CS244

Advanced Topics in Networking

Lecture 7: SDN

Chang Kim

"OpenFlow: Enabling Innovation in Campus Networks"
[A bunch of networking profs, CCR 2008]
"Network Virtualization in Multi-tenant Datacenters"
[T. Koponen, et al, NSDI 2014]
"From Ethane to SDN and Beyond"
[Martin Casado et al, CCR 2019]

Spring 2022

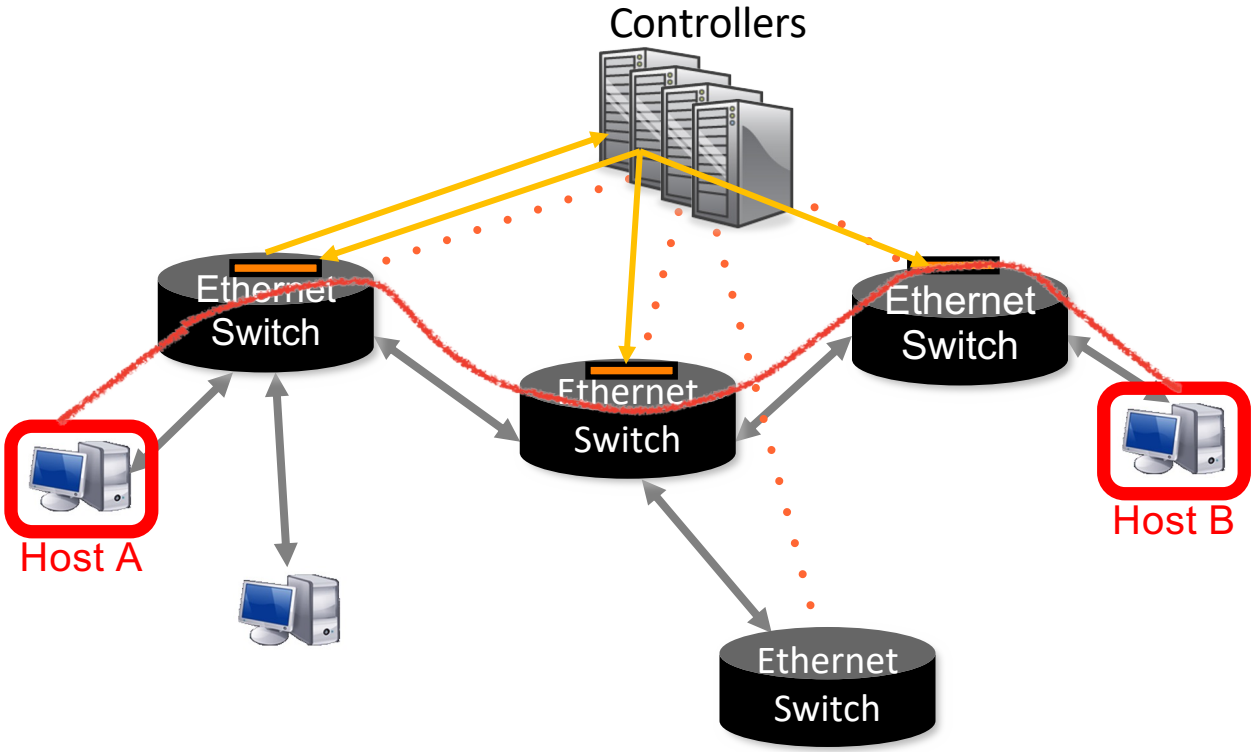
How difficult is it to define all network operations in software, outside the datapath?

2006

35,000 users
10,000 new flows/sec
137 network policies

2,000 switches
2,000 switch CPUs

Extreme thought experiment: What if software decides whether to accept each flow, and how to route it?



A question the team had:
How many $400 servers do we need for 35,000 users?

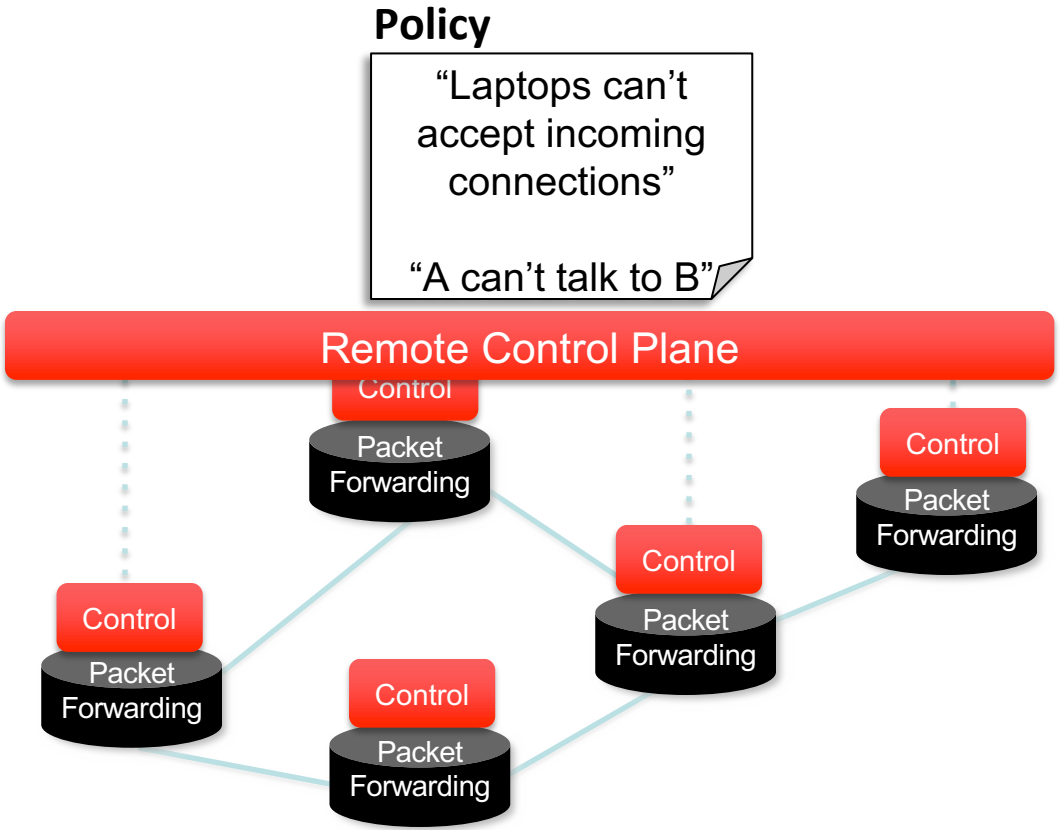
Answer: less than one

If we can control the network centrally
then (eventually) we will.

With replication for
fault-tolerance and performance scaling.

Q: Why might we want to control them centrally?
Q: How does this compare to how networks are controlled today?

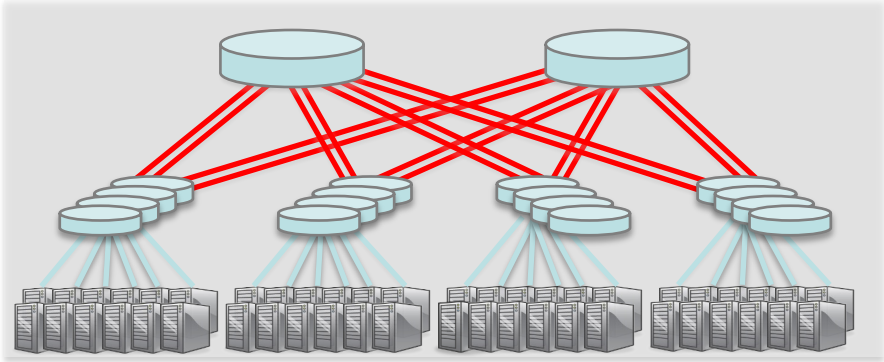
Ethane and Network Policy



The approach was starting elsewhere…

1. Public WANs: Route reflectors decide routes centrally, and download to datapath
   - AT&T Backbone
2. WiFi: CAPWAP and Meraki; Ubiquiti
3. Cable TV: Docsis
4. Disaggregation: Datacenter owners were considering build their own networking equipment.

