阅读笔记

L1

E2E Arg (Saltzer, 1984)

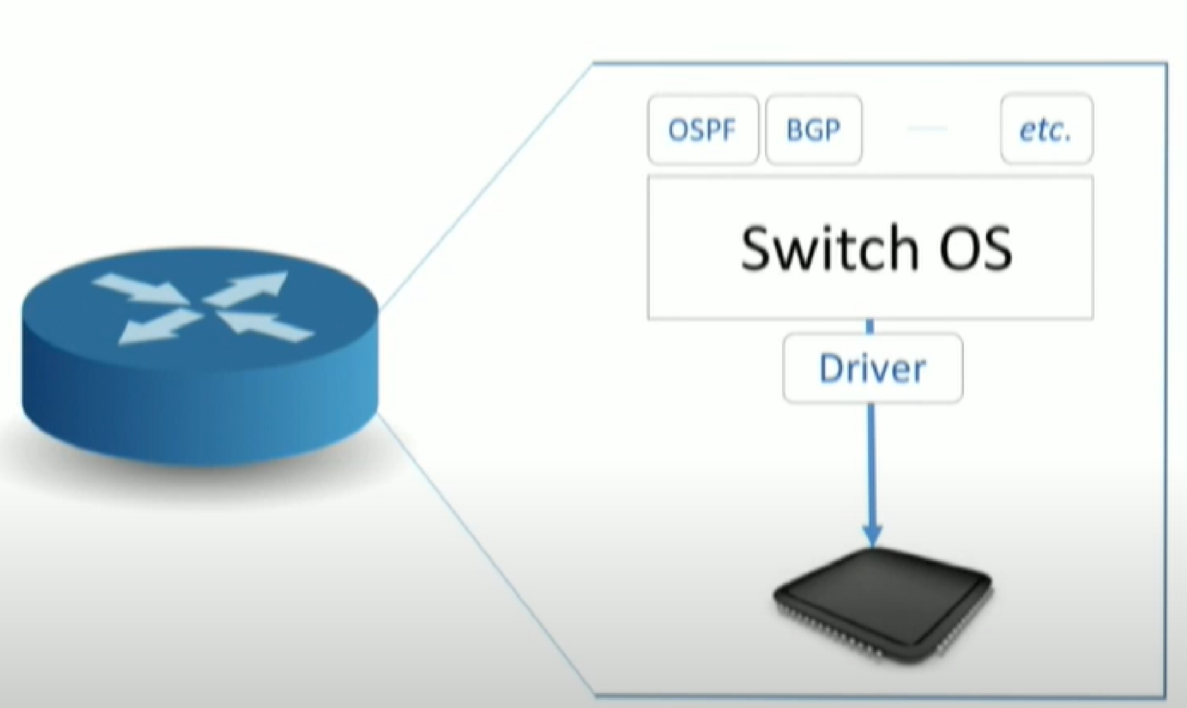
导读：https://blog.acolyer.org/2014/11/14/end-to-end-arguments-in-system-design/

