SDN

[1] , , 2005
[2] , , 2008
[3]Research in Anti-Spam Method Based on Bayesian Filtering PACIIA2008
[4]Improvement of Chinese Spam filtering method based on Bayesian Classification ICFCC2010
[5]New intrusion detection method based on partition CECN2014
[6]KDDCUP99 2014

[7]An effective flame Segmentation method based on OHTA color space ISAM2012
[8]Research on Background Subtraction Based on Center-symmetric Local Binary Patterns IMSNA2013
[9] 2014
[10] 2015
[11] Vibe 2017

[12] 2017

一、什么是SDN

二、SDN应用

三、实验环境

SDN

(Software Defined Network, SDN)

2006 Nick McKeown Clean Slate Openflow

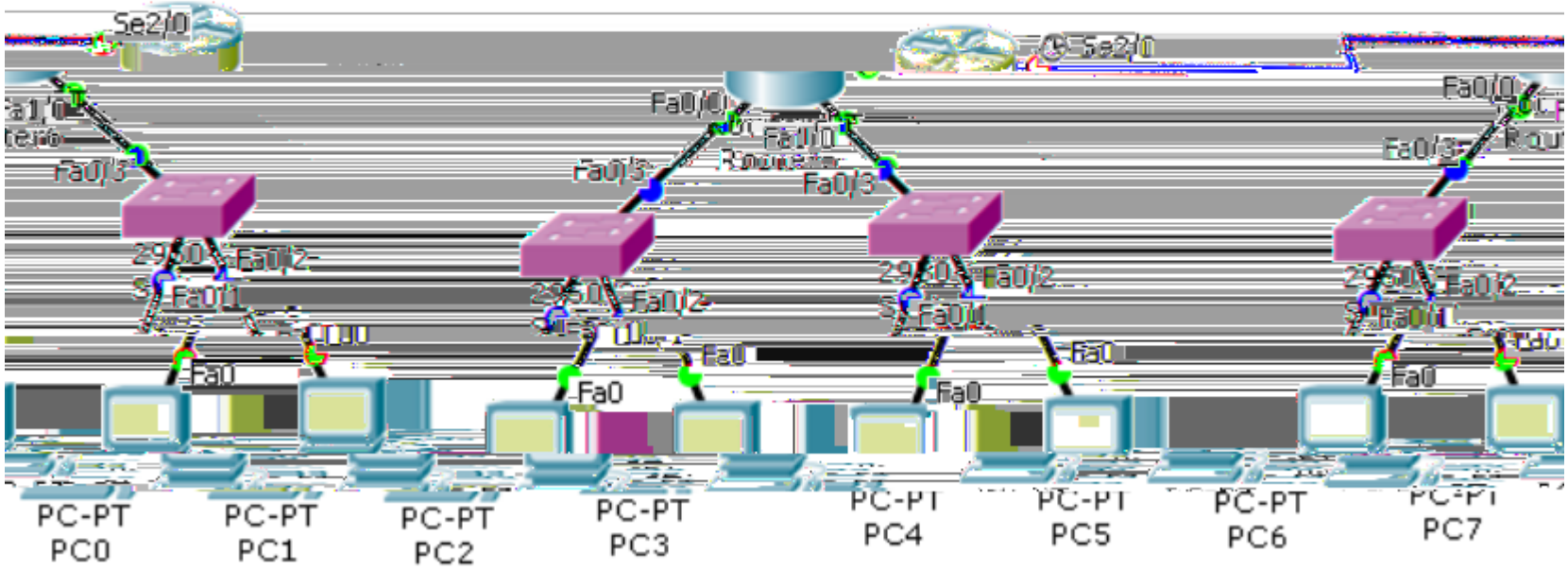
2007 Martin Casado Ethane

2008 Nick OpenFlow ACM SIGCOMM OpenFlow: Enabling Innovation in Campus Networks”

2012 863”

2012 SDN

SDN ”


Se2/0
Fa0/3
Fa0/2
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3
PC-PT
PC4
PC-PT
PC5
PC-PT
PC6
PC-PT
PC7

