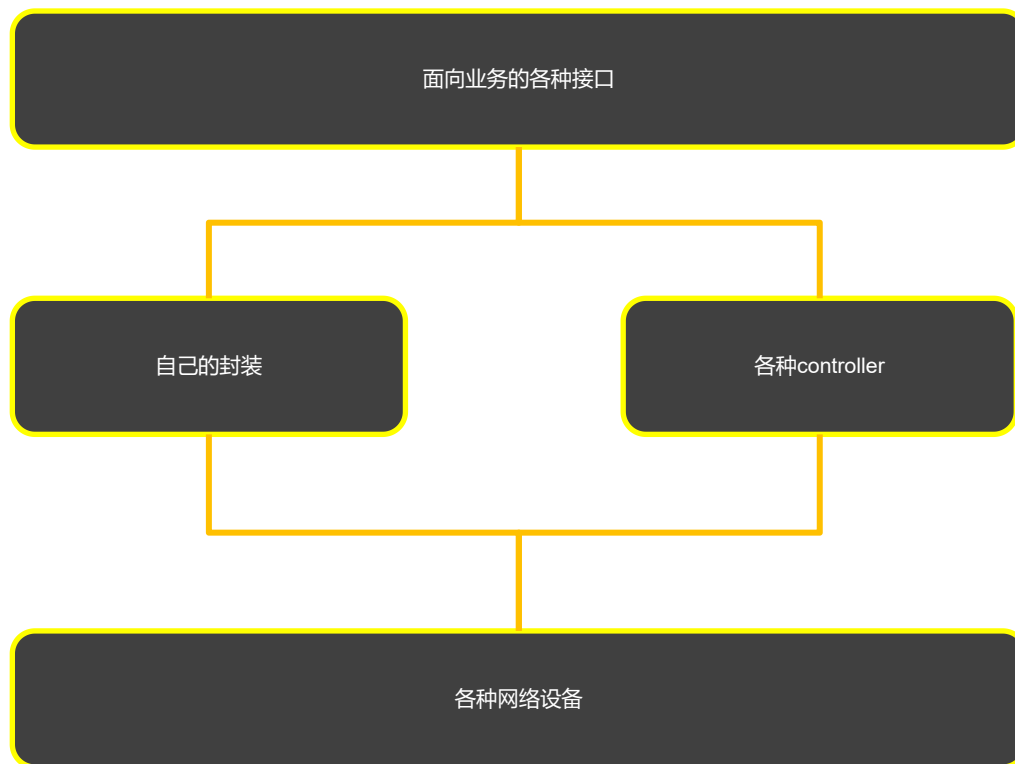
# 正式上线的Openstack拓扑





# 从另外一个角度看SDN





将网络从服务向产品转型。以后面向业务不再提供人工服务，而是提供接口服务。

# 技能转变



```
[mahongwei@ ~]$  
[mahongwei@ ~]$  
[mahongwei@ ~]$  
[mahongwei@ ~]$  
[mahongwei@ ~]$  
[mahongwei@ ~]$ telnet 1.1.1.1
```



```
伪代码  
def connect():  
    con = netconf.con('1.1.1.1')  
    return con
```