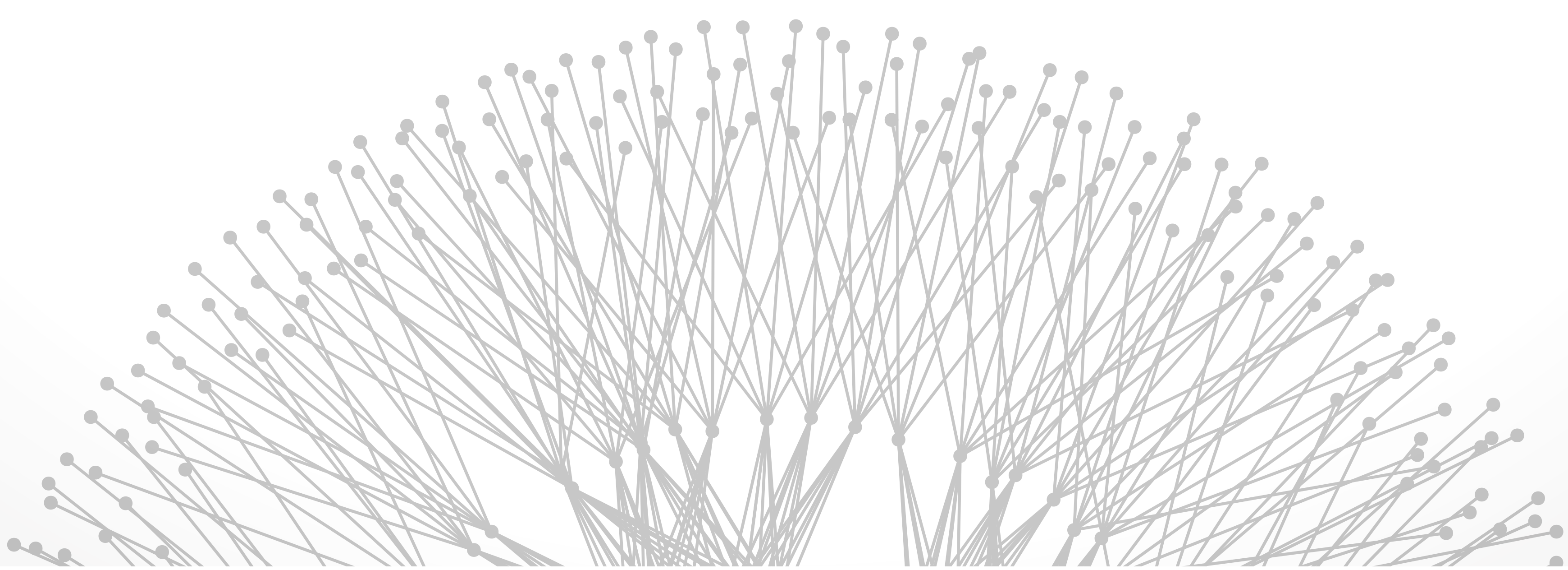
Software Defined Networking Architecture

Brighten Godfrey

CS 538 October 8 2013



The Problem

Networks are complicated

- Just like any computer system
- Worse: it's distributed
- Even worse: no clean programming APIs, only "knobs and dials"