Example: Big Data Center



Cost

500,000 servers
25,000 switches
$10k per legacy switch = $250M
$2k disaggregated switch = $50M
Savings in 5 data centers = $1Bn

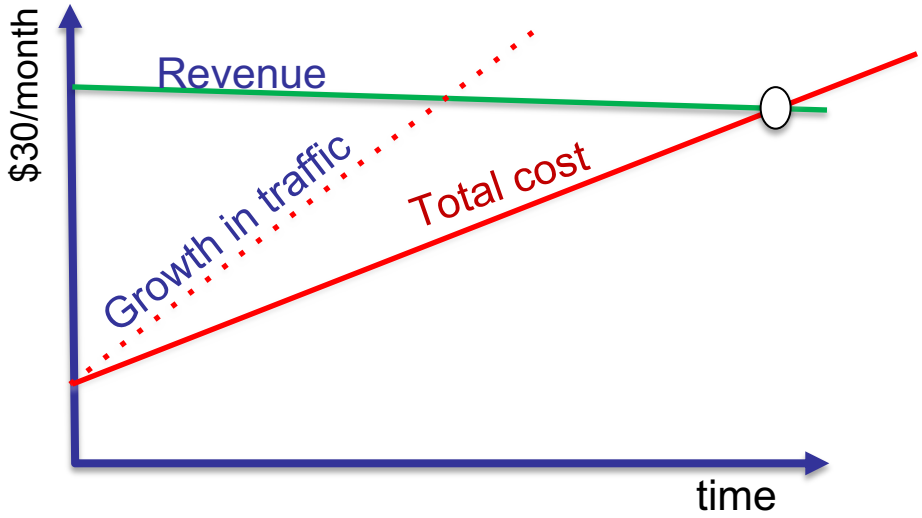
Control

Centralized remote control is easier
"Centralize if you can, distribute if you can't"
Customized, differentiated network
Home grown traffic engineering
50% utilization → 95% utilization

By 2008, Google, Microsoft, and Amazon were starting to write their own software

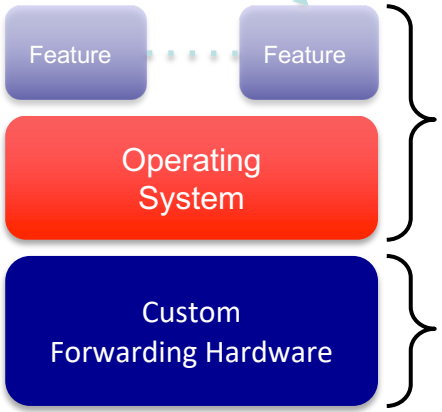
Internet Service Providers (ISPs)

- Global IP traffic growing 40-50% per year
- End-customer monthly bill unchanged
- Therefore, CAPEX and OPEX need to reduce 40-50% per Gb/s per year
- But in practice, reduces by ~20% per year



What a big Internet router looked like

Routing, management, mobility management, access control, VPNs, …



Million of lines of source code

7,000 Internet RFCs

Billions of gates

Bloated

Power Hungry

- Overly complex
- Mainframe mentality
- Too expensive

After Ethane: What was next?

Microsoft: “Come on in….”

Cisco: “It will never work…”

Raw nerve.

We must be onto something.

“The Future of Networking and the Past of Protocols”

Scott Shenker 2011

Networks today are run by

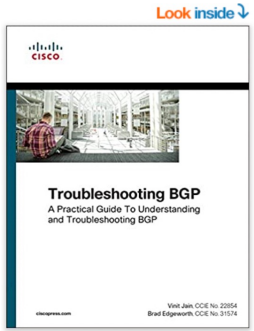
"Masters of Complexity"

Case in point: Understanding BGP

Troubleshooting BGP: A Practical Guide to Understanding and Troubleshooting BGP (Networking Technology) 1st Edition

by Vinit Jain (Author), Brad Edgeworth (Author)

★★★★½ 36 ratings



Kindle $40.39 | **Paperback** $54.71 - $63.49 | Other Sellers See all 2 versions

Buy used: $54.71

Buy new: $63.49

Only 13 left in stock (more on the way).

Ships from and sold by Amazon.com.

May be available at a lower price from other sellers, potentially without free Prime shipping.

List Price: ~~$69.99~~ Details
Save: $6.50 (9%)
prime

FREE delivery **Tuesday, April 19**.
Order within 6 hrs 28 mins

Deliver to kyungle - Palo Alto 94306

Product details

Publisher : Cisco Press; 1st edition (December 20, 2016)

Language : English

Paperback : 832 pages

ISBN-10 : 1587144646

ISBN-13 : 978-1587144646

Item Weight : 2.95 pounds

Dimensions : 7.3 x 1.8 x 9 inches

Best Sellers Rank: #1,099,762 in Books (See Top 100 in Books)
#956 in Computer Networking (Books)
#1,385 in Computer Networks, Protocols & APIs (Books)

Customer Reviews: ★★★★½ 36 ratings

Case in point: Understanding BGP

RFC Editor

RFC Number (or Subseries Number):

Title/Keyword: BGP

Show Abstract Show Keywords

Additional Criteria

Search Clear all

218 results (Show **25** | All)

Number	Files	Title	Authors	Date	More Info
RFC 1105	ASCII, PDF, HTML	**Border Gateway Protocol (BGP)**	K. Lougheed, Y. Rekhter	June 1989	Obsoleted by RFC 1163
RFC 1163	ASCII, PDF, HTML	**Border Gateway Protocol (BGP)**	K. Lougheed, Y. Rekhter	June 1990	Obsoletes RFC 1105, Obsoleted by RFC 1267
RFC 1164	ASCII, PDF, HTML	**Application of the Border Gateway Protocol in the Internet**	J.C. Honig, D. Katz, M. Mathis, Y. Rekhter, J.Y. Yu	June 1990	Obsoleted by RFC 1268
RFC 1265	ASCII, PDF, HTML	**BGP Protocol Analysis**	Y. Rekhter	October	

1. **Prefer the route with the highest weight**
   The weight attribute is proprietary to Cisco and is local to the router only.
   By default it is set to 0 for routes that were not originated by this router.
2. **Prefer the route with the highest local preference value**
   The local preference is used within an autonomous system
   By default it is set to 100 for all networks.
3. **Prefer the route that the local router originated**
   A locally originated route has a next hop of 0.0.0.0 in the BGP table.
4. **Prefer the route with the shortest autonomous system path**
5. **Prefer the lowest origin code**
   (IGP < EGP < incomplete)
6. **Prefer the path with the lowest MED**
   (The MED is exchanged between autonomous systems.) The MED comparison is made only if the neighboring autonomous system is the same for all routes considered, unless the *bgp always-compare-med* command is enabled
7. **Prefer external paths (EBGP) to internal paths (IBGP)**
8. **Prefer the path through the closest IGP neighbor**, which means that the router prefers the shortest internal path within the autonomous system to reach the destination (the shortest path to the BGP next hop)
9. **Select the oldest route to minimize the effect of routes going up and down (flapping)**
10. **…e lowest neighbor BGP router ID value.**
11. **Prefer the router with the lowest neighbor IP address**

Oh, by the way, this path selection logic is NOT specified in any of these 218 RFCs covering BGP.

Abstractions in computer systems

Virtual memory: Abstract illusion of infinite, private physical memory

File system: Uniform illusion of read/write data store.

Operating system: Shields user from CPU scheduling and peripheral sharing.

…

“Modularity based on abstraction is the way things are done!”