OVS

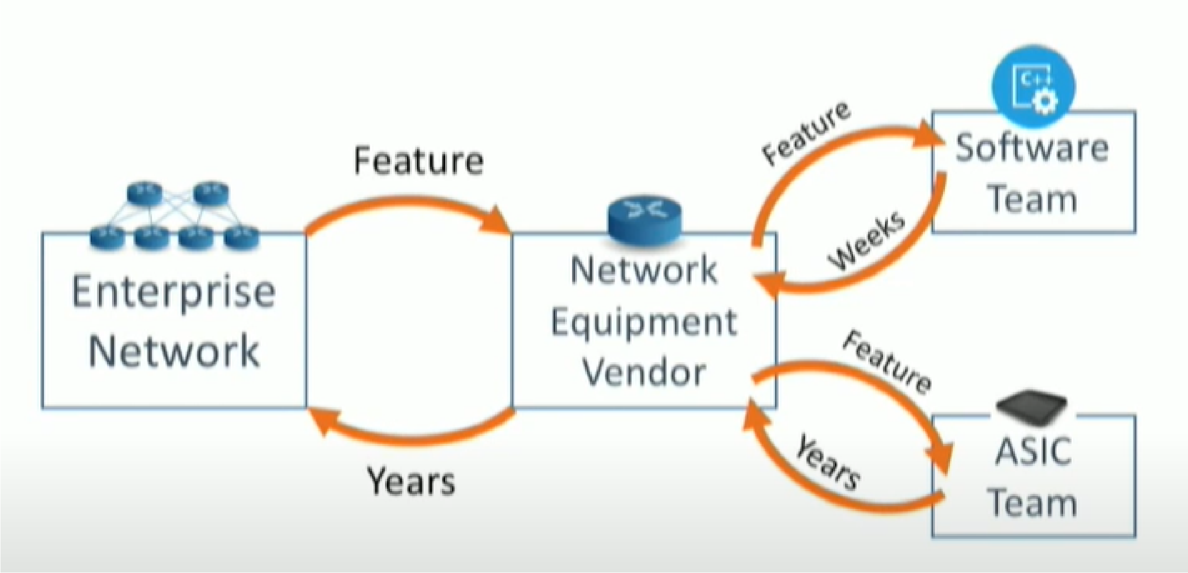
```
试图删除所有queue时提示仍有reference
解决：先清空QoS table
sudo ovs-vsctl -- --all destroy QoS
sudo ovs-vsctl -- --all destroy Queue

试图为新建的br0添加eth0提示出错
解决：需要设置为内部类型
ovs-vsctl add-port br0 port0 -- set Interface port0 type=internal
ref: https://github.com/openvswitch/ovs-issues/issues/110

add flow template:
sudo ovs-ofctl add-flow S3
"priority=10,in_port=3,dl_type=0x0800,nw_src=10.0.0.3,nw_dst=10.0.0.6,actions=set_queue:123,output:1" --protocol=OpenFlow13

show flow template:
sudo ovs-ofctl dump-flows S2 --protocol=OpenFlow13
```

video

2:30 - why is it take so long to add new features to network equipments



4:00 - 5:00 bottleneck of adding new features



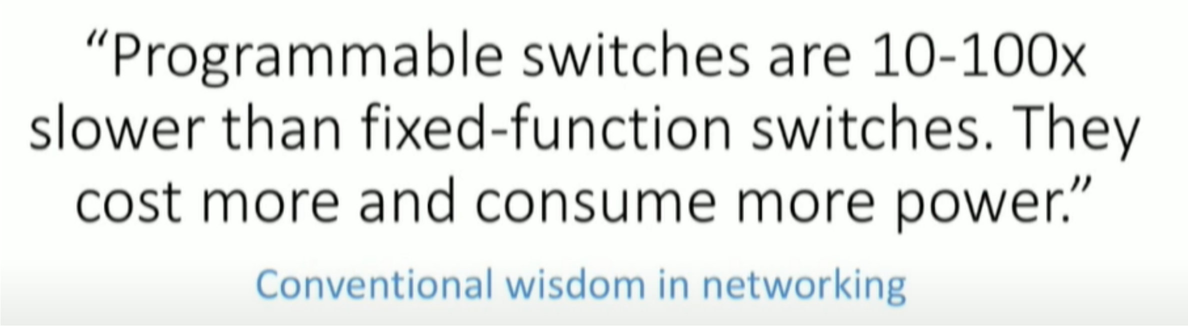
6:20 network systems are buit bottoms-up

(new feature implemented under constraint from chip datasheets)

7:50 newtorks will be programmed top-down

(programmable switches)

8:35 but why aren't all systems built this way?