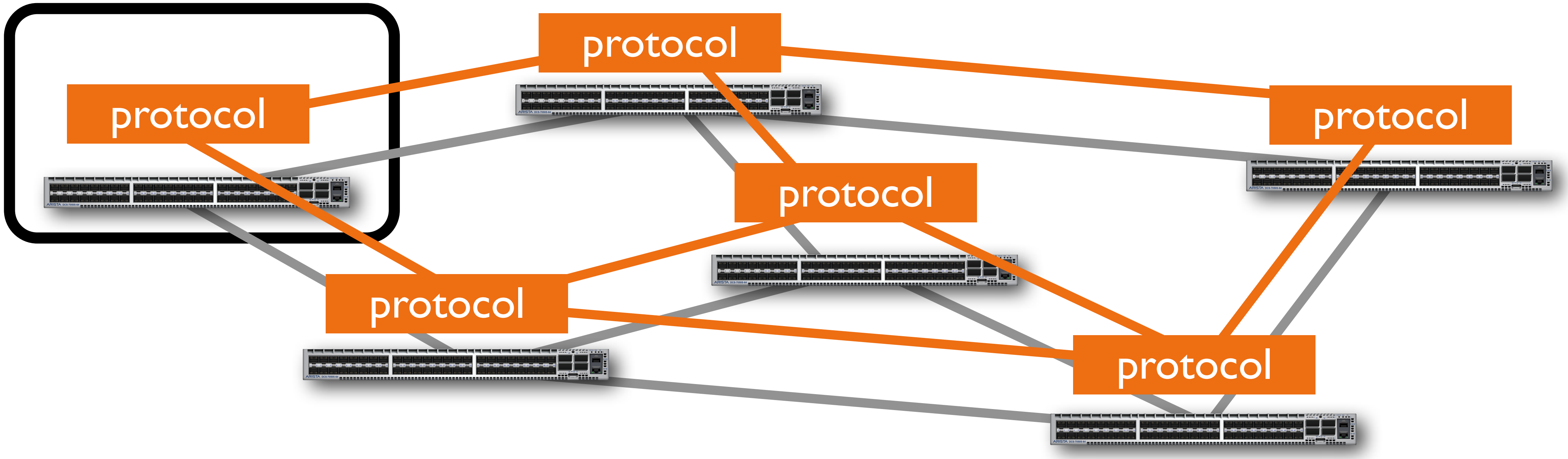
Network equipment is proprietary

- Integrated solutions (software, configuration, protocol implementations, hardware) from major vendors (Cisco, Juniper, etc.)

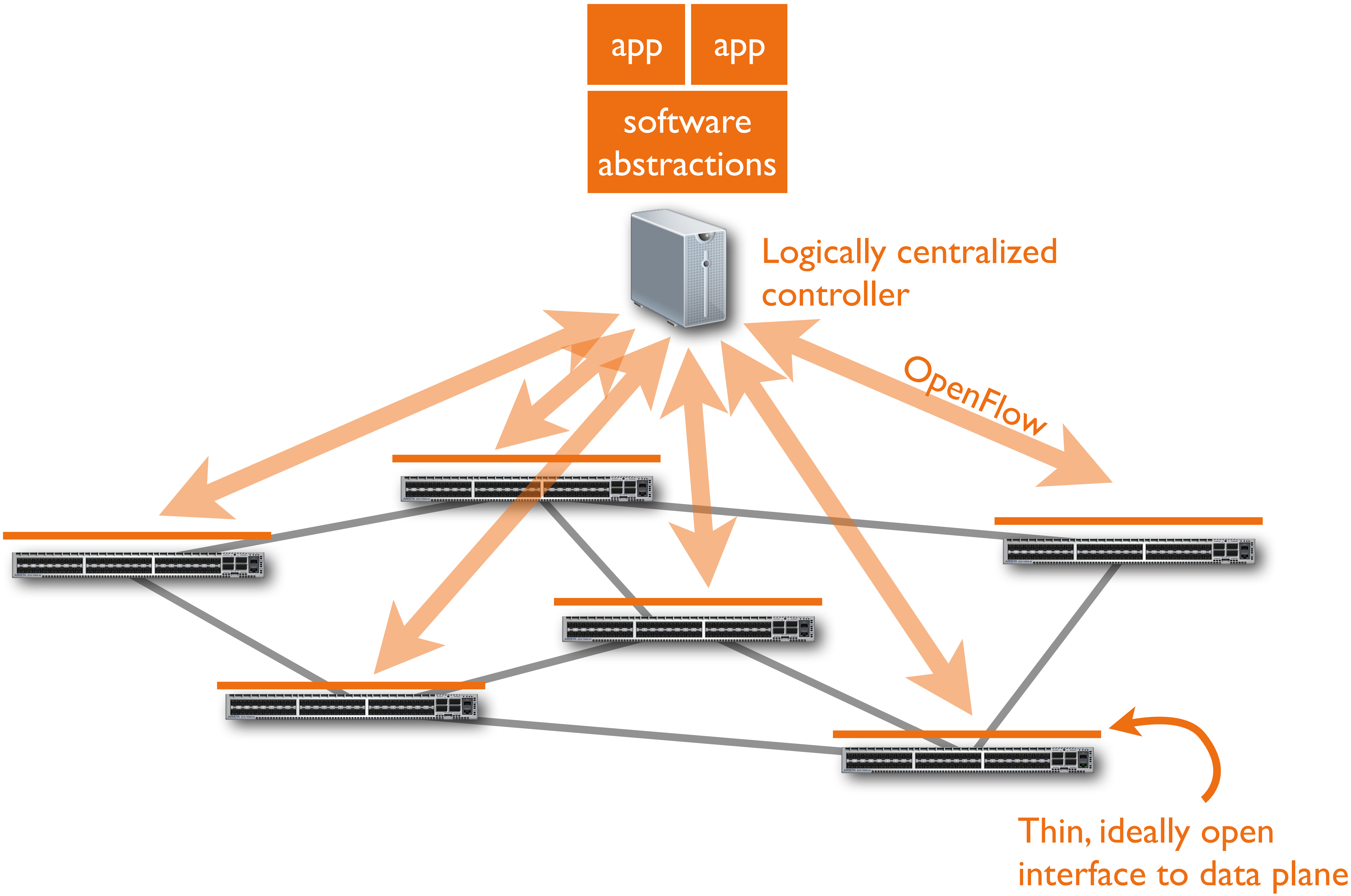
Result: Hard to innovate and modify networks