Barbara Liskov (MIT)
Turing Award Lecture 2009

SDN: An early definition

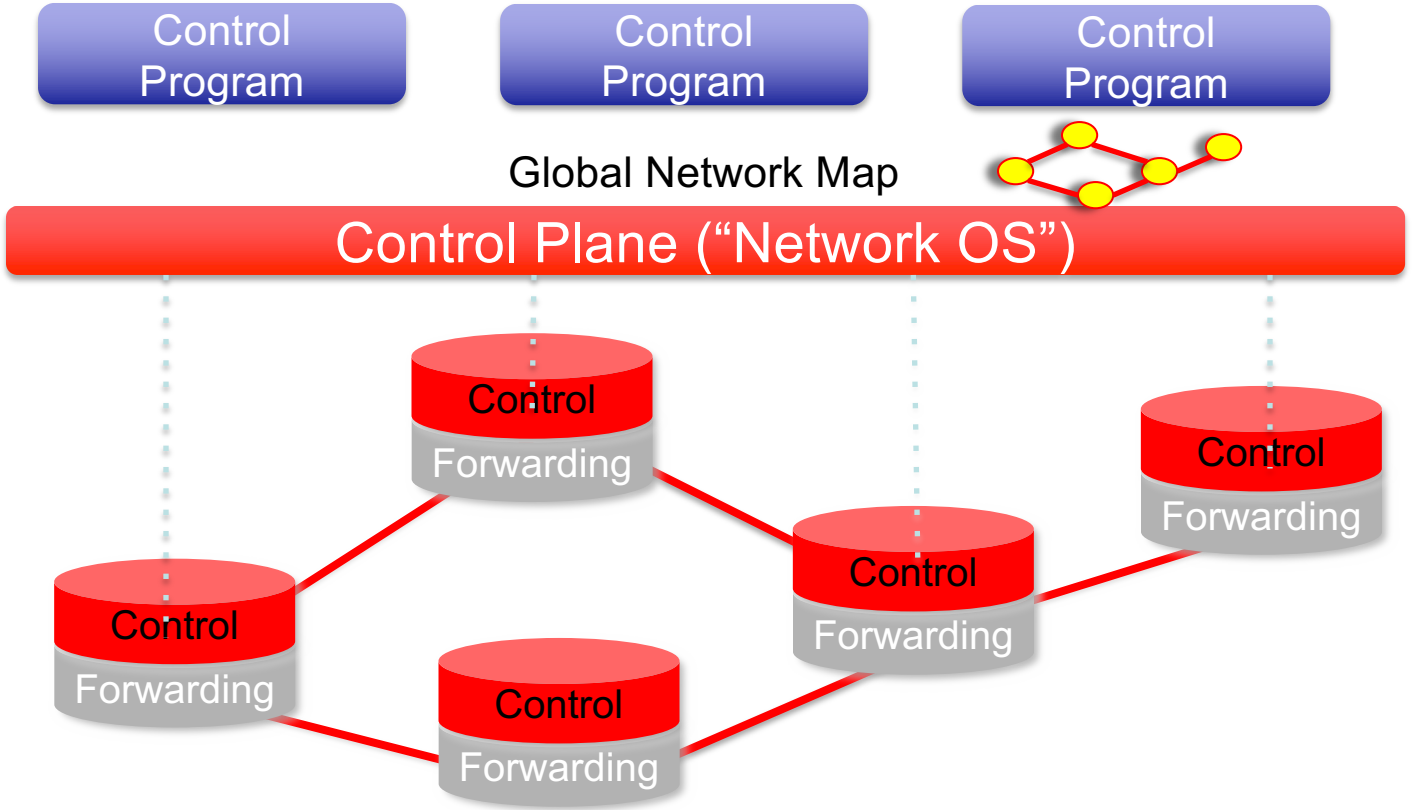
A network in which the control plane is physically separate from the forwarding plane.

and

A single control plane controls several forwarding devices.

(Evolved over time)

Software Defined Network (SDN)



OpenFlow

Motivation for OpenFlow

“Thus, the commercial solutions are too closed and inflexible, and the research solutions either have insufficient performance or fanout, or are too expensive. It seems unlikely that the research solutions, with their complete generality, can overcome their performance or cost limitations. A more promising approach is to compromise on generality and to seek a degree of switch flexibility that is:

1. Amenable to high-performance and low-cost implementations.
2. Capable of supporting a broad range of research.
3. Assured to isolate experimental traffic from production traffic.
4. Consistent with vendors’ need for closed platforms."

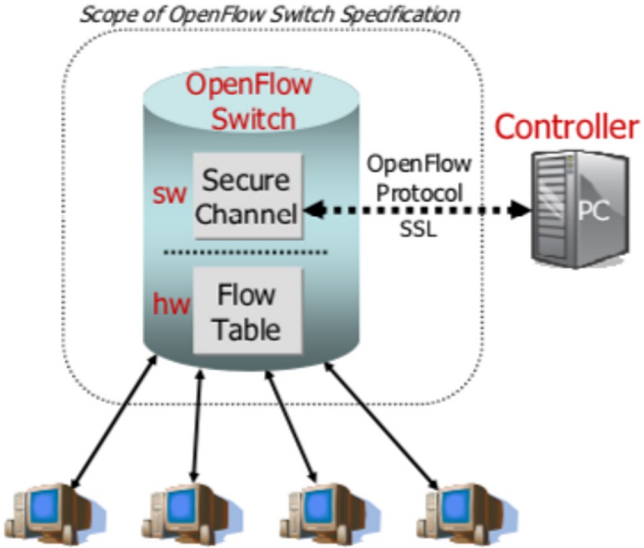


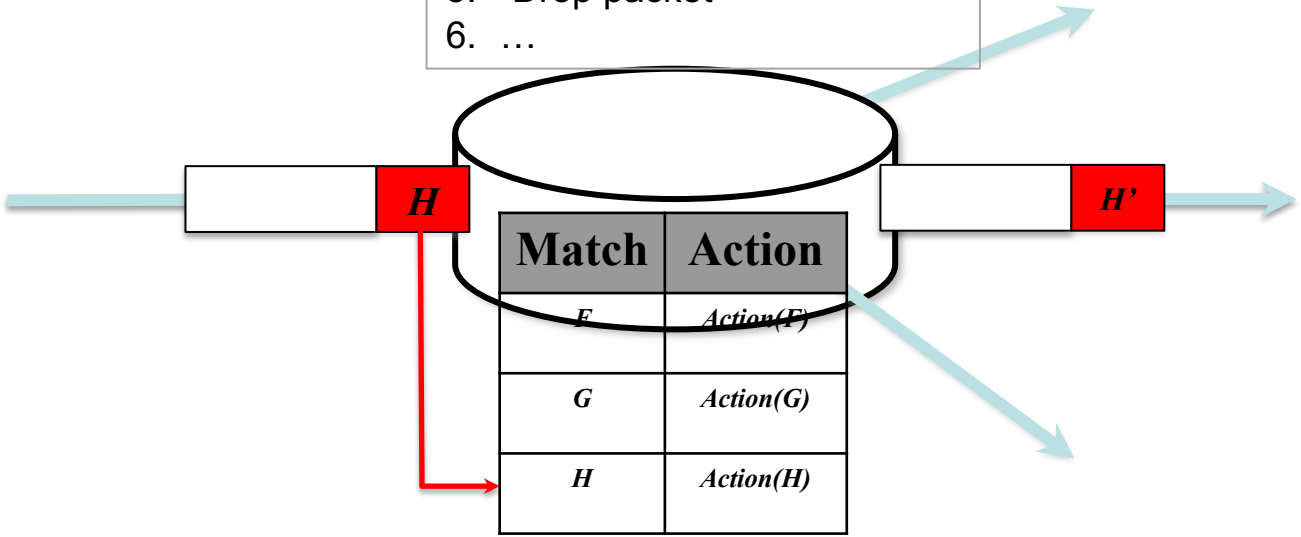
Figure 1: Idealized OpenFlow Switch. The Flow Table is controlled by a remote controller via the Secure Channel.