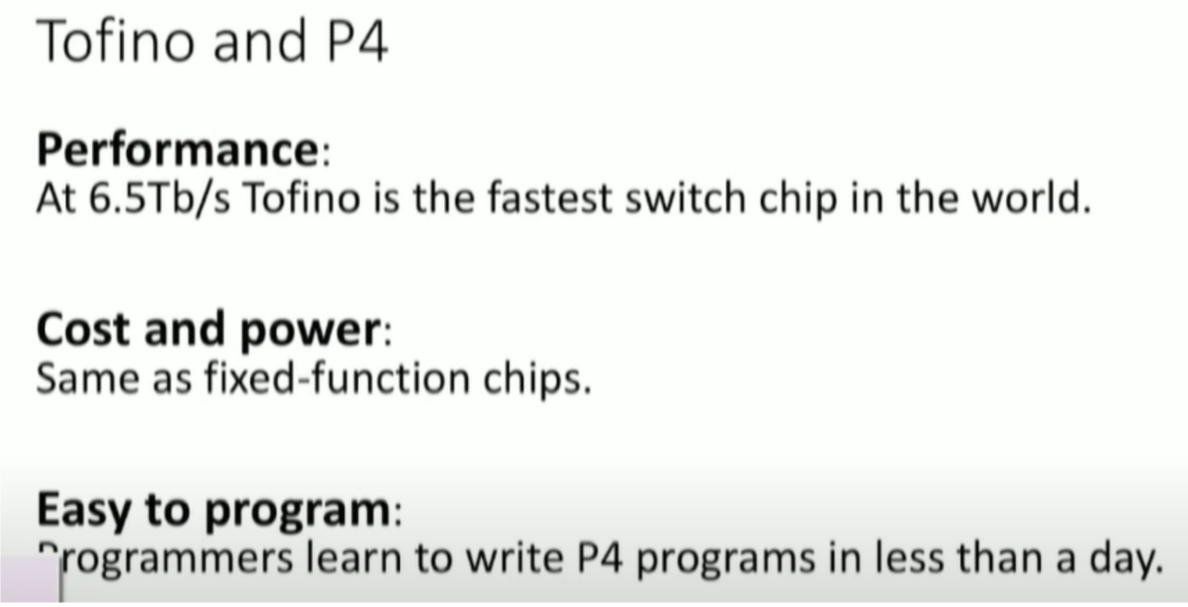


"Programmable switches are 10-100x slower than fixed-function switches. They cost more and consume more power."

Conventional wisdom in networking

today this is not true!

9:23 comparison



Tofino and P4

Performance:
At 6.5Tb/s Tofino is the fastest switch chip in the world.

Cost and power:
Same as fixed-function chips.

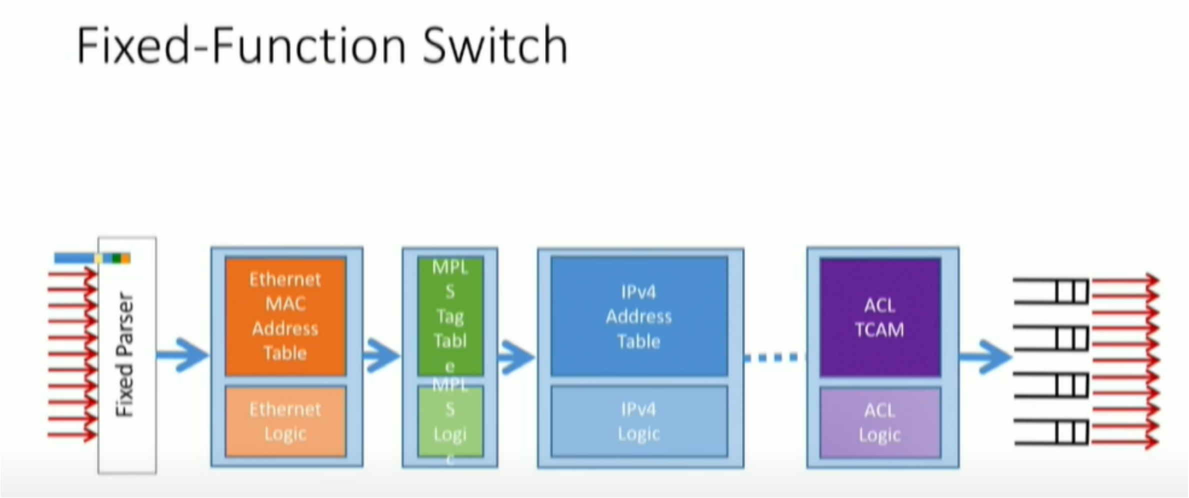
Easy to program:
Programmers learn to write P4 programs in less than a day.