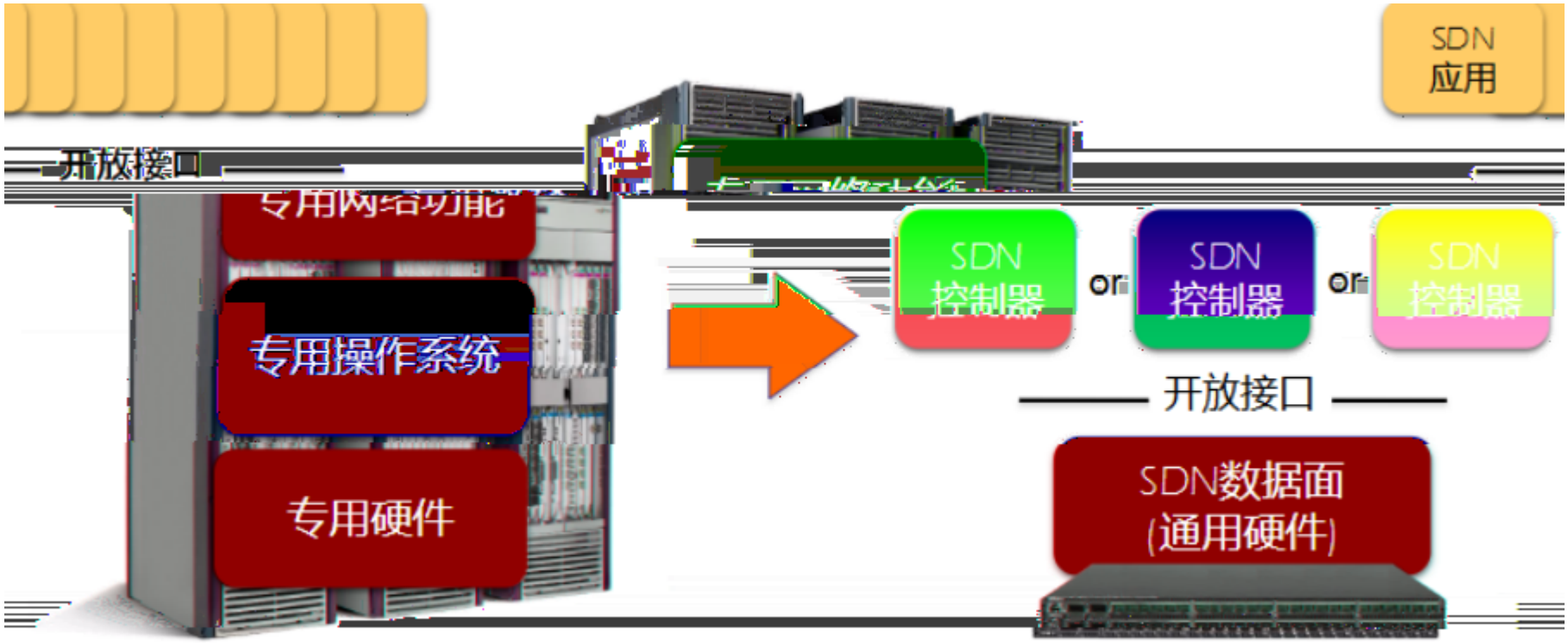
网络功能

厂商操作系统

定制化硬件

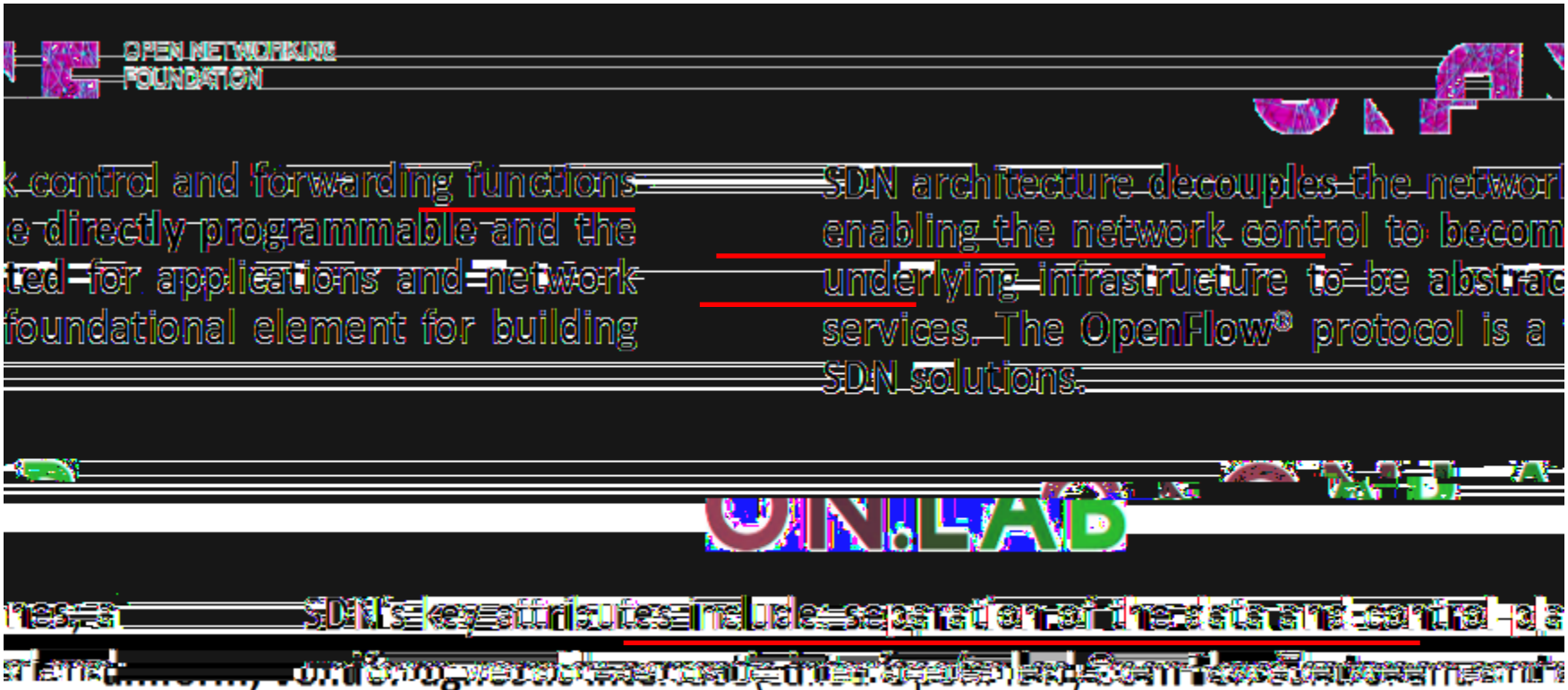
负责　　闭　统　　业创

传统　络　SDN


SDN
应用
开放接口
专用网络功能
专用操作系统
专用硬件
SDN
控制器
or
SDN
控制器
or
SDN
控制器
开放接口
SDN数据面
(通用硬件)