Match-Action Forwarding Abstraction

Action Primitives

1. “Forward to ports 4 & 5”
2. “Push header Y after bit 12”
3. “Pop header bits 8-12”
4. “Decrement bits 13-18”
5. “Drop packet”
6. …

“Plumbing primitives”



OpenFlow Goals

(as described at the time)

Short-term, backward compatability

Match: include well-known header fields.

Action: necessary set for existing protocols.

- Support existing protocols on existing switch chips.

Q: How well was each goal met?

Long-term

Match: Very general, not protocol specific.

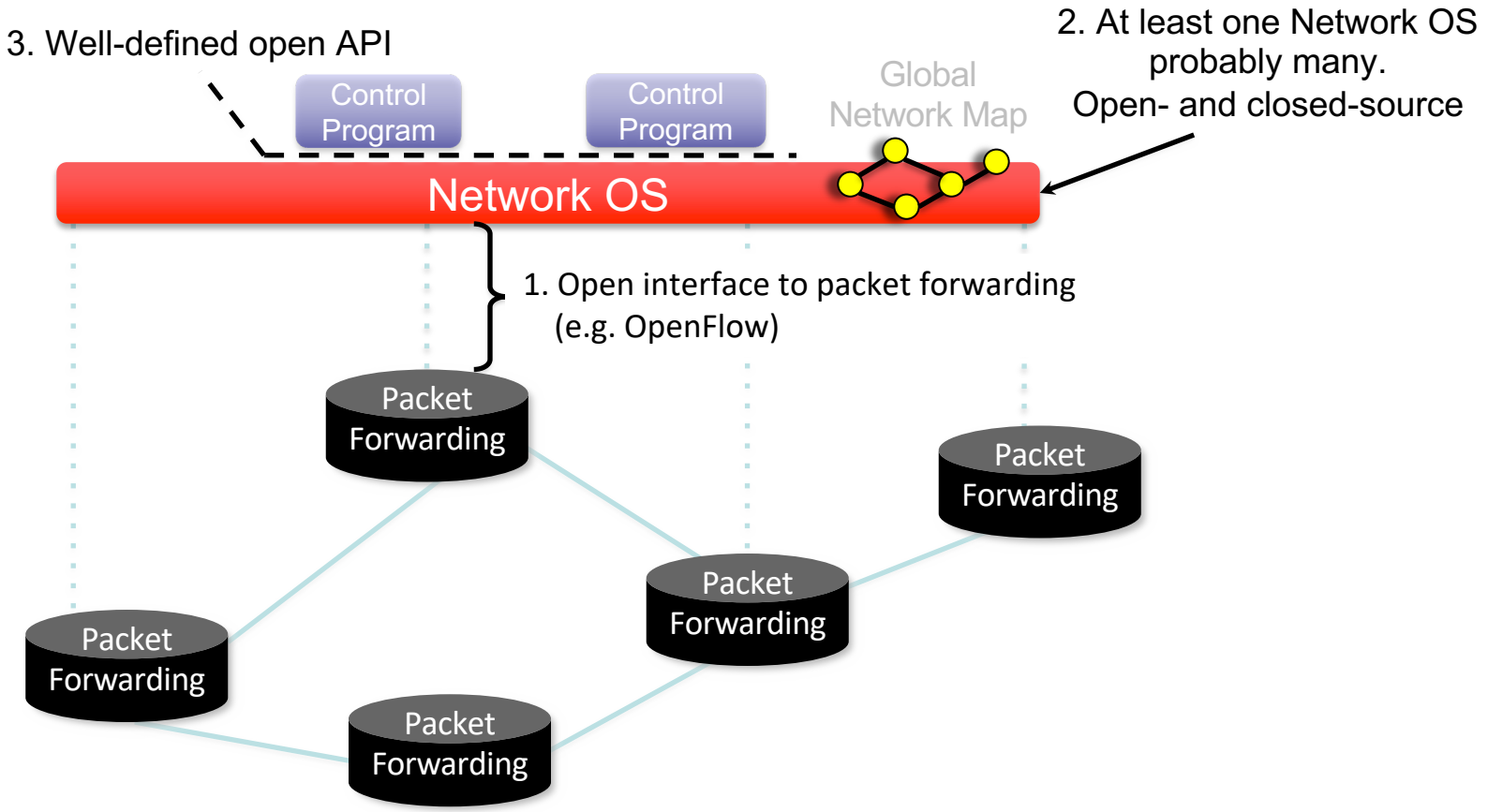
Action: Small instruction set, not protocol specific.

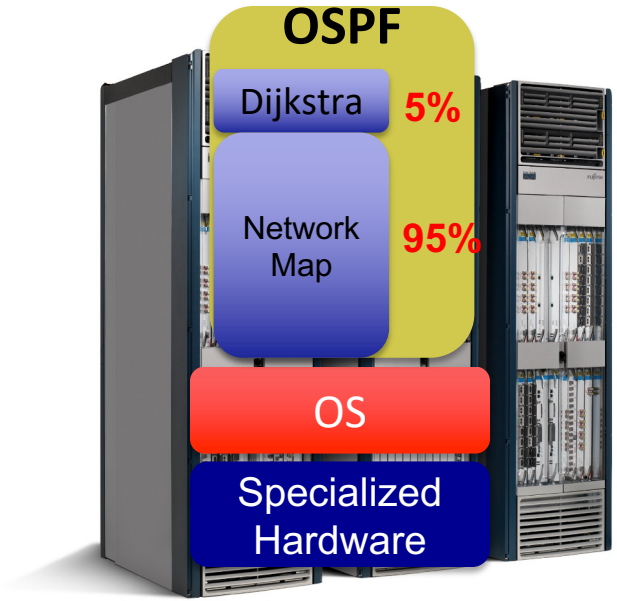
- Make it easy to add new headers and actions.
- Any network (packet, circuit, radio).

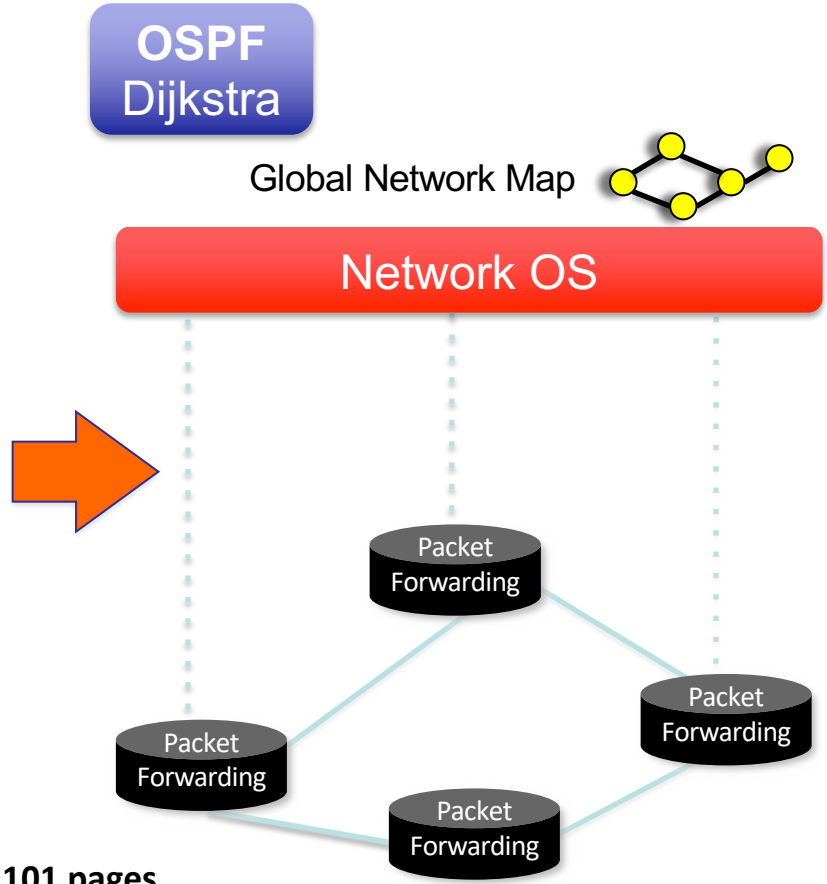
OpenFlow: Control Abstraction

1. Control plane can run on modern servers
2. Can adopt software engineering best-practices
3. Easier to add new control programs
4. …or customize locally
5. Solve distributed systems problem once, rather than for every protocol