Traditional networking

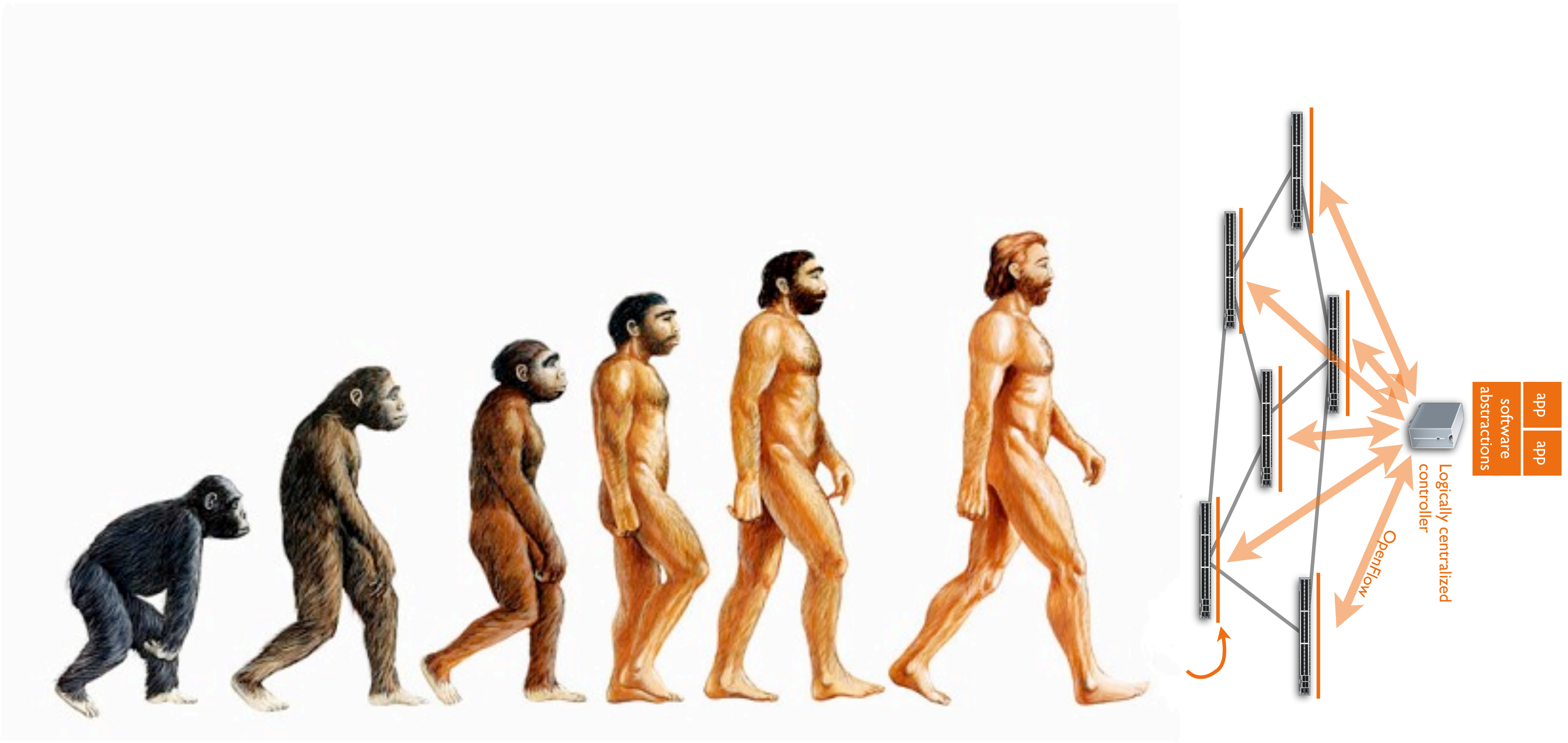
monolithic,
proprietary,
distributed



Software Defined Networking