闭　统　　统

SDN 定义



data planes; **a logically centralized control plane** that offers a consistent, network-wide view to programmers or operators.

SDN 读

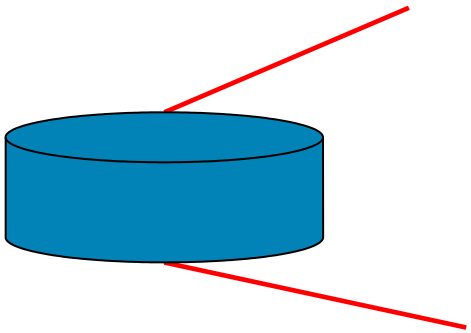
网络开放可编程

数控分离

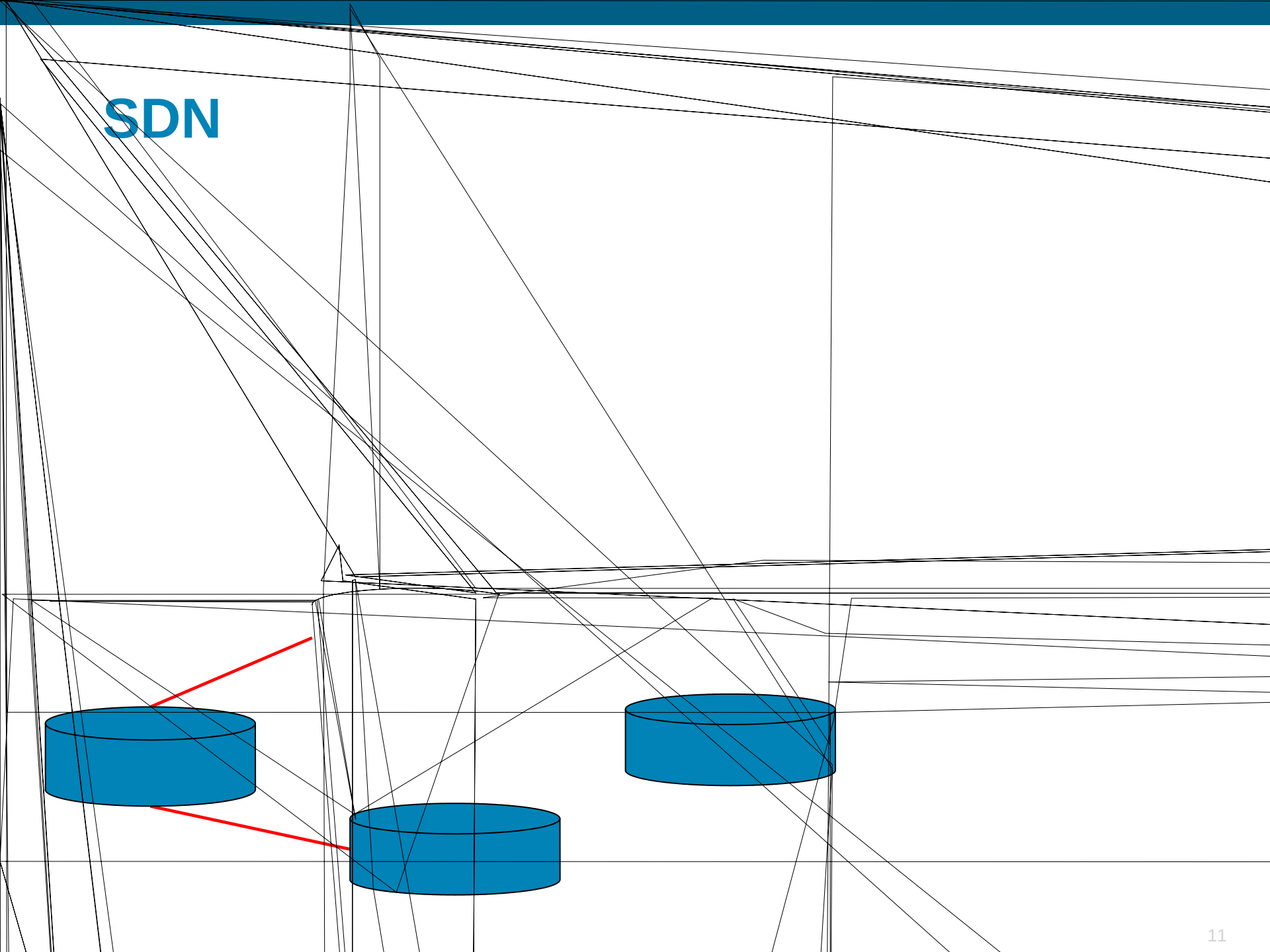
逻辑上集中控制

络 结

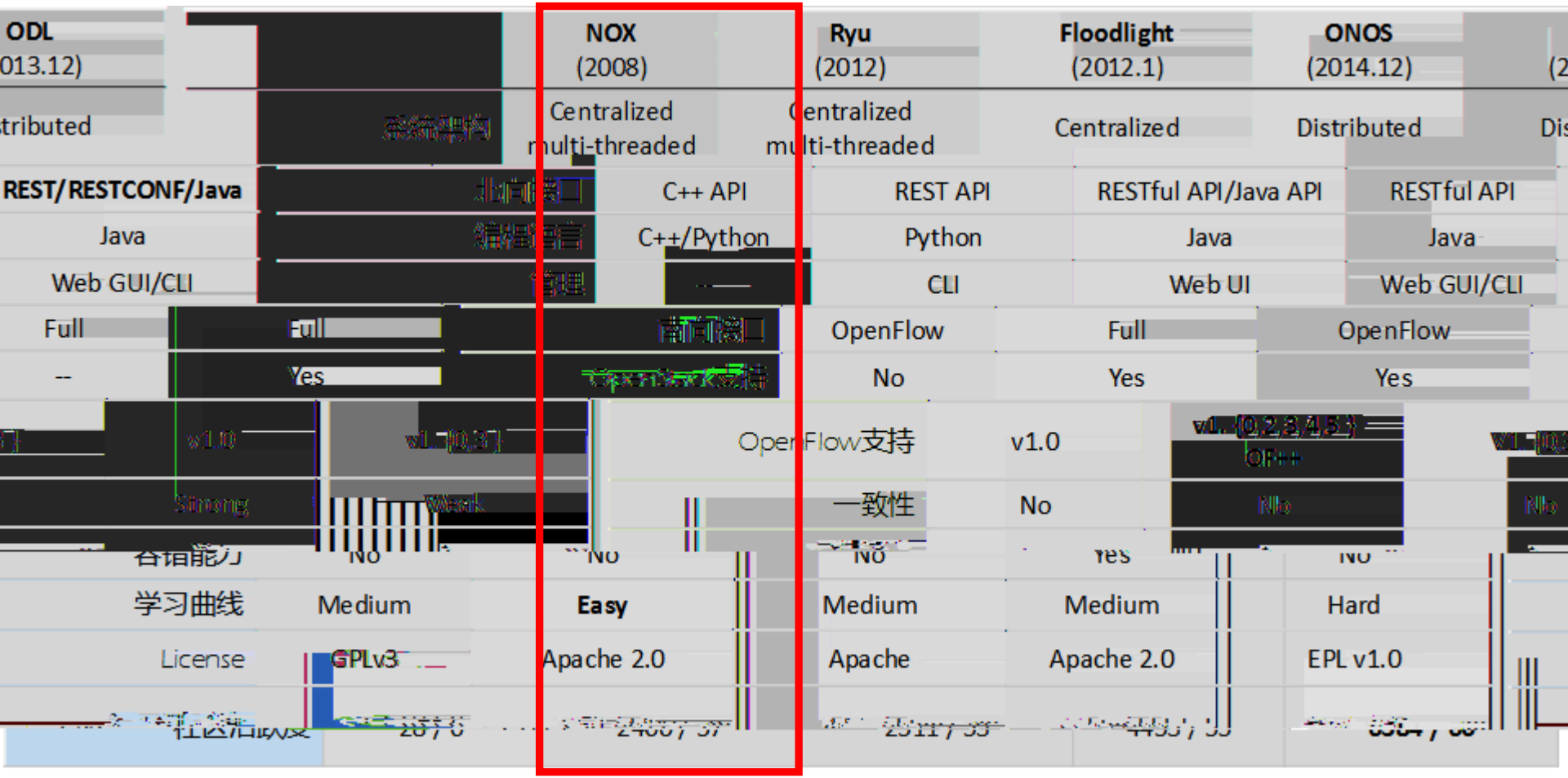