SDN: Software Defined Networks







RFC 2328: **245 pages**

Distributed System

Builds consistent, up-to-date map of the network: **101 pages**

Dijkstra's Algorithm: **1 page**

OpenFlow: Forwarding Abstraction

1. Vendor-agnostic interface to forwarding plane
2. Simpler, lower-cost, lower-power hardware

Match + Action abstraction

Pros

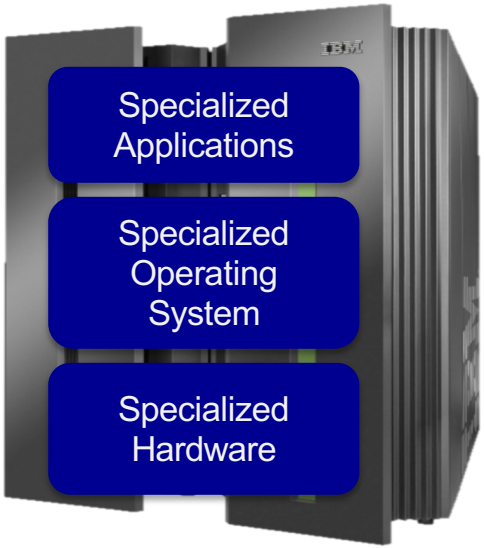
- Simple abstraction of stateless forwarding
  (e.g. Ethernet, IPv4, IPv6, VLAN, VPNs, …)
- Add/delete table entries: If a packet matches a field, then perform actions.
- Allows one API to control multiple protocols
- Enabled multiple controllers: NOX, POX, ONIX, Beacon, Floodlight, …
- Easy to add to existing switches or new disaggregated switches
  (hence Google adoption)

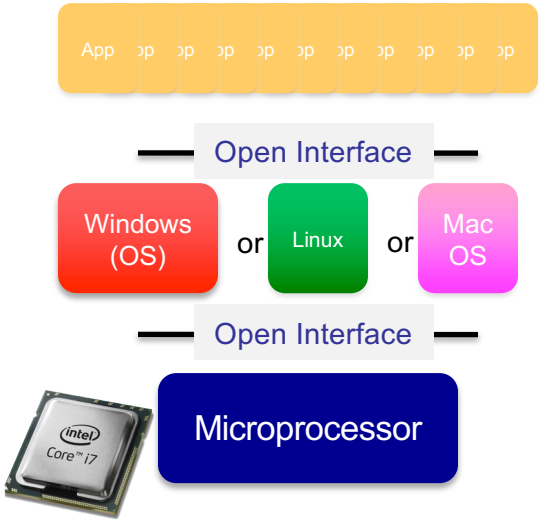
Cons

- Underlying functions were fixed, hard to add or evolve (hence P4 later)
- Hard to introduce new versions of API
- Switch vendors very reluctant to support

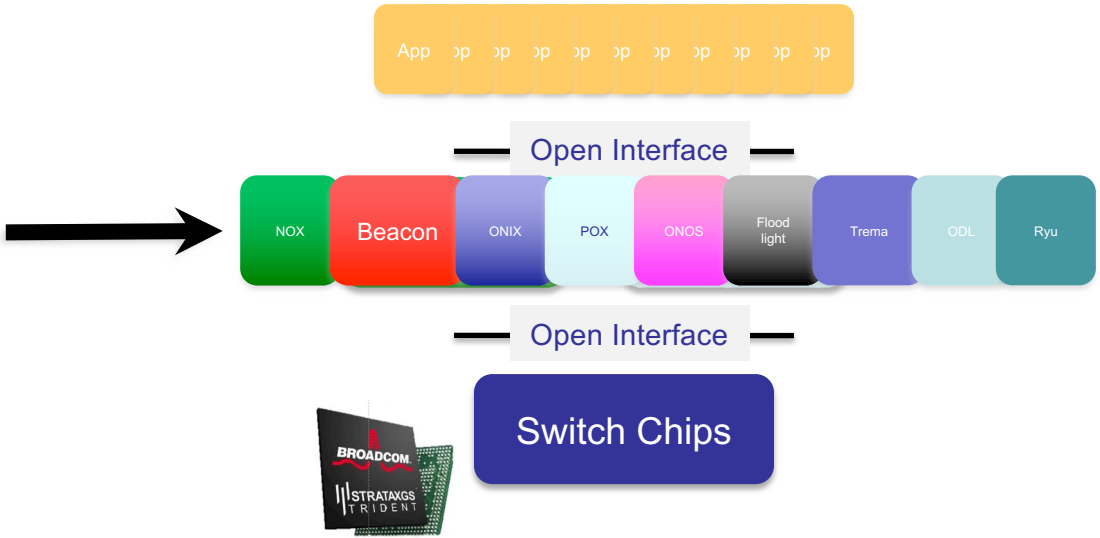
In the context of bigger networking industry changes