12:00 domain specific processors

13:40 PISA(protocol independent switch architecture)

14:40 question from audience: networking is relatively small market

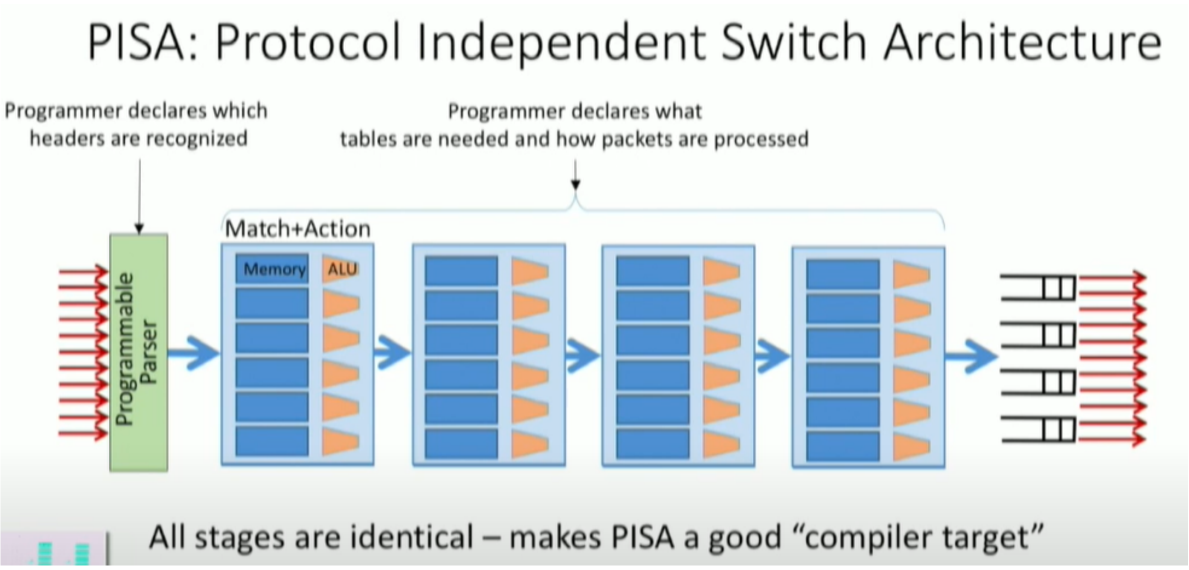
16:00 fixed function switch



Fixed-Function Switch

18:00-20:00 question from audience: how could programmable switches become faster? - because of intelligent abstraction of instructions

20:00 PISA



PISA: Protocol Independent Switch Architecture

Programmer declares which headers are recognized

Programmer declares what tables are needed and how packets are processed

All stages are identical – makes PISA a good "compiler target"