SDN 层 结



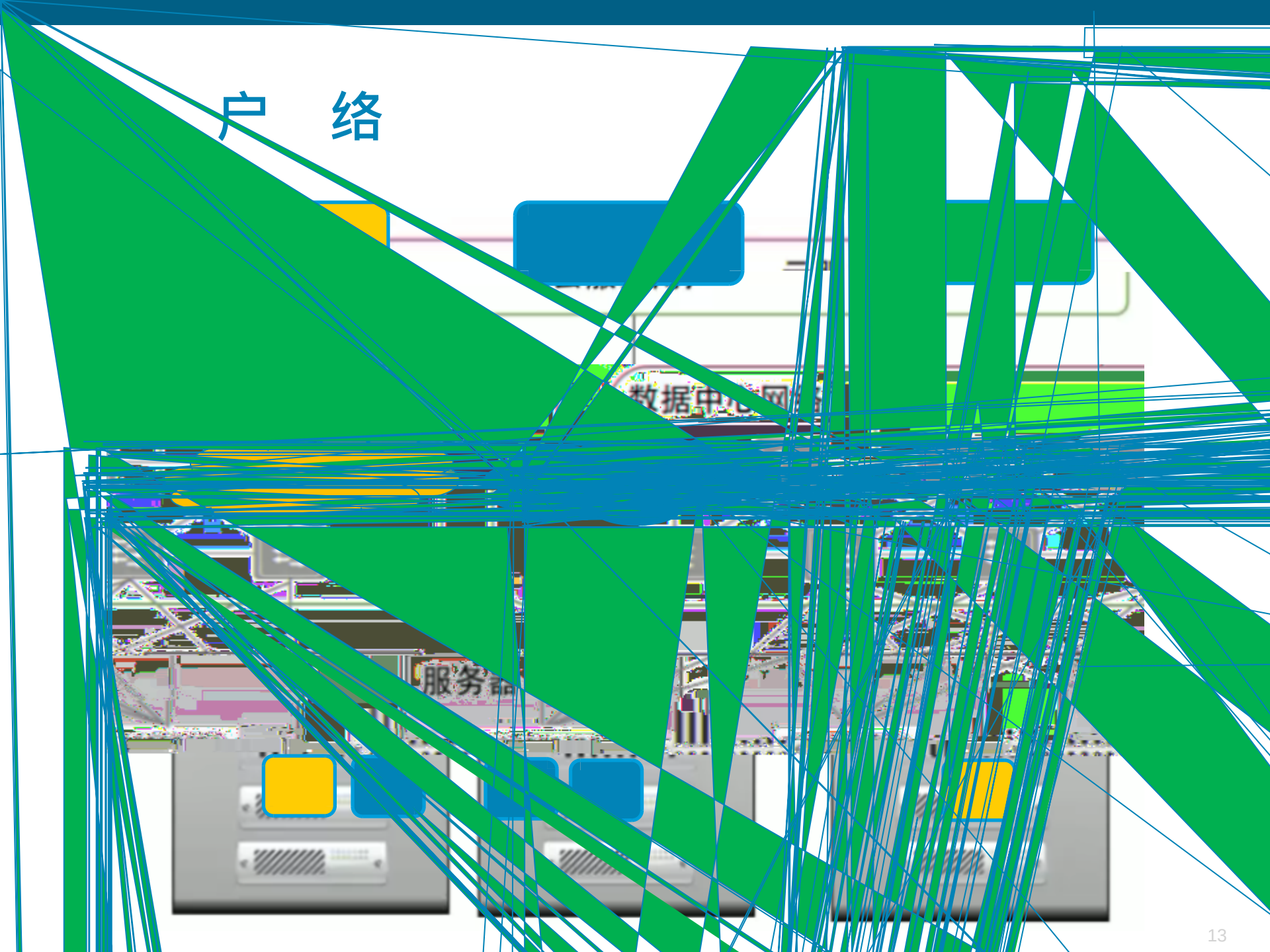
SDN



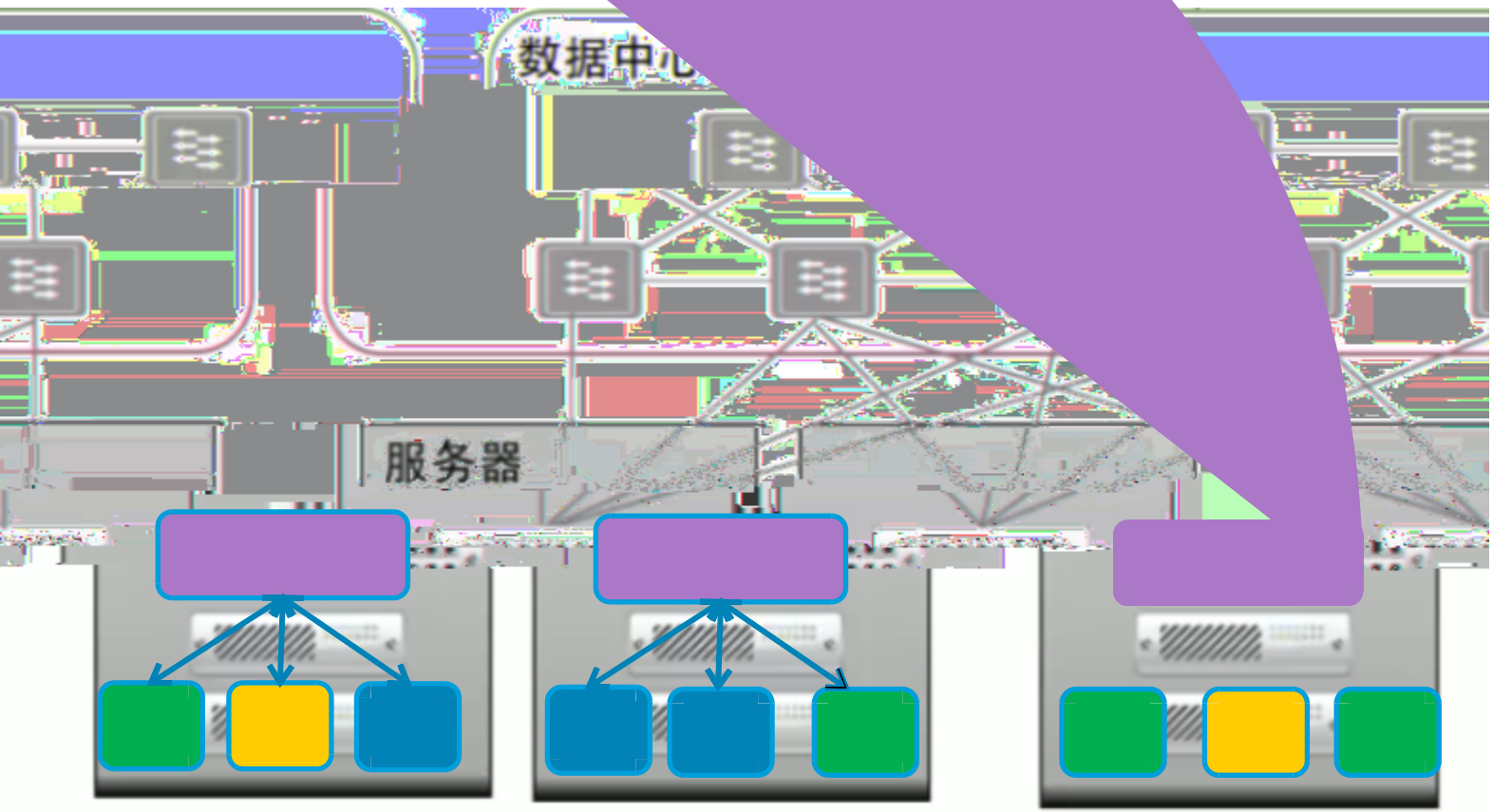
SDN


ODL
013.12)
NOX
(2008)
Ryu
(2012)
Floodlight
(2012.1)
ONOS
(2014.12)
tributed
Centralized
multi-threaded
Centralized
multi-threaded
Centralized
Distributed
REST/RESTCONF/Java
C++ API
REST API
RESTful API/Java API
RESTful API
Java
C++/Python
Python
Java
Java
Web GUI/CLI
CLI
Web UI
Web GUI/CLI
Full
Full
OpenFlow
Full
OpenFlow
--
Yes
No
Yes
Yes
v1.0
OpenFlow支持
v1.0
Strong
Weak
一致性
No
No
No
No
No
学习曲线
Medium
Easy
Medium
Medium
Hard
License
GPLv3
Apache 2.0
Apache
Apache 2.0
EPL v1.0