Computer Industry



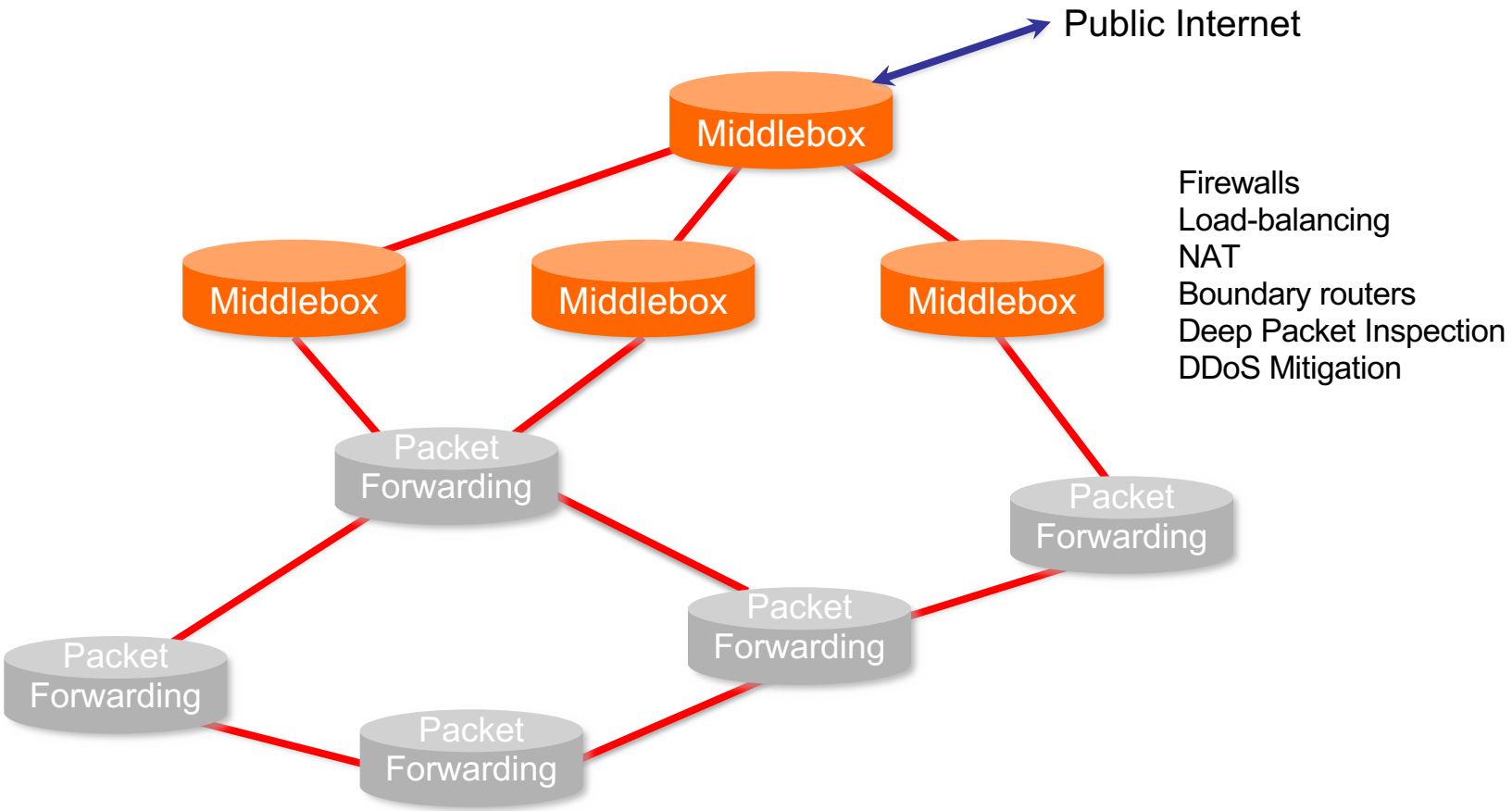


Networking Industry

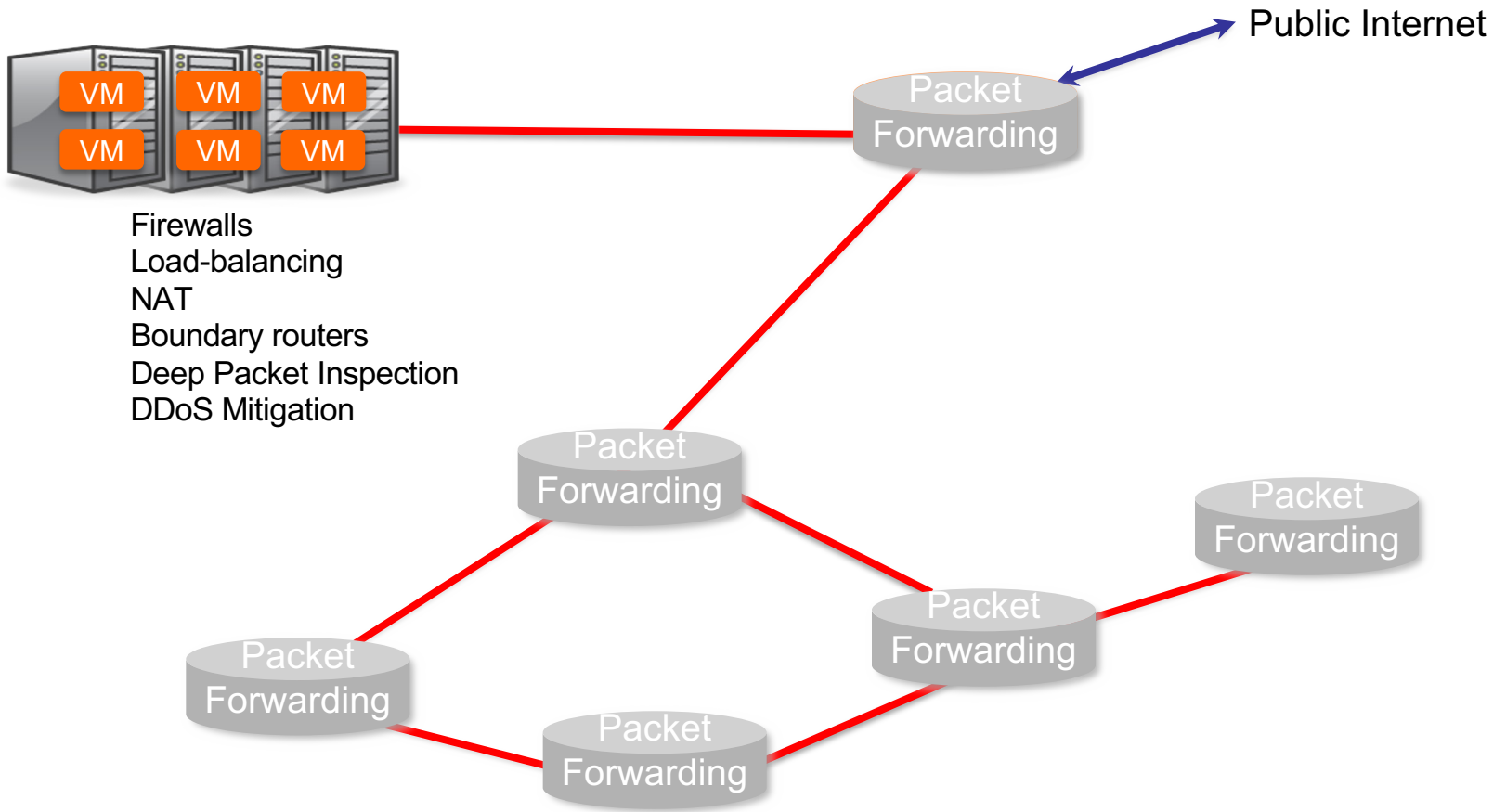


"Software is eating the world (of networking)"

Network Function Virtualization (NFV)



Network Function Virtualization (NFV)



With hindsight, disaggregation and SDN were inevitable

Part of a bigger trend towards the owners and operators of networks taking control of how they work

Inevitable because…

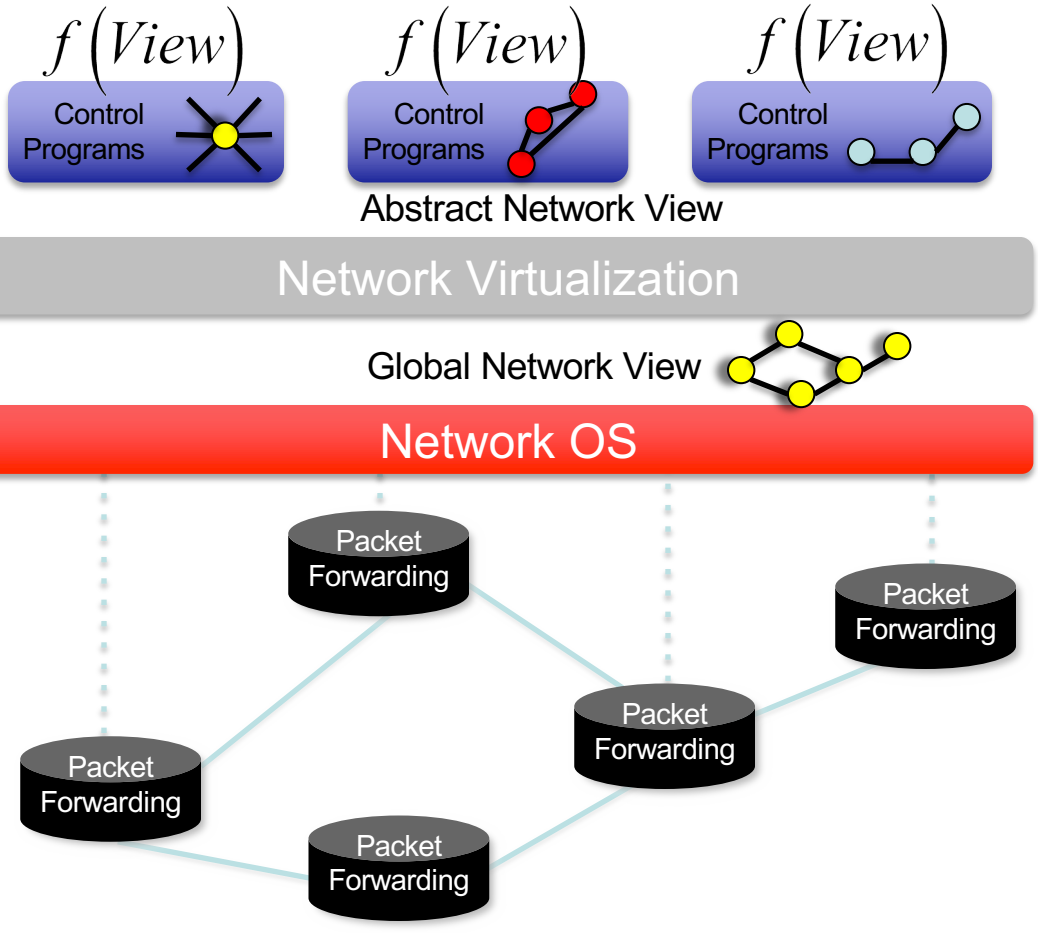
1. Rise of Linux.
2. Rise of baremetal servers and data centers.
3. SDN: Rise of merchant switching silicon.