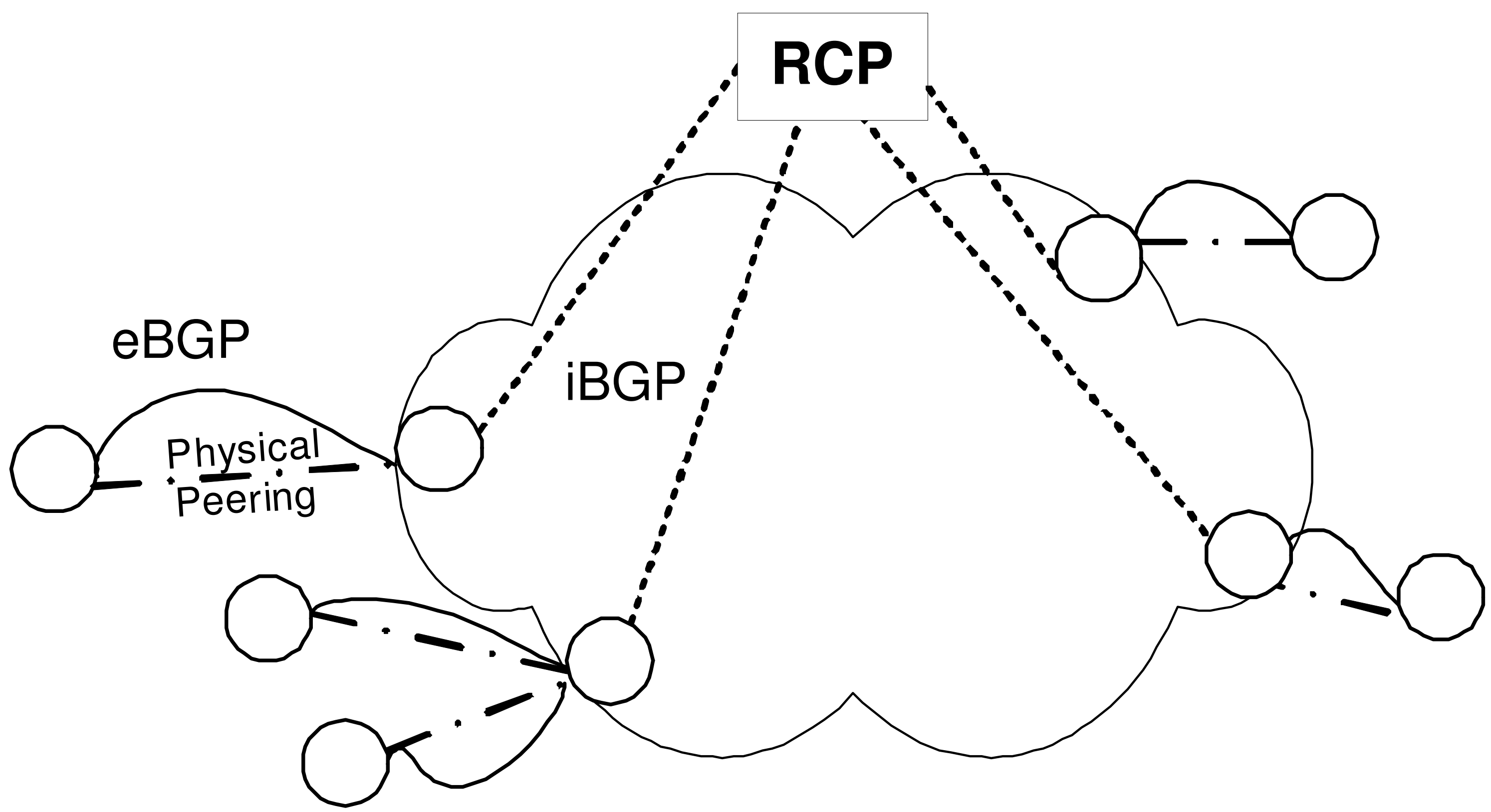
Evolution of SDN



Evolution of SDN

Routing Control Platform (2005)