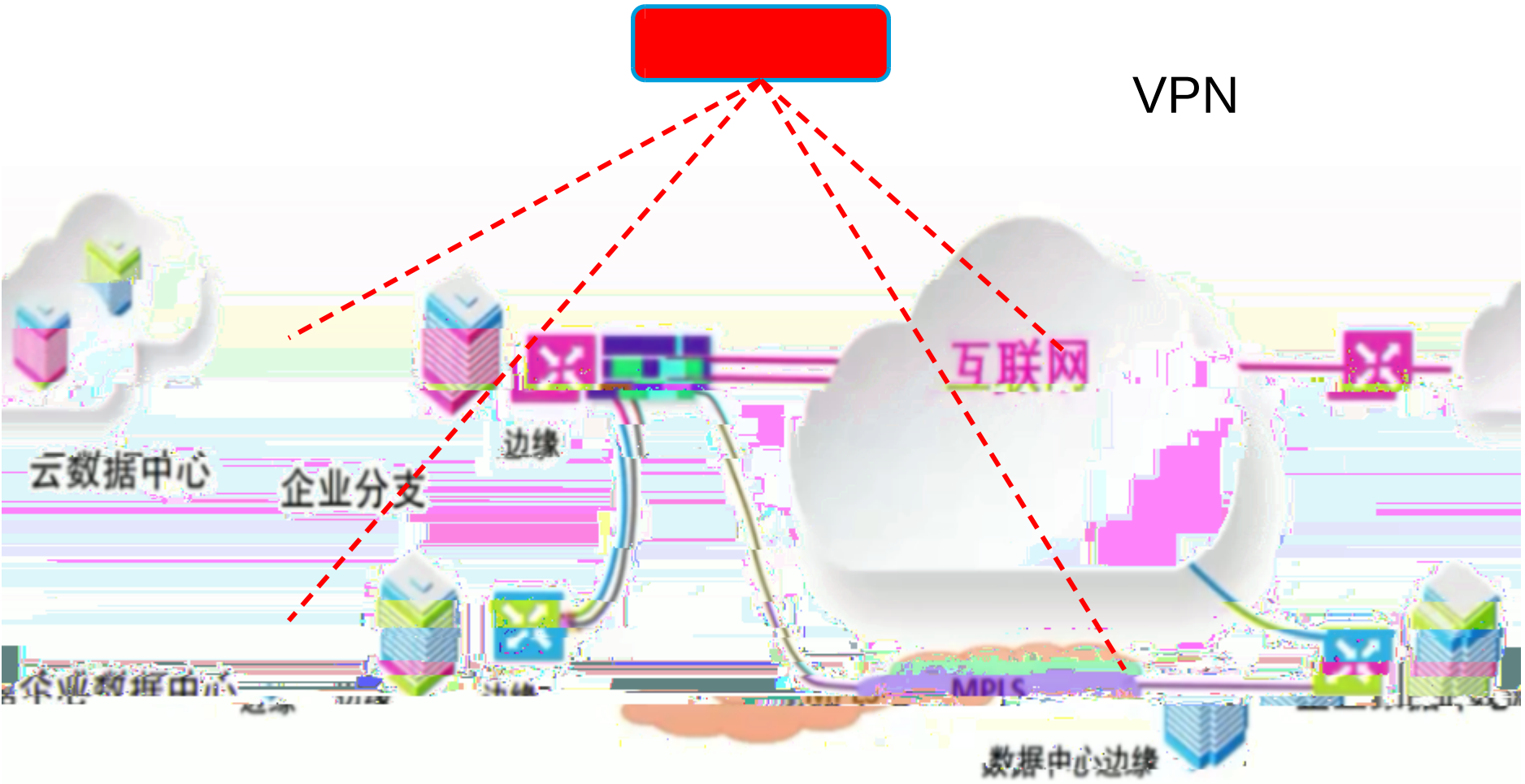
户　络



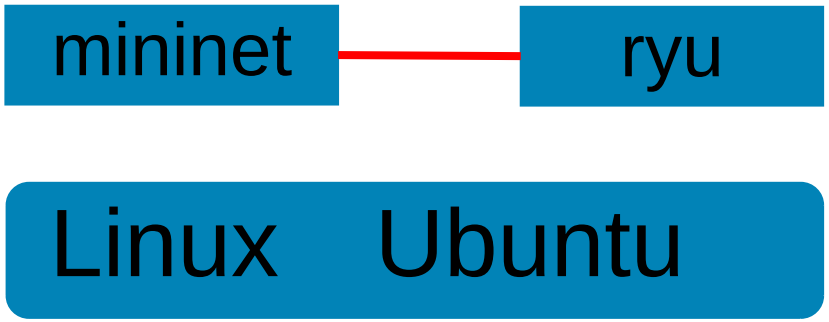
SD



软　　义　　——SDWAN



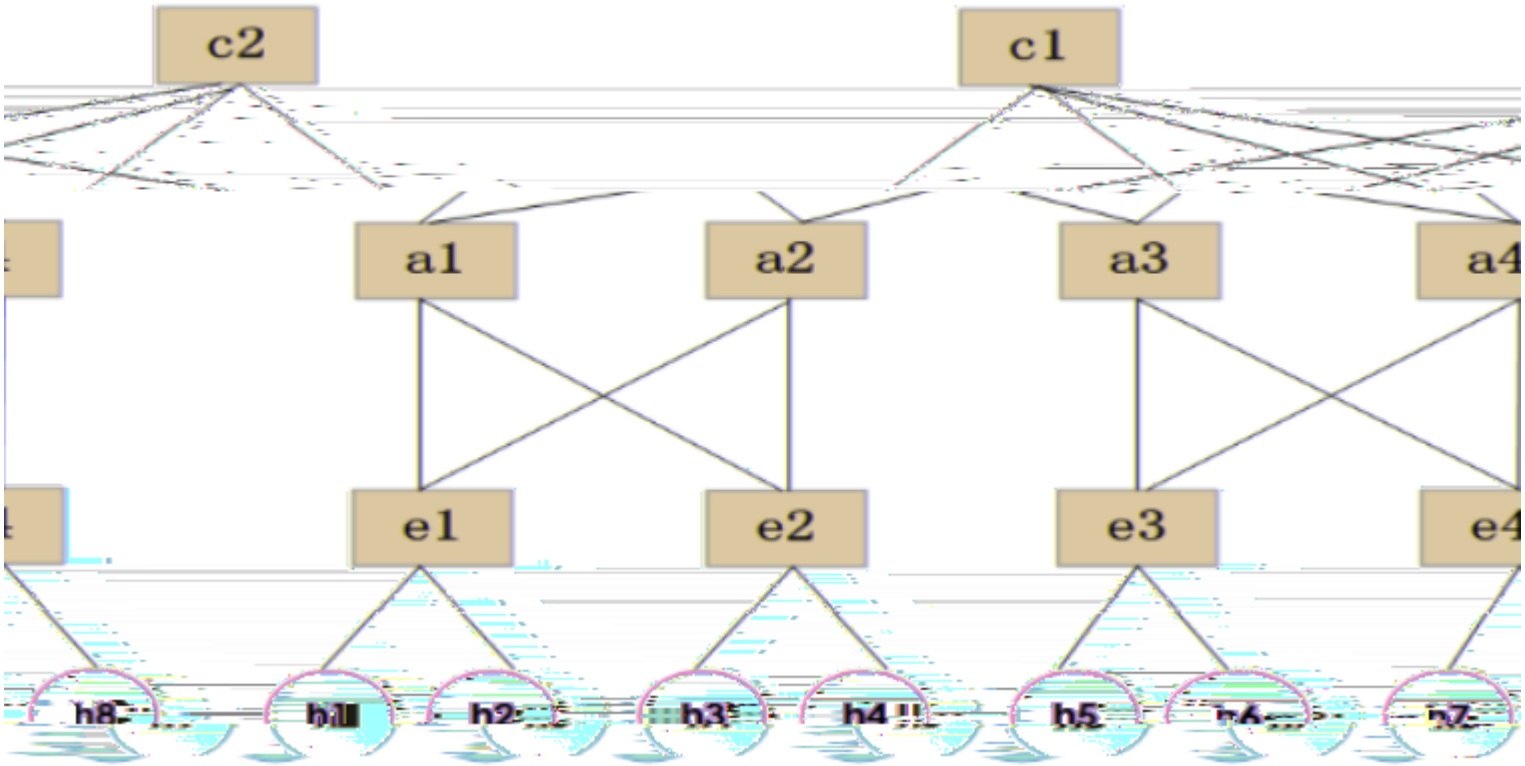
实验环



mininet Ryu SDN

Python

拟

- 

拟

fattree.py

```
#!/usr/bin/python
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.node import RemoteController
from mininet.link import TCLink
from mininet.util import dumpNodeConnections
class MyTopo(Topo):
        def __init__(self):
                Topo.__init__(self)
                L1 = 2
                L2 = L1 * 2
                L3 = L2
                c = []
                a = []
                e = []
```

拟

fattree.py

```
):                                                    for i in range( L1
ddSwitch('c{}'.format(i+1))                               sw=self.a
(sw)                                                      c.append(

):                                                    for i in range( L2
ddSwitch('a{}'.format(L1+i+1))                            sw=self.a
(sw)                                                      a.append(

            for i in range( L3 ):
                sw=self.addSwitch('e{}'.format(L2+L1+i+1))
                e.append(sw)
```

拟

fattree.py

```
for i in range(0, L1):
        sw1 = c[i]
        for sw2 in a[i/2::L1/2]:
                self.addLink(sw2,sw1)

for i in range(0, L2, 2):
        for  sw1 in a[i:i+2]:
                for sw2 in e[i:i+2]:
                        self.addLink(sw2,sw1)
```

拟

fattree.py

```
        count =1
        for sw1 in e:
                for i in range(2):