Today

Most networking equipment is disaggregating

- Enterprise network equipment: switch, router, firewall
- WiFi APs
- Intra- and inter-datacenter networks
- ISP routers and switches
- Cellular basestations (4G, 5G…)
- Residential broadband access

Network Virtualization



You said

Hannes

Given that NVP is focused on providing virtualization capability to enterprise workloads specifically, rather than mega-data centers, what considerations or changes would need to be implemented at the design level to allow for similar levels of virtualization at that scale?

Agata

The authors describe that virtualization can be achieved by making switches and routers directly programmable, but it would require commercial vendors' buy-in - has that happened?

You said

Leo

Are the actions provided in the flow-table given as an executable format for the switch's processor to execute, or must the switches add the simple actions to their hardware? Does this limit the complexity of actions if line-rate processing is desired?

Since the network is virtualized in software, does this mean that the network is susceptible to non-deterministic tail latency (i.e. problems with scheduling or contention)?

Kathleen

Why are forwarding pipelines necessary/beneficial over a single forwarding/flow table?

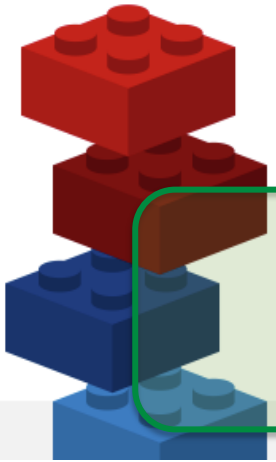
You said

Many of you …

How widely has OpenFlow been adopted?

ABOUT PROJECTS SOLUTIONS SOFTWARE DEFINED STANDARDS GET INVOLVED

SDN OVERVIEW PRODUCT CERTIFICATION SKILLS CERTIFICATION CORD LEARNING LABS NEW



Software-Defined Networking (SDN) Definition

What is SDN? The physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices.

What is SDN in plain English?

- Ideally at the level for college freshmen
  - Because, if you can't, you are not really understanding it! [Feynman's guiding principle]

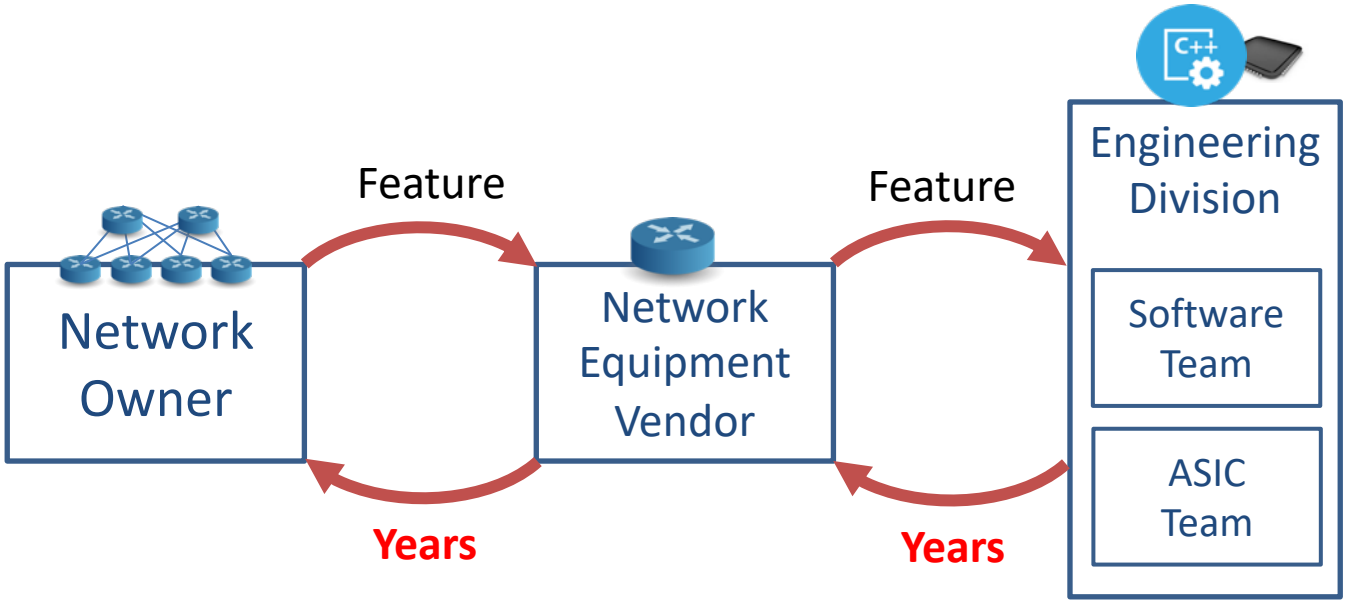
"Making programming networks as easy as programming computers."

Natural questions that follow

"Making programming networks as easy as programming computers."

- Why should we program a network?
  - To realize some "beautiful ideas" easily, preferably on our own
- What are those "beautiful ideas"?
  - Any impactful or intriguing apps in particular?
- Why couldn't we do this easily in the pre-SDN era?
  - Any fundamental shifts happening?