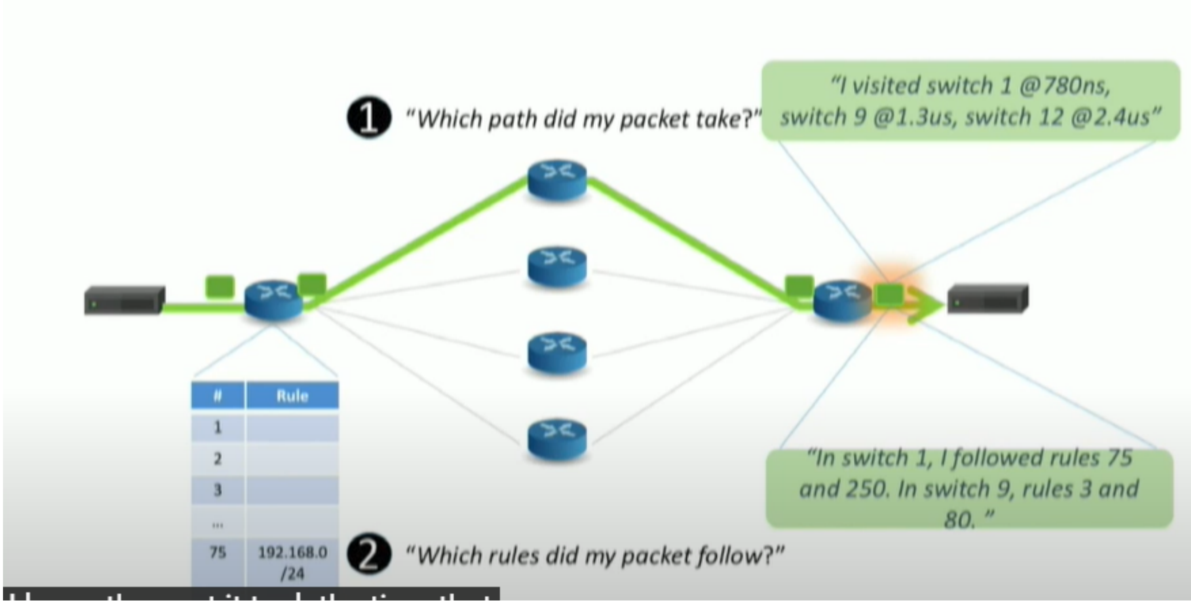
23:00 quesiton from audience: we only look at headers instead of the payload (L3 and L4 instead of L7)? - yes. it's a trade off

23:00-26:00 some other questions on table designing and physical limitations

28:09 the killer app

29:00 embarassing state of measurement in networking

30:00 a smarter way of debuging



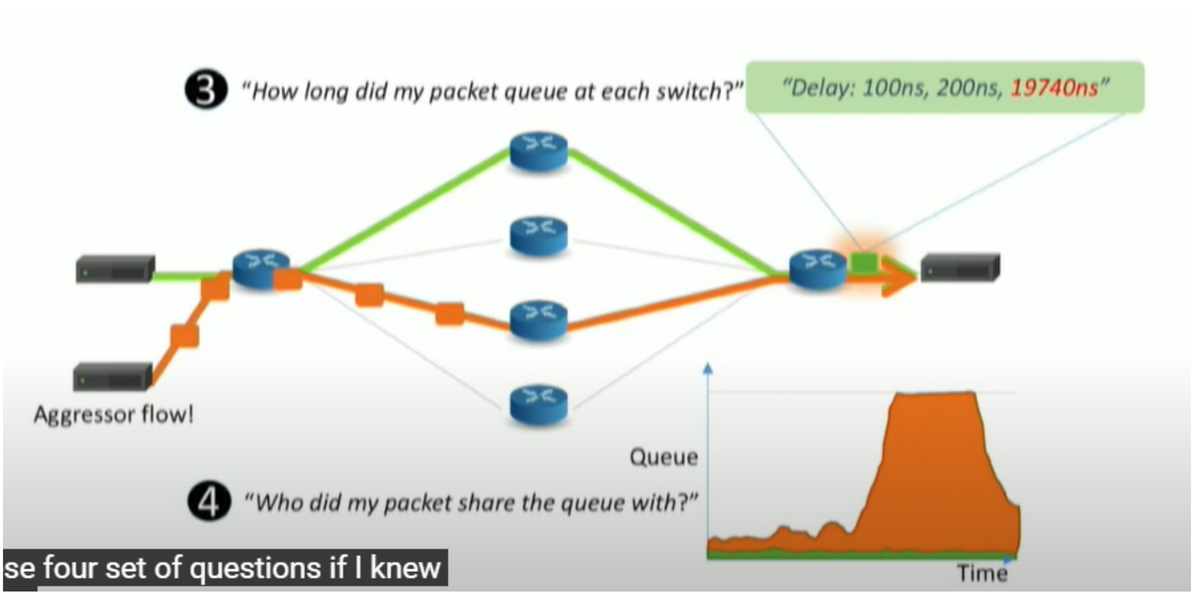
1 "Which path did my packet take?"