                        host = self.addHost('h{}'.format( count ))
                        self.addLink( sw1,host)
                        count += 1

topos = { 'mytopo' : (lambda:MyTopo()) }
```

拟

net.py iperf_single

```
iperf_single( self ,hosts=None, udpBw='10M', period=60, port=5001):      def
    if not hosts:
            return
    else:
            assert len(hosts) ==2
    client,server=hosts
    filename=client.name[1:]+'.out'
    output('*** Iperf:testing bandwidth between ')
    output("%s and %s\n" %(client.name, server.name))
    iperfArgs = 'iperf -u '
    bwArgs = '-b '+udpBw + ' '
    print "***start server***"
            server.cmd( iperfArgs + '-s -i 1' + '>/home/wjs/log/' + filename + '&')
            print "***start client ***"
            client.cmd(
                    iperfArgs + '-t ' +str(period) + ' -c '+server.IP() + ' ' +bwArgs
                    +' > /home/wjs/log' + 'client' + filename +'&')
```

拟

net.py iperfMulti

```
def iperfMulti(self, bw , period =60):
        base_port =5001
        server_list = []
        client_list = [h for h in self.hosts ]
        host_list = []
        host_list = [h for h in self.hosts]

        cli_outs = []
        ser_outs = []
        _len = len(host_list)
        for i in xrange(0, _len):
                client = host_list[i]
                server = client
                while(server == client ):
                        server = random.choice(host_list)
                server_list.append(server)
                self.iperf_single(hosts = [client,server], udpBw=bw;