Pre-SDN state of the network industry

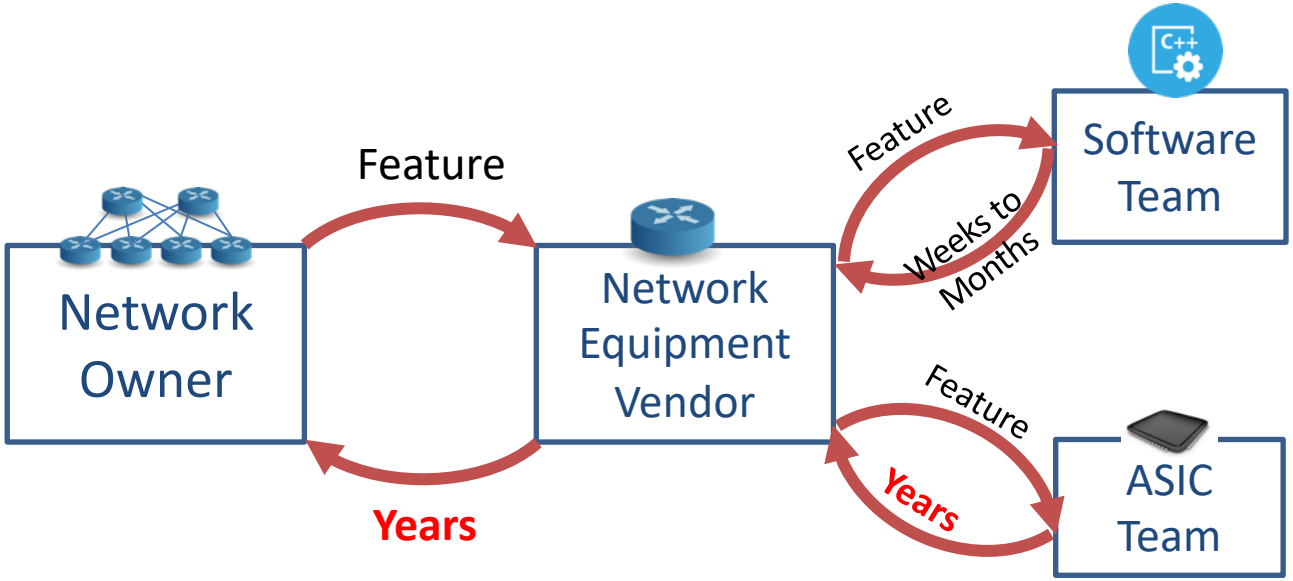


Compared to other industries, this is very unnatural

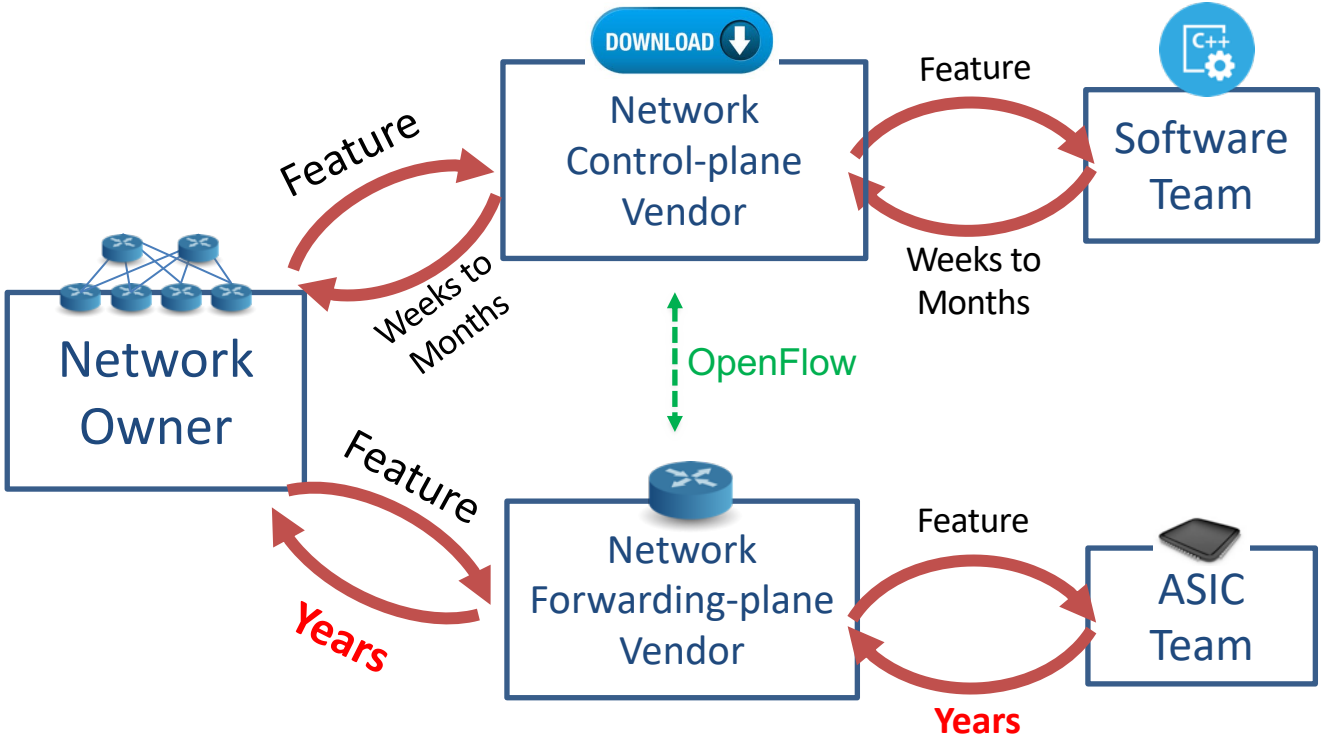
- Because we all know how to realize our own ideas by programming CPUs, GPUs, TPUs, etc.
  - Programs used in every phase (implement, verify, test, deploy, and maintain)
  - Extremely fast iteration and differentiation
  - We own our own ideas
  - A sustainable ecosystem where all participants benefit

Can we replicate this healthy, sustainable ecosystem for networking?

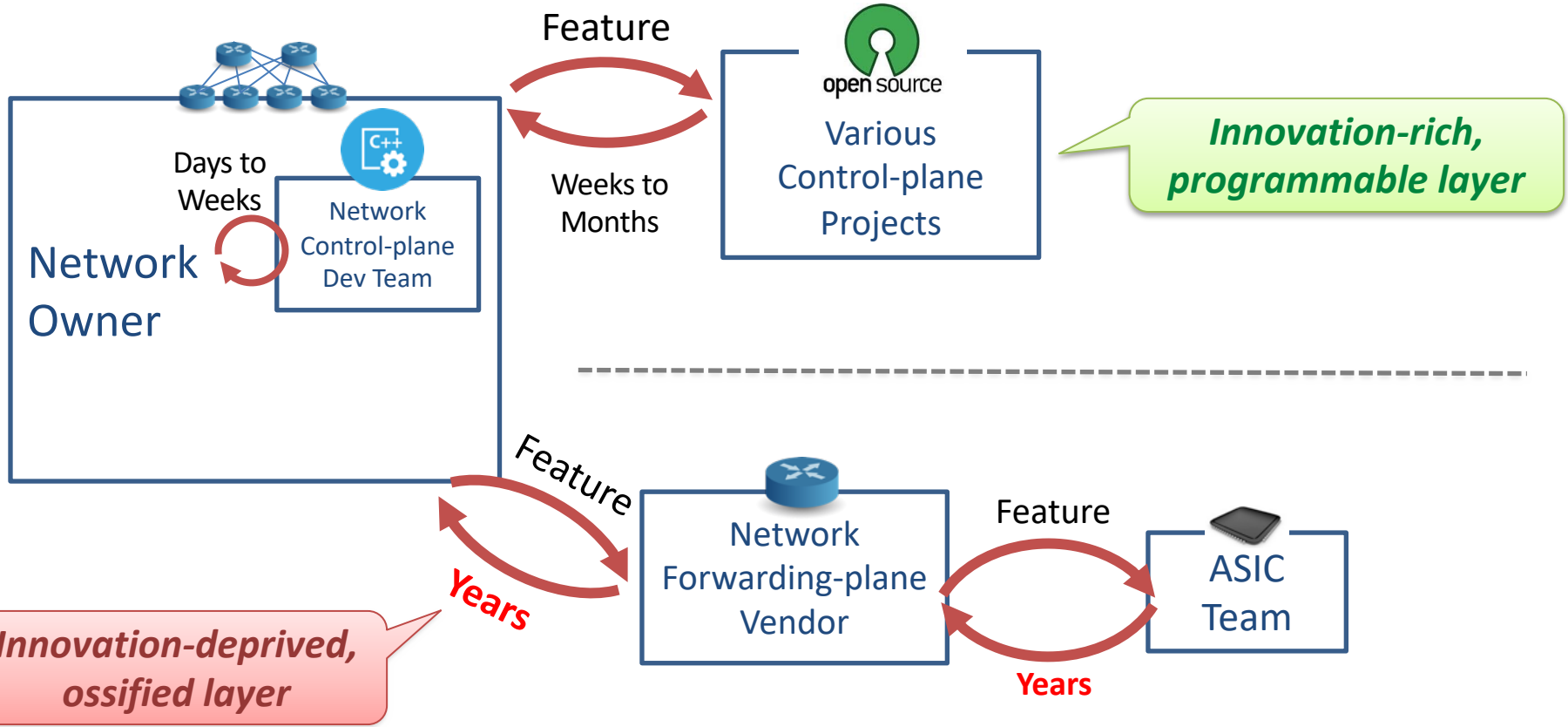
What SDN pioneers had realized …



And, SDN started to unfold …



And, SDN started to unfold …



End.