- [Caesar, Caldwell, Feamster, Rexford, Shaikh, van der Merwe, NSDI 2005]
- Centralized computation of BGP routes, pushed to border routers via iBGP

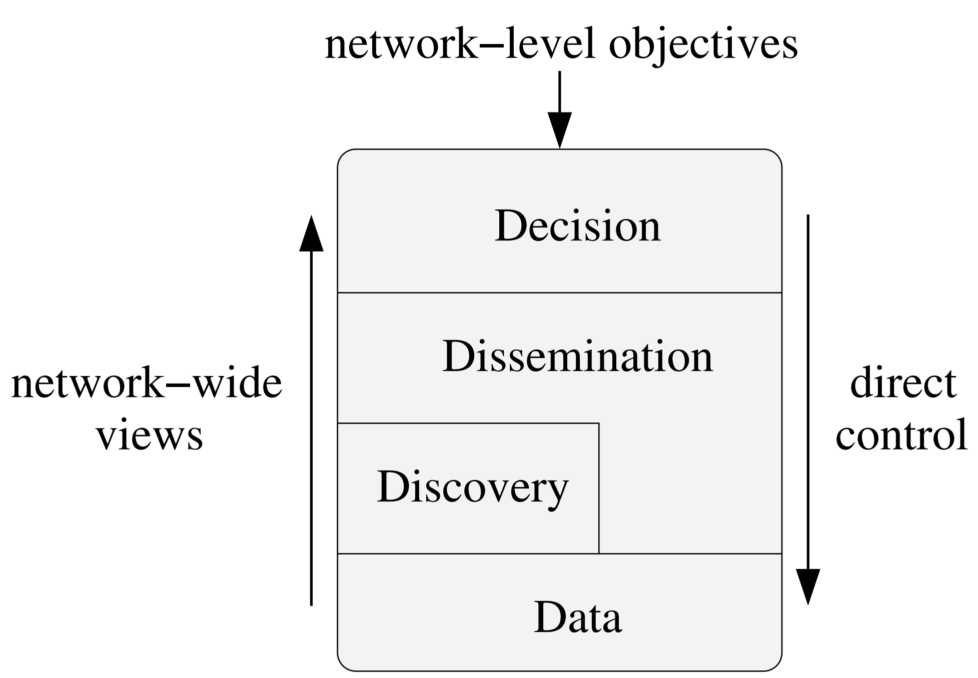


Evolution of SDN

Routing Control Platform (2005)

4D architecture (2005)

- A Clean Slate 4D Approach to Network Control and Management [Greenberg, Hjalmtysson, Maltz, Myers, Rexford, Xie, Yan, Zhan, Zhang, CCR Oct 2005]
- Logically centralized "decision plane" separated from data plane



Evolution of SDN

Routing Control Platform (2005)

4D architecture (2005)

Ethane (2007)

- [Casado, Freedman, Pettit, Luo, McKeown, Shenker, SIGCOMM 2007]
- Centralized controller enforces enterprise network Ethernet forwarding policy using existing hardware

Evolution of SDN

Routing Control Platform (2005)

4D architecture (2005)

Ethane (2007)

- [Casado, Freedman, Pet
  SIGCOMM 2007]
- Centralized controller
  Ethernet forwarding po

```
# Groups —
desktops = ["griffin","roo"];
laptops = ["glaptop","rlaptop"];
phones = ["gphone","rphone"];
server = ["http_server","nfs_server"];
private = ["desktops","laptops"];
computers = ["private","server"];
students = ["bob","bill","pete"];
profs = ["plum"];
group = ["students","profs"];
waps = ["wap1","wap2"];
%%
# Rules —
[(hsrc=in("server")∧(hdst=in("private"))] : deny;
# Do not allow phones and private computers to communicate
[(hsrc=in("phones")∧(hdst=in("computers"))] : deny;
[(hsrc=in("computers")∧(hdst=in("phones"))] : deny;
# NAT-like protection for laptops
[(hsrc=in("laptops")] : outbound-only;
# No restrictions on desktops communicating with each other
[(hsrc=in("desktops")∧(hdst=in("desktops"))] : allow;
# For wireless, non-group members can use http through
# a proxy. Group members have unrestricted access.
[(apsrc=in("waps"))∧(user=in("group"))] :allow;
[(apsrc=in("waps"))∧(protocol="http)] : waypoints("http-proxy");
[(apsrc=in("waps"))] : deny;
[]: allow; # Default-on: by default allow flows
```