                                        period=period, port=base_port)
                sleep(.05)
                base_port +=1
    sleep(period)
    print "test has done"
```

拟

cli.py　　iperfMulti

```
def do_iperfmulti( self, line):
        args = line.split()
        if len(args) == 1:
                udpBw = args[ 0 ]
                self.mn.iperfMulti(udpBw)
        elif len(args) ==2:
                udpBw = args[ 0 ]
                period = args[ 1 ]
                err = False
                self.mn.iperfMulti(udpBw, float(period))
        else:
                error('invalid number of args: iperfmulti udpBw period\n' +
                                        'udpBw examples:1M 120\n')
```

拟

mn

rfudp','iperfmulti') }

iperfmulti

```
TESTS = { name: True
    for name in ( 'pingall', 'pingpair', 'iperf', 'ipe
CLI = None  # Set below if needed
# Locally defined tests
def allTest( net ):
    "Run ping and iperf tests"
    net.waitConnected()
    net.start()
    net.ping()
    net.iperf()
def nullTest( _net ):
    "Null test (does nothing)"
    pass
TESTS.update( all=allTest, none=nullTest, build=nullTest )
# Map to alternate spellings of Mininet() methods
ALTSPELLING = { 'pingall': 'pingAll', 'pingpair': 'pingPair',
                'iperfudp': 'iperfUdp', 'iperfmulti':'iperfMulti' }
```

拟

mininet

ryu

#ryu-manager simple_switch_stp.py

mininet

#mn –custom fattree.py --topo mytopo
--controller=remote,ip=127.0.0.1,port=6633

iperf h1 h2
iperf h1 h3
iperf h1 h5
iperfmulti 0.025m

Thank You!

2018 05 12