"I visited switch 1 @780ns, switch 9 @1.3us, switch 12 @2.4us"

#	Rule
1	
2	
3	
...	
75	192.168.0 /24

2 "Which rules did my packet follow?"

"In switch 1, I followed rules 75 and 250. In switch 9, rules 3 and 80. "



3 "How long did my packet queue at each switch?"

"Delay: 100ns, 200ns, 19740ns"

4 "Who did my packet share the queue with?"

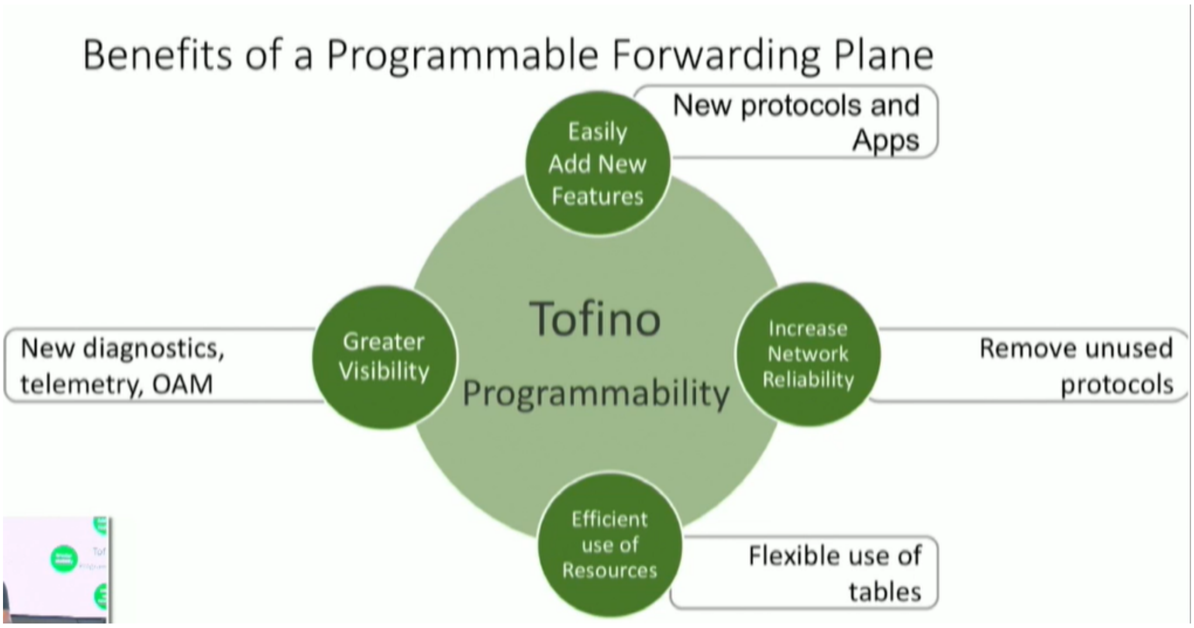
se four set of questions if I knew

32:30 the 4 questions above are what the network should answer when debugging...

but no network can do that today

but Tofino and P4 can

33:00 benefits of programmable forwarding plane



Benefits of a Programmable Forwarding Plane

33:30+ Q&A

Fat Tree

每个 switch 有 k 个端口。核心层 switch 为 `(k/2)**2` 个，Pod 有 k 个，每个 Pod 包含 (k/2) 个汇聚层 switch 和 (k/2) 个边缘层 switch，Pod 内每个汇聚层 switch 与每个边缘层 switch 连接（各自消耗 k/2 个端口）。每个汇聚层 switch 剩余 k/2 个端口与 k/2 个核心层 switch 相连，于是每个 Pod 都有 `(k/2)**2` 个端口与核心层 switch 相连，各个端口恰连接各不相同的核心层 switch。每个边缘层的剩余端口都用于连接 host，共能连接 k*(k/2)*(k/2) 个 host。

```
# of port per switch = k
# of pod = k
# of switch = k**2 + (k/2)**2 = k**2 * 5/4
# of server = k*(k/2)*(k/2) = k**3 /4
```

REF: https://www.cnblogs.com/zhuting/p/8880475.html