Figure 4: A sample policy file using *Pol-Eth*

Evolution of SDN

Routing Control Platform (2005)

4D architecture (2005)

Ethane (2007)

OpenFlow (2008)

- [McKeown, Anderson, Balakrishnan, Parulkar, Peterson, Rexford, Shenker, Turner, CCR 2008]
- Thin, standardized interface to data plane
- General-purpose programmability at controller

Evolution of SDN

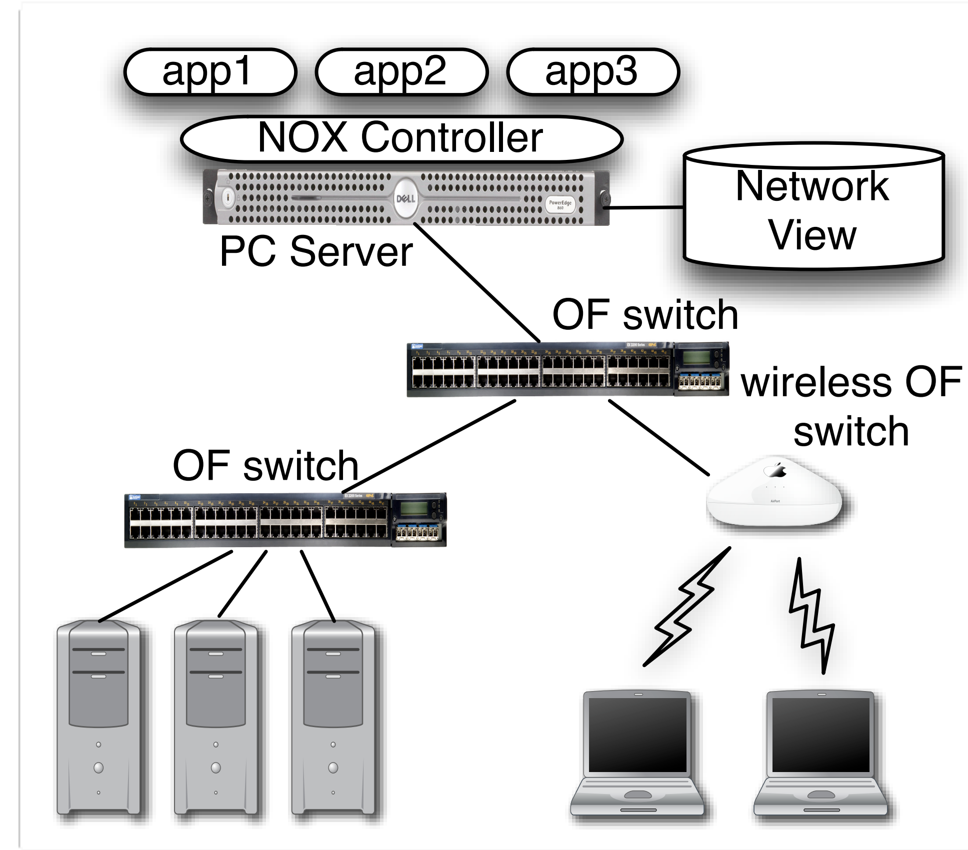
Routing Control Platform (2005)

4D architecture (2005)

Ethane (2007)

OpenFlow (2008)

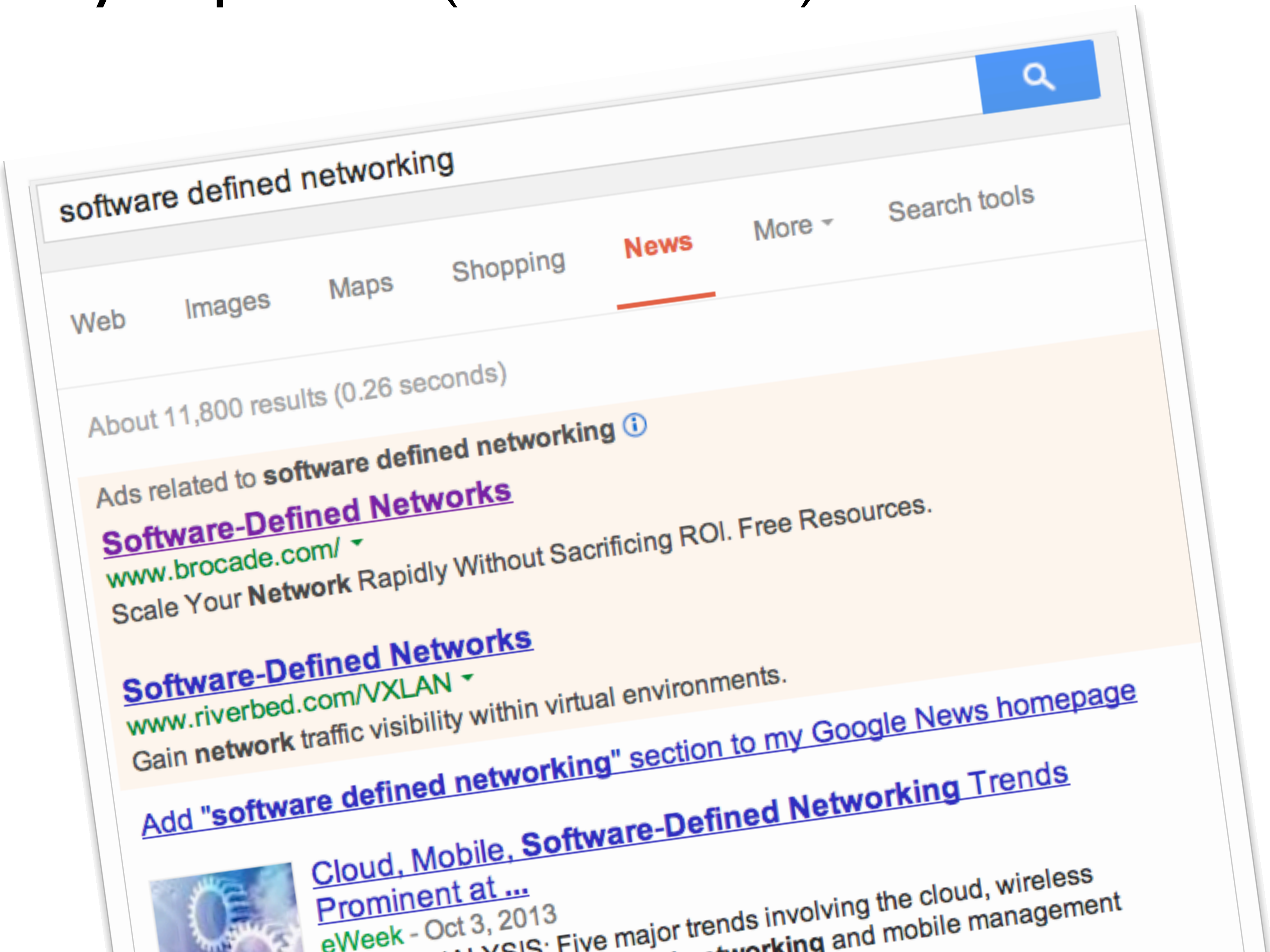
NOX (2008)



- [Gude, Koponen, Pettit, Pfaff, Casado, McKeown, Shenker, CCR 2008]
- First OF controller: centralized network view provided to multiple control apps as a database
- Behind the scenes, handles state collection & distribution

Evolution of SDN

Industry explosion (~2009-2010)



Opportunities

Open data plane interface

- Hardware: easier for operators to change hardware, and for vendors to enter market
- Software: can finally directly access device behavior

Centralized controller

- Direct programmatic control of network

Software abstractions on the controller

- Solve distributed systems problems only once, then just write algorithms
- Libraries/languages to help programmers write net apps

Opportunities

Open data plane interface

- Hardware: easier for operators to change hardware, and for vendors to enter market
- Software: can finally directly access device behavior

Centralized controller

- Direct programmatic control of network

Software abstractions on the cont

- Solve distributed systems proble write algorithms
- Libraries/languages to help programme

All active areas of current research!

Challenges for SDN

Scalability (controller is bottleneck)

Single point of failure (or small number)

Latency to controller

Needs new hardware or software

Distributed system challenges still present

- Imperfect knowledge of network state
- Consistency issues between controllers

Q: Drivers of early deployment?

What drove early deployment of OpenFlow & SDN? [Gourav]

Access control in enterprises? Net research?

- Good ideas, are already valuable (e.g. NSF GENI)
- But not the "killer apps" for initial large-scale deployment

Q: Drivers of early deployment?

Cloud virtualization

- Create separate virtual networks for tenants
- Allow flexible placement and movement of VMs

WAN traffic engineering

- Drive utilization to near 100% when possible
- Protect critical traffic from congestion

Key characteristics of the above

- Special-purpose deployments with less diverse hardware
- Existing solutions aren't just annoying, *they don't work!*

Q: When do you control the net?

When does the SDN controller send instructions to switches?

- ...in the OpenFlow paper? reactive
- ...other options? proactive

Q: How does SDN affect reliability?

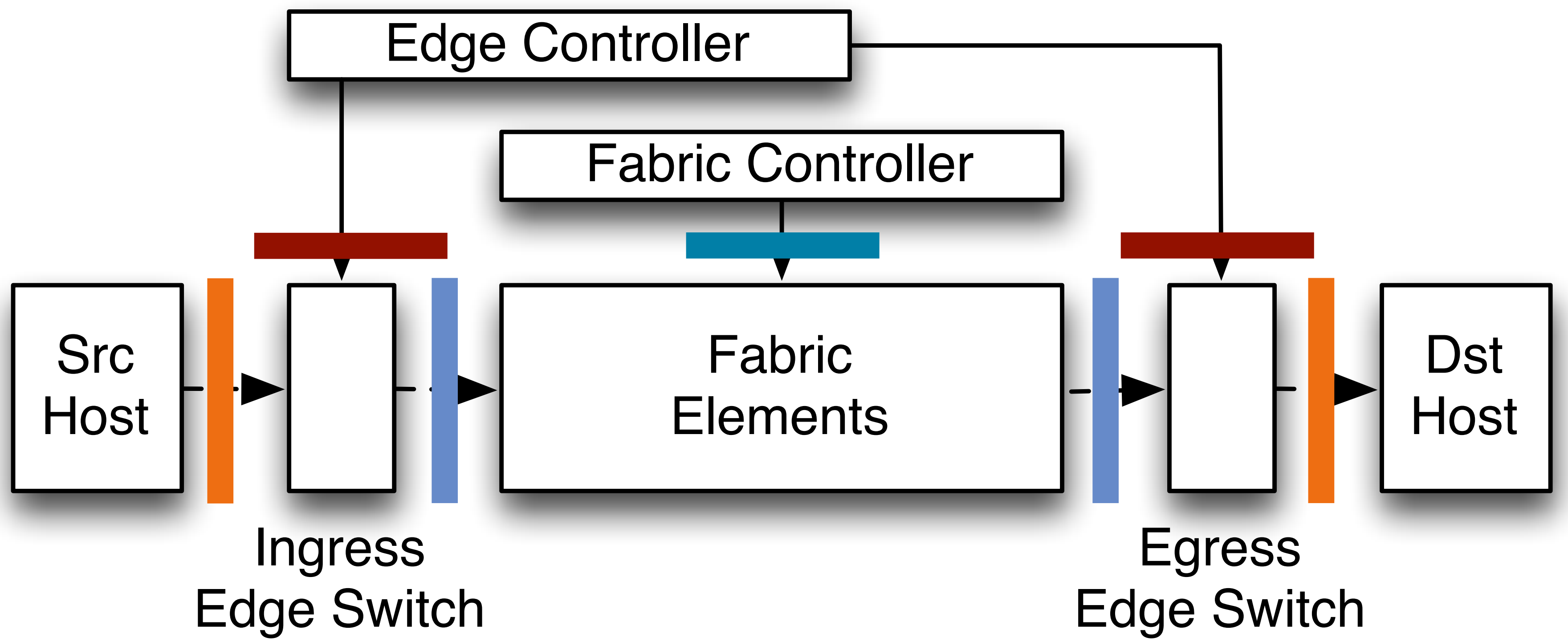
More bugs in the network, or fewer?

From SDN to Fabric

[Casado,Koponen,Shenker,Tootoonchian, HotSDN'12]

Separate interfaces:

- Host-network (external-to-internal data plane)
- Operator-network
- Packet-switch (internal data plane)



Fabric common comments

1. Is edge layer scalable?
2. No experimental results :-)
3. Is a two-layer design hard to manage?

How do established router vendors approach SDN? [Jereme Lamps]

Announcements

Poster Session Dec 17, 1:30 - 4:30

- All must attend
- Conflict? Describe situation to me by Monday Oct 14
- Online students will do separate presentation, to be scheduled

Thursday

- SDN applications
- Reading: B4 [SIGCOMM 2013]